Hi All,

Sharing the assignment **@Ashutosh, @Pawan , @Ranjeet** will be your mentor for this assignment.

Evaluation Process : Evaluation would NOT be done on your system. It would be a clean sandbox on a different machine. Code would be checked out from github, built on the command line and executed. With that, your code would be reviewed, graded and feedback provided during review.

==========Assignment ==========

MILESTONE 1: 12th Jan 1 PM
==========

REST application of user management using Spring Boot and Spring Data. This application will perform basic CRUD(Create, Read , Update , Delete) operations on the User table.

1.API which will create a user in the user table.

- url:http://localhost:8080/user
- METHOD : POST
- input:userName , firstName, lastName, mobileNumber , emailID, address1, address2
- Validations : same email id, userName or phone number (user already exists)

2.API which will read data from the database.

- url:http://localhost:8080/user?userId=<userID>
- METHOD: GET
- Result: ashutosh,rathor, 999999999, ashutosh.rathor@paytm.com

3.Update API

- url:http://localhost:8080/user
- requestParam: userID
- METHOD:(PUT)
- validation: user should exist

4.API which will delete data in the user table.

- url:http://localhost:8080/user
- requestParam: userID
- validation: user should exist

Hint:
sample link https://www.journaldev.com/17034/spring-data-jpa

Expectations:
==========

1. **Flow Diagram**
2. **Schema Design**
3. **Workable Code with proper comment**
4. **Proper Test Cases**

MILESTONE 2: 14th Jan 5 PM
==========

Wallet Management

1. Create Wallet: API which will create wallet for a user

- url:http://localhost:8080/wallet
- METHOD : POST
- input: phone number
- Authentication Token {{JWT}}
- Validations : phone number should exist , only one wallet for a user.
- After creation push event in kafka

2.API to transfer money from one wallet to another wallet (p2p).

- url:http://localhost:8080/transaction
- METHOD : POST
- input:{payer_phone_number,payee_phone_number,amount}
- Validations : payer and payee both should exist, payer should have sufficient balance.
- After transfer push event in kafka

3.Transaction Summary API

- url:http://localhost:8080/transaction?userId=<userId>
- METHOD: GET
- Validations: userId should exists
- Note : this api should return in a pagination way.

4.Transaction Status

- url:http://localhost:8080/transaction?txnId=<txnID>
- Method :GET
- Validation: TransactionId should exists

Expectations:
==========

1. **Flow Diagram (Visual Paradigm)**
2. **Schema Design**
3. **All Apis should have authentication {{JWT}}**
4. **Code with proper comment**
5. **Junit Test cases**
6. **Debugging through log4j**

Milestone 3: 18 Jan 5 PM
==========

1. Serve Traffic through load Balancer i.e Nginx.
2. Ingest Data in elasticSearch through Kafka.
3. Write consumers in flink pipeline.
4. Serve transaction through ElasticSearch
5. Fuzzy searching on phone number , name and amount.

    url:http://localhost:8080/transaction?userId=<userId>
    METHOD: GET
    Validations: userId should exist
    Note : this api should return in a pagination way.

Good Luck. 😊