Assignment:

1. Front end and backend technology of your choice and you should show working code during interview:

Use case implemented and code shared to <Harika through Github>:

**Modules implement:**

1. User registration and authentication: Users should be able to create an account and log in securely to the app.
2. Vehicle listing: Car owners can list their vehicles for rent, specifying details such as the type of car, availability, and rental price.

| **Technology** | **Why to Choose:** |
| --- | --- |

|  |  |
| --- | --- |
| **Spring Boot** | : Microservices-friendly, fast development, auto-configuration, scalable. |

|  |  |
| --- | --- |
| **React.js** | : Fast UI updates, reusable component-based, easy state management |

|  |  |
| --- | --- |
| **MySQL** | : Reliable, scalable, widely supported, strong data integrity. |

|  |  |
| --- | --- |
| **Spring Security + JWT** | : Secure authentication, token-based stateless security, role-based access. |

|  |  |
| --- | --- |
| **Axios API** | : Easy async API calls, built-in error handling |

1. Prepare a DB Design for the whole use case: Checked in <DB Design>

Event Driven Microservice Architecture.

|  |  |  |
| --- | --- | --- |
| **Service Name** | Desc | Responsibility |
| **Authentication sevice** | Handles user authentication and authorization. | - User registration & login. - Password management (reset/change). - JWT-based authentication. - Role-based access control (RBAC). |
| **Vehicle Service** | Manages vehicle listings | - Car owners can add/update/delete vehicles |
| - Stores vehicle details (availability, type, pricing) |
| - Vehicle search & filtering |
| **Booking Service** | Handles ride booking | - Users book available vehicles |
| - Manages booking status transitions (PENDING, CONFIRMED, CANCELLED, COMPLETED) |
| - Publishes booking events to Kafka |
| **Payment Service** | Manages payments & transactions | - Processes payments |
| - Stores transaction records |
| - Handles refunds and failures |
| **Notification Service** | Sends real-time notifications | - Listens to Kafka events (booking confirmed, canceled, completed) |
| - Sends notifications |
| **Review & Rating Service** | Collects & manages user feedback | - Users can rate & review cars & owners |
| - Stores historical rating data |
| - Affects vehicle recommendations |
| **Admin Panel Service** | Provides dashboard for admins | - User management (ban, enable, disable) |
| - View system logs and reports |
| - Handle disputes |
| **Search & Recommendation Service** | Enhances user experience | - Provides vehicle recommendations based on past bookings |