**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

|  |  |
| --- | --- |
| Date | 16 April 2025 |
| Team ID | SWTID1744468652 |
| Project Name | I-movies : movie ticket booking system |
| Maximum Marks | 4 Marks |

**Technical Architecture:**

This MERN stack-based application is designed using a modular, component-driven architecture to ensure scalability, maintainability, and real-time responsiveness. The architecture follows the traditional three-tier pattern:

* Frontend Layer: Built using React.js, this layer handles the client-side logic, including UI rendering, routing, and communicating with the backend via RESTful APIs. Users can register, login, browse data, perform actions, and view updates dynamically.
* Backend Layer: Powered by Node.js and Express.js, this layer processes requests from the frontend, manages business logic, and performs operations such as authentication, CRUD operations, and validations.
* Database Layer: MongoDB is used as a NoSQL document-based database to store application data such as user details, session info, posts/products/tasks (depending on your project type), and more.

All parts of the system are hosted on cloud platforms like Vercel (for frontend) and Railway (for backend), providing continuous integration, deployment, and availability.

**Components & Technologies:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **Component** | **Description** | **Technology** |
| **1** | Frontend | UI for user interaction, forms, views, and routing | **React.js**, HTML, CSS, JavaScript |
| **2** | Backend | API endpoints, routing, controllers, business logic | **Node.js**, **Express.js** |
| 3 | Authentication | User login, signup, secure token-based access | **JWT**, **bcrypt.js** |
| **4** | Database | Stores users, items/tasks/posts, sessions | **MongoDB**, **Mongoose** |
| **5** | File Storage | If project includes image/file uploads | Multer, Local Uploads Folder / Cloudinary |
| **6** | Hosting & Deployment | Continuous deployment for frontend & backend | **Vercel (frontend)**, **Railway (backend)** |
| **7** | API Testing/Tools | For testing endpoints and debugging | Postman, Thunder Client |

**Application Characteristics:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **Characteristics** | **Description** | **Technology** |
| **1** | Full-Stack App | Complete end-to-end functionality from UI to DB | MERN Stack |
| **2** | |  | | --- | |  |  |  | | --- | | Open Source & Modular | | Built using open-source tools with clean separation of concerns | |  | | --- | | separation of concerns |  |  | | --- | | React components, Express routes | |
| **3** | Security | Encrypted passwords, protected routes, role-based access (if applied) | |  | | --- | |  |  |  | | --- | | JWT, bcrypt, Helme | |
| **4** | Scalable | |  | | --- | |  |  |  | | --- | | Easily extendable architecture with REST APIs | | CRUD architecture, modular file structure |
| **5** | Performance | |  | | --- | |  |  |  | | --- | | Fast frontend rendering, lazy loading (if applicable), optimized queries | | MongoDB Indexing, React optimization |