# Project Design Phase-ll
# Technology Stack (Architecture & Stack)

| | |
|---|---|
| **Date** | **17 April 2025** |
| **Team ID** | **SWTID1743955267** |
| **Project Title:** | **RideEase** |
| **Maximum Marks** | **4 Marks** |

## Technical Architecture

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

**Example:**

**Architecture Summary**

Users (Riders, Drivers, Admin) interact via React UI
MongoDB Atlas serves as your cloud-hosted data layer
Express.js handles API routes and business logic
Node.js powers the backend server environment
JWT manages authentication and authorization
The entire MERN stack can be deployed locally or via cloud services (like Vercel/Render)

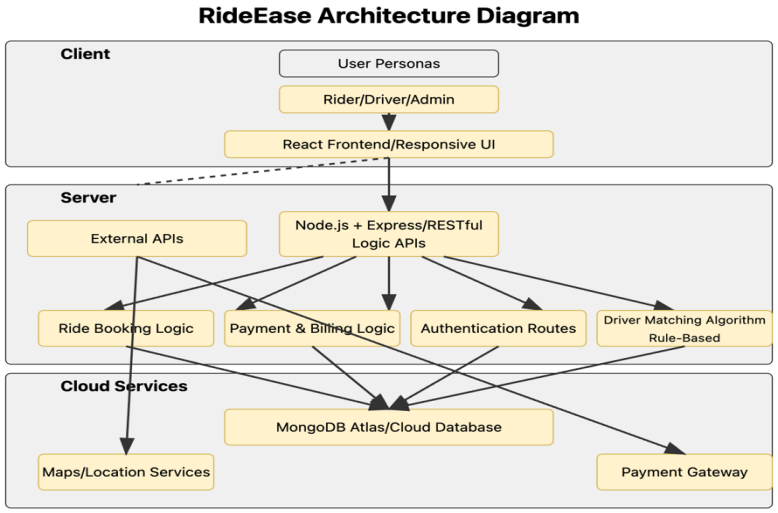

**Table-1 : Components & Technologies**

| S.No | Component | Description | Technology |
|---|---|---|---|
| 1. | User Interface | Web/mobile interface for booking rides and managing accounts | HTML, CSS, JavaScript, React.js |
| 2. | Application Logic-1 | User Authentication and Role-based Access Management | Node.js, Express.js, JWT |
| 3. | Application Logic-2 | Ride Booking and Management System | Node.js, Express.js |
| 4. | Application Logic-3 | Payment Processing and Transaction Management | Node.js, Express.js |
| 5. | Application Logic-4 | Route Optimization and Fare Calculation | Node.js, Express.js |
| 6. | Database | Stores user profiles, ride history, driver data, transactions | MongoDB (NoSQL) |
| 7. | Cloud Database | Cloud-hosted database instance | MongoDB Atlas |
| 8. | Location Services | Maps integration for pickup/dropoff locations | Google Maps API / Mapbox |
| 9. | External API-1 | Payment gateway integration | Stripe / PayPal / Razorpay |
| 10. | External API-2 | SMS/Email notifications for ride updates | Twilio / SendGrid |
| 11. | Infrastructure (Server/Cloud) | Deployed on cloud platform for scalability | Render / Vercel for frontend, Railway / Heroku for backend |

**Table-2: Application Characteristics**

| S.No | Characteristics | Description | Technology |
|---|---|---|---|
| 1. | Open-Source Frameworks | Utilizes widely adopted open-source technologies for frontend, backend, and database management | React.js, Node.js, Express.js, MongoDB, Tailwind CSS |
| 2. | Security Implementations | JWT authentication, route protection, secure payment processing | JWT, HTTPS, Helmet.js, bcrypt |
| 3. | Scalable Architecture | 3-tier architecture (Frontend-Backend-Database) with cloud deployment for horizontal scaling | MERN Stack, Render/Vercel, MongoDB Atlas (auto-scaling) |
| 4. | Availability | Hosted on cloud with automatic failover and high availability using distributed architecture | MongoDB Atlas (multi-region), Render/Vercel |

| S.No | Characteristics | Description | Technology |
|---|---|---|---|
| 5. | Performance | React virtual DOM for fast UI rendering, API caching, optimized database queries | React.js, Express.js, MongoDB indexes, Redux for state management |
| 6. | Real-time Features | Live tracking of drivers, instant ride status updates | Socket.io, WebSockets |
| 7. | Mobile Responsiveness | Adaptive design for various screen sizes | React with responsive CSS, Media Queries |
| 8. | Offline Capabilities | Basic functionality during intermittent connectivity | Progressive Web App (PWA) features, localStorage |