## WEB DEVELOPMENT CAPSTONE PROJECT GUIDELINES | JANUARY 2025 COHORT\*

### TABLE OF CONTENTS

[Project Overview](#_1uytvnn98ofq)

[Expectations by Module](#_4vhqws69lmqm)

[Project Requirements](#_c5r3ieqk3z5y)

[Feature List](#_4rsrrxk1ywys)

[Review process](#_q72oogyeen5c)

[Additional Resources](#_h1f02ypxzk5x)

## Project Overview

### **Project Deadline:** Submission by 11:59 AM (Noon) EST on the last Friday of Module 4.

Your projects are your number one way to show off your skills and land a job as a developer. This document will provide an outline of the minimum requirements to pass Code:You. We encourage you to go well beyond what is detailed here, show your progress to your mentors regularly, and strive for something you want to use to impress employers.

Please read this document carefully and be sure to ask any questions you may have early on.

#### 

## Expectations by Module:

**Module 3:** Develop your project plan during this phase. Regularly consult with mentors to refine your ideas and implementation strategies. Turn in the project plan assignment.

**Module 4 (Capstone):** The "Turn In Form" for your project will have been distributed. Submit your project before the due date. You can continue working up until the deadline. Ensure your project is thoroughly tested and reviewed by a mentor.

## Project Requirements

To pass your project, ensure all the following criteria are met. Maintain regular communication with your mentors to stay aligned with these guidelines.

### **Capstone Presentation and Interview**

**1. Capstone Project Overview:**

* Be prepared to explain your capstone project.
* Discuss the problem you aimed to solve and how your project addresses it.

**2. Code Functionality and Use:**

* Demonstrate how your code supports your project’s functionality.
* Highlight a specific section of your code that you’re particularly proud of or found challenging to implement.

**3. Future Plans and Reflection:**

* Reflect on your experience with Code:You.
* Share a key skill or lesson you’ve learned and how it will influence your future projects or career.

### **GitHub Repository and Commits**

Source control is a cornerstone of development, enabling version tracking, collaboration, and project management. GitHub provides developers with tools to track changes, collaborate, and present their work professionally.

**Requirements:**

* Upload your project to a GitHub repository with at least **10 distinct commits**.
* **Commits and pushes must be done through the terminal/command line**. Commits pushed through Git Hub’s “File Upload” button will not be counted.
* Use Git commands consistently to demonstrate ongoing progress, not just a final upload.

### **README File**

A README file serves as a roadmap for your project, providing essential details to ensure clarity and accessibility for collaborators and reviewers.

**Requirements:**

* Write a detailed README file explaining your project in at least one paragraph.
* Identify and describe **three or more features** you integrated from the provided list.
* Make a fetch request to an API and use the response in a meaningful way
* Include any special instructions for reviewers to run your project.

### **Visual Appeal**

In development, the user experience (UX) and visual design are just as critical as technical functionality. Your project should be engaging, visually cohesive, and aligned with industry standards.

**Requirements:**

* Design your project to be **visually appealing**, following current industry trends.
* Ensure text is correctly spelled, legible, and consistent across all components.
* Use a clear and consistent design for headings, buttons, forms, and interactive elements.
* Research other applications and websites for inspiration. Emulate styles and functionalities that enhance usability and appeal.
* Choose a **color palette** and **font stack** to create a polished and professional design.

By meeting these standards, your project will demonstrate both technical proficiency and design sensibility, making it attractive to mentors, reviewers, and potential employers.

## Web Development Project Requirements

### Responsive Design:

Implement responsive design using media queries, CSS Grid, Flexbox, etc.

Your application should adapt to at least two screen sizes (mobile and desktop).

Integrate responsive design as a testament to your JavaScript, HTML, and CSS integration skills.  
  
Feature Implementation:

* **Integrate a third-party API into your project (MANDATORY)**. **Using a Weather API will not count.**
* **At least one media query to make your site responsive.**
* Choose at least 3 items from the first table. You may substitute requirements from the first table with requirements from the second table.
* Consider adding a 4th feature as a backup plan.
* Failure to meet all requirements will result in incomplete status for the class.

## Web Development Capstone Features List

1. **Choose at least three of the following:**

| **FEATURE** | **DIFFICULTY** |
| --- | --- |
| Analyze data that is stored in arrays, objects, sets or maps and display information about it in your app. | Easy |
| Use a regular expression to validate user input and either prevent the invalid input or inform the user about it (in all cases prevent invalid input from being stored or saved). | Easy |
| Create a function that accepts two or more input parameters and returns a value that is calculated or determined by the inputs. Basic math functions don’t count (e.g. addition, etc). | Easy |
| Visualize data in a user friendly way. (e.g. graph, chart, etc)  This can include using libraries like ChartJS | Easy/Intermediate |
| Convert user input between two formats and display the result. (e.g. Fahrenheit to Celcius, kilograms to pounds, etc) | Easy/Intermediate |
| Calculate and display data based on an external factor (ex: get the current date, and display how many days remaining until some event) | Intermediate |
| Persist data to an internal API and make the stored data accessible in your app. (including after reload/refresh).  This can be achieved either by using local storage or building your own API that stores data into a JSON file. | Intermediate/Hard\* |

1. **Can swap with items from section one:**

| **FEATURE** | **DIFFICULTY** |
| --- | --- |
| Create a node.js web server using a modern framework such as Express.js. | Easy/Intermediate |
| Interact with a SQLite database to store and retrieve information | Intermediate/Hard |
| Implement modern interactive UI features (e.g. table/data sorting, autocomplete, drag-and-drop, calendar-date-picker, etc). | Intermediate |
| Develop your project using a common JavaScript framework such as React, Svelte, or Vue. | Intermediate - Hard |
| Create 3 or more unit tests for your application (and document how to run them) | Intermediate - Hard |

#### Review Process:

Our project reviewers consist of mentors, the training team and other staff. We will evaluate the project based on a general rubric for capstone projects. Projects must meet all the requirements to pass the session. Having a passing project is only part of the requirements to graduate a session or class.

