Final Assignment: Banking System Development and Deployment

Project Narrative:

Welcome to the culminating assignment of your BFSI application development course! In this comprehensive project, you'll demonstrate your proficiency in various topics, including unit testing, collaborative Git workflows, and Docker containerization. Your mission is to develop, test, and deploy a Python-based banking system that encompasses authentication, transaction processing, and account management functionalities.

Use Case:
You are tasked with developing a Python-based banking system for a fictitious financial institution. The system should allow users to authenticate, perform transactions such as deposits, withdrawals, and transfers, and manage their accounts effectively.

Tasks:

1. Unit Testing:
    - Implement comprehensive unit tests using both the unittest and Pytest frameworks to validate the following components:
        - User authentication functionality.
        - Transaction processing logic for deposits, withdrawals, and transfers.
        - Account management features such as balance retrieval and statement generation.
2. Collaborative Git Workflow:
    - Collaborate with your team members on a shared Git repository for the banking system project.
    - Contribute to the project by implementing new features or fixing bugs in existing modules.
    - Create pull requests, participate in code reviews, and ensure timely integration of changes into the main project repository.
3. Dockerization:
    - Containerize the Python Flask application for the banking system using Docker.

  - Create Dockerfiles for each microservice (authentication, transaction, account management).
  - Utilize Docker Compose to define a multi-container environment for running the microservices.
  - Test the Dockerized application locally to verify functionality and compatibility.

Expected Outcomes:

- A fully functional Python-based banking system with robust authentication, transaction processing, and account management capabilities.
- Comprehensive unit tests ensuring the reliability and correctness of critical system functionalities.
- Seamless collaboration and version control management using Git, facilitating effective teamwork and code integration.
- Dockerized microservices enabling easy deployment and scalability of the banking system in containerized environments.

Deliverables:

- GitHub repository URL containing the Python project code.
- Unit test files demonstrating test cases for each module/functionality.
- Documentation detailing the setup instructions, usage guidelines, and Dockerization process.
- Dockerfiles for each microservice in the Python Flask application.
- Docker Compose file defining the multi-container environment.
- Docker images pushed to a container registry with proper tagging and versioning.

Evaluation Criteria:

Aspect	**Granular Tasks**	**Marks**
Unit Testing	Implementation of comprehensive unit tests using unittest and Pytest frameworks for all critical functionalities.	20

	- Test coverage for authentication, transaction processing, and account management functionalities.	
	- Use of assertion statements to verify expected outcomes.	
	- Handling of edge cases and error scenarios in test cases.	
Collaborative Git Workflow	Active participation in the collaborative Git workflow, including contributions, pull requests, and code reviews.	15
	- Meaningful commit messages and clear pull request descriptions.	
	- Constructive feedback provided during code reviews.	
	- Resolution of merge conflicts and timely integration of changes.	
Dockerization	Successful containerization of the Python Flask application with Dockerfiles, Docker Compose, and local testing.	15
	- Dockerfiles properly defining dependencies and commands for each microservice.	
	- Docker Compose file configuring inter-service communication and network settings.	
	- Local testing to ensure functionality and compatibility within Docker containers.	
Documentation Quality	Clear and detailed documentation covering setup instructions, usage guidelines, and Dockerization process.	5