**CI Task Description: Enhancing the Flask Calculator Application**

**Objective:**

You will enhance an existing Flask-based calculator application by adding a new feature or improving an existing one. The main branch contains a basic calculator that performs addition, subtraction, multiplication, and division. Your task is to fork the repository, create a new branch, and implement your enhancement. You will use GitHub Actions to test your code, ensure code linting, and submit a pull request (PR) for review.

---

**Task Steps:**

1. Fork the Repository:

- Fork the main repository to your GitHub account.

- Clone your forked repository to your local machine.

2. Create a New Branch:

- Create a new branch in your forked repository with a descriptive name (e.g., `feature/add-exponentiation`).

3. Install Dependencies:

- Install the required dependencies by running:

4. Enhance the Calculator:

- Add a new feature or improve an existing one. For example:

- Add exponentiation (`a^b`).

- Add a square root function.

- Improve error handling for division by zero.

- Add a history feature to track previous calculations.

- Ensure your code follows PEP 8 guidelines for linting.

5. Commit Code at Logical Checkpoints:

- Commit your code at logical checkpoints with meaningful commit messages (e.g., "Added exponentiation feature" or "Fixed division by zero error").

6. Set Up GitHub Actions:

- Create a GitHub Actions workflow file (`.github/workflows/ci.yml`) in your branch to:

- Install dependencies.

- Run unit tests.

- Perform code linting using `flake8` or `pylint`.

7. Test Your Code:

- Push your changes to your branch and ensure the GitHub Actions workflow runs successfully.

- Fix any issues identified by the tests or linting.

8. Submit a Pull Request:

- Once your code is ready, submit a PR from your branch to the `main` branch of the original repository.

- Include a detailed description of your changes in the PR.

9. Await Review:

- You do not have permission to merge your code into the `main` branch. A reviewer will assess your PR and provide feedback.

**Requirements:**

- Your code must pass all GitHub Actions workflows (tests and linting).

- Commit messages must be clear and descriptive.

- Your PR description must explain the changes and how they enhance the application.

- Follow PEP 8 guidelines for Python code.

**Deliverables:**

1. A new branch in your forked repository with the enhanced feature.

2. A GitHub Actions workflow file that runs tests and linting.

3. A pull request to the main repository with a detailed description of your changes.