Wall Township Public Schools	Board Approval Date:

Course Title: WEB DESIGN AND MOBILE APP DEVELOPMENT

Course Description: This course fulfills the graduation requirement for career education/practical arts. In Web Design and Mobile App Development, students will unleash their creativity while learning to build responsive websites and fun mobile applications using HTML, CSS, JavaScript, and the React Native framework. Perfect for beginners, this course combines coding with imaginative projects that inspire problem-solving and innovation. Students will dive into exciting topics like designing eye-catching user interfaces, managing app states, and crafting organized layouts. They'll create a vibrant portfolio website to showcase their unique projects and develop mobile apps through interactive coding exercises. This dynamic and engaging course promises a fun learning experience while laying a strong foundation for future developers.

New Jersey Student Learning Standards
• 2020 New Jersey Student Learning Standards – Computer Science and Design Thinking Introduction
• 2020 New Jersey Student Learning Standards – Career Readiness, Life Literacies, and Key Skills

Course Sequence & Pacing	
Unit Title	**Estimated Number of Instructional Periods**
Unit 1: Web Design	30
Unit 2: Intro to Programming with Karel	25
Unit 3: JavaScript and Graphics	8
Unit 4: Intro to Mobile Apps & Components and Stylesheet	12
Unit 5: Buttons and Text Boxes	8
Unit 6: Advanced Layouts and Images	20
Unit 7: Event and State	10
Unit 8: Creating Multiple Screens	7
Unit 9: Working with Conditionals	15

Unit 1

Desired Results
Unit 1 Title: Web Design

Unit Summary:

In this unit, students are introduced to the fundamentals of web design, focusing on both the structure and presentation of web pages. They learn to create web pages using HTML elements such as links, images, lists, and tables, and enhance their designs with CSS styling through rules, classes, and IDs. Students also explore how web pages are requested and delivered across the internet.

Learning Standards

NJSLS:

- 8.1.12.CS.1: Describe ways in which integrated systems hide underlying implementation details to simplify user experiences.
- 8.1.12.CS.2: Model interactions between application software, system software, and hardware.
- 8.1.12.CS.3: Compare the functions of application software, system software, and hardware.
- 8.1.12.CS.4: Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.
- 8.1.12.NI.1: Evaluate the scalability and reliability of networks, by describing the relationship between routers, switches, servers, topology, and addressing.
- 8.1.12.NI.2: Evaluate security measures to address various common security threats.
- 8.1.12.NI.3: Explain how the needs of users and the sensitivity of data determine the level of security implemented.
- 8.1.12.NI.4: Explain how decisions on methods to protect data are influenced by whether the data is at rest, in transit, or in use.
- 8.1.12.IC.1: Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.
- 8.1.12.IC.2: Test and refine computational artifacts to reduce bias and equity deficits.
- 8.1.12.IC.3: Predict the potential impacts and implications of emerging technologies on larger social, economic, and political structures, using evidence from credible sources.
- 8.1.12.DA.1: Create interactive data visualizations using software tools to help others better understand real world phenomena, including climate change.
- 8.1.12.DA.2: Describe the trade-offs in how and where data is organized and stored.
- 8.1.12.DA.3: Translate between decimal numbers and binary numbers.

- 8.1.12.DA.4: Explain the relationship between binary numbers and the storage and use of data in a computing device.
- 8.1.12.DA.5: Create data visualizations from large data sets to summarize, communicate, and support different interpretations of real-world phenomena.
- 8.1.12.DA.6: Create and refine computational models to better represent the relationships among different elements of data collected from a phenomenon or process.
- 8.1.12.AP.1: Design algorithms to solve computational problems using a combination of original and existing algorithms.
- 8.1.12.AP.2: Create generalized computational solutions using collections instead of repeatedly using simple variables.
- 8.1.12.AP.3: Select and combine control structures for a specific application based upon performance and readability, and identify trade-offs to justify the choice.
- 8.1.12.AP.4: Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue.
- 8.1.12.AP.5: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
- 8.1.12.AP.6: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.
- 8.1.12.AP.7: Collaboratively design and develop programs and artifacts for broad audiences by incorporating feedback from users.
- 8.1.12.AP.8: Evaluate and refine computational artifacts to make them more usable and accessible.
- 8.1.12.AP.9: Collaboratively document and present design decisions in the development of complex programs.
- 8.2.12.ED.1: Use research to design and create a product or system that addresses a problem and make modifications based on input from potential consumers.
- 8.2.12.ED.2: Create scaled engineering drawings for a new product or system and make modification to increase optimization based on feedback.
- 8.2.12.ED.3: Evaluate several models of the same type of product and make recommendations for a new design based on a cost benefit analysis.

- 8.2.12.ED.4: Design a product or system that addresses a global problem and document decisions made based on research, constraints, trade-offs, and aesthetic and ethical considerations and share this information with an appropriate audience.
- 8.2.12.ED.5: Evaluate the effectiveness of a product or system based on factors that are related to its requirements, specifications, and constraints (e.g., safety, reliability, economic considerations, quality control, environmental concerns, manufacturability, maintenance and repair, ergonomics).
- 8.2.12.ED.6: Analyze the effects of changing resources when designing a specific product or system (e.g., materials, energy, tools, capital, labor).
- 8.2.12.ITH.1: Analyze a product to determine the impact that economic, political, social, and/or cultural factors have had on its design, including its design constraints.
- 8.2.12.ITH.2: Propose an innovation to meet future demands supported by an analysis of the potential costs, benefits, trade-offs, and risks related to the use of the innovation.
- 8.2.12.ITH.3: Analyze the impact that globalization, social media, and access to open source technologies has had on innovation and on a society's economy, politics, and culture.
- 8.2.12.NT.1: Explain how different groups can contribute to the overall design of a product.
- 8.2.12.NT.2: Redesign an existing product to improve form or function.
- 8.2.12.ETW.1: Evaluate ethical considerations regarding the sustainability of environmental resources that are used for the design, creation, and maintenance of a chosen product.
- 8.2.12.ETW.2: Synthesize and analyze data collected to monitor the effects of a technological product or system on the environment.
- 8.2.12.ETW.3: Identify a complex, global environmental or climate change issue, develop a systemic plan of investigation, and propose an innovative sustainable solution.
- 8.2.12.EC.1: Analyze controversial technological issues and determine the degree to which individuals, businesses, and governments have an ethical role in decisions that are made.
- 8.2.12.EC.2: Assess the positive and negative impacts of emerging technologies on developing countries and evaluate how individuals, non-profit organizations, and governments have responded.
- 8.2.12.EC.3: Synthesize data, analyze trends, and draw conclusions regarding the effect of a technology on the individual, culture, society, and environment and share this information with the appropriate audience.

- 8.2.12.ETW.4: Research historical tensions between environmental and economic considerations as driven by human needs and wants in the development of a technological product and present the competing viewpoints.

Interdisciplinary Connections

Career Readiness, Life Literacies, & Key Skills *(CLKS)*:

- 9.4.12.CI.1: Demonstrate the ability to reflect, analyze, and use creative skills and ideas (e.g., 1.1.12prof.CR3a).
- 9.4.12.CI.2: Identify career pathways that highlight personal talents, skills, and abilities (e.g., 1.4.12prof.CR2b, 2.2.12.LF.8).
- 9.4.12.CI.3: Investigate new challenges and opportunities for personal growth, advancement, and transition (e.g., 2.1.12.PGD.1).
- 9.4.12.CT.1: Identify problem-solving strategies used in the development of an innovative product or practice (e.g., 1.1.12acc.C1b, 2.2.12.PF.3).
- 9.4.12.CT.2: Explain the potential benefits of collaborating to enhance critical thinking and problem solving (e.g., 1.3E.12profCR3.a).
- 9.4.12.CT.3: Enlist input from a variety of stakeholders (e.g., community members, experts in the field) to design a service learning activity that addresses a local or global issue (e.g., environmental justice).
- 9.4.12.CT.4: Participate in online strategy and planning sessions for course-based, school-based, or other project and determine the strategies that contribute to effective outcomes.
- 9.4.12.DC.1: Explain the beneficial and harmful effects that intellectual property laws can have on the creation and sharing of content (e.g., 6.1.12.CivicsPR.16.a).
- 9.4.12.DC.2: Compare and contrast international differences in copyright laws and ethics.
- 9.4.12.DC.3: Evaluate the social and economic implications of privacy in the context of safety, law, or ethics (e.g., 6.3.12.HistoryCA.1).
- 9.4.12.DC.4: Explain the privacy concerns related to the collection of data (e.g., cookies) and generation of data through automated processes that may not be evident to users (e.g., 8.1.12.NI.3).
- 9.4.12.DC.5: Debate laws and regulations that impact the development and use of software.
- 9.4.12.DC.6: Select information to post online that positively impacts personal image and future college and career opportunities.

- 9.4.12.DC.7: Evaluate the influence of digital communities on the nature, content and responsibilities of careers, and other aspects of society (e.g., 6.1.12.CivicsPD.16.a).
- 9.4.12.DC.8: Explain how increased network connectivity and computing capabilities of everyday objects allow for innovative technological approaches to climate protection.
- 9.4.12.GCA.1: Collaborate with individuals to analyze a variety of potential solutions to climate change effects and determine why some solutions (e.g., political, economic, cultural) may work better than others (e.g., SL.11-12.1, HS-ETS1-1, HS-ETS1-2, HS-ETS1-4, 6.3.12.GeoGI.1, 7.1.IH.IPERS.6, 7.1.IL.IPERS.7, 8.2.12.ETW.3).
- 9.4.12.IML.1: Compare search browsers and recognize features that allow for filtering of information.
- 9.4.12.IML.2: Evaluate digital sources for timeliness, accuracy, perspective, credibility of the source, and relevance of information, in media, data, or other resources (e.g., NJSLSA.W8, SocialStudies Practice: Gathering and Evaluating Sources).
- 9.4.12.IML.3: Analyze data using tools and models to make valid and reliable claims, or to determine optimal design solutions (e.g., S-ID.B.6a, 8.1.12.DA.5, 7.1.IH.IPRET.8).
- 9.4.12.IML.4: Assess and critique the appropriateness and impact of existing data visualizations for an intended audience (e.g., S-ID.B.6b, HS-LS2-4).
- 9.4.12.IML.5: Evaluate, synthesize, and apply information on climate change from various sources appropriately (e.g., 2.1.12.CHSS.6, S.IC.B.4, S.IC.B.6, 8.1.12.DA.1, 6.1.12.GeoHE.14.a, 7.1.AL.PRSNT.2).
- 9.4.12.IML.6: Use various types of media to produce and store information on climate change for different purposes and audiences with sensitivity to cultural, gender, and age diversity (e.g., NJSLSA.SL5).
- 9.4.12.IML.7: Develop an argument to support a claim regarding a current workplace or societal/ethical issue such as climate change (e.g., NJSLSA.W1, 7.1.AL.PRSNT.4).
- 9.4.12.IML.8: Evaluate media sources for point of view, bias, and motivations (e.g., NJSLSA.R6, 7.1.AL.IPRET.6).
- 9.4.12.IML.9: Analyze the decisions creators make to reveal explicit and implicit messages within information and media (e.g., 1.5.12acc.C2a, 7.1.IL.IPRET.4).
- 9.4.12.TL.1: Assess digital tools based on features such as accessibility options, capacities, and utility for accomplishing a specified task (e.g., W.11-12.6).
- 9.4.12.TL.2: Generate data using formula-based calculations in a spreadsheet and draw conclusions about the data.
- 9.4.12.TL.3: Analyze the effectiveness of the process and quality of collaborative environments.

- 9.4.12.TL.4: Collaborate in online learning communities or social networks or virtual worlds to analyze and propose a resolution to a real-world problem (e.g., 7.1.AL.IPERS.6).

Interdisciplinary Connections:

Science and Engineering Practices:

- Develop a model based on evidence to illustrate the relationships between systems or between components of a system.
- Use a model to predict the relationships between systems or between components of a system.
- Apply scientific principles and evidence to provide an explanation of phenomena and solve design problems, taking into account possible unanticipated effects.
- Analyze data using tools, technologies, and/or models (e.g., computational, mathematical) in order to make valid and reliable scientific claims or determine an optimal design solution.
- Create a computational model or simulation of a phenomenon, designed device, process, or system.
- Design, evaluate, and/or refine a solution to a complex real-world problem, based on scientific knowledge, student-generated sources of evidence, prioritized criteria, and tradeoff considerations.
- Evaluate the claims, evidence, and reasoning behind currently accepted explanations or solutions to determine the merits of arguments.
- Evaluate the validity and reliability of multiple claims that appear in scientific and technical texts or media reports, verifying the data when possible.
- Use mathematical and/or computational representations of phenomena or design solutions to support explanations.
- Make and defend a claim based on evidence about the natural world that reflects scientific knowledge, and student-generated evidence.
- Create or revise a simulation of a phenomenon, designed device, process, or system.
- Apply scientific reasoning to link evidence to the claims to assess the extent to which the reasoning and data support the explanation or conclusion.
- Construct an oral and written argument or counter-arguments based on data and evidence.

- Evaluate a solution to a complex real-world problem, based on scientific knowledge, student-generated sources of evidence, prioritized criteria, and tradeoff considerations.

Engineering Design Standards:

- HS-ETS1-1 Analyze a major global challenge to specify qualitative and quantitative criteria and constraints for solutions that account for societal needs and wants.
- HS-ETS1-2 Design a solution to a complex real-world problem by breaking it down into smaller, more manageable problems that can be solved through engineering.
- HS-ETS1-3 Evaluate a solution to a complex real-world problem based on prioritized criteria and trade-offs that account for a range of constraints, including cost, safety, reliability, and aesthetics, as well as possible social, cultural, and environmental impacts.
- HS-ETS1-4 Use a computer simulation to model the impact of proposed solutions to a complex real-world problem with numerous criteria and constraints on interactions within and between systems relevant to the problem.

Learning Targets

Unit Essential Questions:

- How does HTML provide the structure for a webpage?
- In what ways does CSS enhance the design and usability of a webpage?
- How do links, images, lists, and tables improve the functionality and user experience of a website?
- How are web pages requested, delivered, and displayed on the internet?
- Why is it important to combine creativity with technical skills when designing websites?

Unit Enduring Understandings:

- Webpages are built using HTML, which provides the foundation for organizing and displaying content.
- CSS is essential for controlling the appearance and style of a website, creating a visually appealing and user-friendly experience.
- Interactive and well-structured features like links, images, lists, and tables make websites more engaging and informative.

	• Understanding the process of how web pages are served and accessed is crucial to becoming an informed and effective web designer. • Effective web design requires a balance of technical knowledge and creative design, allowing students to express ideas while meeting user needs.
Vocabulary: HTML (HyperText Markup Language), CSS (Cascading Style Sheets), Element, Tag, Attribute, Selector, Class, ID, Property, Value, Accessibility, Responsive Design, User Experience (UX), User Interface (UI), Hyperlink, Domain, URL, Server, Client, Browser, Debugging, Validation, Hierarchy, Layout, Typography, Color Theory, Wireframe, Prototype	
Students will be able to: • Identify the purpose and applications of HTML • Create their first simple web page • Discern the various parts of an HTML page • Create fully formed HTML pages • Apply formatting tags in order to modify the appearance of text and make web pages look clear and aesthetically pleasing • Add and utilize hyperlinks on their webpages • Embed an image in HTML • Incorporate different kinds of lists to their web pages • Create tables in their web pages • Explain the benefits of including tables on web pages • Compare various ways of displaying information and choose the appropriate format • Apply HTML styling to make their web pages more visually appealing and unique • Describe how CSS adds styling to HTML pages • Use CSS tag selectors to select all elements of the same kind and give them all the same style • Use CSS class selectors to apply CSS styling to all HTML units that share a specified class	

- Use CSS Selectors by ID to select a single element to format on a webpage
- Describe the process that occurs when typing in a URL, from sending a request and response over the Internet to viewing a webpage
- Create their own website from scratch, hosted at their own custom domain

Learning Activities & Differentiation: *(Amistad & Holocaust, Asian American & Pacific Islanders, Climate Change, LGBTQ and People with Disabilities, Diversity, Equity & Inclusion [DEI])*

Learning Activities:

- Inclusive Homepage Design: Students create a personal homepage that highlights their identity and values, incorporating diverse voices such as underrepresented leaders in technology, historical figures, or cultural influences.
- Digital Storytelling with Images: Students design a collage or showcase page using HTML and CSS to tell a story connected to a social issue (e.g., climate change awareness, accessibility in technology, or representation in media).
- Representation in Web Tables: Students build tables that present information about innovators from diverse backgrounds, including Amistad figures in STEM, AAPI pioneers, Holocaust-era technologists, and LGBTQ+ or disabled leaders in digital fields.
- Accessibility Challenge: Students apply CSS to improve readability and usability (font size, contrast, alt text for images) and test their webpages with accessibility tools or simulated vision impairments.
- Understanding How the Web Works: Students trace the process of a URL request and delivery, discussing how the digital divide impacts different communities' ability to access and contribute to the web.

Differentiation Strategies:

- For Advanced Learners: Extend projects to include responsive design using CSS media queries, advanced layout techniques, or research on inclusive web standards.
- For Struggling Learners: Provide structured templates, step-by-step coding guides, and color-coded annotations to scaffold HTML and CSS learning, with simplified tasks leading up to the final project.

- For English Language Learners: Offer bilingual glossaries of HTML and CSS terminology, vocabulary banks with visual cues, and access to captioned tutorial videos.
- Universal Design for Learning (UDL): Provide multiple ways to learn content (articles, videos, tutorials), flexible ways to demonstrate knowledge (website walkthroughs, digital posters, or podcasts), and allow varied pacing with checkpoints and peer support.

Education Connections:

- Highlight diverse voices and contributions in web development and technology, including innovators from historically underrepresented groups.
- Discuss how accessibility features in web design (alt text, contrast, navigation tools) support equity and inclusion for people with disabilities and LGBTQ+ individuals.
- Explore the importance of presenting information ethically and accurately online, connecting to historical and social justice movements such as Amistad and Holocaust remembrance.
- Examine how climate change and social justice issues can be amplified through digital storytelling and accessible web design.
- Encourage critical conversations about who has access to web publishing tools, how design choices can reinforce inequality, and how inclusive design can empower marginalized communities.

