1. Describe the CI/CD process for the project in terms of workflow diagrams.You can provide a general, high-level workflow diagram that illustrates the typical CI/CD process without going into the specific implementation details.

CI/CD Workflow diagram

1> Developent Phase :
- 1> code commit, Developer commit code to their local repositories
- 2> code is push to a vesrsion control system (github)

2> Contious Integration (CI):
- 1> Trigger CI pipleline, Push to the repository triggers the pipleline automatically.
- 2> Build Stage:
  - Frontend: build React Native application using tools like Webpack or expo.
  - Backend : compile java spring-boot microservices and package them into Docker images.
- 3> Unit testS:
  - Frontend: Run unit tests using tools like Jest.
  - Backend: Run unit tests using JUnit
- 4> code anyalysis: Perform static code analysis and linting for qulity and style adherence.
- 5> Artifcat Storage: Store build and Docker images in atifact repository (Nexus)

3> Countious Deployment(CD):
- 1> Depoly to Staging:
  - Frontend: Deploy the React Native application to a staging environment.
  - Backend: Deploy Docker containers to a staging environment using orchestration tools like Kubernetes.
  - Integration Testing: Execute integration tests in the staging environment to ensure that all components work together.
  - Manual Approval (Optional): In some setups, manual approval may be required before deploying to production.
  - Deploy to Production:
    - Frontend: Deploy the React Native application to the Google Play Store and Apple App Store.
    - Backend: Deploy Docker containers to the production environment using Kubernetes orchestration platform.
  - Monitoring and Alerts:
    - Monitor the deployed application for performance, errors, and health using tools like Prometheus, Grafana
    - Set up alerts for critical issues.

4> Feedback Loop:
- 1> Issue Detection: Detect issues or bugs in the production environment through monitoring and user feedback.
- 2> Rollback: if Necessary, Roll back to the previous stable version if critical issues are detected in production.
- 3> Continuous Improvement: Iterate on code improvements and updates based on feedback and performance data.