**Capstone Project: DevOps for Banking Microservices -Mid term 1/2 day**

**Project Overview:** ABC Software Company is currently following Agile methodology for a banking client. Most of the Software Development Life Cycle (SDLC) steps are manual. Currently they are looking for central repo mgmt, also having config mgmt. tools in the system with containerization

**Project Requirements**

**Git Requirements**

1. **Repository Creation:**
   * Create a GitHub repository named **BankingMicroservice**.
2. **Branch Management:**
   * As a code reviewer, create individual branches for each developer involved in the project. Assume there are 2 developers
   * Developers will push their code to their respective branches upon completing their work.
   * Review the code changes and merge approved changes into the **Master** branch, which contains the latest stable code.

Note: as the implementation part, make sure to showcase all the work in the perspective of Reviewer , developers and approver.

1. **Commit Tracking:**
   * Record all commits in the local repository.
   * Push changes to the remote GitHub repository **BankingMicroservice** once development is complete.
2. **Documentation:**
   * Create a README.md file in the remote repository, serving as documentation for the project.

**Maven Requirements**

1. **Build Automation:**
   * The project is a Maven-based microservice featuring a main file named **App.java**.
   * A unit test program is located in the src/test/ folder.
   * Code should be built automatically as soon as updates are made to the **Master** branch.
   * Required dependencies, such as the JUnit framework, must be added automatically.
   * A JAR file should be generated in the target folder upon successful builds.

**Production Server Preparation**

1. **Server Environment:**
   * Prepare 2 ( Ansible controller and Client) Linux production server (Ubuntu or Rocky Linux).
2. **Software Installation:**
   * Use Ansible to install Java and Apache on the production server.
   * Start the Apache service on the ansible Client.
   * Install Docker and docker-compose on the Ansible Controller Linux machines.
3. **Docker Container Deployment:**
   * Run the Docker container on the Ansible controller production server
   * Repeat the Docker requirements step for the production machine to ensure deployment consistency.

**Docker Requirements**

1. **Docker Configuration:**
   * Create a docker-compose file in the **BankingMicroservice** repository.
   * Place the WAR/JAR file on a folder in the host machine.
   * Use the base tomcat:latest image
   * Map the volume and the port of the container (8000:8080) and create a new network in the docker-compose file.

**Output expectation**

In the repo

Java Code with pom.xml

2 ansible playbook

Dockerfile (if required)

Docker-compose.yaml

.gitignore ( to block unwanted file to move to the repo).

Tomcat default page using the Client linux

Maven project output on the ansible controller public ip on port 8000.

Note: -- Make sure to use only the IDE tool ( Visual studio code ) for all the above work.