Projects are evaluated through the following steps:

* The project reviewer clones your project using Git.
* They follow your readme instructions to set up/run your project.
* They will test your app for the requirements in this document and also look through the code
* Reviewers will try only the most minimal troubleshooting steps if your project does not run “out of the box”

### Testing Your Project:

Sharing and testing your capstone project offers invaluable benefits for both personal and professional growth. By sharing your project with peers, mentors, or industry professionals, you gain valuable feedback and insights that can help refine and improve your work. Embracing the principles of sharing and testing not only enhances the quality of your project but also cultivates essential skills in communication, collaboration, and problem-solving, paving the way for success in your future endeavors.

Successful testing includes:

* Identification of potential issues or bugs
* Verification of functionality and performance
* Ensuring compatibility across different devices and platforms
* Evaluation of user experience and interface usability
* Confirmation of security measures and data integrity
* Validation of adherence to project requirements and specifications
* Gathering feedback for improvement and refinement

The reviewers **MUST** be able to run your project for it to pass. For your project to pass review, ensure that reviewers can run it smoothly. Test thoroughly and provide clear instructions for setup. Maintain modularity and portability, avoiding database dependencies like SQLServer. Opt for portable solutions like SQLite to enhance accessibility and ease of use.

### Mentor Engagement:

Regularly consult with mentors about your project. Share your ideas early and ensure you're on track to a successful project.

### **Demo Day:**

After the completion of the pathways, Code:You will host a Demo Day for graduates of the program to show off their projects to the community. Employers, mentors, and other members of the tech community will be invited to see the projects and meet you - the developer of that project!

Presentation slots for the event will be invite-only, as we will not have enough time for every Code:You student to participate (sorry, there’s hundreds of projects!). To incentivize you to do as well as you can on your project, invites to present at Demo Day will be based on the most impressive projects to employers. What determines that selection is ultimately subjective - how do you compare the visual presentation of one project to the technical skill level of another? But we will strive to be as fair as we can during the process.

### **Policy for using AI/Machine Learning Prompts for writing code:**

AI should not replace essential human elements of writing and reviewing code. Students are expected to use AI technologies ethically, respecting copyright laws, privacy norms, and the intellectual property rights of others. AI resources should be used responsibly and not abused for the ease of use in creating original content. AI should not be used to engage in plagiarism, cheating, or any form of dishonesty. AI work must not be submitted in the place of student work.

AI tools should be used as an aid rather than a substitute for human expertise. Students are accountable for the code generated with AI assistance, and it must adhere to existing coding standards and practices. All AI-generated code should be scrutinized to ensure that it aligns with project requirements, just like any manually written code.

Failing to adhere to these rules may result in not passing the class. 0

**Documentation**

Proper documentation is vital for any well-maintained coding project, but it’s imperative when introducing AI-generated code into your work.

Documenting the AI-generated code helps make the project more transparent. It allows staff and mentors to understand how AI models are utilized within the codebase.

**If AI tools are used to write code, it must be documented in the Read-Me and in comments in the code.** This documentation should cover the purpose of the code and the manner of usage.

**Best Practice: Encapsulate AI-Generated Code**

Encapsulate AI-generated code into defined modules or functions to improve readability and usability immediately. Code encapsulation means wrapping a piece of code, typically a set of statements or functions, within a higher-level structure such as a function, class, or module. The encapsulated code is then treated as a single unit with a well-defined interface and can be invoked or interacted with as a cohesive entity.

Encapsulation promotes modular design by breaking down complex systems into smaller, manageable units. Each encapsulated unit can be developed, tested, and maintained independently, making the codebase more modular and easier to understand.

Encapsulated units can also be reused in different parts of a program or other projects.

### Additional Resources

For additional support in your coding project, consider tapping into online communities like [Stack Overflow](https://stackoverflow.com/) for problem-solving, using version control systems such as Git and GitHub for collaboration, and consulting comprehensive documentation like [MDN](https://developer.mozilla.org/) for guidance. Exploring relevant open-source projects can also offer inspiration and resources. Integrating these tools and resources will streamline your project's development process and enhance its quality.

* **The Ultimate Design Principles Guide For Developers -** <https://bootcamp.uxdesign.cc/the-ultimate-design-principles-guide-for-developers-d4aa58937283>
* **Adobe Colors** - <https://color.adobe.com/create/color-wheel>
* **Coolors** - <https://coolors.co/>
* **An Interactive Guide to Flexbox** - <https://www.joshwcomeau.com/css/interactive-guide-to-flexbox/>
* **An Interactive Guide to CSS Grid** - <https://www.joshwcomeau.com/css/interactive-guide-to-grid/>
* **Regex Cheat Sheet** - <https://www.keycdn.com/support/regex-cheat-sheet>
* **SOLID: The First 5 Principles of Object Oriented Design** - <https://www.digitalocean.com/community/conceptual-articles/s-o-l-i-d-the-first-five-principles-of-object-oriented-design>