**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

Date 26-05-2025

Team ID : LTVIP2025TMID55745

Project Name Freelance Finder

Maximum Marks 4 Marks

**Technical Architecture:**

**Freelance Finder** is designed with a scalable MERN stack architecture consisting of:

* **Presentation Layer (Frontend):** Interactive user interfaces for clients and freelancers built with React.js
* **Business Logic Layer (Backend):** RESTful API handling project management, bidding system, and user authentication
* **Data Storage Layer:** MongoDB for flexible data storage of user profiles, projects, and transactions

The platform integrates with third-party APIs for payment processing, email notifications, and cloud storage.

**Table-1: Components & Technologies:**

+---+-------------------+-------------------------+-------------------+  
| # | Component | Description | Technology |  
+===+===================+=========================+===================+  
| 1 | User Interface | Responsive web app for | React.js |  
| | (Client) | clients/freelancers | Redux Toolkit |  
| | | | Material UI |  
+---+-------------------+-------------------------+-------------------+  
| 2 | Core API | User authentication, | Node.js |  
| | (Server) | project management | Express.js |  
| | | | JWT Authentication|  
+---+-------------------+-------------------------+-------------------+  
| 3 | Real-time | Chat messaging system | Socket.io |  
| | Communication | | |  
+---+-------------------+-------------------------+-------------------+  
| 4 | Database | Stores all application | MongoDB |  
| | | data | Mongoose ODM |  
+---+-------------------+-------------------------+-------------------+  
| 5 | Payment | Secure transaction | Stripe API |  
| | Processing | processing | |  
+---+-------------------+-------------------------+-------------------+

**Table-2: Application Characteristics:**

+---+-------------------+-------------------------+-------------------+  
| # | Characteristics | Description | Implementation |  
+===+===================+=========================+===================+  
| 1 | Scalability | Microservices-ready | Docker |  
| | | architecture | Kubernetes |  
+---+-------------------+-------------------------+-------------------+  
| 2 | Security | End-to-end encryption | Bcrypt.js |  
| | | and data protection | Helmet.js |  
+---+-------------------+-------------------------+-------------------+  
| 3 | Performance | Optimized for 1000+ | Redis Cache |  
| | | concurrent users | |  
+---+-------------------+-------------------------+-------------------+  
| 4 | CI/CD Pipeline | Automated testing and | GitHub Actions |  
| | | deployment | AWS Elastic Beanstalk|  
+---+-------------------+-------------------------+-------------------+

**Project Setup and Configuration**

Folder Structure

text

Copy

Download

freelance-finder/

├── client/ # Frontend React application

│ ├── public/ # Static assets

│ └── src/ # React components

│ ├── assets/ # Images, styles

│ ├── components/ # Reusable UI

│ ├── pages/ # Route pages

│ └── store/ # Redux configuration

├── server/ # Backend Node.js

│ ├── config/ # DB connection

│ ├── controllers/ # Business logic

│ ├── models/ # MongoDB schemas

│ ├── routes/ # API endpoints

│ └── middleware/ # Auth handlers

└── .env # Environment variables

Installation Guide

1. **Prerequisites:**
   * Node.js v16+
   * MongoDB Atlas account
   * Git
2. **Frontend Setup:**

bash

Copy

Download

cd client

npm install react react-dom @mui/material @emotion/react axios redux react-redux

1. **Backend Setup:**

bash

Copy

Download

cd server

npm install express mongoose bcryptjs jsonwebtoken cors dotenv socket.io stripe

1. **Environment Configuration:**  
   Create .env file with:

text

Copy

Download

MONGO\_URI=mongodb+srv://<username>:<password>@cluster.mongodb.net/freelance-finder

JWT\_SECRET=your\_jwt\_secret\_here

STRIPE\_KEY=your\_stripe\_api\_key

1. **Running the Application:**

bash

Copy

Download

*# Start backend*

cd server && npm start

*# Start frontend*

cd ../client && npm start

**References:**

* [React Documentation](https://react.dev/)
* [Express.js Best Practices](https://expressjs.com/en/advanced/best-practice-performance.html)
* [Mongoose Documentation](https://mongoosejs.com/docs/)
* [Stripe API Docs](https://stripe.com/docs/api)

**Key Improvements:**

1. Replaced medical-specific components with freelance platform requirements
2. Added real-time communication and payment processing technologies
3. Included modern frontend stack (Redux Toolkit, Material UI)
4. Detailed folder structure for MERN application
5. Added security and performance optimization tools
6. Included CI/CD pipeline configuration
7. Provided complete installation guide with environment setup