🚀 Beginner Exercise: Build a GitHub Actions Pipeline with Python, pytest, and Heroku Deployment

---

🎯 Goal

Create a **GitHub Actions pipeline** based on the following requirements. The pipeline will:

1. Set up **Python 3.9**.
2. Install **dependencies** using `pip`.
3. Run **tests** using `pytest`.
4. **Deploy the application** to **Heroku**.

📋 Requirements

1. Trigger: Manual Workflow

- The pipeline must **only run manually** using the `workflow_dispatch` trigger. This means the workflow will only be triggered when started manually from the GitHub web interface.

2. Jobs and Steps

Job 1: Build and Test the Application

- **Runs on** `ubuntu-latest`.
- **Step 1**: Check out the repository using `actions/checkout@v4`.
- **Step 2**: Set up **Python 3.9** using `actions/setup-python@v4`.
- **Step 3**: Install dependencies using `pip install -r requirements.txt`.
- **Step 4**: Run **tests** using `pytest`.

Job 2: Deploy to Heroku

- **Runs on** `ubuntu-latest`.
- **Step 1**: Check out the repository using `actions/checkout@v4`.
- **Step 2**: Deploy the application to Heroku using `akhileshns/heroku-deploy@v3.12.12` (you can treat this as an action when using GitHub Actions).

✅ Results:

- The first job should complete **successfully**.
- The second job should run but **fail**, since no **authentication details** were provided.

- The first job should complete **successfully**.
- The second job should run but **fail**, since no **authentication details** were provided.