Assessment Evidence

Formative Assessments:

- CodeHS exercises
    - Example Exercise: Create a webpage that provides links to your favorite books
    - Example Exercise: Create CSS classes and IDs to apply formatting to a BINGO board
- CodeHS Checks for Understanding

Summative Assessments:

- Unit Project: Build a homepage

Common Benchmark Assessments:

- Unit Project: Build a homepage

Alternative Assessments:

- CodeHS Unit Quiz

Core Instructional & Supplemental Materials

Core Materials:

- CodeHS Web Design curriculum (HTML/CSS modules)
- Built-in CodeHS Web Lab (HTML/CSS editor and live preview environment)
- HTML and CSS reference guides and documentation
- CodeHS tutorials on links, images, tables, lists, and inline styling
- CodeHS lessons on CSS rules, selectors, classes, and IDs

Supplemental Materials:

- Sample HTML/CSS templates for scaffolding assignments
- Visual guides and infographics on webpage structure and CSS hierarchy
- Flowchart organizers for planning webpage layout and navigation
- Peer review checklists and collaborative feedback rubrics
- Example projects for design inspiration (e.g., music pages, cultural collages)
- Accessibility checklists to support inclusive design practices
- Rubrics for evaluating web design projects and personal portfolios
- Glossaries of key web development terms for English Language Learners

Accommodations and Modifications:
Students with Disabilities, English Language Learners, Students at Risk of Failure, Students with 504s, Gifted & Talented Students

Specific Strategies and Practices that Support Students with Disabilities:
Students receiving Special Education programming have specific goals and objectives, as well as accommodations and modifications outlined within their Individualized Education Plans (IEP) due to an identified disability and/or diagnosis. In addition to exposure to the general education curriculum, the instruction is differentiated based upon the student's needs. The IEP acts as a supplemental curriculum guide inclusive of instructional strategies that support each learner.

- Additional time - Review of directions - Extra visual and verbal cues and prompts - Follow a routine/schedule - Verbal and visual cues regarding directions and staying on task - Immediate feedback

Specific Strategies and Practices that Support English Language Learners
English language learner (ELL) is a national-origin-minority student who is limited-English-proficient. This term is often preferred over limited-English-proficient (LEP) as it highlights accomplishments rather than deficits.

- Pre-teaching of vocabulary and concepts - Personal Glossary - Visual learning, including graphic organizers - Teacher modeling - Cooperative learning groups

Students at Risk of Failure

- Alternative Assessments - Learning Contracts - Think-Pair-Share by Readiness - Tiered Level of Supports - Small Group Instruction - Group Work Choice

Students with 504 Plans
Teachers are responsible for implementing designated services and strategies identified on a student's 504 Plan.
504 plans are formal plans that schools develop to give kids the support they need and covers any condition that limits daily activities in a major way. Students' accommodations are outlined in their individual 504 plan.

Specific Strategies and Practices that Support Gifted & Talented Students
It is critical that all teachers are able to recognize a high-ability student who may need more depth and complexity in instruction.

- Small Group In-Depth Instruction - Differentiated Online Practice

Unit 2

Desired Results

Unit 2 Title: Intro to Programming with Karel

Unit Summary:

In this unit, students are introduced to the fundamentals of programming through engaging activities with Karel the Dog. Using JavaScript, they learn how to write commands, create and call methods, and apply control structures such as loops and conditionals. The unit emphasizes proper programming syntax, formatting, and problem-solving strategies like Top Down Design, helping students build a strong foundation for future coding.

Learning Standards

NJSLS:

- 8.1.12.CS.1: Describe ways in which integrated systems hide underlying implementation details to simplify user experiences.
- 8.1.12.CS.2: Model interactions between application software, system software, and hardware.
- 8.1.12.CS.3: Compare the functions of application software, system software, and hardware.
- 8.1.12.CS.4: Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.
- 8.1.12.AP.1: Design algorithms to solve computational problems using a combination of original and existing algorithms.
- 8.1.12.AP.3: Select and combine control structures for a specific application based upon performance and readability, and identify trade-offs to justify the choice.
- 8.1.12.AP.4: Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue.
- 8.1.12.AP.5: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
- 8.1.12.AP.6: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.
- 8.1.12.AP.7: Collaboratively design and develop programs and artifacts for broad audiences by incorporating feedback from users.

- 8.1.12.AP.8: Evaluate and refine computational artifacts to make them more usable and accessible.
- 8.1.12.AP.9: Collaboratively document and present design decisions in the development of complex programs.

Interdisciplinary Connections

Career Readiness, Life Literacies, & Key Skills *(CLKS)*:

- 9.4.12.CI.1: Demonstrate the ability to reflect, analyze, and use creative skills and ideas (e.g., 1.1.12prof.CR3a).
- 9.4.12.CI.2: Identify career pathways that highlight personal talents, skills, and abilities (e.g., 1.4.12prof.CR2b, 2.2.12.LF.8).
- 9.4.12.CI.3: Investigate new challenges and opportunities for personal growth, advancement, and transition (e.g., 2.1.12.PGD.1).
- 9.4.12.CT.1: Identify problem-solving strategies used in the development of an innovative product or practice (e.g., 1.1.12acc.C1b, 2.2.12.PF.3).
- 9.4.12.CT.2: Explain the potential benefits of collaborating to enhance critical thinking and problem solving (e.g., 1.3E.12profCR3.a).
- 9.4.12.CT.3: Enlist input from a variety of stakeholders (e.g., community members, experts in the field) to design a service learning activity that addresses a local or global issue (e.g., environmental justice).
- 9.4.12.CT.4: Participate in online strategy and planning sessions for course-based, school-based, or other project and determine the strategies that contribute to effective outcomes.
- 9.4.12.DC.1: Explain the beneficial and harmful effects that intellectual property laws can have on the creation and sharing of content (e.g., 6.1.12.CivicsPR.16.a).
- 9.4.12.DC.2: Compare and contrast international differences in copyright laws and ethics.
- 9.4.12.DC.3: Evaluate the social and economic implications of privacy in the context of safety, law, or ethics (e.g., 6.3.12.HistoryCA.1).
- 9.4.12.DC.4: Explain the privacy concerns related to the collection of data (e.g., cookies) and generation of data through automated processes that may not be evident to users (e.g., 8.1.12.NI.3).
- 9.4.12.DC.5: Debate laws and regulations that impact the development and use of software.
- 9.4.12.DC.6: Select information to post online that positively impacts personal image and future college and career opportunities.

- 9.4.12.DC.7: Evaluate the influence of digital communities on the nature, content and responsibilities of careers, and other aspects of society (e.g., 6.1.12.CivicsPD.16.a).
- 9.4.12.DC.8: Explain how increased network connectivity and computing capabilities of everyday objects allow for innovative technological approaches to climate protection.
- 9.4.12.GCA.1: Collaborate with individuals to analyze a variety of potential solutions to climate change effects and determine why some solutions (e.g., political. economic, cultural) may work better than others (e.g., SL.11-12.1., HS-ETS1-1, HS-ETS1-2, HS-ETS1-4, 6.3.12.GeoGI.1, 7.1.IH.IPERS.6, 7.1.IL.IPERS.7, 8.2.12.ETW.3).
- 9.4.12.IML.1: Compare search browsers and recognize features that allow for filtering of information.
- 9.4.12.IML.2: Evaluate digital sources for timeliness, accuracy, perspective, credibility of the source, and relevance of information, in media, data, or other resources (e.g., NJSLSA.W8, SocialStudies Practice: Gathering and Evaluating Sources).
- 9.4.12.IML.3: Analyze data using tools and models to make valid and reliable claims, or to determine optimal design solutions (e.g., S-ID.B.6a., 8.1.12.DA.5, 7.1.IH.IPRET.8).
- 9.4.12.IML.4: Assess and critique the appropriateness and impact of existing data visualizations for an intended audience (e.g., S-ID.B.6b, HS-LS2-4).
- 9.4.12.IML.5: Evaluate, synthesize, and apply information on climate change from various sources appropriately (e.g., 2.1.12.CHSS.6, S.IC.B.4, S.IC.B.6, 8.1.12.DA.1, 6.1.12.GeoHE.14.a, 7.1.AL.PRSNT.2).
- 9.4.12.IML.6: Use various types of media to produce and store information on climate change for different purposes and audiences with sensitivity to cultural, gender, and age diversity (e.g., NJSLSA.SL5).
- 9.4.12.IML.7: Develop an argument to support a claim regarding a current workplace or societal/ethical issue such as climate change (e.g., NJSLSA.W1, 7.1.AL.PRSNT.4).
- 9.4.12.IML.8: Evaluate media sources for point of view, bias, and motivations (e.g., NJSLSA.R6, 7.1.AL.IPRET.6).
- 9.4.12.IML.9: Analyze the decisions creators make to reveal explicit and implicit messages within information and media (e.g., 1.5.12acc.C2a, 7.1.IL.IPRET.4).
- 9.4.12.TL.1: Assess digital tools based on features such as accessibility options, capacities, and utility for accomplishing a specified task (e.g., W.11-12.6.).
- 9.4.12.TL.2: Generate data using formula-based calculations in a spreadsheet and draw conclusions about the data.
- 9.4.12.TL.3: Analyze the effectiveness of the process and quality of collaborative environments.

- 9.4.12.TL.4: Collaborate in online learning communities or social networks or virtual worlds to analyze and propose a resolution to a real-world problem (e.g., 7.1.AL.IPERS.6).

Interdisciplinary Connections:

Science and Engineering Practices:

- Develop and use a model based on evidence to illustrate the relationships between systems or between components of a system.
- Plan and conduct an investigation individually and collaboratively to produce data to serve as the basis for evidence, and in the design: decide on types, how much, and accuracy of data needed to produce reliable measurements and consider limitations on the precision of the data (e.g., number of trials, cost, risk, time), and refine the design accordingly.
- Apply scientific principles and evidence to provide an explanation of phenomena and solve design problems, taking into account possible unanticipated effects.
- Analyze data using tools, technologies, and/or models (e.g., computational, mathematical) in order to make valid and reliable scientific claims or determine an optimal design solution.
- Use mathematical and/or computational representations of phenomena or design solutions to support explanations.
- Use mathematical representations of phenomena or design solutions to support and revise explanations.
- Make and defend a claim based on evidence about the natural world that reflects scientific knowledge, and student-generated evidence.
- Apply scientific reasoning to link evidence to the claims to assess the extent to which the reasoning and data support the explanation or conclusion.
- Construct an oral and written argument or counter-arguments based on data and evidence.

Engineering Design Standards:

- HS-ETS1-2 Design a solution to a complex real-world problem by breaking it down into smaller, more manageable problems that can be solved through engineering.

- HS-ETS1-3 Evaluate a solution to a complex real-world problem based on prioritized criteria and trade-offs that account for a range of constraints, including cost, safety, reliability, and aesthetics, as well as possible social, cultural, and environmental impacts.
- HS-ETS1-4 Use a computer simulation to model the impact of proposed solutions to a complex real-world problem with numerous criteria and constraints on interactions within and between systems relevant to the problem.

Learning Targets

Unit Essential Questions:

- How can programming commands be combined to solve complex problems?
- What is the difference between defining a method and calling a method, and why is this distinction important?
- How do control structures like loops and conditionals make programs more efficient and adaptable?
- Why is breaking a large problem into smaller parts (Top Down Design) beneficial in programming?
- How does proper code formatting and commenting improve program readability and maintenance?

Unit Enduring Understandings:

- Programming requires clear and precise commands to control a computer or robot effectively.
- Methods allow programmers to encapsulate behavior, making code reusable and easier to manage.
- Control flow structures such as loops and conditionals enable programs to handle repetition and decision-making.
- Breaking down complex problems into smaller subproblems simplifies coding and debugging.
- Well-commented and organized code improves collaboration, understanding, and future modification.

Vocabulary: commands, method, function, control flow, loop, conditional, precondition, postcondition, top down design, syntax, comment, call, define

Students will be able to:

- Recognize, explain, and use the commands that Karel can be given. These commands are: move(), putBall(), takeBall() and turnLeft().
- Recognize, explain, and use the commands that Karel can be given
- Debug and rework their code as well as someone else's code
- Define and call functions
- Understand what functions are, how they are used and how using them improves programs
- Design and implement their own functions to solve problems
- Explain the importance of writing readable code
- Analyze and compare the readability of different programs
- Use the start function to make their programs more readable
- Break a large problem down into smaller, simpler problems
- Write functions that solve the simpler problems, and use them as building blocks to solve the larger problem
- Compare programs and identify good vs. poor decomposition
- Explain the preconditions and postconditions of a function
- Create clear and readable comments in their code that help the reader understand the code
- Explain the purpose of comments
- Write programs that use SuperKarel instead of Karel
- Utilize the new toolbox of commands that SuperKarel provides over Karel
- Create for loops to repeat code a fixed number of times
- Explain when a for loop would be a useful tool
- Utilize for loops to write programs that would be difficult/impossible without loops
- Use conditions to gather information about Karel's world (is the front clear, is Karel facing north, etc)
- Create if statements to only execute code if a certain condition is true
- Explain the purpose of an If/Else statement
- Create If/Else statements to solve new types of problems
- Identify when an If/Else statement is appropriate to be used
- Explain the purpose of a while loop

- Create while loops to repeat code while a condition is true
- Utilize while loops to solve new types of problems
- Test their solutions on different Karel worlds
- Identify the different control structures that can be used to modify the flow of control through a program
- Combine control structures to solve complicated problems
- Choose the proper control structure for a given problem
- Analyze a solution to a problem and explain why it works
- Use control structures to create general solutions that work on all Karel worlds
- Explain why it is important to indent code
- Identify proper indentation
- Modify a program to have proper indentation
- Write programs with proper indentation
- Define a problem in their own words and plan out a solution to the problem
- Break a large problem down into smaller pieces and solve each of the pieces, then use these solutions as building blocks to solve the larger problem
- Utilize the proper control structures to create general solutions that solve multiple Karel worlds
- Write clear and readable code using control structures, functions, decomposition, and comments

Learning Activities & Differentiation: *(Amistad & Holocaust, Asian American & Pacific Islanders, Climate Change, LGBTQ and People with Disabilities, Diversity, Equity & Inclusion [DEI])*

Learning Activities:

- Commands Practice: Students write programs using Karel's basic commands (move(), putBall(), takeBall(), turnLeft()) to complete structured tasks like building shapes or moving through mazes.
- Function Definition and Calling: Students learn to define their own functions (e.g., turnRight(), makePancakes()) and call them within programs to simplify code and promote reuse.
- Top Down Design: Students break larger Karel problems into smaller functions and build their solutions step-by-step, enhancing problem-solving and modular programming skills.

- Loops and Conditionals: Students implement for loops to repeat actions, and use if/else statements and while loops to control program flow based on conditions in Karel's world.
- Debugging and Code Improvement: Students practice debugging their own and peers' programs to find and fix errors, improving logical thinking and code quality.
- Commenting and Readability: Students learn to write clear comments and format their code for readability, understanding the importance of maintainable programming.
- SuperKarel Commands: Students explore advanced commands offered by SuperKarel to solve more complex tasks efficiently.

Differentiation Strategies:

- For Advanced Learners: Challenge students with complex problems requiring nested loops, multiple functions, and comprehensive use of conditionals and SuperKarel commands. Encourage exploration of algorithm optimization and code refactoring.
- For Struggling Learners: Provide scaffolded templates, step-by-step instructions, and guided debugging activities focusing on core commands and simple functions. Use visual aids like flowcharts to support understanding.
- For English Language Learners: Offer bilingual glossaries for programming terms, annotated code examples, and captioned tutorial videos. Use visuals and diagrams to explain control flow and functions.
- Universal Design for Learning (UDL): Present content through varied formats (written, video, interactive coding), allow different ways to demonstrate learning (code, written explanation, flowcharts), and enable flexible pacing with checkpoints and peer support.

Education Connections:

- Highlight how programming foundational skills like problem decomposition and control flow are essential building blocks for all areas of computer science and technology innovation.
- Discuss the importance of writing clear, readable, and well-documented code to support collaboration, maintainability, and ethical software development.
- Explore how debugging and testing reflect real-world practices in software engineering, emphasizing persistence

and critical thinking in problem-solving.
- Examine how programming concepts such as loops and conditionals enable computers to handle complex tasks efficiently and adapt to changing conditions.
- Encourage critical conversations about the role of computational thinking in everyday technology use and how mastering these skills can empower students to create and innovate in diverse fields.

Assessment Evidence

Formative Assessments:

- CodeHS exercises
    - Example Exercise: Pyramid of Karel

      Write a program to have Karel build a pyramid. There should be three balls on the first row, two in the second row, and one in the third row.
    - Example Exercise: The Two Towers

      In this program, Karel should build two towers of tennis balls. Each tower should be 3 tennis balls high. At the end, Karel should end up on top of the second tower, facing East.
- CodeHS Checks for Understanding

Summative Assessments:

- Karel Challenges

Common Benchmark Assessments:

- Karel Challenges

Alternative Assessments:

- CodeHS Unit Quiz

Core Instructional & Supplemental Materials

Core Materials:

- CodeHS Introduction to Programming with Karel the Dog curriculum (JavaScript modules)
- Built-in CodeHS Karel Editor with live execution environment
- Karel command and syntax reference guides
- CodeHS tutorials on commands, functions, loops, and conditionals
- CodeHS lessons on Top Down Design, commenting, and debugging strategies

Supplemental Materials:

- Sample Karel code templates for scaffolding assignments
- Visual guides and flowcharts illustrating control flow, loops, and conditionals
- Problem decomposition organizers for planning Top Down Design solutions
- Debugging checklists and peer collaboration rubrics
- Example Karel challenge solutions for inspiration (e.g., Pyramid of Karel, Random Hurdles)
- Readability and commenting checklists to support clear, maintainable code
- Rubrics for evaluating programming projects, including functionality, decomposition, and style
- Glossaries of key programming terms for English Language Learners

Accommodations and Modifications: Students with Disabilities, English Language Learners, Students at Risk of Failure, Students with 504s, Gifted & Talented Students

Specific Strategies and Practices that Support Students with Disabilities:

Students receiving Special Education programming have specific goals and objectives, as well as accommodations and modifications outlined within their Individualized Education Plans (IEP) due to an identified disability and/or diagnosis. In addition to exposure to the general education curriculum, the instruction is differentiated based upon the student's needs. The IEP acts as a supplemental curriculum guide inclusive of instructional strategies that support each learner.

