DevOps

Assignment - 01: Collaborative API Development Using Git & GitHub[CLO – 01]

You are required to work in groups of **three students** to develop an API for a **Weather Forecasting Application** using **Flask** (Python). The primary focus is to showcase your ability to collaborate effectively using Git and GitHub. Each group member will have a distinct role, and you will integrate your work into a single repository.

Assignment Tasks:

1. API Development:

- Build a simple API using Flask. For example, an endpoint /weather should return basic weather information such as temperature, city, and description.
- Each student must work on a separate task based on the roles assigned above.

2. Git & GitHub Collaboration:

- Set up a **GitHub repository** and invite your group members as collaborators.
- **Create branches** for each member's individual tasks (e.g., feature-api-development, feature-server-setup, feature-tests).
- Ensure that each member regularly commits their changes and pushes them to their respective branches on GitHub.
- **Collaborate through pull requests**: Once a member's task is complete, they must submit a pull request to merge their branch into the main branch. The other group members must review the code before merging.

3. Handling Merge Conflicts:

- During the integration phase, if there are any **merge conflicts**, work together to resolve them.
- Document how your team handled the conflicts and integrated changes smoothly.

4. Final Testing and API Documentation:

- After all features are merged, ensure the API works as intended by testing each endpoint.
- The student responsible for testing should create unit tests for the API.
- Provide proper **documentation** in the repository's README.md file explaining how the API works and how to run it.

Deliverables:

1. **GitHub Repository:**

- A single repository with all code, commits, and branches clearly showing the collaboration process.
- Pull requests showing each team member's contributions.
- The final, working version of the API code.

2. **Documentation:**
   - A detailed README.md file that explains:
     - The purpose of the API.
     - How to install dependencies and run the API.
     - How each member contributed to the project.
     - Steps for testing the API.

Grading Criteria:

- **API Functionality (40%)**: The API should run without errors and provide correct responses.
- **GitHub Collaboration (40%)**: Demonstrate effective use of Git and GitHub, including branches, commits, pull requests, and merge conflict resolution.
- **Testing and Documentation (20%)**: Ensure proper testing of the API and clear, concise documentation.

Submission Deadline:

- The final submission (repository link) should be uploaded by **October – 12 -2024**

Note: This assignment is focused not only on creating the API but also on how well your group can work together using Git and GitHub. Successful collaboration will be a key factor in your grade.