

# Week 2 Report – Continuous Integration (CI) Setup

## Objectives

- Configure continuous integration using **GitHub Actions**.
- Implement automated build and testing processes.
- Integrate linting and code quality tools.

## Summary of Activities

### 1. CI Configuration

- A `.github/workflows/ci.yml` file was created by **Aine Levi** to automate builds and run tests on each push to the `main` branch.
- The CI workflow installs project dependencies, runs Django unit tests, and builds the React frontend.

### 2. Testing Framework Integration

- **Tusiime Emmanuel** - set up initial Jest tests for the React frontend.
- **Aine Levi** - configured Django test cases for the API endpoints.
- **Otim Ronald** assisted in defining mock test data and verifying API responses.

### 3. Code Quality and Linting

- **Ssentongo** - implemented **ESLint** and **Prettier** in the frontend for code style enforcement.
- Automated linting runs as part of the CI pipeline to ensure code consistency.

### 4. ClickUp Updates

- All CI-related tasks were updated and marked in progress. Each member's assignments reflected task completion and dependencies.

## Deliverables

- GitHub Actions workflow (`ci.yml`)
- Functional CI pipeline that runs automated builds and tests
- ESLint + Prettier integration for code quality

## Challenges & Future Plans

Minor issues arose due to dependency version mismatches in the workflow runner. The team resolved this by pinning dependency versions in `package.json` and `requirements.txt`. Week 3 will focus on automating deployment to a staging environment.