

Part 2 - Homework: AI-Assisted Operations

This assignment transitions from generating code to managing the entire development lifecycle with AI. You will learn to automate the operational tasks that surround coding, such as code reviews and pull requests.

Learning Goals

By completing this assignment, you will learn to:

1. **Implement AI code reviews** to automatically catch inconsistencies, common bugs, and style issues in your pull requests.
2. **Automate simple Git and GitHub operations** by delegating tasks like writing commit messages and creating pull requests to an AI agent.
3. **Exercise precise control over AI agents** to perform well-defined tasks, ensuring you maintain full authority over the process.
4. **Apply a security-first mindset** by using practical safeguards when granting an AI agent the autonomy to interact with your repository.

The Challenge

Your task is to integrate AI into the workflow of the MICE Quotient web application you built in Part 1. Instead of writing new features, you will configure your repository to use AI for quality control and process automation. You will set up an automated AI code reviewer and then direct an agent to perform a cleanup task and submit its own work as a pull request.

What Success Looks Like

- An **AI code review tool** (e.g., GitHub Copilot, Claude) successfully configured on your repository.
- A **pull request created by you** that has received and displayed automated feedback from the AI reviewer.
- A second **pull request created entirely by an AI agent** for a simple cleanup or refactoring task, demonstrating your ability to delegate operational work.

Core Requirements**

Your assignment must meet the following requirements:

1. **Fork the Repository (if you haven't already):** All work should be done in your personal fork of the course repository.

2. **Configure an AI Pull Request Reviewer:** Add and enable an AI-powered tool that automatically reviews new pull requests. GitHub Copilot is a great option, but you can also use alternatives
3. **Submit a PR for AI Review:** Make a small, manual change to your project (e.g., refactor a function, add comments, fix a typo). Create a pull request to see your AI reviewer in action. Analyze its feedback for quality and relevance.
4. **Delegate PR Creation to an Agent:** Instruct an AI agent to perform a small, self-contained task and create a pull request for it.

Key Concepts to Explore

- **AI-Assisted Code Review:** The goal is not to replace human reviewers but to augment them by letting an AI handle the first pass for routine checks. Remember that the quality of a review depends heavily on the context you provide, such as coding standards and the specific focus for the review.
- **Spec-Driven Development:** While this homework focuses on operations, the principle from Lesson 3 remains critical. A clear, specific prompt or “spec” is the most effective way to guide an agent to a successful outcome, whether it’s writing a feature or creating a pull request.

Getting Started

1. **Choose Your Review Tool:** Select an AI code review tool. The easiest to start with is the native functionality in GitHub Copilot.
2. **Configure the Tool:** Follow the documentation to install and configure the tool on your forked repository.
3. **Create Your Manual PR:** Create a new branch (`git checkout -b my-manual-pr`), make a small code improvement, commit your changes, and open a pull request on GitHub.
4. **Observe and Learn:** Wait for the AI reviewer to comment on your PR. Evaluate its suggestions. Are they helpful? Do they align with your project’s standards?
5. **Delegate the Agent PR:** In a github issue, ask your AI assistant to run a cleanup task and create the PR, providing a clear and specific prompt.
6. **Verify the Result:** Check GitHub to ensure the agent-created PR was made correctly, with the right title, description, and changes. Merge both PRs once you are satisfied, getting hands on with the code yourself if needed to ensure quality.