- Small group/One to one - Additional time - Review of directions - Student restates information - Extra visual and verbal cues and prompts - Follow a routine/schedule - Checklists - Immediate feedback

Specific Strategies and Practices that Support English Language Learners

English language learner (ELL) is a national-origin-minority student who is limited-English-proficient. This term is often preferred over limited-English-proficient (LEP) as it highlights accomplishments rather than deficits.

- Pre-teaching of vocabulary and concepts - Personal Glossary - Extended Time - Simplified/Verbal Instructions - Visual learning, including graphic organizers - Teacher modeling - Scaffolding - Think-pair-share - Cooperative learning groups

Students at Risk of Failure

- Alternative Assessments - Supplemental Instruction - Think-Pair-Share by Readiness - Homework Options - Leveled Rubrics - Tiered Level of Supports - Small Group Instruction - Work Alone Choice - Group Work Choice

Students with 504 Plans

Teachers are responsible for implementing designated services and strategies identified on a student's 504 Plan.
504 plans are formal plans that schools develop to give kids the support they need and covers any condition that limits daily activities in a major way. Students' accommodations are outlined in their individual 504 plan.

Specific Strategies and Practices that Support Gifted & Talented Students

It is critical that all teachers are able to recognize a high-ability student who may need more depth and complexity in instruction.

-Small Group In-Depth Instruction - Differentiated Online Practice - Alternative Assessments - Pre-Assess to provide Differentiated Instruction to Accelerate Learning

Unit 3

Desired Results

Unit 3 Title: JavaScript and Graphics

Unit Summary:

In this unit, students transition from Karel to real JavaScript programming. They learn foundational concepts including variables, user input, arithmetic expressions, and basic graphics. Through seven hands-on programming exercises, students practice gathering and using input, performing calculations, and creating visual outputs on the canvas.

Learning Standards

NJSLS:

- 8.1.12.CS.1: Describe ways in which integrated systems hide underlying implementation details to simplify user experiences.
- 8.1.12.CS.2: Model interactions between application software, system software, and hardware.
- 8.1.12.CS.3: Compare the functions of application software, system software, and hardware.
- 8.1.12.CS.4: Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.
- 8.1.12.IC.1: Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.
- 8.1.12.IC.2: Test and refine computational artifacts to reduce bias and equity deficits.
- 8.1.12.IC.3: Predict the potential impacts and implications of emerging technologies on larger social, economic, and political structures, using evidence from credible sources.
- 8.1.12.AP.1: Design algorithms to solve computational problems using a combination of original and existing algorithms.
- 8.1.12.AP.3: Select and combine control structures for a specific application based upon performance and readability, and identify trade-offs to justify the choice.
- 8.1.12.AP.4: Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue.
- 8.1.12.AP.5: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
- 8.1.12.AP.6: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.
- 8.1.12.AP.7: Collaboratively design and develop programs and artifacts for broad audiences by incorporating feedback from users.
- 8.1.12.AP.8: Evaluate and refine computational artifacts to make them more usable and accessible.
- 8.1.12.AP.9: Collaboratively document and present design decisions in the development of complex programs.

Interdisciplinary Connections

Career Readiness, Life Literacies, & Key Skills *(CLKS)*:

- 9.4.12.CI.1: Demonstrate the ability to reflect, analyze, and use creative skills and ideas (e.g., 1.1.12prof.CR3a).
- 9.4.12.CI.2: Identify career pathways that highlight personal talents, skills, and abilities (e.g., 1.4.12prof.CR2b, 2.2.12.LF.8).
- 9.4.12.CI.3: Investigate new challenges and opportunities for personal growth, advancement, and transition (e.g., 2.1.12.PGD.1).
- 9.4.12.CT.1: Identify problem-solving strategies used in the development of an innovative product or practice (e.g., 1.1.12acc.C1b, 2.2.12.PF.3).
- 9.4.12.CT.2: Explain the potential benefits of collaborating to enhance critical thinking and problem solving (e.g., 1.3E.12profCR3.a).
- 9.4.12.CT.3: Enlist input from a variety of stakeholders (e.g., community members, experts in the field) to design a service learning activity that addresses a local or global issue (e.g., environmental justice).
- 9.4.12.CT.4: Participate in online strategy and planning sessions for course-based, school-based, or other project and determine the strategies that contribute to effective outcomes.
- 9.4.12.DC.1: Explain the beneficial and harmful effects that intellectual property laws can have on the creation and sharing of content (e.g., 6.1.12.CivicsPR.16.a).
- 9.4.12.DC.2: Compare and contrast international differences in copyright laws and ethics.
- 9.4.12.DC.3: Evaluate the social and economic implications of privacy in the context of safety, law, or ethics (e.g., 6.3.12.HistoryCA.1).
- 9.4.12.DC.4: Explain the privacy concerns related to the collection of data (e.g., cookies) and generation of data through automated processes that may not be evident to users (e.g., 8.1.12.NI.3).
- 9.4.12.DC.5: Debate laws and regulations that impact the development and use of software.
- 9.4.12.DC.6: Select information to post online that positively impacts personal image and future college and career opportunities.
- 9.4.12.DC.7: Evaluate the influence of digital communities on the nature, content and responsibilities of careers, and other aspects of society (e.g., 6.1.12.CivicsPD.16.a).
- 9.4.12.DC.8: Explain how increased network connectivity and computing capabilities of everyday objects allow for innovative technological approaches to climate protection.

- 9.4.12.GCA.1: Collaborate with individuals to analyze a variety of potential solutions to climate change effects and determine why some solutions (e.g., political, economic, cultural) may work better than others (e.g., SL.11-12.1., HS-ETS1-1, HS-ETS1-2, HS-ETS1-4, 6.3.12.GeoGI.1, 7.1.IH.IPERS.6, 7.1.IL.IPERS.7, 8.2.12.ETW.3).
- 9.4.12.IML.1: Compare search browsers and recognize features that allow for filtering of information.
- 9.4.12.IML.2: Evaluate digital sources for timeliness, accuracy, perspective, credibility of the source, and relevance of information, in media, data, or other resources (e.g., NJSLSA.W8, SocialStudies Practice: Gathering and Evaluating Sources).
- 9.4.12.IML.3: Analyze data using tools and models to make valid and reliable claims, or to determine optimal design solutions (e.g., S-ID.B.6a., 8.1.12.DA.5, 7.1.IH.IPRET.8).
- 9.4.12.IML.4: Assess and critique the appropriateness and impact of existing data visualizations for an intended audience (e.g., S-ID.B.6b, HS-LS2-4).
- 9.4.12.IML.5: Evaluate, synthesize, and apply information on climate change from various sources appropriately (e.g., 2.1.12.CHSS.6, S.IC.B.4, S.IC.B.6, 8.1.12.DA.1, 6.1.12.GeoHE.14.a, 7.1.AL.PRSNT.2).
- 9.4.12.IML.6: Use various types of media to produce and store information on climate change for different purposes and audiences with sensitivity to cultural, gender, and age diversity (e.g., NJSLSA.SL5).
- 9.4.12.IML.7: Develop an argument to support a claim regarding a current workplace or societal/ethical issue such as climate change (e.g., NJSLSA.W1, 7.1.AL.PRSNT.4).
- 9.4.12.IML.8: Evaluate media sources for point of view, bias, and motivations (e.g., NJSLSA.R6, 7.1.AL.IPRET.6).
- 9.4.12.IML.9: Analyze the decisions creators make to reveal explicit and implicit messages within information and media (e.g., 1.5.12acc.C2a, 7.1.IL.IPRET.4).
- 9.4.12.TL.1: Assess digital tools based on features such as accessibility options, capacities, and utility for accomplishing a specified task (e.g., W.11-12.6.).
- 9.4.12.TL.2: Generate data using formula-based calculations in a spreadsheet and draw conclusions about the data.
- 9.4.12.TL.3: Analyze the effectiveness of the process and quality of collaborative environments.
- 9.4.12.TL.4: Collaborate in online learning communities or social networks or virtual worlds to analyze and propose a resolution to a real-world problem (e.g., 7.1.AL.IPERS.6).

Interdisciplinary Connections:

Science and Engineering Practices:

- Develop and use a model based on evidence to illustrate the relationships between systems or between components of a system.
- Use a model to predict the relationships between systems or between components of a system.
- Analyze data using tools, technologies, and/or models (e.g., computational, mathematical) in order to make valid and reliable scientific claims or determine an optimal design solution.
- Use mathematical and/or computational representations of phenomena or design solutions to support explanations.
- Apply concepts of statistics and probability (including determining function fits to data, slope, intercept, and correlation coefficient for linear fits) to scientific and engineering questions and problems, using digital tools when feasible.
- Design, evaluate, and/or refine a solution to a complex real-world problem, based on scientific knowledge, student-generated sources of evidence, prioritized criteria, and tradeoff considerations.
- Communicate scientific and technical information (e.g. about the process of development and the design and performance of a proposed process or system) in multiple formats (including orally, graphically, textually, and mathematically).

Engineering Design Standards:

- HS-ETS1-2 Design a solution to a complex real-world problem by breaking it down into smaller, more manageable problems that can be solved through engineering.
- HS-ETS1-4 Use a computer simulation to model the impact of proposed solutions to a complex real-world problem with numerous criteria and constraints on interactions within and between systems relevant to the problem.

Learning Targets	
Unit Essential Questions:	**Unit Enduring Understandings:**

• How can variables be used to store and manipulate information in a program? • Why is user input important for creating interactive programs? • In what ways can arithmetic expressions influence the behavior of a program? • How can graphics be used to represent ideas and enhance user experiences? • What strategies can programmers use to break down problems when creating programs with both logic and visuals?	• Variables allow programmers to store, update, and use data flexibly within their programs. • User input makes programs interactive and responsive to individual users. • Arithmetic expressions enable calculations that drive dynamic and meaningful program behavior. • Graphics in JavaScript provide a way to visually communicate ideas, patterns, and designs. • Programming requires breaking problems into smaller, manageable parts and combining logic with creativity to produce engaging results.
Vocabulary: variables, user input, arithmetic expressions, graphics, canvas, rectangle, syntax, data, program, code, interactive, logic, expression, operator, value, string, number, function, statement, output	
Students will be able to: • Write a JavaScript program by typing commands with proper syntax in the start function • Write a program that prints out a message to the user • Explain what variables are and what they are used for • Create their own variables • Print out the values stored in variables • Create programs that ask the user for input • Store user input in variables and print it back to the user • Choose the proper input function to use depending on the type of information needed • Describe the different mathematical operators we can use in programs • Create programs that use basic math to compute useful things • Create programs that take in user input, do simple computations with the input, and produce useful output • Create graphical JavaScript programs that draw shapes on the canvas	

- Locate points on the graphics canvas using (x, y) coordinates

Learning Activities & Differentiation: *(Amistad & Holocaust, Asian American & Pacific Islanders, Climate Change, LGBTQ and People with Disabilities, Diversity, Equity & Inclusion [DEI])*

Learning Activities:

- Variables in Action: Students create programs that store and display information using variables (e.g., favorite foods, hobbies, or cultural traditions), promoting diverse personal expression.
- Interactive Programs with User Input: Students design input-driven programs, such as a Dinner Plans exercise, that personalize interactions and highlight respect for different perspectives and lifestyles.
- Arithmetic Expressions for Real-World Contexts: Students apply mathematical operators to solve problems, such as calculating carbon footprints, promoting awareness of climate change and sustainable practices.
- Graphics on the Canvas: Students build visual programs such as drawing national flags, cultural symbols, or awareness posters for social causes, fostering respect for global diversity and inclusion.
- Debugging and Code Collaboration: Students work in pairs or groups to troubleshoot and improve each other's programs, emphasizing collaboration, empathy, and persistence in problem-solving.
- Commenting and Code Readability: Students practice writing clear comments and formatting their code for readability, reinforcing the importance of accessible and inclusive communication.

Differentiation Strategies:

- For Advanced Learners: Assign multi-step projects combining user input, arithmetic, and graphics (e.g., interactive calculators or custom flag generators). Encourage extensions such as using loops or custom functions.
- For Struggling Learners: Provide starter code and guided practice focusing on one concept at a time (variables, input, or graphics). Use visual organizers like pseudocode charts to support sequencing.
- For English Language Learners: Offer bilingual glossaries, step-by-step annotated code, and visual guides of the canvas coordinate system. Incorporate culturally relevant examples to build engagement.
- Universal Design for Learning (UDL): Deliver instruction in multiple formats (live coding, video demos, diagrams). Allow students to demonstrate mastery through different mediums (code, flowcharts, presentations). Provide

checkpoints and peer feedback opportunities.

Education Connections:

- Highlight how variables and user input mirror real-world data collection, from surveys to apps that adapt to users' needs.
- Show how graphics can be used to raise awareness of important issues (e.g., climate change infographics, pride symbols, cultural flags).
- Connect debugging and problem-solving to resilience and persistence, skills essential in both computer science and everyday life.
- Discuss the ethical importance of clear, accessible code that can be shared, read, and built upon by diverse communities.
- Explore how computational thinking empowers students to design inclusive, meaningful digital experiences that reflect and respect diverse perspectives.

Assessment Evidence

Formative Assessments:

- CodeHS exercises
    - Example Exercise: Dinner Plans

      Prompt the user for their name, then ask them what time you should meet for dinner.

      Greet them by name and tell them you will meet them at the time they specified!
    - Example Exercise: Flag of the Netherlands

      This program should draw the flag for the Netherlands. The top third of the canvas is red, the middle third is white, and the bottom third is blue, as shown below. You will need to use Rectangle objects in this program.
- CodeHS Checks for Understanding

Summative Assessments:

- Graphics Challenges

Common Benchmark Assessments:

- Graphics Challenges

Alternative Assessments:

- CodeHS Unit Quiz

Core Instructional & Supplemental Materials

Core Materials:

- CodeHS JavaScript and Graphics module
- Built-in CodeHS JavaScript Editor with live execution environment
- JavaScript syntax and variable reference guides
- CodeHS tutorials on variables, user input, arithmetic expressions, and graphics
- CodeHS lessons on debugging, commenting, and program readability
- Canvas coordinate system guides and shape-drawing tutorials

Supplemental Materials:

- Sample JavaScript code templates for variables, input, arithmetic, and graphics
- Visual guides illustrating the canvas grid, (x, y) coordinates, and shape placement
- Problem decomposition organizers for planning interactive and graphics-based programs
- Debugging checklists and peer collaboration rubrics
- Example graphics challenge solutions for inspiration (e.g., Flag of the Netherlands, custom interactive messages)
- Readability and commenting checklists to support clear, maintainable code
- Rubrics for evaluating programming projects, including correctness, creativity, and clarity
- Glossaries of key programming terms for English Language Learners

Accommodations and Modifications:
Students with Disabilities, English Language Learners,

Students at Risk of Failure, Students with 504s, Gifted & Talented Students

Specific Strategies and Practices that Support Students with Disabilities:

Students receiving Special Education programming have specific goals and objectives, as well as accommodations and modifications outlined within their Individualized Education Plans (IEP) due to an identified disability and/or diagnosis. In addition to exposure to the general education curriculum, the instruction is differentiated based upon the student's needs. The IEP acts as a supplemental curriculum guide inclusive of instructional strategies that support each learner.

- Small group/One to one - Additional time - Review of directions - Extra visual and verbal cues and prompts - Checklists - Immediate feedback

Specific Strategies and Practices that Support English Language Learners

English language learner (ELL) is a national-origin-minority student who is limited-English-proficient. This term is often preferred over limited-English-proficient (LEP) as it highlights accomplishments rather than deficits.

- Pre-teaching of vocabulary and concepts - Personal Glossary - Simplified/Verbal Instructions - Visual learning, including graphic organizers - Teacher modeling - Scaffolding - Think-pair-share - Cooperative learning groups

Students at Risk of Failure

- Alternative Assessments - Supplemental Instruction - Think-Pair-Share by Readiness - Homework Options - Differentiated Online Practice - Tiered Activities - Small Group Instruction - Group Work Choice

Students with 504 Plans

Teachers are responsible for implementing designated services and strategies identified on a student's 504 Plan.
504 plans are formal plans that schools develop to give kids the support they need and covers any condition that limits daily activities in a major way. Students' accommodations are outlined in their individual 504 plan.

Specific Strategies and Practices that Support Gifted & Talented Students

It is critical that all teachers are able to recognize a high-ability student who may need more depth and complexity in instruction.

-Small Group In-Depth Instruction - Differentiated Online Practice - Alternative Assessments

Unit 4

Desired Results

Unit 4 Title: Intro to Mobile Apps & Components and Stylesheet

Unit Summary:

In this unit, students are introduced to the fundamentals of mobile app development using React Native and the Expo platform. The unit begins with an overview of the course structure, program syntax, and the essential tools needed to build and run apps. Students explore why mobile apps are important, how React Native is structured, and how Expo enables app testing and execution. The unit then transitions to a focus on components—the building blocks of React Native applications—and the Stylesheet API used for customizing app appearance. Students learn how to import, debug, and structure components, as well as how to style both View and Text components using layout properties like justification, alignment, and flex direction.

Learning Standards

NJSLS:

- 8.1.12.CS.1: Describe ways in which integrated systems hide underlying implementation details to simplify user experiences.
- 8.1.12.CS.2: Model interactions between application software, system software, and hardware.
- 8.1.12.CS.3: Compare the functions of application software, system software, and hardware.
- 8.1.12.CS.4: Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.
- 8.1.12.IC.1: Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.
- 8.1.12.IC.2: Test and refine computational artifacts to reduce bias and equity deficits.
- 8.1.12.IC.3: Predict the potential impacts and implications of emerging technologies on larger social, economic, and political structures, using evidence from credible sources.
- 8.1.12.AP.1: Design algorithms to solve computational problems using a combination of original and existing algorithms.

- 8.1.12.AP.3: Select and combine control structures for a specific application based upon performance and readability, and identify trade-offs to justify the choice.
- 8.1.12.AP.4: Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue.
- 8.1.12.AP.5: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
- 8.1.12.AP.6: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.
- 8.1.12.AP.7: Collaboratively design and develop programs and artifacts for broad audiences by incorporating feedback from users.
- 8.1.12.AP.8: Evaluate and refine computational artifacts to make them more usable and accessible.
- 8.1.12.AP.9: Collaboratively document and present design decisions in the development of complex programs.

