# Report 4: AI + CI/CD + Testing Report (Integration Phase)

## AI Feature Integration

Describe the AI functionality integrated (e.g., recommendation, automation).

Explain the algorithms, models, or APIs used.

Include screenshots or code snippets if applicable.

## CI/CD Pipeline Setup

A Continuous Integration/Continuous Deployment (CI/CD) pipeline is set up using GitHub Actions to automate the build and testing process.

* Tools: Git, GitHub, and GitHub Actions.
* Pipeline Steps: The workflow is defined in the .github/workflows/dotnet.yml file and executes on every push to the main branch or pull request.
  1. Checkout: The pipeline checks out the latest source code.
  2. Setup .NET: It installs the required .NET 8 SDK.
  3. Restore Dependencies: It runs dotnet restore on the solution to download all necessary NuGet packages.
  4. Build: It compiles the entire solution (PRN\_Project\_Coffee\_Shop.sln) in Release mode using dotnet build.
  5. Test: It runs all unit tests in the CoffeeShop.Tests project using dotnet test. If any test fails, the pipeline fails.
* Pipeline Configuration Link: The pipeline is configured in the repository at the following path: .github/workflows/dotnet.yml.

## Deployment Workflow

Explain how the system was deployed to staging or production.

Include deployment frequency and automation levels.

## Collaboration and Automation

* Collaboration: As a solo project, coordination is straightforward. The CI/CD pipeline serves as an automated quality gate. Pull requests are used to ensure that changes pass all automated checks before being merged into the main branch.
* Automation: GitHub Actions automatically runs the build and test suite for every change, providing immediate feedback on whether a commit introduces a regression. This removes the need for the developer to manually run all tests before every commit.

## Lessons Learned

* AI Integration: Integrating even simple AI features requires careful consideration of the data model. The current recommendation logic works but could be slow on a large dataset. A key lesson is that performance must be a primary consideration, and more advanced techniques (like pre-calculating recommendations in a background job) would be needed for a production system.

 Automated Testing: Setting up the CI pipeline early was crucial. It helps catch issues where a change in one part of the code unexpectedly breaks a feature elsewhere. The main takeaway is that a robust test suite is essential for maintaining code quality. The current test coverage is a known weakness, and more effort is needed to write comprehensive tests, especially for business logic..

## Appendix (Optional)

Include full test logs, model performance charts, or pipeline files if needed.