Plan for the Assignment

Phase 1: Learning Phase (30-40 mins)

1. **Objective:**
    - Gain foundational knowledge of GitHub Actions and CI/CD practices.
2. **Steps:**
    - Watch the provided video tutorial.
    - Take notes on:
        - How to set up a GitHub Actions workflow.
        - Key components: triggers (on), jobs, steps, runners.
        - How to implement automated testing and deployment.
    - Discuss with your team after watching to clarify concepts.

Phase 2: Implementation Phase (80 mins)

Step 1: Team Planning (10-15 mins)

- **Divide Responsibilities:**
    - Assign team members to focus on:
        1. **CI/CD Pipeline Setup:** Setting up the .yml file for GitHub Actions.
        2. **Testing:** Writing or verifying automated test cases using pytest.
        3. **Feature Development:** Implementing or enhancing features from your sprint backlog.
        4. **Code Review & Linting:** Ensuring code quality via flake8 or similar tools.

Step 2: CI/CD Pipeline Implementation (20-30 mins)

- **Objective:** Set up a basic CI/CD pipeline to automate:
    1. **Code Integration:**
        - Ensure every code change pushed to the repository triggers the pipeline.
    2. **Automated Testing:**
        - Use pytest to validate critical functionality.
    3. **Deployment:**
        - (Optional for this phase) Simulate or deploy to a mock environment.
- **Pipeline Tasks:**
    - Create a .github/workflows/ci-cd-pipeline.yml file.
    - Define these steps in the pipeline:

1. **Set up Python Environment:** Install Python dependencies (requirements.txt).
2. **Linting and Formatting:** Run flake8 to check code quality.
3. **Automated Testing:** Use pytest to execute test cases.
4. **Deployment:** Add a placeholder for deployment (e.g., SCP or Docker).

Step 3: Feature Development (20-30 mins)

- Implement or enhance functionalities from your sprint backlog.
    - Examples for your project:

        - Add new stock management features.
        - Improve GUI usability.
        - Fix bugs related to database interactions or user authentication.

Step 4: Testing and Debugging (15 mins)

- Run the CI/CD pipeline to identify issues.
- Debug errors in tests, code, or pipeline steps.

Step 5: Documentation and Submission (10 mins)

- Write the **Reflection Document** covering:
    - Challenges faced during setup (e.g., syntax issues in .yml).
    - Learning outcomes from CI/CD implementation.
    - Benefits observed in your development workflow.
- Push the final implementation and reflection document to GitHub.

Phase 3: Reflection and Submission

- Finalize the reflection document with team input.
- Review the GitHub repository for completeness.
- Submit the repository link and reflection document.