Austin Walter
Scott Rostron
Onan Chew

Project Owner: Dr. Warren-Seeger

University of South Carolina Aiken

UX Development and Design Plan

Modular Platform Design in Web Development

September 4, 2023

Requirements Gathering:

- Scope Definition:
  - Understand the scope of the project. What are the key features required in the UI? What are the user's roles? What action can they perform?
- Content Audit:
  - Review any existing interfaces, content, or brand guidelines that the new design should adhere to.

UI Design in Figma.

- Set up a New Project in Figma
  - Use Figma's team libraries to keep components consistent.
- Design The wireframes and Sitemap/User Flow Diagram:
  - Create a Sitemap to demonstrate the structure of the webpage.
  - Create User Flow Diagram to demonstrate the capabilities of each user type.
  - Create low-fidelity sketches of the user interface. Should represent the skeletal framework of the UI.
- Design High-Fidelity Mockups
  - Based on feedback from the wireframes, design high fidelity mockups with detailed design specifications
- Design Interactive Prototypes:
  - Link the mockups together to create a clickable prototype. This helps in understanding user flows and interactions.
- Gather Feedback:
  - Share protype with Project Owner and potential users (Students, and Professors)
- Prepare Assets for Development:
  - Make sure to export any assets (e.g., images, icons) needed for development. Plugins like 'Figmify' to convert Figma designs to React components.

Setting Up React:

- Initialize a New React APP, GitHub repository.
- Set up Necessary Libraries

- Directory Structure Definitions:
    - Organize file structure for components, assets, utilities, and services in a logical and agreed manner.

Developing the UI with React

- Break Down Figma Designs: Identify reusable components from Figma design. For instance, button input fields, modals, etc.
- Develop Base Components: Create these reusable components in React. Try to make them as modular and reusable as possible.
- Integrate Components to Build Views: Use the base components to construct full views/pages in React.
- Style the Components: Use css-in-js libraries or traditional css/scss. Aim to maintain consistency with the Figma design. Use global style variables to maintain consistency and to enable theme changes.
- Interactivity: Implement user interaction such as form validation, animation, and state transitions.
- Connect to backend: Set up connection to the backend using external libraries.

Testing and Review

- Functional Testing: Ensure that all functions (buttons, forms, links) work as expected.
- Visual Testing: Confirm that the developed UI matches the Figma design. Tools like "Percy" can help automate visual testing.
- Performance Testing: Use tools like Google's Lighthouse to ensure the React app performs well, especially on mobile devices.
- Accessibility Testing: Ensure your UI is accessible to all users, including those with disabilities. Use tools like 'axe'

Deployment

- Optimize Build: Minify CSS and JavaScript, optimize images, and ensure lazy loading if necessary.
- Deployment: Use platforms like Vercel, Netlify or traditional cloud providers like AWS, Azure, etc.