Interdisciplinary Connections

Career Readiness, Life Literacies, & Key Skills *(CLKS)*:

- 9.4.12.CI.1: Demonstrate the ability to reflect, analyze, and use creative skills and ideas (e.g., 1.1.12prof.CR3a).
- 9.4.12.CI.2: Identify career pathways that highlight personal talents, skills, and abilities (e.g., 1.4.12prof.CR2b, 2.2.12.LF.8).
- 9.4.12.CI.3: Investigate new challenges and opportunities for personal growth, advancement, and transition (e.g., 2.1.12.PGD.1).
- 9.4.12.CT.1: Identify problem-solving strategies used in the development of an innovative product or practice (e.g., 1.1.12acc.C1b, 2.2.12.PF.3).
- 9.4.12.CT.2: Explain the potential benefits of collaborating to enhance critical thinking and problem solving (e.g., 1.3E.12profCR3.a).
- 9.4.12.CT.3: Enlist input from a variety of stakeholders (e.g., community members, experts in the field) to design a service learning activity that addresses a local or global issue (e.g., environmental justice).
- 9.4.12.CT.4: Participate in online strategy and planning sessions for course-based, school-based, or other project and determine the strategies that contribute to effective outcomes.

- 9.4.12.DC.1: Explain the beneficial and harmful effects that intellectual property laws can have on the creation and sharing of content (e.g., 6.1.12.CivicsPR.16.a).
- 9.4.12.DC.2: Compare and contrast international differences in copyright laws and ethics.
- 9.4.12.DC.3: Evaluate the social and economic implications of privacy in the context of safety, law, or ethics (e.g., 6.3.12.HistoryCA.1).
- 9.4.12.DC.4: Explain the privacy concerns related to the collection of data (e.g., cookies) and generation of data through automated processes that may not be evident to users (e.g., 8.1.12.NI.3).
- 9.4.12.DC.5: Debate laws and regulations that impact the development and use of software.
- 9.4.12.DC.6: Select information to post online that positively impacts personal image and future college and career opportunities.
- 9.4.12.DC.7: Evaluate the influence of digital communities on the nature, content and responsibilities of careers, and other aspects of society (e.g., 6.1.12.CivicsPD.16.a).
- 9.4.12.DC.8: Explain how increased network connectivity and computing capabilities of everyday objects allow for innovative technological approaches to climate protection.
- 9.4.12.GCA.1: Collaborate with individuals to analyze a variety of potential solutions to climate change effects and determine why some solutions (e.g., political. economic, cultural) may work better than others (e.g., SL.11-12.1., HS-ETS1-1, HS-ETS1-2, HS-ETS1-4, 6.3.12.GeoGI.1, 7.1.IH.IPERS.6, 7.1.IL.IPERS.7, 8.2.12.ETW.3).
- 9.4.12.IML.1: Compare search browsers and recognize features that allow for filtering of information.
- 9.4.12.IML.2: Evaluate digital sources for timeliness, accuracy, perspective, credibility of the source, and relevance of information, in media, data, or other resources (e.g., NJSLSA.W8, SocialStudies Practice: Gathering and Evaluating Sources).
- 9.4.12.IML.3: Analyze data using tools and models to make valid and reliable claims, or to determine optimal design solutions (e.g., S-ID.B.6a., 8.1.12.DA.5, 7.1.IH.IPRET.8).
- 9.4.12.IML.4: Assess and critique the appropriateness and impact of existing data visualizations for an intended audience (e.g., S-ID.B.6b, HS-LS2-4).
- 9.4.12.IML.5: Evaluate, synthesize, and apply information on climate change from various sources appropriately (e.g., 2.1.12.CHSS.6, S.IC.B.4, S.IC.B.6, 8.1.12.DA.1, 6.1.12.GeoHE.14.a, 7.1.AL.PRSNT.2).

- 9.4.12.IML.6: Use various types of media to produce and store information on climate change for different purposes and audiences with sensitivity to cultural, gender, and age diversity (e.g., NJSLSA.SL5).
- 9.4.12.IML.7: Develop an argument to support a claim regarding a current workplace or societal/ethical issue such as climate change (e.g., NJSLSA.W1, 7.1.AL.PRSNT.4).
- 9.4.12.IML.8: Evaluate media sources for point of view, bias, and motivations (e.g., NJSLSA.R6, 7.1.AL.IPRET.6).
- 9.4.12.IML.9: Analyze the decisions creators make to reveal explicit and implicit messages within information and media (e.g., 1.5.12acc.C2a, 7.1.IL.IPRET.4).
- 9.4.12.TL.1: Assess digital tools based on features such as accessibility options, capacities, and utility for accomplishing a specified task (e.g., W.11-12.6.).
- 9.4.12.TL.2: Generate data using formula-based calculations in a spreadsheet and draw conclusions about the data.
- 9.4.12.TL.3: Analyze the effectiveness of the process and quality of collaborative environments.
- 9.4.12.TL.4: Collaborate in online learning communities or social networks or virtual worlds to analyze and propose a resolution to a real-world problem (e.g., 7.1.AL.IPERS.6).

Interdisciplinary Connections:

Science and Engineering Practices:

- Develop and use a model based on evidence to illustrate the relationships between systems or between components of a system.
- Create a computational model or simulation of a phenomenon, designed device, process, or system.
- Design, evaluate, and/or refine a solution to a complex real-world problem, based on scientific knowledge, student-generated sources of evidence, prioritized criteria, and tradeoff considerations.
- Communicate technical information or ideas (e.g. about phenomena and/or the process of development and the design and performance of a proposed process or system) in multiple formats (including orally, graphically, textually, and mathematically).
- Use mathematical and/or computational representations of phenomena or design solutions to support explanations.

- Design or refine a solution to a complex real-world problem, based on scientific knowledge, student-generated sources of evidence, prioritized criteria, and tradeoff considerations.
- Evaluate a solution to a complex real-world problem, based on scientific knowledge, student-generated sources of evidence, prioritized criteria, and tradeoff considerations.

Engineering Design Standards:

- HS-ETS1-2 Design a solution to a complex real-world problem by breaking it down into smaller, more manageable problems that can be solved through engineering.
- HS-ETS1-3 Evaluate a solution to a complex real-world problem based on prioritized criteria and trade-offs that account for a range of constraints, including cost, safety, reliability, and aesthetics, as well as possible social, cultural, and environmental impacts.
- HS-ETS1-4 Use a computer simulation to model the impact of proposed solutions to a complex real-world problem with numerous criteria and constraints on interactions within and between systems relevant to the problem.

Learning Targets

Unit Essential Questions:

- Mobile applications are a primary way people interact with technology, making their design and usability highly impactful.
- React Native allows developers to build mobile apps using a component-based structure that promotes efficiency and reusability.
- The Expo platform provides tools that simplify the process of running and testing mobile applications.

Unit Enduring Understandings:

- Why are mobile apps important in today's digital world, and how do they influence everyday life?
- How does the structure of a React Native program support the development of mobile apps?
- What role does Expo play in the creation and testing of mobile applications?
- How can the Stylesheet API be used to improve the design and usability of an app?

• Styling with the Stylesheet API is essential for creating visually appealing and user-friendly app interfaces. • Debugging and refining components are critical skills in ensuring that mobile apps function as intended.	• Why is debugging an essential step in mobile app development, and what strategies can help identify and fix issues?
Vocabulary: react native, expo, component, stylesheet, api, view, text, justify, align, flex direction, syntax, debug, editor, attribute, nested views, user interface, mobile app, program structure, layout, customization	
Students will be able to: • Articulate what React Native is, what it is used for, and the benefits of the framework • Download and run the Expo app on a device in order to physically view apps as they create them throughout the course • Create a basic app using view and text components • Nest child components inside a parent view component • Define style rules using a stylesheet • Apply a specific style rule to a particular component • Position different components relative to each other • Control the size and position of components • Style a Text component's font size, weight, and color using the Stylesheet API	
Learning Activities & Differentiation: *(Amistad & Holocaust, Asian American & Pacific Islanders, Climate Change, LGBTQ and People with Disabilities, Diversity, Equity & Inclusion [DEI])* **Learning Activities:** • Components in Action: Students build simple apps with View and Text components that share personal or cultural stories, highlight environmental issues, or celebrate diversity (e.g., a "My Cultural Favorites" app). • Nested Components Project: Students nest child components within parent Views to create layouts such as digital	

posters for social causes (e.g., Holocaust remembrance, AAPI heritage, LGBTQ Pride, disability awareness).
- Styling for Impact: Students use the Stylesheet API to design accessible, inclusive interfaces, experimenting with font size, weight, and color to ensure readability for all users.
- Designing for Inclusion: Students position and style components to build awareness tools like carbon footprint trackers, news articles, or cultural information cards.
- Collaborative Debugging: Students pair up to debug each other's apps, promoting collaboration, empathy, and shared problem-solving.

Differentiation Strategies:

- For Advanced Learners: Assign multi-screen app projects that combine nested components, custom styles, and layouts (e.g., an interactive awareness app or a digital cultural showcase). Encourage the use of advanced styling attributes and flexbox alignment.
- For Struggling Learners: Provide starter code with partially built components. Focus on one skill at a time (running Expo, nesting Views, styling Text). Use color-coded pseudocode or visual flowcharts to guide understanding.
- For English Language Learners: Provide bilingual glossaries of React Native and styling terms. Include annotated code samples and visual guides showing how components are positioned. Encourage cultural connections by allowing students to design apps that reflect their backgrounds.
- Universal Design for Learning (UDL): Present lessons through live coding, annotated videos, and visual diagrams of app structure. Allow multiple means of expression: students may demonstrate learning through functioning code, visual mockups, or presentations explaining their design choices. Provide checkpoints and structured peer review opportunities.

Education Connections:

- Highlight how React Native apps mirror real-world digital tools that adapt to user needs, from accessibility features to cultural resources.
- Connect styling and design choices to inclusive communication, ensuring apps are accessible and welcoming to diverse audiences.

- Show how Expo testing reflects real-world app development, where collaboration and adaptation are essential.
- Emphasize that debugging and persistence are transferable life skills, supporting problem-solving and resilience.
- Explore how computational thinking and mobile app design empower students to create meaningful digital experiences that respect diversity, equity, and inclusion.

Assessment Evidence

Formative Assessments:

- CodeHS exercises
  - Example activity: Connecting Components to the Stylesheet
  - Example activity: Nested Views
- CodeHS Checks for Understanding

Summative Assessments:

- Challenge: Scorekeeper Pt 1

Common Benchmark Assessments:

- Challenge: Scorekeeper Pt 1

Alternative Assessments:

- CodeHS Unit Quiz

Core Instructional & Supplemental Materials

Core Materials:

- CodeHS Introduction to Mobile Apps modules on React Native and Expo
- Built-in CodeHS React Native Editor with live mobile preview environment
- React Native syntax and component reference guides
- CodeHS tutorials on components, the Stylesheet API, and styling techniques

- CodeHS lessons on debugging components, customizing text, and program readability
- Guides to using the Stylesheet API for layout, alignment, and flex direction
- Expo documentation for running and testing apps on devices

Supplemental Materials:

- Sample React Native code templates for View, Text, and nested components
- Visual guides illustrating component hierarchy, parent-child nesting, and flexbox layout
- Organizers for planning app design, including wireframes and style rule charts
- Debugging checklists and peer collaboration rubrics for component troubleshooting
- Example styling challenge solutions for inspiration (e.g., News Article layout, Favorites List app)
- Accessibility and readability checklists to ensure inclusive and clear design choices
- Rubrics for evaluating app projects, including correctness, creativity, usability, and design clarity
- Bilingual glossaries of key React Native and styling terms to support English Language Learners

Accommodations and Modifications: Students with Disabilities, English Language Learners, Students at Risk of Failure, Students with 504s, Gifted & Talented Students

Specific Strategies and Practices that Support Students with Disabilities:

Students receiving Special Education programming have specific goals and objectives, as well as accommodations and modifications outlined within their Individualized Education Plans (IEP) due to an identified disability and/or diagnosis. In addition to exposure to the general education curriculum, the instruction is differentiated based upon the student's needs. The IEP acts as a supplemental curriculum guide inclusive of instructional strategies that support each learner.

- Additional time - Review of directions - Space for movement or breaks - Extra visual and verbal cues and prompts - Preferential seating - Rest breaks - Verbal and visual cues regarding directions and staying on task - Checklists - Immediate feedback

Specific Strategies and Practices that Support English Language Learners

English language learner (ELL) is a national-origin-minority student who is limited-English-proficient. This term is often preferred over limited-English-proficient (LEP) as it highlights accomplishments rather than deficits.

- Extended Time - Simplified/Verbal Instructions - Frequent Breaks - Visual learning, including graphic organizers - Teacher modeling - Scaffolding - Think-pair-share - Cooperative learning groups

Students at Risk of Failure

- Think-Pair-Share by Readiness - Homework Options - Differentiated Online Practice - Leveled Rubrics - Tiered Level of Supports - Small Group Instruction

Students with 504 Plans

Teachers are responsible for implementing designated services and strategies identified on a student's 504 Plan.
504 plans are formal plans that schools develop to give kids the support they need and covers any condition that limits daily activities in a major way. Students' accommodations are outlined in their individual 504 plan.

Specific Strategies and Practices that Support Gifted & Talented Students

It is critical that all teachers are able to recognize a high-ability student who may need more depth and complexity in instruction.

-Homogenous Grouping - Small Group In-Depth Instruction

Unit 5

Desired Results

Unit 5 Title: Buttons and Text Boxes

Unit Summary:

In this unit, students learn to create interactive buttons using TouchableHighlight and handle taps with the onPress function. They also explore the TextInput component to allow user text entry. Activities include building a homepage, a search page, and a feedback form.

Learning Standards

NJSLS:

- 8.1.12.CS.1: Describe ways in which integrated systems hide underlying implementation details to simplify user

experiences.

- 8.1.12.CS.2: Model interactions between application software, system software, and hardware.
- 8.1.12.CS.3: Compare the functions of application software, system software, and hardware.
- 8.1.12.CS.4: Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.
- 8.1.12.IC.2: Test and refine computational artifacts to reduce bias and equity deficits.
- 8.1.12.AP.3: Select and combine control structures for a specific application based upon performance and readability, and identify trade-offs to justify the choice.
- 8.1.12.AP.4: Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue.
- 8.1.12.AP.5: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
- 8.1.12.AP.6: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.
- 8.1.12.AP.7: Collaboratively design and develop programs and artifacts for broad audiences by incorporating feedback from users.
- 8.1.12.AP.8: Evaluate and refine computational artifacts to make them more usable and accessible.

Interdisciplinary Connections

Career Readiness, Life Literacies, & Key Skills *(CLKS)*:

- 9.4.12.CI.1: Demonstrate the ability to reflect, analyze, and use creative skills and ideas (e.g., 1.1.12prof.CR3a).
- 9.4.12.CI.2: Identify career pathways that highlight personal talents, skills, and abilities (e.g., 1.4.12prof.CR2b, 2.2.12.LF.8).
- 9.4.12.CI.3: Investigate new challenges and opportunities for personal growth, advancement, and transition (e.g., 2.1.12.PGD.1).
- 9.4.12.CT.1: Identify problem-solving strategies used in the development of an innovative product or practice (e.g., 1.1.12acc.C1b, 2.2.12.PF.3).

- 9.4.12.CT.2: Explain the potential benefits of collaborating to enhance critical thinking and problem solving (e.g., 1.3E.12profCR3.a).
- 9.4.12.CT.3: Enlist input from a variety of stakeholders (e.g., community members, experts in the field) to design a service learning activity that addresses a local or global issue (e.g., environmental justice).
- 9.4.12.CT.4: Participate in online strategy and planning sessions for course-based, school-based, or other project and determine the strategies that contribute to effective outcomes.
- 9.4.12.DC.1: Explain the beneficial and harmful effects that intellectual property laws can have on the creation and sharing of content (e.g., 6.1.12.CivicsPR.16.a).
- 9.4.12.DC.2: Compare and contrast international differences in copyright laws and ethics.
- 9.4.12.DC.3: Evaluate the social and economic implications of privacy in the context of safety, law, or ethics (e.g., 6.3.12.HistoryCA.1).
- 9.4.12.DC.4: Explain the privacy concerns related to the collection of data (e.g., cookies) and generation of data through automated processes that may not be evident to users (e.g., 8.1.12.NI.3).
- 9.4.12.DC.5: Debate laws and regulations that impact the development and use of software.
- 9.4.12.DC.6: Select information to post online that positively impacts personal image and future college and career opportunities.
- 9.4.12.DC.7: Evaluate the influence of digital communities on the nature, content and responsibilities of careers, and other aspects of society (e.g., 6.1.12.CivicsPD.16.a).
- 9.4.12.DC.8: Explain how increased network connectivity and computing capabilities of everyday objects allow for innovative technological approaches to climate protection.
- 9.4.12.GCA.1: Collaborate with individuals to analyze a variety of potential solutions to climate change effects and determine why some solutions (e.g., political, economic, cultural) may work better than others (e.g., SL.11-12.1., HS-ETS1-1, HS-ETS1-2, HS-ETS1-4, 6.3.12.GeoGI.1, 7.1.IH.IPERS.6, 7.1.IL.IPERS.7, 8.2.12.ETW.3).
- 9.4.12.IML.1: Compare search browsers and recognize features that allow for filtering of information.
- 9.4.12.IML.2: Evaluate digital sources for timeliness, accuracy, perspective, credibility of the source, and relevance of information, in media, data, or other resources (e.g., NJSLSA.W8, SocialStudies Practice: Gathering and Evaluating Sources).

