Experiment 8: Implement CI/CD pipeline for automated deployment of an AI application

Learning Objective: Students should be able to understand how to implement a Continuous Integration and Continuous Deployment (CI/CD) pipeline for automating the deployment of an AI application.

Tools: GitHub Actions, Jenkins, Docker, Python, VS Code, AWS/GCP/Azure (optional)

Theory:

CI/CD is a software engineering practice that automates the process of integrating code changes, testing, and deploying applications. Continuous Integration (CI) ensures that code changes are automatically tested and merged into a shared repository, while Continuous Deployment (CD) automates the deployment of the application to production or staging environments.

In this experiment, we will build a simple AI application, set up a CI/CD pipeline using GitHub Actions, and automate the process of building, testing, and deploying the application using Docker containers. This ensures fast and reliable deployment while minimizing human intervention.

CI/CD stands for **Continuous Integration** and **Continuous Deployment**. These are modern software engineering practices that automate parts of the development process, ensuring faster and more reliable software delivery.

- **Continuous Integration (CI)**:
  - The practice of frequently integrating code changes into a shared repository.
  - Every time a developer commits new code, it is automatically tested and merged to detect any issues early.
  - This ensures that new features or bug fixes do not break the existing application.
  - CI includes tasks such as building the code, running tests, and reporting errors.
- **Continuous Deployment (CD)**:
  - A practice where code that passes the testing phase is automatically deployed to production or staging environments without manual intervention.
  - This makes deployments more frequent and allows faster delivery of new features, bug fixes, or updates to users.

Significance of CI/CD in AI Development:

CI/CD becomes particularly important when developing and deploying AI applications because of the complexity involved in training models, running tests, and ensuring that the models work in production environments. This experiment will show how CI/CD pipelines can be utilized to automate the deployment of an AI model and improve development efficiency, reduce errors, and accelerate delivery.

The goal is to:

- Automate the testing of the AI model.
- Build Docker images for easy deployment and replication.
- Deploy the application into production using a cloud platform

Command & Output:

1. **Set up Git repository and push the AI application**

```
git init
git add .
git commit -m "Initial commit of AI application"
git remote add origin <repository-url>
git push -u origin main
```

2. **Create a GitHub Actions workflow (`.github/workflows/cicd.yml`)**

```
name: CI/CD Pipeline
on:
 push:
    branches:
   - main
jobs:
  build:
   runs-on: ubuntu-latest
    steps:
      - name: Checkout code
        uses: actions/checkout@v2
     - name: Set up Python
     uses: actions/setup-python@v2
        with:
       python-version: '3.9'
   - name: Install dependencies
        run: pip install -r requirements.txt
   - name: Run tests
        run: pytest
  deploy:
    needs: build
    runs-on: ubuntu-latest
   steps:
  - name: Deploy to Server
        run: echo "Deploying AI application..."
```

3. **Create a Dockerfile for containerizing the AI application**

```
FROM python:3.9
WORKDIR /app
COPY . /app
RUN pip install -r requirements.txt
CMD ["python", "app.py"]
```

4. **Build and run the Docker container**

```
docker build -t ai-app .
docker run -p 5000:5000 ai-app
```

5. **Automate deployment to a cloud platform (AWS/GCP/Azure) using CI/CD pipeline**
   - Configure cloud credentials in GitHub Secrets.
   - Add deployment scripts to the CI/CD workflow.

Conclusion:

Viva Questions:

1. What is CI/CD, and why is it important?
2. What are the key differences between Continuous Integration and Continuous Deployment?
3. How does GitHub Actions automate the deployment process?
4. What is the purpose of a Dockerfile in a CI/CD pipeline?
5. How can CI/CD be integrated with cloud platforms like AWS, GCP, or Azure?

For Faculty Use:

Correction Parameters	**Formative Assessment [40%]**	**Timely completion of Practical [40%]**	**Attendance/ Learning Attitude [20%]**	
Marks Obtained				