

CPSC 5260 Foundations of Software Engineering

Winter Quarter 2026

Individual Project

RefactorBot – An AI-Powered Code Refactoring Engine

1. Project Vision and Overview

In modern software development, technical debt and legacy code are significant challenges. This project aims to tackle this problem head-on by leveraging Artificial Intelligence. You will design and build a **web application** that acts as an intelligent code refactoring assistant.

The core functionality is simple yet powerful: A user provides a snippet of source code, and the application returns a refactored, improved version of that code. The goal is not just to make code work, but to make it cleaner, more efficient, and more maintainable according to established software engineering principles.

2. Learning Objectives

Upon completion of this project, you will be able to:

- Design and architect a full-stack web application.
 - Integrate a third-party AI/ML API into a software product.
 - Write clean, maintainable code and, crucially, refactor code effectively.
-

3. Project Requirements & Features

Core User Stories:

- **Story 1:** As a user, I want to paste my source code into a text area on a web page, so that I can submit it for refactoring.
 - **Story 2:** As a user, I want to click a "Refactor" button and see the refactored code displayed clearly on the same page, so that I can immediately review the changes.
 - **Story 3:** As a user, I want to upload a source code file directly, so that I don't have to copy and paste large files.
 - **Story 4:** As a user, I want to read a brief explanation from the AI justifying the refactoring changes, so that I can learn from the process.
-

- **Story 5: As a system, I must** send the user's code to a configured AI service, **so that** I can receive a refactored version.
- **Story 6: As a user, I want to** see a side-by-side diff view of the original and refactored code, **so that I** can easily identify the changes.

IMPORTANT: You cannot apply any AI-tools to build the application. You can only call the API of AI tools to refactor your input code.

4. Project Grading Breakdown

Final Presentation and Demo Breakdown	Points
1. Core user story 1 is implemented	2
2. Core user story 2 is implemented	2
3. Core user story 3 is implemented	2
4. Core user story 4 is implemented	2
5. Core user story 5 is implemented.	2
6. Core user story 6 is implemented.	2
7. Clear UI is provided.	3
Total	15

5. Project Grading

On February 11, 2026, we will hold an individual mid-point project check-in. This is not a grading activity, but you are expected to have started your project by that date (regardless of how much progress you have made). By the mid-point check-in, you should ideally have a basic UI implemented and have begun working on the back end of your application.

Zero implementation for your individual project during the mid-point check will result in a 0 for your individual project in the end.

During the final week of the quarter, you will present your implementation in a 1-on-1 meeting with Nayan. A detailed meeting schedule will be released during the second-to-last week of the quarter.