- 9.4.12.IML.3: Analyze data using tools and models to make valid and reliable claims, or to determine optimal design solutions (e.g., S-ID.B.6a., 8.1.12.DA.5, 7.1.IH.IPRET.8).
- 9.4.12.IML.4: Assess and critique the appropriateness and impact of existing data visualizations for an intended audience (e.g., S-ID.B.6b, HS-LS2-4).
- 9.4.12.IML.5: Evaluate, synthesize, and apply information on climate change from various sources appropriately (e.g., 2.1.12.CHSS.6, S.IC.B.4, S.IC.B.6, 8.1.12.DA.1, 6.1.12.GeoHE.14.a, 7.1.AL.PRSNT.2).
- 9.4.12.IML.6: Use various types of media to produce and store information on climate change for different purposes and audiences with sensitivity to cultural, gender, and age diversity (e.g., NJSLSA.SL5).
- 9.4.12.IML.7: Develop an argument to support a claim regarding a current workplace or societal/ethical issue such as climate change (e.g., NJSLSA.W1, 7.1.AL.PRSNT.4).
- 9.4.12.IML.8: Evaluate media sources for point of view, bias, and motivations (e.g., NJSLSA.R6, 7.1.AL.IPRET.6).
- 9.4.12.IML.9: Analyze the decisions creators make to reveal explicit and implicit messages within information and media (e.g., 1.5.12acc.C2a, 7.1.IL.IPRET.4).
- 9.4.12.TL.1: Assess digital tools based on features such as accessibility options, capacities, and utility for accomplishing a specified task (e.g., W.11-12.6.).
- 9.4.12.TL.2: Generate data using formula-based calculations in a spreadsheet and draw conclusions about the data.
- 9.4.12.TL.3: Analyze the effectiveness of the process and quality of collaborative environments.
- 9.4.12.TL.4: Collaborate in online learning communities or social networks or virtual worlds to analyze and propose a resolution to a real-world problem (e.g., 7.1.AL.IPERS.6).

Interdisciplinary Connections:

Science and Engineering Practices:

- Develop a model based on evidence to illustrate the relationships between systems or between components of a system.
- Use a model to predict the relationships between systems or between components of a system.
- Construct and revise an explanation based on valid and reliable evidence obtained from a variety of sources (including students' own investigations, models, theories, simulations, peer review) and the assumption that

theories and laws that describe the natural world operate today as they did in the past and will continue to do so in the future.

- Analyze data using tools, technologies, and/or models (e.g., computational, mathematical) in order to make valid and reliable scientific claims or determine an optimal design solution.
- Use mathematical and/or computational representations of phenomena or design solutions to support explanations.
- Apply scientific ideas to solve a design problem, taking into account possible unanticipated effects.
- Communicate scientific and technical information (e.g. about the process of development and the design and performance of a proposed process or system) in multiple formats (including orally, graphically, textually, and mathematically).
- Design, evaluate, and/or refine a solution to a complex real-world problem, based on scientific knowledge, student-generated sources of evidence, prioritized criteria, and tradeoff considerations.
- Evaluate a solution to a complex real-world problem, based on scientific knowledge, student-generated sources of evidence, prioritized criteria, and tradeoff considerations.
- Create a computational model or simulation of a phenomenon, designed device, process, or system.

Engineering Design Standards:

- HS-ETS1-2 Design a solution to a complex real-world problem by breaking it down into smaller, more manageable problems that can be solved through engineering.
- HS-ETS1-3 Evaluate a solution to a complex real-world problem based on prioritized criteria and trade-offs that account for a range of constraints, including cost, safety, reliability, and aesthetics, as well as possible social, cultural, and environmental impacts.
- HS-ETS1-4 Use a computer simulation to model the impact of proposed solutions to a complex real-world problem with numerous criteria and constraints on interactions within and between systems relevant to the problem.

Learning Targets

Unit Essential Questions:	**Unit Enduring Understandings:**
• How do buttons enhance user interaction in an app? • What is the role of the TouchableHighlight component in creating buttons? • How does the onPress function enable functionality for buttons? • How can text input be captured and used within an app? • Why is user input important for interactive applications?	• Buttons are key elements that enable users to interact with and control app behavior. • The TouchableHighlight component provides a way to create customizable, tappable buttons. • The onPress function connects button taps to specific actions in the app. • Text input components allow users to provide information that the app can process and respond to. • Collecting user input is essential for creating dynamic, personalized app experiences.
Vocabulary: touchablehighlight, onpress function, button, textinput, user input, interactive component, tap event, input field, event handler, feedback form, homepage, search page	
Students will be able to: • create an alert when a particular component is pressed • make any component behave like a button • use the TextInput component to allow users to enter text into their apps • set state values that will control the initial values in their text boxes • concatenate strings with state objects to personalize alert messages	
Learning Activities & Differentiation: *(Amistad & Holocaust, Asian American & Pacific Islanders, Climate Change, LGBTQ and People with Disabilities, Diversity, Equity & Inclusion [DEI])* **Learning Activities:** • TouchableHighlight Buttons in Action: Students build simple apps using TouchableHighlight components as	

buttons to navigate or trigger alerts. For example, creating a Homepage that highlights cultural heritage or social justice topics, such as AAPI history or climate change awareness.

- Exploring the onPress Function: Through hands-on labs, students implement the onPress function to activate button actions that share important information or prompt reflection on DEI themes.
- Using TextInput for User Interaction: Students design apps like a Google Search Page or Feedback Form where users enter text. These inputs can gather opinions on topics like LGBTQ inclusion or disability access.
- Personalizing Alerts with State and TextInput: Students practice setting and updating state values to personalize alert messages based on user input, reinforcing how technology can create tailored, inclusive experiences.
- Debugging & Peer Collaboration: Students pair up to troubleshoot code, sharing diverse perspectives and supporting each other's understanding of interactive components.

Differentiation Strategies:

- Advanced Learners: Create multi-screen apps combining TouchableHighlight buttons with TextInput forms that dynamically change content based on user input, focusing on interactive awareness campaigns or cultural showcases.
- Struggling Learners: Use partially completed starter apps focusing on one core skill at a time (e.g., adding a TouchableHighlight button or TextInput field). Provide visual flowcharts and stepwise pseudocode.
- English Language Learners (ELLs): Provide annotated examples and bilingual glossaries for key terms like "TouchableHighlight," "onPress," and "TextInput." Encourage apps that reflect students' own cultural or social experiences.
- Universal Design for Learning (UDL): Combine live coding demonstrations, visual app component maps, and interactive labs. Offer multiple ways to show learning—working apps, annotated screenshots, or brief presentations.

Education Connections:

- Show how TouchableHighlight buttons and TextInput fields reflect real-world app features that promote accessible communication and user engagement.

- Connect app development with social awareness by building apps focused on DEI, cultural heritage, and environmental issues.
- Emphasize debugging and collaboration as key skills that foster empathy and teamwork in technology and beyond.
- Illustrate how personalized user interactions in apps can amplify diverse voices and perspectives.

Assessment Evidence

Formative Assessments:

- CodeHS exercises
  - Example activity: Build a Homepage
  - Example activity: Google Search Page
- CodeHS Checks for Understanding

Summative Assessments:

- Challenge: Scorekeeper Pt 2

Common Benchmark Assessments:

- Challenge: Scorekeeper Pt 2

Alternative Assessments:

- CodeHS Unit Quiz

Core Instructional & Supplemental Materials

Core Materials:

- Built-in CodeHS React Native Editor with live mobile preview environment
- React Native syntax and component reference guides

- CodeHS tutorials on components like TouchableHighlight and TextInput, plus the Stylesheet API and styling techniques
- CodeHS lessons on debugging interactive components, customizing text inputs, and improving program readability
- Guides to using the Stylesheet API for layout, alignment, and flex direction
- Expo documentation for running and testing apps on physical devices

Supplemental Materials:

- Sample React Native code templates for TouchableHighlight buttons, TextInput fields, and simple app pages (e.g., Homepage, Google Search, Feedback Form)
- Visual guides illustrating component hierarchy, parent-child nesting, and flexbox layout principles
- Organizers for planning app designs, including wireframes and style rule charts focusing on buttons and text input layouts
- Debugging checklists and peer collaboration rubrics for troubleshooting interactive components
- Example styling challenge solutions (e.g., button styles, input form layouts) for student inspiration
- Accessibility and readability checklists to ensure inclusive, clear, and user-friendly interface designs
- Rubrics for evaluating app projects on correctness, creativity, usability, and design clarity
- Bilingual glossaries of key React Native, TouchableHighlight, TextInput, and styling terms to support English Language Learners

Accommodations and Modifications: Students with Disabilities, English Language Learners, Students at Risk of Failure, Students with 504s, Gifted & Talented Students

Specific Strategies and Practices that Support Students with Disabilities:

Students receiving Special Education programming have specific goals and objectives, as well as accommodations and modifications outlined within their Individualized Education Plans (IEP) due to an identified disability and/or diagnosis. In addition to exposure to the general education curriculum, the instruction is differentiated based upon the student's needs. The IEP acts as a supplemental curriculum guide inclusive of instructional strategies that support each learner.

- Review of directions - Extra visual and verbal cues and prompts - Preferential seating - Follow a routine/schedule - Rest breaks - Verbal and visual cues regarding directions and staying on task - Checklists - Immediate feedback

Specific Strategies and Practices that Support English Language Learners

English language learner (ELL) is a national-origin-minority student who is limited-English-proficient. This term is often preferred over limited-English-proficient (LEP) as it highlights accomplishments rather than deficits.

- Teacher modeling - Pairing students with beginning English language skills with students who have more advanced English language skills - Scaffolding - Think-pair-share

Students at Risk of Failure

- Learning Contracts - Supplemental Instruction - Think-Pair-Share by Readiness - Homework Options - Differentiated Online Practice

Students with 504 Plans

Teachers are responsible for implementing designated services and strategies identified on a student's 504 Plan.
504 plans are formal plans that schools develop to give kids the support they need and covers any condition that limits daily activities in a major way. Students' accommodations are outlined in their individual 504 plan.

Specific Strategies and Practices that Support Gifted & Talented Students

It is critical that all teachers are able to recognize a high-ability student who may need more depth and complexity in instruction.

-Homogenous Grouping - Alternative Assessments

Unit 6

Desired Results

Unit 6 Title: Advanced Layouts and Images

Unit Summary:

In this unit, students deepen their skills in mobile app development by learning to integrate images and advanced layout techniques. They explore the use of the Image and ImageBackground components to customize app visuals, while mastering flexible layouts with Flex values and the Dimensions API to create responsive designs that adapt to various screen sizes.

Learning Standards

NJSLS:

- 8.1.12.CS.1: Describe ways in which integrated systems hide underlying implementation details to simplify user experiences.
- 8.1.12.CS.2: Model interactions between application software, system software, and hardware.
- 8.1.12.CS.3: Compare the functions of application software, system software, and hardware.
- 8.1.12.CS.4: Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.
- 8.1.12.NI.2: Evaluate security measures to address various common security threats.
- 8.1.12.IC.1: Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.
- 8.1.12.IC.2: Test and refine computational artifacts to reduce bias and equity deficits.
- 8.1.12.AP.3: Select and combine control structures for a specific application based upon performance and readability, and identify trade-offs to justify the choice.
- 8.1.12.AP.4: Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue.
- 8.1.12.AP.5: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
- 8.1.12.AP.6: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.
- 8.1.12.AP.7: Collaboratively design and develop programs and artifacts for broad audiences by incorporating

feedback from users.

- 8.1.12.AP.8: Evaluate and refine computational artifacts to make them more usable and accessible.
- 8.1.12.AP.9: Collaboratively document and present design decisions in the development of complex programs.

Interdisciplinary Connections

Career Readiness, Life Literacies, & Key Skills *(CLKS)*:

- 9.4.12.CI.1: Demonstrate the ability to reflect, analyze, and use creative skills and ideas (e.g., 1.1.12prof.CR3a).
- 9.4.12.CI.2: Identify career pathways that highlight personal talents, skills, and abilities (e.g., 1.4.12prof.CR2b, 2.2.12.LF.8).
- 9.4.12.CI.3: Investigate new challenges and opportunities for personal growth, advancement, and transition (e.g., 2.1.12.PGD.1).
- 9.4.12.CT.1: Identify problem-solving strategies used in the development of an innovative product or practice (e.g., 1.1.12acc.C1b, 2.2.12.PF.3).
- 9.4.12.CT.2: Explain the potential benefits of collaborating to enhance critical thinking and problem solving (e.g., 1.3E.12profCR3.a).
- 9.4.12.CT.3: Enlist input from a variety of stakeholders (e.g., community members, experts in the field) to design a service learning activity that addresses a local or global issue (e.g., environmental justice).
- 9.4.12.CT.4: Participate in online strategy and planning sessions for course-based, school-based, or other project and determine the strategies that contribute to effective outcomes.
- 9.4.12.DC.1: Explain the beneficial and harmful effects that intellectual property laws can have on the creation and sharing of content (e.g., 6.1.12.CivicsPR.16.a).
- 9.4.12.DC.2: Compare and contrast international differences in copyright laws and ethics.
- 9.4.12.DC.3: Evaluate the social and economic implications of privacy in the context of safety, law, or ethics (e.g., 6.3.12.HistoryCA.1).
- 9.4.12.DC.4: Explain the privacy concerns related to the collection of data (e.g., cookies) and generation of data through automated processes that may not be evident to users (e.g., 8.1.12.NI.3).
- 9.4.12.DC.5: Debate laws and regulations that impact the development and use of software.
- 9.4.12.DC.6: Select information to post online that positively impacts personal image and future college and career opportunities.

- 9.4.12.DC.7: Evaluate the influence of digital communities on the nature, content and responsibilities of careers, and other aspects of society (e.g., 6.1.12.CivicsPD.16.a).
- 9.4.12.DC.8: Explain how increased network connectivity and computing capabilities of everyday objects allow for innovative technological approaches to climate protection.
- 9.4.12.GCA.1: Collaborate with individuals to analyze a variety of potential solutions to climate change effects and determine why some solutions (e.g., political, economic, cultural) may work better than others (e.g., SL.11-12.1., HS-ETS1-1, HS-ETS1-2, HS-ETS1-4, 6.3.12.GeoGI.1, 7.1.IH.IPERS.6, 7.1.IL.IPERS.7, 8.2.12.ETW.3).
- 9.4.12.IML.1: Compare search browsers and recognize features that allow for filtering of information.
- 9.4.12.IML.2: Evaluate digital sources for timeliness, accuracy, perspective, credibility of the source, and relevance of information, in media, data, or other resources (e.g., NJSLSA.W8, SocialStudies Practice: Gathering and Evaluating Sources).
- 9.4.12.IML.3: Analyze data using tools and models to make valid and reliable claims, or to determine optimal design solutions (e.g., S-ID.B.6a., 8.1.12.DA.5, 7.1.IH.IPRET.8).
- 9.4.12.IML.4: Assess and critique the appropriateness and impact of existing data visualizations for an intended audience (e.g., S-ID.B.6b, HS-LS2-4).
- 9.4.12.IML.5: Evaluate, synthesize, and apply information on climate change from various sources appropriately (e.g., 2.1.12.CHSS.6, S.IC.B.4, S.IC.B.6, 8.1.12.DA.1, 6.1.12.GeoHE.14.a, 7.1.AL.PRSNT.2).
- 9.4.12.IML.6: Use various types of media to produce and store information on climate change for different purposes and audiences with sensitivity to cultural, gender, and age diversity (e.g., NJSLSA.SL5).
- 9.4.12.IML.7: Develop an argument to support a claim regarding a current workplace or societal/ethical issue such as climate change (e.g., NJSLSA.W1, 7.1.AL.PRSNT.4).
- 9.4.12.IML.8: Evaluate media sources for point of view, bias, and motivations (e.g., NJSLSA.R6, 7.1.AL.IPRET.6).
- 9.4.12.IML.9: Analyze the decisions creators make to reveal explicit and implicit messages within information and media (e.g., 1.5.12acc.C2a, 7.1.IL.IPRET.4).
- 9.4.12.TL.1: Assess digital tools based on features such as accessibility options, capacities, and utility for accomplishing a specified task (e.g., W.11-12.6.).
- 9.4.12.TL.2: Generate data using formula-based calculations in a spreadsheet and draw conclusions about the data.
- 9.4.12.TL.3: Analyze the effectiveness of the process and quality of collaborative environments.

- 9.4.12.TL.4: Collaborate in online learning communities or social networks or virtual worlds to analyze and propose a resolution to a real-world problem (e.g., 7.1.AL.IPERS.6).

Interdisciplinary Connections:

Science and Engineering Practices:

- Develop a model based on evidence to illustrate the relationships between systems or between components of a system.
- Use a model to predict the relationships between systems or between components of a system.
- Plan and conduct an investigation individually and collaboratively to produce data to serve as the basis for evidence, and in the design: decide on types, how much, and accuracy of data needed to produce reliable measurements and consider limitations on the precision of the data (e.g., number of trials, cost, risk, time), and refine the design accordingly.
- Apply scientific principles and evidence to provide an explanation of phenomena and solve design problems, taking into account possible unanticipated effects.
- Analyze data using tools, technologies, and/or models (e.g., computational, mathematical) in order to make valid and reliable scientific claims or determine an optimal design solution.
- Communicate scientific and technical information (e.g. about the process of development and the design and performance of a proposed process or system) in multiple formats (including orally, graphically, textually, and mathematically).
- Create a computational model or simulation of a phenomenon, designed device, process, or system.
- Design, evaluate, and/or refine a solution to a complex real-world problem, based on scientific knowledge, student-generated sources of evidence, prioritized criteria, and tradeoff considerations.
- Evaluate a solution to a complex real-world problem, based on scientific knowledge, student-generated sources of evidence, prioritized criteria, and tradeoff considerations.

Engineering Design Standards:

- HS-ETS1-2 Design a solution to a complex real-world problem by breaking it down into smaller, more manageable problems that can be solved through engineering.
- HS-ETS1-3 Evaluate a solution to a complex real-world problem based on prioritized criteria and trade-offs that account for a range of constraints, including cost, safety, reliability, and aesthetics, as well as possible social, cultural, and environmental impacts.
- HS-ETS1-4 Use a computer simulation to model the impact of proposed solutions to a complex real-world problem with numerous criteria and constraints on interactions within and between systems relevant to the problem.

Learning Targets

Unit Essential Questions:

- How do images enhance the visual design and user experience of a mobile app?
- What role do Flex layouts play in creating responsive and adaptive mobile app interfaces?
- How can the Dimensions API be used to ensure app components scale properly across different devices?
- What strategies can be used to effectively plan and build a functional app screen using multiple components?
- How does combining images, layouts, and interactive elements contribute to the overall usability of a mobile app?

