Internal Hire Assignment

Problem Statement:

Design and Implement a driver suggestion microservice using Spring Boot, Kafka, MySQL as a tech stack.

Service should be able to store Walmart Store's GPS location along with store ID. The service should be able to capture the drivers lat/long. Then for a given store, service should be able to return a list of drivers sorted by their distance from that store i.e. the driver closest to the store should be the first entry in the list.

- Service should be able to take driver's current (latest) location via Kafka event.
    - Structure of the Kafka event in json format:

```
{
 "driverID": "m123@gmail.com",
 "latitude": 27.876,
 "longitude": -128.33
}
```

- Service should be able to take store configuration via REST API.
    - Structure of the rest API request in json format:

```
{
 "storeID": "1234",
 "latitude": 27.876,
 "longitude": -128.33
}
```

- Service should expose a GET API to fetch N drivers around a store. StoreID and N should be taken as a query parameter to the API.
    - To calculate distance between store and driver's latest location, please use straight line distance (crow-fly distance). For simplicity, you can use coordinate geometry distance calculation formula.
- Service should expose a POST API to take Kafka event payload and publish in Kafka.
- Service should be able to consume Kafka message from the topic even if the message is published through Kafka command line.

System Design Guidelines:

- You are free to choose the structure of the microservice and its constituent modules which will interact with DB, Kafka, API etc.
- There is a docker compose file provided along with this document. This file already has required containers for Kafka, Zookeeper and MySQL.
    - Zookeeper will be running at: **localhost:2181**
    - Kafka will be running at: **localhost:9092**
    - MySQL will be running at: **localhost:3306**
        - Login username **= user**
        - Login password = **user123**
- Spring boot service should be running on port number **9080**.
- Use Kafka topic name "**driver_location**" to capture driver's current location.
- You choose appropriate class names, modules, Rest API endpoints, DB schema.

Assignment Submission Guidelines:

- Edit the docker compose file and add entry for spring boot application such that if evaluator runs "docker-compose up" then service and all the required components should be up and running.
- Submit a zip file containing updated docker compose file. This zip file should contain all the required jars and dependencies to run the service such that evaluator can cd into the extracted folder and directly run "docker-compose up" command to start all the components and service.

You will be evaluated on:

- How efficiently you solve the given problem in terms of time/space complexity.
- Latency of REST APIs.
- Coding standards, Junits coverage, REST compliance, structure of multi-module maven project, database design (for scale).
- Ability to start the service in working condition just by running "docker-compose up" without any errors.