**Report on FastAPI Deployment and Selenium Testing Sessions**

**Session 1: FastAPI Deployment on Render**

This session focused on deploying a portfolio website built using FastAPI on the Render platform. The key activities included:

**1. Setting Up the Virtual Environment**

- A virtual environment was created using the following commands:
- python -m venv ~/profile
- source ~/profile/scripts/activate
- FastAPI and its standard dependencies were installed using:
- python -m pip install "fastapi[standard]"

**2. Building the FastAPI Application**

- Navigated to the project directory and created the application file:
- touch app.py
- Opened the file using Visual Studio Code:
- code app.py
- Ran the FastAPI development server with:
- uvicorn app:app --reload
- Created a templates folder and added an index.html file for the front-end template.

**3. Deployment on Render**

- The following commands were used for deployment configuration on Render:
    - **Build Command**:
    - pip install fastapi "uvicorn[standard]"
    - **Start Command**:
    - uvicorn app:app --host 0.0.0.0 --port $PORT

**4. Version Control with Git**

- Used git diff to check the differences before committing changes.

**5. Introduction to Selenium**

- Selenium was installed with:

- python -m pip install selenium
- The Chrome browser was launched and controlled programmatically.
- Web elements were located using find_element, and actions were performed using send_keys.

**6. Virtual Environment Management**

- Practiced creating and deleting multiple virtual environments.
- Used the deactivate command to exit the active virtual environment.

---

**Session 2: Testing and Continuous Integration with GitHub Actions**

This session focused on automating testing and CI/CD integration using GitHub Actions.

**1. Automated Testing with Selenium and FastAPI**

- Installed the **FinTest Pro** extension.
- Added a file named test_main.py in the project folder.
- Used the TestClient from fastapi.testclient to write and run tests for the homepage.

**2. GitHub Actions for CI/CD**

- Created a GitHub Actions workflow file named main.yml.
- Configured it to run on push events to the main branch or any other branch.
- The workflow included steps to:
    - Install necessary dependencies
    - Run tests using:
    - pytest test_main.py

**3. Managing Dependencies**

- Updated the requirements.txt file to include:
    - fastapi
    - pytest

---

**Summary**

Across the two sessions, participants gained practical experience in deploying a FastAPI application on the Render platform, writing and executing automated tests using Selenium,

and implementing continuous integration workflows with GitHub Actions. These activities provided a foundational understanding of modern web development, testing, and deployment practices.