Unit Enduring Understandings:

- Images are essential elements that improve app aesthetics and help communicate information visually.
- Flex layouts provide a flexible and efficient way to organize and size app components across various screen sizes.
- The Dimensions API allows developers to create responsive designs by dynamically adapting component sizes to device dimensions.
- Careful planning and step-by-step development are crucial for building functional and visually appealing app screens.
- Integrating images, layouts, and interactive components enhances both the look and functionality, leading to better user engagement.

Vocabulary: image component, imagebackground component, flex layout, flex values, flex ratio, dimensions api, responsive design, screen width, screen height, app layout, component sizing, interactive elements, app functionality, mobile app development, user interface, visual design, layout planning, flexible box model, scaling, user experience

Students will be able to:

- incorporate images into their app designs and size them appropriately
- explain what the aspect ratio of an image is
- use ImageBackground components to layer various components on top of an image
- use Image components as buttons in their apps
- size components relative to each other and the screen size using flex
- size components relative to each other and the screen size using the Dimensions API
- note the reasons we would use Dimensions over flex values to create responsive designs
- replicate an app screen design using the skills they've learned up to this point
- think critically about the layout and style of an app screen

Learning Activities & Differentiation: *(Amistad & Holocaust, Asian American & Pacific Islanders, Climate Change, LGBTQ and People with Disabilities, Diversity, Equity & Inclusion [DEI])*

Learning Activities:

- TouchableHighlight Buttons in Action: Students build simple apps using TouchableHighlight components as buttons to navigate or trigger alerts. For example, creating a Homepage that highlights cultural heritage or social justice topics, such as AAPI history or climate change awareness.
- Exploring the onPress Function: Through hands-on labs, students implement the onPress function to activate button actions that share important information or prompt reflection.
- Using TextInput for User Interaction: Students design apps like a Google Search Page or Feedback Form where users enter text. These inputs can gather opinions on various topics.
- Personalizing Alerts with State and TextInput: Students practice setting and updating state values to personalize alert messages based on user input, reinforcing how technology can create tailored experiences.

- Debugging & Peer Collaboration: Students pair up to troubleshoot code, sharing perspectives and supporting each other's understanding of interactive components.

Differentiation Strategies:

- Advanced Learners: Create multi-screen apps combining TouchableHighlight buttons with TextInput forms that dynamically change content based on user input, focusing on interactive awareness campaigns or cultural showcases.
- Struggling Learners: Use partially completed starter apps focusing on one core skill at a time (e.g., adding a TouchableHighlight button or TextInput field). Provide visual flowcharts and stepwise pseudocode.
- English Language Learners (ELLs): Provide annotated examples and bilingual glossaries for key terms like "TouchableHighlight," "onPress," and "TextInput." Encourage apps that reflect students' own cultural or social experiences.
- Universal Design for Learning (UDL): Combine live coding demonstrations, visual app component maps, and interactive labs. Offer multiple ways to show learning—working apps, annotated screenshots, or brief presentations.

Education Connections:

- Show how TouchableHighlight buttons and TextInput fields reflect real-world app features that promote accessible communication and user engagement.
- Connect app development with social awareness by building apps focused on DEI, cultural heritage, and environmental issues such as Amistad & Holocaust remembrance, Asian American & Pacific Islander history, climate change activism, LGBTQ inclusion, and accessibility for people with disabilities.
- Emphasize debugging and collaboration as key skills that foster empathy and teamwork in technology and beyond.
- Illustrate how personalized user interactions in apps can amplify diverse voices and perspectives.

Assessment Evidence

Formative Assessments:

- CodeHS exercises
  - Example activity: Simple Flag with Flex
  - Example activity: SnapChat Login Page using Dimensions
- CodeHS Checks for Understanding

Summative Assessments:

- App Build: Recreate a Popular App
- Challenge: Scorekeeper Pt 3

Common Benchmark Assessments:

- App Build: Recreate a Popular App
- Challenge: Scorekeeper Pt 3

Alternative Assessments:

- CodeHS Unit Quiz

Core Instructional & Supplemental Materials

Core Materials:

- Built-in CodeHS React Native Editor with live mobile preview environment
- React Native syntax and component reference guides
- CodeHS tutorials on components like TouchableHighlight and TextInput, plus the Stylesheet API and styling techniques
- CodeHS lessons on debugging interactive components, customizing text inputs, and improving program readability
- Guides to using the Stylesheet API for layout, alignment, and flex direction
- Expo documentation for running and testing apps on physical devices

Supplemental Materials:

- Sample React Native code templates for TouchableHighlight buttons, TextInput fields, and simple app pages (e.g., Homepage, Google Search, Feedback Form)
- Visual guides illustrating component hierarchy, parent-child nesting, and flexbox layout principles
- Organizers for planning app designs, including wireframes and style rule charts focusing on buttons and text input layouts
- Debugging checklists and peer collaboration rubrics for troubleshooting interactive components
- Example styling challenge solutions (e.g., button styles, input form layouts) for student inspiration
- Accessibility and readability checklists to ensure inclusive, clear, and user-friendly interface designs
- Rubrics for evaluating app projects on correctness, creativity, usability, and design clarity
- Bilingual glossaries of key React Native, TouchableHighlight, TextInput, and styling terms to support English Language Learners

Accommodations and Modifications: Students with Disabilities, English Language Learners, Students at Risk of Failure, Students with 504s, Gifted & Talented Students

Specific Strategies and Practices that Support Students with Disabilities:

Students receiving Special Education programming have specific goals and objectives, as well as accommodations and modifications outlined within their Individualized Education Plans (IEP) due to an identified disability and/or diagnosis. In addition to exposure to the general education curriculum, the instruction is differentiated based upon the student's needs. The IEP acts as a supplemental curriculum guide inclusive of instructional strategies that support each learner.

- Small group/One to one - Additional time - Student restates information - Extra visual and verbal cues and prompts - Preferential seating - Rest breaks - Checklists - Immediate feedback

Specific Strategies and Practices that Support English Language Learners

English language learner (ELL) is a national-origin-minority student who is limited-English-proficient. This term is often preferred over limited-English-proficient (LEP) as it highlights accomplishments rather than deficits.

- Pre-teaching of vocabulary and concepts - Personal Glossary - Extended Time - Simplified/Verbal Instructions - Visual learning, including graphic organizers - Teacher modeling - Scaffolding - Think-pair-share

Students at Risk of Failure

- Home/School Connection - Alternative Assessments - Learning Contracts - Think-Pair-Share by Readiness - Homework Options - Small Group Instruction - Group Work Choice

Students with 504 Plans

Teachers are responsible for implementing designated services and strategies identified on a student's 504 Plan.
504 plans are formal plans that schools develop to give kids the support they need and covers any condition that limits daily activities in a major way. Students' accommodations are outlined in their individual 504 plan.

Specific Strategies and Practices that Support Gifted & Talented Students

It is critical that all teachers are able to recognize a high-ability student who may need more depth and complexity in instruction.

-Homogenous Grouping - Small Group In-Depth Instruction - Alternative Assessments

Unit 7

Desired Results

Unit 7 Title: Event and State

Unit Summary:

In this unit, students explore how to use and update state values to build dynamic and responsive apps. They learn foundational concepts such as what state is, how to manage it using JSX, and how to update state through user interactions like button presses (onPress). The module covers updating numerical state values with mathematical operations and manipulating string states using methods. Hands-on activities include setting location with state, converting names to nicknames, creating a simple calculator, and building a spellchecker.

Learning Standards

NJSLS:

- 8.1.12.CS.1: Describe ways in which integrated systems hide underlying implementation details to simplify user experiences.
- 8.1.12.CS.2: Model interactions between application software, system software, and hardware.
- 8.1.12.CS.3: Compare the functions of application software, system software, and hardware.
- 8.1.12.CS.4: Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.
- 8.1.12.IC.1: Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.
- 8.1.12.IC.2: Test and refine computational artifacts to reduce bias and equity deficits.
- 8.1.12.IC.3: Predict the potential impacts and implications of emerging technologies on larger social, economic, and political structures, using evidence from credible sources.
- 8.1.12.AP.1: Design algorithms to solve computational problems using a combination of original and existing algorithms.
- 8.1.12.AP.3: Select and combine control structures for a specific application based upon performance and readability, and identify trade-offs to justify the choice.
- 8.1.12.AP.4: Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue.
- 8.1.12.AP.5: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
- 8.1.12.AP.6: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.
- 8.1.12.AP.7: Collaboratively design and develop programs and artifacts for broad audiences by incorporating feedback from users.
- 8.1.12.AP.8: Evaluate and refine computational artifacts to make them more usable and accessible.
- 8.1.12.AP.9: Collaboratively document and present design decisions in the development of complex programs.

Interdisciplinary Connections

Career Readiness, Life Literacies, & Key Skills *(CLKS)*:

- 9.4.12.CI.1: Demonstrate the ability to reflect, analyze, and use creative skills and ideas (e.g., 1.1.12prof.CR3a).
- 9.4.12.CI.2: Identify career pathways that highlight personal talents, skills, and abilities (e.g., 1.4.12prof.CR2b, 2.2.12.LF.8).
- 9.4.12.CI.3: Investigate new challenges and opportunities for personal growth, advancement, and transition (e.g., 2.1.12.PGD.1).
- 9.4.12.CT.1: Identify problem-solving strategies used in the development of an innovative product or practice (e.g., 1.1.12acc.C1b, 2.2.12.PF.3).
- 9.4.12.CT.2: Explain the potential benefits of collaborating to enhance critical thinking and problem solving (e.g., 1.3E.12profCR3.a).
- 9.4.12.CT.3: Enlist input from a variety of stakeholders (e.g., community members, experts in the field) to design a service learning activity that addresses a local or global issue (e.g., environmental justice).
- 9.4.12.CT.4: Participate in online strategy and planning sessions for course-based, school-based, or other project and determine the strategies that contribute to effective outcomes.
- 9.4.12.DC.1: Explain the beneficial and harmful effects that intellectual property laws can have on the creation and sharing of content (e.g., 6.1.12.CivicsPR.16.a).
- 9.4.12.DC.2: Compare and contrast international differences in copyright laws and ethics.
- 9.4.12.DC.3: Evaluate the social and economic implications of privacy in the context of safety, law, or ethics (e.g., 6.3.12.HistoryCA.1).
- 9.4.12.DC.4: Explain the privacy concerns related to the collection of data (e.g., cookies) and generation of data through automated processes that may not be evident to users (e.g., 8.1.12.NI.3).
- 9.4.12.DC.5: Debate laws and regulations that impact the development and use of software.
- 9.4.12.DC.6: Select information to post online that positively impacts personal image and future college and career opportunities.
- 9.4.12.DC.7: Evaluate the influence of digital communities on the nature, content and responsibilities of careers, and other aspects of society (e.g., 6.1.12.CivicsPD.16.a).
- 9.4.12.DC.8: Explain how increased network connectivity and computing capabilities of everyday objects allow for innovative technological approaches to climate protection.

- 9.4.12.GCA.1: Collaborate with individuals to analyze a variety of potential solutions to climate change effects and determine why some solutions (e.g., political, economic, cultural) may work better than others (e.g., SL.11-12.1, HS-ETS1-1, HS-ETS1-2, HS-ETS1-4, 6.3.12.GeoGI.1, 7.1.IH.IPERS.6, 7.1.IL.IPERS.7, 8.2.12.ETW.3).
- 9.4.12.IML.1: Compare search browsers and recognize features that allow for filtering of information.
- 9.4.12.IML.2: Evaluate digital sources for timeliness, accuracy, perspective, credibility of the source, and relevance of information, in media, data, or other resources (e.g., NJSLSA.W8, SocialStudies Practice: Gathering and Evaluating Sources).
- 9.4.12.IML.3: Analyze data using tools and models to make valid and reliable claims, or to determine optimal design solutions (e.g., S-ID.B.6a., 8.1.12.DA.5, 7.1.IH.IPRET.8).
- 9.4.12.IML.4: Assess and critique the appropriateness and impact of existing data visualizations for an intended audience (e.g., S-ID.B.6b, HS-LS2-4).
- 9.4.12.IML.5: Evaluate, synthesize, and apply information on climate change from various sources appropriately (e.g., 2.1.12.CHSS.6, S.IC.B.4, S.IC.B.6, 8.1.12.DA.1, 6.1.12.GeoHE.14.a, 7.1.AL.PRSNT.2).
- 9.4.12.IML.6: Use various types of media to produce and store information on climate change for different purposes and audiences with sensitivity to cultural, gender, and age diversity (e.g., NJSLSA.SL5).
- 9.4.12.IML.7: Develop an argument to support a claim regarding a current workplace or societal/ethical issue such as climate change (e.g., NJSLSA.W1, 7.1.AL.PRSNT.4).
- 9.4.12.IML.8: Evaluate media sources for point of view, bias, and motivations (e.g., NJSLSA.R6, 7.1.AL.IPRET.6).
- 9.4.12.IML.9: Analyze the decisions creators make to reveal explicit and implicit messages within information and media (e.g., 1.5.12acc.C2a, 7.1.IL.IPRET.4).
- 9.4.12.TL.1: Assess digital tools based on features such as accessibility options, capacities, and utility for accomplishing a specified task (e.g., W.11-12.6.).
- 9.4.12.TL.2: Generate data using formula-based calculations in a spreadsheet and draw conclusions about the data.
- 9.4.12.TL.3: Analyze the effectiveness of the process and quality of collaborative environments.
- 9.4.12.TL.4: Collaborate in online learning communities or social networks or virtual worlds to analyze and propose a resolution to a real-world problem (e.g., 7.1.AL.IPERS.6).

Interdisciplinary Connections:

Science and Engineering Practices:

- Develop a model based on evidence to illustrate the relationships between systems or between components of a system.
- Use a model to predict the relationships between systems or between components of a system.
- Plan and conduct an investigation individually and collaboratively to produce data to serve as the basis for evidence, and in the design: decide on types, how much, and accuracy of data needed to produce reliable measurements and consider limitations on the precision of the data (e.g., number of trials, cost, risk, time), and refine the design accordingly.
- Apply scientific principles and evidence to provide an explanation of phenomena and solve design problems, taking into account possible unanticipated effects.
- Analyze data using tools, technologies, and/or models (e.g., computational, mathematical) in order to make valid and reliable scientific claims or determine an optimal design solution.
- Use mathematical representations of phenomena to describe explanations.
- Communicate scientific and technical information (e.g. about the process of development and the design and performance of a proposed process or system) in multiple formats (including orally, graphically, textually, and mathematically).
- Create a computational model or simulation of a phenomenon, designed device, process, or system.
- Design, evaluate, and/or refine a solution to a complex real-world problem, based on scientific knowledge, student-generated sources of evidence, prioritized criteria, and tradeoff considerations.
- Evaluate a solution to a complex real-world problem, based on scientific knowledge, student-generated sources of evidence, prioritized criteria, and tradeoff considerations.

Engineering Design Standards:

- HS-ETS1-2 Design a solution to a complex real-world problem by breaking it down into smaller, more manageable problems that can be solved through engineering.

- HS-ETS1-3 Evaluate a solution to a complex real-world problem based on prioritized criteria and trade-offs that account for a range of constraints, including cost, safety, reliability, and aesthetics, as well as possible social, cultural, and environmental impacts.
- HS-ETS1-4 Use a computer simulation to model the impact of proposed solutions to a complex real-world problem with numerous criteria and constraints on interactions within and between systems relevant to the problem.

Learning Targets

Unit Essential Questions:

- What is state in an app, and why is it important for creating dynamic programs?
- How can user interactions trigger changes in an app's state?
- In what ways can mathematical operations be used to update state values?
- How do string methods help in manipulating the state of text within an app?
- How does managing state effectively improve the user experience in interactive applications?

Unit Enduring Understandings:

- State represents the current data or condition of an app that can change over time.
- Updating state in response to events allows apps to react dynamically to user input.
- Mathematical equations can be used to modify numerical state values to perform calculations.
- String methods provide tools to modify text-based state, enabling functions like spellchecking or formatting.
- Proper state management is essential for building responsive, interactive, and user-friendly applications.

Vocabulary: state, state object, jsx, onpress, update state, mathematical equations, string methods, dynamic programs, user interaction, event handler, variable, function, nickname, calculator, spellchecker

Students will be able to:

- set multiple state values
- use state values inside their programs

- create and use functions that are called using the onPress function to update state values
- use mathematical equations to update numerical state values in their programs
- use string methods to update the value of string states

Learning Activities & Differentiation: *(Amistad & Holocaust, Asian American & Pacific Islanders, Climate Change, LGBTQ and People with Disabilities, Diversity, Equity & Inclusion [DEI])*

Learning Activities:

- TouchableHighlight Buttons in Action: Students build simple apps using TouchableHighlight components as buttons to navigate or trigger alerts. For example, creating a Homepage that highlights important topics or messages.
- Exploring the onPress Function: Through hands-on labs, students implement the onPress function to activate button actions that share information or prompt reflection.
- Using TextInput for User Interaction: Students design apps like a Google Search Page or Feedback Form where users enter text, gathering input or feedback.
- Personalizing Alerts with State and TextInput: Students practice setting and updating state values to personalize alert messages based on user input, reinforcing how technology can create tailored experiences.
- Debugging & Peer Collaboration: Students pair up to troubleshoot code, sharing perspectives and supporting each other's understanding of interactive components.

Differentiation Strategies:

- Advanced Learners: Create multi-screen apps combining TouchableHighlight buttons with TextInput forms that dynamically change content based on user input, focusing on interactive campaigns or showcases.
- Struggling Learners: Use partially completed starter apps focusing on one core skill at a time (e.g., adding a TouchableHighlight button or TextInput field). Provide visual flowcharts and stepwise pseudocode.
- English Language Learners (ELLs): Provide annotated examples and bilingual glossaries for key terms like "TouchableHighlight," "onPress," and "TextInput." Encourage apps that reflect students' own cultural or social experiences.

- Universal Design for Learning (UDL): Combine live coding demonstrations, visual app component maps, and interactive labs. Offer multiple ways to show learning—working apps, annotated screenshots, or brief presentations.

