

## CIT 312 Final Project Documentation

Kayton McKinney

Date: May 7, 2025

### 1. How Less/SCSS has been used

I created a style.less file to manage branding, layout, and consistent design choices. I defined variables for all Purdue colors (like @boilermakerGold and @steam) and font stacks (like @font-primary for Franklin Gothic). I used LESS to nest rules and avoid repetitive code, which made the stylesheet easier to maintain. For example, I added hover states using darken(@black, 10%) to make buttons interactive.

### 2. How responsiveness has been achieved

I used Bootstrap 5's grid system, layout utilities, and responsive navbar components. I structured most content with container, row, and col-md-\* classes to ensure the layout adapts across screen sizes. I also used utilities like img-fluid, d-flex, and justify-content-center to keep images and text aligned and scaled. The navbar uses navbar-expand-lg and navbar-toggler to turn into a hamburger menu on smaller screens.

### 3. The rationale behind creation of the favicon

I created my favicon using an online generator to represent web programming for CIT Purdue students. I used Purdue's brand colors to keep the look consistent with my project. I wanted something that related to web design or programming, so it resembles a webpage.

### 4. The rationale behind use of the 4 chosen Bootstrap features

I used Bootstrap to help speed up development and maintain accessibility and responsiveness. Here's why I chose each feature:

- Navbar with Toggler: I used this to help users navigate easily on mobile and desktop.
- Cards with Ratio: I used cards to display each CIT pathway in a clean, consistent way, and ratio classes to keep icons square and centered.
- Modals: I used modals so students could learn about each pathway without leaving the page.
- Buttons: I styled all buttons using btn, btn-dark, and my LESS variables to make them match Purdue's colors and feel interactive.

### 5. How I Used Semantic HTML and Accessibility

I wrote my HTML with semantic structure in mind using <main>, <section>, <footer>, <form>, and more. I also included aria-label attributes and role="button" for better screen reader

support. I did this because I wanted to follow best practices for accessibility and make the site usable for everyone.

## 6. How I Built the Interactivity

I used plain JavaScript to connect the quiz to the homepage. When someone finishes the quiz, I save their result in `localStorage`, and when the homepage loads, the corresponding modal opens automatically. I also let users click on each quadrant or page in the navbar to learn more about each pathway. This approach makes the app more engaging and informative.