**Project Design Phase**

**Solution Architecture**

|  |  |
| --- | --- |
| Date | 9 APRIL 2025 |
| Team ID | SWTID1742640402 |
| Project Name | MyRide |
| Maximum Marks | 4 Marks |

**Solution Architecture**

A **scalable, real-time** system built on the MERN stack with secure APIs and optimized for mobile users.

**1. Client Layer (Frontend)**

* **Framework**: React.js (PWA for offline support).
* **Key Libraries**:
  + Axios: HTTP requests to backend.
  + Google Maps API: Real-time ride tracking.
  + Tailwind CSS: Mobile-first responsive design.
* **Features**:
  + Booking interface with fare estimation.
  + Interactive map for live driver tracking.
  + Payment gateway integration (Razorpay/Stripe).

**2. Server Layer (Backend)**

* **Framework**: Node.js + Express.js.
* **Core Modules**:
  + RESTful APIs for booking, payments, and auth.
  + Socket.io: Real-time driver-passenger communication.
  + Rate Limiter: Prevent API abuse.

**3. Business Logic Layer**

* **Dynamic Dispatch**: AI algorithm for driver matching (proximity + rating).
* **Pricing Engine**:
  + Distance-based fare calculation (Google Maps Distance Matrix).
  + Surge pricing during peak demand.
* **Role-Based Access**:
  + Passengers: Book rides, rate drivers.
  + Drivers: Accept rides, track earnings.
  + Admins: Monitor system, resolve disputes.

**4. Database Layer**

* **MongoDB Atlas**:
  + users: Passenger/driver profiles (geospatial index for location).
  + bookings: Ride details (status, timestamps, payment IDs).
  + transactions: Payment logs.
* **Redis** (Optional): Cache frequent queries (e.g., nearby drivers).

**5. Geolocation & Mapping**

* **Google Maps Platform**:
  + Places API: Auto-complete pickup/destination.
  + Directions API: Route optimization for drivers.
* **Geospatial Queries**: MongoDB $near for driver discovery.

**6. Authentication & Security**

* **JWT**: Stateless auth for passengers/drivers.
* **Bcrypt**: Password hashing.
* **Input Sanitization**: Prevent NoSQL/SQL injection.
* **HTTPS**: End-to-end encryption.

**7. Analytics & Monitoring**

* **Admin Dashboard**:
  + Demand heatmaps (MongoDB Charts).
  + Ride completion/failure rates.
* **Error Tracking**: Sentry for crash reporting.
* **Performance**: New Relic for API latency monitoring.

**8. Storage & CDN**

* **AWS S3**: Store driver documents (license, vehicle RC).
* **CDN**: CloudFront for fast static asset delivery globally.