Education Connections:

- Show how TouchableHighlight buttons and TextInput fields reflect real-world app features that promote accessible communication and user engagement.
- Connect app development with social awareness by building apps focused on DEI, cultural heritage, and environmental issues such as Amistad & Holocaust remembrance, Asian American & Pacific Islander history, climate change activism, LGBTQ inclusion, and accessibility for people with disabilities.
- Emphasize debugging and collaboration as key skills that foster empathy and teamwork in technology and beyond.
- Illustrate how personalized user interactions in apps can amplify diverse voices and perspectives.

Assessment Evidence

Formative Assessments:

- CodeHS exercises
  - Example activity: Setting the Location with State
  - Example activity: Spellchecker
- CodeHS Checks for Understanding

Summative Assessments:

- Challenge: Scorekeeper Pt 4

Common Benchmark Assessments:

- Challenge: Scorekeeper Pt 4

Alternative Assessments:

- CodeHS Unit Quiz

Core Instructional & Supplemental Materials

Core Materials:

- Built-in CodeHS React Native Editor with live mobile preview environment
- React Native syntax and component reference guides
- CodeHS tutorials on components like TouchableHighlight and TextInput, plus the Stylesheet API and styling techniques
- CodeHS lessons on debugging interactive components, customizing text inputs, and improving program readability
- Guides to using the Stylesheet API for layout, alignment, and flex direction
- Expo documentation for running and testing apps on physical devices

Supplemental Materials:

- Sample React Native code templates for TouchableHighlight buttons, TextInput fields, and simple app pages (e.g., Homepage, Google Search, Feedback Form)
- Visual guides illustrating component hierarchy, parent-child nesting, and flexbox layout principles
- Organizers for planning app designs, including wireframes and style rule charts focusing on buttons and text input layouts
- Debugging checklists and peer collaboration rubrics for troubleshooting interactive components
- Example styling challenge solutions (e.g., button styles, input form layouts) for student inspiration
- Accessibility and readability checklists to ensure inclusive, clear, and user-friendly interface designs
- Rubrics for evaluating app projects on correctness, creativity, usability, and design clarity
- Bilingual glossaries of key React Native, TouchableHighlight, TextInput, and styling terms to support English Language Learners

Accommodations and Modifications:
Students with Disabilities, English Language Learners,

Students at Risk of Failure, Students with 504s, Gifted & Talented Students

Specific Strategies and Practices that Support Students with Disabilities:

Students receiving Special Education programming have specific goals and objectives, as well as accommodations and modifications outlined within their Individualized Education Plans (IEP) due to an identified disability and/or diagnosis. In addition to exposure to the general education curriculum, the instruction is differentiated based upon the student's needs. The IEP acts as a supplemental curriculum guide inclusive of instructional strategies that support each learner.

- Small group/One to one - Space for movement or breaks - Follow a routine/schedule - Rest breaks - Verbal and visual cues regarding directions and staying on task - Immediate feedback

Specific Strategies and Practices that Support English Language Learners

English language learner (ELL) is a national-origin-minority student who is limited-English-proficient. This term is often preferred over limited-English-proficient (LEP) as it highlights accomplishments rather than deficits.

- Pre-teaching of vocabulary and concepts - Personal Glossary - Simplified/Verbal Instructions - Frequent Breaks - Scaffolding - Word walls

Students at Risk of Failure

- Homogenous Grouping - Supplemental Instruction - Think-Pair-Share by Readiness - Tiered Level of Supports - Tiered Activities - Small Group Instruction - Group Work Choice

Students with 504 Plans

Teachers are responsible for implementing designated services and strategies identified on a student's 504 Plan.
504 plans are formal plans that schools develop to give kids the support they need and covers any condition that limits daily activities in a major way. Students' accommodations are outlined in their individual 504 plan.

Specific Strategies and Practices that Support Gifted & Talented Students

It is critical that all teachers are able to recognize a high-ability student who may need more depth and complexity in instruction.

- Small Group In-Depth Instruction - Differentiated Online Practice - Alternative Assessments

Unit 8

Desired Results

Unit 8 Title: Creating Multiple Screens

Unit Summary:

In this unit, students learn how to enhance their apps by implementing multiple screens with interactive navigation. Key skills include creating a navigation bar (navbar), using buttons and images as interactive elements, and managing app state to switch between screens. Example projects include building an iPhone-style call navbar and an RSVP tracker app that demonstrates screen navigation through button interactions.

Learning Standards

NJSLS:

- 8.1.12.CS.1: Describe ways in which integrated systems hide underlying implementation details to simplify user experiences.
- 8.1.12.CS.2: Model interactions between application software, system software, and hardware.
- 8.1.12.CS.3: Compare the functions of application software, system software, and hardware.
- 8.1.12.CS.4: Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.
- 8.1.12.IC.1: Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.
- 8.1.12.AP.1: Design algorithms to solve computational problems using a combination of original and existing algorithms.
- 8.1.12.AP.3: Select and combine control structures for a specific application based upon performance and readability, and identify trade-offs to justify the choice.
- 8.1.12.AP.4: Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue.
- 8.1.12.AP.5: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.

- 8.1.12.AP.6: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.
- 8.1.12.AP.7: Collaboratively design and develop programs and artifacts for broad audiences by incorporating feedback from users.
- 8.1.12.AP.8: Evaluate and refine computational artifacts to make them more usable and accessible.
- 8.1.12.AP.9: Collaboratively document and present design decisions in the development of complex programs.

Interdisciplinary Connections

Career Readiness, Life Literacies, & Key Skills *(CLKS)*:

- 9.4.12.CI.1: Demonstrate the ability to reflect, analyze, and use creative skills and ideas (e.g., 1.1.12prof.CR3a).
- 9.4.12.CI.2: Identify career pathways that highlight personal talents, skills, and abilities (e.g., 1.4.12prof.CR2b, 2.2.12.LF.8).
- 9.4.12.CI.3: Investigate new challenges and opportunities for personal growth, advancement, and transition (e.g., 2.1.12.PGD.1).
- 9.4.12.CT.1: Identify problem-solving strategies used in the development of an innovative product or practice (e.g., 1.1.12acc.C1b, 2.2.12.PF.3).
- 9.4.12.CT.2: Explain the potential benefits of collaborating to enhance critical thinking and problem solving (e.g., 1.3E.12profCR3.a).
- 9.4.12.CT.3: Enlist input from a variety of stakeholders (e.g., community members, experts in the field) to design a service learning activity that addresses a local or global issue (e.g., environmental justice).
- 9.4.12.CT.4: Participate in online strategy and planning sessions for course-based, school-based, or other project and determine the strategies that contribute to effective outcomes.
- 9.4.12.DC.1: Explain the beneficial and harmful effects that intellectual property laws can have on the creation and sharing of content (e.g., 6.1.12.CivicsPR.16.a).
- 9.4.12.DC.2: Compare and contrast international differences in copyright laws and ethics.
- 9.4.12.DC.3: Evaluate the social and economic implications of privacy in the context of safety, law, or ethics (e.g., 6.3.12.HistoryCA.1).
- 9.4.12.DC.4: Explain the privacy concerns related to the collection of data (e.g., cookies) and generation of data through automated processes that may not be evident to users (e.g., 8.1.12.NI.3).

- 9.4.12.DC.5: Debate laws and regulations that impact the development and use of software.
- 9.4.12.DC.6: Select information to post online that positively impacts personal image and future college and career opportunities.
- 9.4.12.DC.7: Evaluate the influence of digital communities on the nature, content and responsibilities of careers, and other aspects of society (e.g., 6.1.12.CivicsPD.16.a).
- 9.4.12.DC.8: Explain how increased network connectivity and computing capabilities of everyday objects allow for innovative technological approaches to climate protection.
- 9.4.12.GCA.1: Collaborate with individuals to analyze a variety of potential solutions to climate change effects and determine why some solutions (e.g., political. economic, cultural) may work better than others (e.g., SL.11-12.1., HS-ETS1-1, HS-ETS1-2, HS-ETS1-4, 6.3.12.GeoGI.1, 7.1.IH.IPERS.6, 7.1.IL.IPERS.7, 8.2.12.ETW.3).
- 9.4.12.IML.1: Compare search browsers and recognize features that allow for filtering of information.
- 9.4.12.IML.2: Evaluate digital sources for timeliness, accuracy, perspective, credibility of the source, and relevance of information, in media, data, or other resources (e.g., NJSLSA.W8, SocialStudies Practice: Gathering and Evaluating Sources).
- 9.4.12.IML.3: Analyze data using tools and models to make valid and reliable claims, or to determine optimal design solutions (e.g., S-ID.B.6a., 8.1.12.DA.5, 7.1.IH.IPRET.8).
- 9.4.12.IML.4: Assess and critique the appropriateness and impact of existing data visualizations for an intended audience (e.g., S-ID.B.6b, HS-LS2-4).
- 9.4.12.IML.5: Evaluate, synthesize, and apply information on climate change from various sources appropriately (e.g., 2.1.12.CHSS.6, S.IC.B.4, S.IC.B.6, 8.1.12.DA.1, 6.1.12.GeoHE.14.a, 7.1.AL.PRSNT.2).
- 9.4.12.IML.6: Use various types of media to produce and store information on climate change for different purposes and audiences with sensitivity to cultural, gender, and age diversity (e.g., NJSLSA.SL5).
- 9.4.12.IML.7: Develop an argument to support a claim regarding a current workplace or societal/ethical issue such as climate change (e.g., NJSLSA.W1, 7.1.AL.PRSNT.4).
- 9.4.12.IML.8: Evaluate media sources for point of view, bias, and motivations (e.g., NJSLSA.R6, 7.1.AL.IPRET.6).
- 9.4.12.IML.9: Analyze the decisions creators make to reveal explicit and implicit messages within information and media (e.g., 1.5.12acc.C2a, 7.1.IL.IPRET.4).

- 9.4.12.TL.1: Assess digital tools based on features such as accessibility options, capacities, and utility for accomplishing a specified task (e.g., W.11-12.6.).
- 9.4.12.TL.2: Generate data using formula-based calculations in a spreadsheet and draw conclusions about the data.
- 9.4.12.TL.3: Analyze the effectiveness of the process and quality of collaborative environments.
- 9.4.12.TL.4: Collaborate in online learning communities or social networks or virtual worlds to analyze and propose a resolution to a real-world problem (e.g., 7.1.AL.IPERS.6).

Interdisciplinary Connections:

Science and Engineering Practices:

- Develop a model based on evidence to illustrate the relationships between systems or between components of a system.
- Use a model to predict the relationships between systems or between components of a system.
- Analyze data using tools, technologies, and/or models (e.g., computational, mathematical) in order to make valid and reliable scientific claims or determine an optimal design solution.
- Use mathematical and/or computational representations of phenomena or design solutions to support explanations.
- Create a computational model or simulation of a phenomenon, designed device, process, or system.
- Design, evaluate, and/or refine a solution to a complex real-world problem, based on scientific knowledge, student-generated sources of evidence, prioritized criteria, and tradeoff considerations.
- Evaluate a solution to a complex real-world problem, based on scientific knowledge, student-generated sources of evidence, prioritized criteria, and tradeoff considerations.
- Communicate scientific and technical information (e.g. about the process of development and the design and performance of a proposed process or system) in multiple formats (including orally, graphically, textually, and mathematically).

Engineering Design Standards:

- HS-ETS1-1 Analyze a major global challenge to specify qualitative and quantitative criteria and constraints for solutions that account for societal needs and wants.
- HS-ETS1-2 Design a solution to a complex real-world problem by breaking it down into smaller, more manageable problems that can be solved through engineering.
- HS-ETS1-3 Evaluate a solution to a complex real-world problem based on prioritized criteria and trade-offs that account for a range of constraints, including cost, safety, reliability, and aesthetics, as well as possible social, cultural, and environmental impacts.
- HS-ETS1-4 Use a computer simulation to model the impact of proposed solutions to a complex real-world problem with numerous criteria and constraints on interactions within and between systems relevant to the problem.

Learning Targets

Unit Essential Questions:

- How can multiple screens improve the user experience in an app?
- What are the key components needed to create effective navigation between screens?
- How can buttons and images be used to facilitate screen navigation?
- What role does app state play in managing different screens?
- How can designing a navbar help organize app content and functionality?

Unit Enduring Understandings:

- Multiple screens enable apps to organize content and interactivity in a way that enhances usability and user engagement.
- Navigation bars (navbars) are a fundamental UI element that provide clear pathways for users to move through an app.
- Buttons and images can serve as intuitive interactive elements to trigger navigation between screens.
- Managing app state is essential for controlling which screen is displayed and maintaining app behavior across interactions.
- Thoughtful design of navigation structures helps create seamless and accessible app experiences.

Vocabulary: navbar, button, navigation, screen, state, interactivity, user interface, app state, image button, rsvp tracker, multiple screens, functionality, ui element, user experience, screen transition

Students will be able to:

- create a layout which includes a component for each screen's information and a container that acts as a navbar
- develop apps that use buttons placed in a navbar to navigate between multiple screens

Learning Activities & Differentiation: *(Amistad & Holocaust, Asian American & Pacific Islanders, Climate Change, LGBTQ and People with Disabilities, Diversity, Equity & Inclusion [DEI])*

Learning Activities:

- Building a Navbar: Students design and implement a container that acts as a navigation bar, organizing buttons or images to represent each screen.
- Button Navigation Labs: Through guided labs, students use buttons within the navbar to switch between multiple screens, exploring how button presses update app state to control screen transitions.
- Customizing Navbar Buttons: Students experiment with using images or text as buttons in the navbar, enhancing app usability and visual appeal.
- State Management for Screens: Hands-on activities focus on managing app state variables that determine which screen is active, reinforcing the logic behind screen navigation.
- Peer Debugging and Code Sharing: Students collaborate to troubleshoot navigation issues, exchanging ideas to deepen understanding of multi-screen interactivity.

Differentiation Strategies:

- Advanced Learners: Challenge to create multi-screen apps with dynamic content that changes based on user input or other state variables, integrating personalized navigation experiences.
- Struggling Learners: Provide scaffolded starter projects focusing on creating a simple navbar with one or two screens. Use flowcharts and pseudocode to clarify navigation logic step-by-step.
- English Language Learners (ELLs): Supply annotated examples of navbars and navigation buttons, including

bilingual glossaries for key terms such as “navbar,” “state,” and “navigation.” Encourage projects that reflect students’ cultural backgrounds.
- Universal Design for Learning (UDL): Use live demonstrations, visual diagrams of screen flow, and multiple output options like working apps, annotated screenshots, or presentations to engage diverse learners.

Education Connections:
- Highlight how navigation bars and multi-screen apps mirror real-world applications that improve user experience and accessibility.
- Connect app navigation design with social and cultural themes by encouraging apps that organize content related to diversity, equity, and inclusion, or community awareness.
- Emphasize collaboration and debugging as essential skills that support empathy, communication, and teamwork within technology projects and broader contexts.
- Show how designing intuitive navigation can empower users of all abilities, promoting inclusivity and personalized experiences.

Assessment Evidence

Formative Assessments:
- CodeHS exercises
  - Example activity: iPhone Call Navbar
  - Example activity: SRSVP Tracker
- CodeHS Checks for Understanding

Summative Assessments:
- Challenge: Scorekeeper Pt 5

Common Benchmark Assessments:
- Challenge: Scorekeeper Pt 5

Alternative Assessments:

- CodeHS Unit Quiz

Core Instructional & Supplemental Materials

Core Materials:

- Built-in CodeHS React Native Editor with live mobile preview environment
- React Native syntax and component reference guides
- CodeHS tutorials on creating navigation bars, using buttons for navigation, and managing app state
- CodeHS lessons on debugging multi-screen navigation and improving program readability
- Guides to using the Stylesheet API for flexbox layouts, alignment, and responsive design
- Expo documentation for running and testing apps with multiple screens on physical devices

Supplemental Materials:

- Sample React Native code templates for navigation bars, screen containers, and button-based screen transitions
- Visual guides showing screen flow diagrams, navbar structures, and component hierarchies
- Organizers for planning multi-screen app designs, including wireframes and navigation flowcharts
- Debugging checklists and peer collaboration rubrics focused on troubleshooting screen navigation and state logic
- Example multi-screen app solutions (e.g., iPhone Call Navbar, RSVP Tracker) to provide inspiration and modeling
- Accessibility and usability checklists to ensure navbars and buttons are clear, consistent, and user-friendly
- Rubrics for evaluating app projects on correctness, navigation functionality, creativity, and design clarity
- Bilingual glossaries of key React Native terms such as “navbar,” “state,” “navigation,” and “screen transition” to support English Language Learners

Accommodations and Modifications: Students with Disabilities, English Language Learners, Students at Risk of Failure, Students with 504s, Gifted & Talented Students

Specific Strategies and Practices that Support Students with Disabilities:

Students receiving Special Education programming have specific goals and objectives, as well as accommodations and modifications outlined within their Individualized Education Plans (IEP) due to an identified disability and/or diagnosis. In addition to exposure to the general education curriculum, the instruction is differentiated based upon the student's needs. The IEP acts as a supplemental curriculum guide inclusive of instructional strategies that support each learner.

- Space for movement or breaks - Extra visual and verbal cues and prompts - Preferential seating - Rest breaks - Verbal and visual cues regarding directions and staying on task

Specific Strategies and Practices that Support English Language Learners

English language learner (ELL) is a national-origin-minority student who is limited-English-proficient. This term is often preferred over limited-English-proficient (LEP) as it highlights accomplishments rather than deficits.

- Pre-teaching of vocabulary and concepts - Extended Time - Frequent Breaks - Visual learning, including graphic organizers - Use of cognates to increase comprehension - Think-pair-share - Cooperative learning groups

Students at Risk of Failure

- Home/School Connection - Learning Contracts - Homogenous Grouping - Think-Pair-Share by Readiness - Leveled Rubrics - Tiered Level of Supports - Small Group Instruction - Work Alone Choice - Group Work Choice

Students with 504 Plans

Teachers are responsible for implementing designated services and strategies identified on a student's 504 Plan.
504 plans are formal plans that schools develop to give kids the support they need and covers any condition that limits daily activities in a major way. Students' accommodations are outlined in their individual 504 plan.

Specific Strategies and Practices that Support Gifted & Talented Students

It is critical that all teachers are able to recognize a high-ability student who may need more depth and complexity in instruction.

