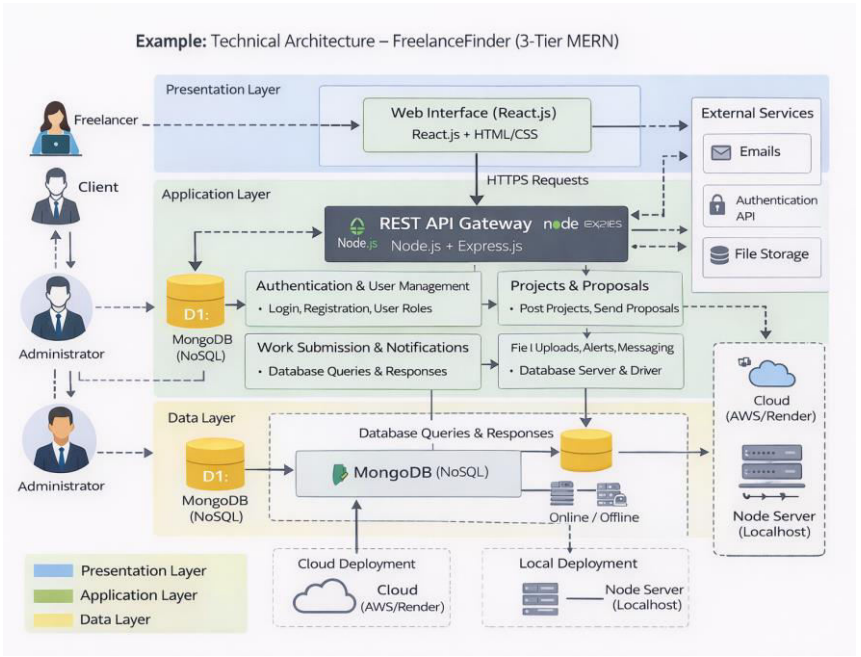
Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	10 February 2026
Team ID	LTVIP2026TMIDS80676
Project Name	FreelanceFinder: Discovering Opportunities, Unlocking Potential
Maximum Marks	4 Marks

Technical Architecture:

The FreelanceFinder platform follows a 3-Tier MERN Architecture consisting of a React-based frontend, Node.js/Express backend APIs, and MongoDB database. The user interface interacts with backend services through REST APIs. Application logic handles authentication, project posting, proposals, messaging, and submission tracking. Data is stored in MongoDB collections, and file uploads are handled through server storage. The system can be deployed locally or on cloud infrastructure with scalable server configuration.



Architecture Layers:

- Presentation Layer: React.js Web Interface
- Application Layer: Node.js + Express.js APIs
- Data Layer: MongoDB Database
- External Interfaces: Email services, Authentication APIs
- Infrastructure: Local Server / Cloud Deployment

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1	User Interface	Web-based dashboard for freelancers, clients, and admin	React.js, HTML, CSS, JavaScript
2	Application Logic-1	Authentication & User Management	Node.js, Express.js
3	Application Logic-2	Project & Proposal Management	Node.js REST APIs
4	Application Logic-3	Work Submission & Notifications	Express.js Middleware
5	Database	Stores users, projects, proposals, submissions	MongoDB (NoSQL)
6	Cloud Database	Optional hosted DB	MongoDB Atlas
7	File Storage	Stores uploaded work files	Local Storage / Cloud Storage
8	External API-1	Email notifications	Nodemailer / SMTP Service
9	External API-2	Authentication services	JWT Authentication
10	Machine Learning Model	Not Applicable (Optional future enhancement)	—
11	Infrastructure (Server / Cloud)	Application deployment	Localhost, Node Server, Cloud (AWS/Render)

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1	Open-Source Frameworks	MERN stack open-source technologies	React.js, Node.js, Express.js, MongoDB
2	Security Implementations	Secure login, token authentication, protected routes	JWT, Password Hashing, Role-based Access
3	Scalable Architecture	Modular backend APIs with separate frontend	3-Tier MERN Architecture
4	Availability	Continuous system access via web deployment	Cloud Hosting / Node Server
5	Performance	Optimized API calls and database queries	REST APIs, MongoDB Indexing