-Homogenous Grouping - Small Group In-Depth Instruction - Alternative Assessments

Unit 9

Desired Results

Unit 9 Title: Working with Conditionals

Unit Summary:

In this unit, students learn to use conditionals and parameters to make apps more flexible and responsive. They practice with activities like checking Venmo balances and determining winners using parameters. They then apply these skills in a Currency Converter App project, combining components, events, state, conditionals, and styling to create a user-friendly app. Students plan with pseudocode, design effective layouts, and present their finished apps.

Learning Standards

NJSLS:

- 8.1.12.CS.1: Describe ways in which integrated systems hide underlying implementation details to simplify user experiences.
- 8.1.12.CS.2: Model interactions between application software, system software, and hardware.
- 8.1.12.CS.3: Compare the functions of application software, system software, and hardware.
- 8.1.12.CS.4: Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.
- 8.1.12.NI.1: Evaluate the scalability and reliability of networks, by describing the relationship between routers, switches, servers, topology, and addressing.
- 8.1.12.NI.2: Evaluate security measures to address various common security threats.
- 8.1.12.NI.3: Explain how the needs of users and the sensitivity of data determine the level of security implemented.
- 8.1.12.NI.4: Explain how decisions on methods to protect data are influenced by whether the data is at rest, in transit, or in use.
- 8.1.12.IC.1: Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.
- 8.1.12.IC.2: Test and refine computational artifacts to reduce bias and equity deficits.
- 8.1.12.IC.3: Predict the potential impacts and implications of emerging technologies on larger social, economic, and political structures, using evidence from credible sources.
- 8.1.12.DA.1: Create interactive data visualizations using software tools to help others better understand real world phenomena, including climate change.

- 8.1.12.DA.2: Describe the trade-offs in how and where data is organized and stored.
- 8.1.12.DA.3: Translate between decimal numbers and binary numbers.
- 8.1.12.DA.4: Explain the relationship between binary numbers and the storage and use of data in a computing device.
- 8.1.12.DA.5: Create data visualizations from large data sets to summarize, communicate, and support different interpretations of real-world phenomena.
- 8.1.12.DA.6: Create and refine computational models to better represent the relationships among different elements of data collected from a phenomenon or process.
- 8.1.12.AP.1: Design algorithms to solve computational problems using a combination of original and existing algorithms.
- 8.1.12.AP.2: Create generalized computational solutions using collections instead of repeatedly using simple variables.
- 8.1.12.AP.3: Select and combine control structures for a specific application based upon performance and readability, and identify trade-offs to justify the choice.
- 8.1.12.AP.4: Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue.
- 8.1.12.AP.5: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
- 8.1.12.AP.6: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.
- 8.1.12.AP.7: Collaboratively design and develop programs and artifacts for broad audiences by incorporating feedback from users.
- 8.1.12.AP.8: Evaluate and refine computational artifacts to make them more usable and accessible.
- 8.1.12.AP.9: Collaboratively document and present design decisions in the development of complex programs.
- 8.2.12.ED.1: Use research to design and create a product or system that addresses a problem and make modifications based on input from potential consumers.
- 8.2.12.ED.2: Create scaled engineering drawings for a new product or system and make modification to increase optimization based on feedback.

- 8.2.12.ED.3: Evaluate several models of the same type of product and make recommendations for a new design based on a cost benefit analysis.
- 8.2.12.ED.4: Design a product or system that addresses a global problem and document decisions made based on research, constraints, trade-offs, and aesthetic and ethical considerations and share this information with an appropriate audience.
- 8.2.12.ED.5: Evaluate the effectiveness of a product or system based on factors that are related to its requirements, specifications, and constraints (e.g., safety, reliability, economic considerations, quality control, environmental concerns, manufacturability, maintenance and repair, ergonomics).
- 8.2.12.ED.6: Analyze the effects of changing resources when designing a specific product or system (e.g., materials, energy, tools, capital, labor).
- 8.2.12.ITH.1: Analyze a product to determine the impact that economic, political, social, and/or cultural factors have had on its design, including its design constraints.
- 8.2.12.ITH.2: Propose an innovation to meet future demands supported by an analysis of the potential costs, benefits, trade-offs, and risks related to the use of the innovation.
- 8.2.12.ITH.3: Analyze the impact that globalization, social media, and access to open source technologies has had on innovation and on a society's economy, politics, and culture.
- 8.2.12.NT.1: Explain how different groups can contribute to the overall design of a product.
- 8.2.12.NT.2: Redesign an existing product to improve form or function.
- 8.2.12.ETW.1: Evaluate ethical considerations regarding the sustainability of environmental resources that are used for the design, creation, and maintenance of a chosen product.
- 8.2.12.ETW.2: Synthesize and analyze data collected to monitor the effects of a technological product or system on the environment.
- 8.2.12.ETW.3: Identify a complex, global environmental or climate change issue, develop a systemic plan of investigation, and propose an innovative sustainable solution.
- 8.2.12.EC.1: Analyze controversial technological issues and determine the degree to which individuals, businesses, and governments have an ethical role in decisions that are made.
- 8.2.12.EC.2: Assess the positive and negative impacts of emerging technologies on developing countries and evaluate how individuals, non-profit organizations, and governments have responded.

- 8.2.12.EC.3: Synthesize data, analyze trends, and draw conclusions regarding the effect of a technology on the individual, culture, society, and environment and share this information with the appropriate audience.
- 8.2.12.ETW.4: Research historical tensions between environmental and economic considerations as driven by human needs and wants in the development of a technological product and present the competing viewpoints.

Interdisciplinary Connections

Career Readiness, Life Literacies, & Key Skills *(CLKS)*:

- 9.4.12.CI.1: Demonstrate the ability to reflect, analyze, and use creative skills and ideas (e.g., 1.1.12prof.CR3a).
- 9.4.12.CI.2: Identify career pathways that highlight personal talents, skills, and abilities (e.g., 1.4.12prof.CR2b, 2.2.12.LF.8).
- 9.4.12.CI.3: Investigate new challenges and opportunities for personal growth, advancement, and transition (e.g., 2.1.12.PGD.1).
- 9.4.12.CT.1: Identify problem-solving strategies used in the development of an innovative product or practice (e.g., 1.1.12acc.C1b, 2.2.12.PF.3).
- 9.4.12.CT.2: Explain the potential benefits of collaborating to enhance critical thinking and problem solving (e.g., 1.3E.12profCR3.a).
- 9.4.12.CT.3: Enlist input from a variety of stakeholders (e.g., community members, experts in the field) to design a service learning activity that addresses a local or global issue (e.g., environmental justice).
- 9.4.12.CT.4: Participate in online strategy and planning sessions for course-based, school-based, or other project and determine the strategies that contribute to effective outcomes.
- 9.4.12.DC.1: Explain the beneficial and harmful effects that intellectual property laws can have on the creation and sharing of content (e.g., 6.1.12.CivicsPR.16.a).
- 9.4.12.DC.2: Compare and contrast international differences in copyright laws and ethics.
- 9.4.12.DC.3: Evaluate the social and economic implications of privacy in the context of safety, law, or ethics (e.g., 6.3.12.HistoryCA.1).
- 9.4.12.DC.4: Explain the privacy concerns related to the collection of data (e.g., cookies) and generation of data through automated processes that may not be evident to users (e.g., 8.1.12.NI.3).
- 9.4.12.DC.5: Debate laws and regulations that impact the development and use of software.

- 9.4.12.DC.6: Select information to post online that positively impacts personal image and future college and career opportunities.
- 9.4.12.DC.7: Evaluate the influence of digital communities on the nature, content and responsibilities of careers, and other aspects of society (e.g., 6.1.12.CivicsPD.16.a).
- 9.4.12.DC.8: Explain how increased network connectivity and computing capabilities of everyday objects allow for innovative technological approaches to climate protection.
- 9.4.12.GCA.1: Collaborate with individuals to analyze a variety of potential solutions to climate change effects and determine why some solutions (e.g., political, economic, cultural) may work better than others (e.g., SL.11-12.1., HS-ETS1-1, HS-ETS1-2, HS-ETS1-4, 6.3.12.GeoGI.1, 7.1.IH.IPERS.6, 7.1.IL.IPERS.7, 8.2.12.ETW.3).
- 9.4.12.IML.1: Compare search browsers and recognize features that allow for filtering of information.
- 9.4.12.IML.2: Evaluate digital sources for timeliness, accuracy, perspective, credibility of the source, and relevance of information, in media, data, or other resources (e.g., NJSLSA.W8, SocialStudies Practice: Gathering and Evaluating Sources).
- 9.4.12.IML.3: Analyze data using tools and models to make valid and reliable claims, or to determine optimal design solutions (e.g., S-ID.B.6a., 8.1.12.DA.5, 7.1.IH.IPRET.8).
- 9.4.12.IML.4: Assess and critique the appropriateness and impact of existing data visualizations for an intended audience (e.g., S-ID.B.6b, HS-LS2-4).
- 9.4.12.IML.5: Evaluate, synthesize, and apply information on climate change from various sources appropriately (e.g., 2.1.12.CHSS.6, S.IC.B.4, S.IC.B.6, 8.1.12.DA.1, 6.1.12.GeoHE.14.a, 7.1.AL.PRSNT.2).
- 9.4.12.IML.6: Use various types of media to produce and store information on climate change for different purposes and audiences with sensitivity to cultural, gender, and age diversity (e.g., NJSLSA.SL5).
- 9.4.12.IML.7: Develop an argument to support a claim regarding a current workplace or societal/ethical issue such as climate change (e.g., NJSLSA.W1, 7.1.AL.PRSNT.4).
- 9.4.12.IML.8: Evaluate media sources for point of view, bias, and motivations (e.g., NJSLSA.R6, 7.1.AL.IPRET.6).
- 9.4.12.IML.9: Analyze the decisions creators make to reveal explicit and implicit messages within information and media (e.g., 1.5.12acc.C2a, 7.1.IL.IPRET.4).
- 9.4.12.TL.1: Assess digital tools based on features such as accessibility options, capacities, and utility for accomplishing a specified task (e.g., W.11-12.6.).

- 9.4.12.TL.2: Generate data using formula-based calculations in a spreadsheet and draw conclusions about the data.
- 9.4.12.TL.3: Analyze the effectiveness of the process and quality of collaborative environments.
- 9.4.12.TL.4: Collaborate in online learning communities or social networks or virtual worlds to analyze and propose a resolution to a real-world problem (e.g., 7.1.AL.IPERS.6).

Interdisciplinary Connections:

Science and Engineering Practices:

- Develop a model based on evidence to illustrate the relationships between systems or between components of a system.
- Use a model to predict the relationships between systems or between components of a system.
- Plan and conduct an investigation individually and collaboratively to produce data to serve as the basis for evidence, and in the design: decide on types, how much, and accuracy of data needed to produce reliable measurements and consider limitations on the precision of the data (e.g., number of trials, cost, risk, time), and refine the design accordingly.
- Apply scientific principles and evidence to provide an explanation of phenomena and solve design problems, taking into account possible unanticipated effects.
- Analyze data using tools, technologies, and/or models (e.g., computational, mathematical) in order to make valid and reliable scientific claims or determine an optimal design solution.
- Use mathematical representations of phenomena to describe explanations.
- Communicate scientific and technical information (e.g. about the process of development and the design and performance of a proposed process or system) in multiple formats (including orally, graphically, textually, and mathematically).
- Design, evaluate, and/or refine a solution to a complex real-world problem, based on scientific knowledge, student-generated sources of evidence, prioritized criteria, and tradeoff considerations.
- Evaluate a solution to a complex real-world problem, based on scientific knowledge, student-generated sources of evidence, prioritized criteria, and tradeoff considerations.
- Create a computational model or simulation of a phenomenon, designed device, process, or system.

Engineering Design Standards:

- HS-ETS1-2 Design a solution to a complex real-world problem by breaking it down into smaller, more manageable problems that can be solved through engineering.
- HS-ETS1-3 Evaluate a solution to a complex real-world problem based on prioritized criteria and trade-offs that account for a range of constraints, including cost, safety, reliability, and aesthetics, as well as possible social, cultural, and environmental impacts.

Learning Targets

Unit Essential Questions:

- How do conditionals make apps more interactive and adaptable?
- In what ways can parameters influence the behavior of a function in an app?
- Why is managing state important when building functional mobile apps?
- How does planning with pseudocode and layouts improve the app development process?
- What design choices make an app both functional and user-friendly?

Unit Enduring Understandings:

- Conditionals allow apps to respond dynamically to user input and changing data.
- Parameters make functions more flexible by allowing customization and reuse.
- Effective state management ensures apps run smoothly and provide accurate feedback.
- Careful planning with pseudocode and layout design leads to clearer, more efficient development.
- A balance of functionality and design is essential for creating engaging, user-friendly apps.

Vocabulary: conditionals, state, parameters, functions, components, events, pseudocode, stylesheet, layout, user interface, app planning, currency converter, dynamic behavior, logic, responsiveness

Students will be able to:

- use conditionals in their programs to tell the computer to complete one action over another

- use parameters inside functions to control conditional blocks
- create an app that converts between multiple currencies
- think critically about the user experience while using their app

Learning Activities & Differentiation: *(Amistad & Holocaust, Asian American & Pacific Islanders, Climate Change, LGBTQ and People with Disabilities, Diversity, Equity & Inclusion [DEI])*

Learning Activities:

- Conditional Balance Checks: Students build apps that use conditionals to check and update balances (e.g., Venmo Transfer activity), reinforcing how logic drives app behavior.
- Parameters in Action: Guided labs where students pass parameters into functions to determine outcomes, such as the Conditional Winner activity.
- Currency Converter Project: Students design and code a currency converter app using conditionals, parameters, and stylesheets to create an accurate and user-friendly interface.
- User Experience Planning: Students write pseudocode and sketch layouts to plan the most intuitive app flow for users.
- Peer Testing & Debugging: Students collaborate by testing each other's apps, offering feedback on usability and troubleshooting issues together.

Differentiation Strategies:

- Advanced Learners: Extend the Currency Converter project to include live exchange rates or multiple themes, integrating more complex conditional logic and layout customization.
- Struggling Learners: Provide partially completed app templates focusing on basic conditionals and single-parameter functions. Use flowcharts and graphic organizers to visualize logic paths.
- English Language Learners (ELLs): Offer bilingual glossaries for key terms such as conditionals, state, parameter, conversion, layout. Include annotated code examples with step-by-step explanations.
- Universal Design for Learning (UDL): Incorporate live coding demonstrations, visual diagrams of conditional flows, and multiple output formats such as functioning apps, annotated screenshots, or narrated video walkthroughs.

Education Connections:

- Connect conditionals in app design to real-world decision-making, showing how apps adapt based on user input.
- Encourage projects that highlight cultural awareness or global themes (e.g., converting currencies from different countries to discuss global trade or travel).
- Emphasize peer collaboration and debugging as skills that build teamwork, empathy, and problem-solving in both technology and community contexts.
- Showcase how designing inclusive, user-friendly apps supports accessibility and empowers diverse users.

Assessment Evidence

Formative Assessments:

- CodeHS exercises
  - Example activity: Venmo Balance Transfer: Check Balance
  - Example activity: Conditional Winner
- CodeHS Checks for Understanding

Summative Assessments:

- App Build: Currency Converter App
- Challenge: Scorekeeper Pt 6

Common Benchmark Assessments:

- App Build: Currency Converter App
- Challenge: Scorekeeper Pt 6

Alternative Assessments:

- CodeHS Unit Quiz

Core Instructional & Supplemental Materials

Core Materials:

- Built-in CodeHS React Native Editor with live mobile preview environment
- React Native syntax and component reference guides
- CodeHS tutorials on using conditionals, passing parameters into functions, and managing state updates
- CodeHS lessons on debugging conditional logic and improving program readability
- Guides to using the Stylesheet API for layout design, alignment, and currency conversion formatting
- Expo documentation for running and testing currency converter apps on physical devices

Supplemental Materials:

- Sample React Native code templates demonstrating conditionals, parameterized functions, and state changes
- Visual guides showing logic flow diagrams, pseudocode examples, and conditional branching structures
- Organizers for planning app logic and layouts, including wireframes and currency conversion flowcharts
- Debugging checklists and peer collaboration rubrics focused on troubleshooting conditional statements and state logic
- Example currency conversion and logic-driven app solutions to provide modeling and inspiration
- Accessibility and usability checklists to ensure layouts, labels, and conversion options are clear and user-friendly
- Rubrics for evaluating app projects on correctness, conditional logic, creativity, and design clarity
- Bilingual glossaries of key React Native terms such as “conditional,” “parameter,” “state,” “conversion,” and “layout” to support English Language Learners

Accommodations and Modifications: Students with Disabilities, English Language Learners, Students at Risk of Failure, Students with 504s, Gifted & Talented Students

Specific Strategies and Practices that Support Students with Disabilities:

Students receiving Special Education programming have specific goals and objectives, as well as accommodations and modifications outlined within their Individualized Education Plans (IEP) due to an identified disability and/or diagnosis. In

addition to exposure to the general education curriculum, the instruction is differentiated based upon the student's needs. The IEP acts as a supplemental curriculum guide inclusive of instructional strategies that support each learner.

- Additional time - Review of directions - Extra visual and verbal cues and prompts - Preferential seating - Follow a routine/schedule - Immediate feedback

Specific Strategies and Practices that Support English Language Learners

English language learner (ELL) is a national-origin-minority student who is limited-English-proficient. This term is often preferred over limited-English-proficient (LEP) as it highlights accomplishments rather than deficits.

- Pre-teaching of vocabulary and concepts - Extended Time - Simplified/Verbal Instructions - Frequent Breaks - Visual learning, including graphic organizers - Scaffolding - Word walls - Think-pair-share - Cooperative learning groups

Students at Risk of Failure

- Home/School Connection - Alternative Assessments - Learning Contracts - Homogenous Grouping - Supplemental Instruction - Think-Pair-Share by Readiness - Homework Options

Students with 504 Plans

Teachers are responsible for implementing designated services and strategies identified on a student's 504 Plan.
504 plans are formal plans that schools develop to give kids the support they need and covers any condition that limits daily activities in a major way. Students' accommodations are outlined in their individual 504 plan.

Specific Strategies and Practices that Support Gifted & Talented Students

It is critical that all teachers are able to recognize a high-ability student who may need more depth and complexity in instruction.

-Homogenous Grouping - Small Group In-Depth Instruction - Alternative Assessments