**Freelancing Application MERN**

**SB works done by Team ID:**

NM2024TMID12188

**Team leader : Harish Raghav prasad R**

**: Yogavel k**

**: Dhanaraj s**

**: Ashok ram J**

# Introduction

A freelancing application is a software platform designed to connect freelancers with clients seeking their services. It serves as a centralized platform for freelancers to showcase their skills, apply for projects, and receive payments. Similarly, clients can post project requirements, review freelancer profiles, and manage their hired talent.The primary goal of this application is to streamline the freelancing process, enabling seamless interaction between clients and freelancers while ensuring secure transactions and efficient project management.

Key features include:

1.\*User Registration:\* Freelancers and clients can create accounts to access the platform’s features. 2.\*Project Listings:\* Clients can post detailed descriptions of projects, including deadlines, budget, and requirements.

3.\*Bidding and Proposals:\* Freelancers can bid on projects by submitting tailored proposals.

4.\*Chat System:\* Real-time communication between freelancers and clients.

5.Payment Integration:\* Secure payment gateways to facilitate transactions between clients and freelancers.

6.\*Review and Ratings:\* Mutual feedback mechanism for clients and freelancers to build trust and improve service quality.

**Technical Architecture:**

The freelancing application adopts a \*MERN (MongoDB, Express.js, React.js, Node.js)\* stack to build a robust and scalable architecture:

**Frontend:** React.js is used to develop a dynamic user interface, leveraging libraries like Material-UI and Bootstrap for responsive and user-friendly designs

**Backend:** Express.js and Node.js are employed to create APIs, manage server-side logic, and facilitate efficient communication between the frontend and database.

**Database:** MongoDB is used to store user profiles, project details, proposals, and payment records, offering flexibility and scalability.

**APIs:** RESTful APIs enable smooth data exchange between the client and server.

**Authentication:** Secure user authentication is implemented using JWT (JSON Web Tokens).

**Real-Time Features:** Socket.io is used for implementing real-time chat functionality.

**Payment Gateway:** Stripe or PayPal is integrated to ensure secure and seamless payment processing.

**ER Diagram:**

The ER Diagram illustrates the relationships between key entities such as Users, Projects, Proposals, and Payments.

* A **User** (client or freelancer) can create or bid on multiple **Projects**.
* **Projects** are associated with multiple **Proposals** submitted by freelancers.
* **Payments** are linked to both **Users** and **Projects**, ensuring accountability and transaction traceability.

# Project Structure

## Frontend Structure

The frontend folder contains:

**Components:** Reusable React components for UI development.

**Pages:** Key application pages like Home, Dashboard, Project Details, etc.

**Services:** Axios-based services for API communication. **Assets:** Static files like images, CSS, and icons.

**Backend Structure**

The backend folder contains:

* **Models:** MongoDB schemas for Users, Projects, Proposals, and Payments.
* **Routes:** RESTful API endpoints.
* **Controllers:** Business logic for handling requests.
* **Config:** Environment variables and database configuration.

# Pre-Requisites

**Tools and Technologies:**

**Node.js and npm:** For server-side JavaScript execution.

**Express.js:** Simplifies backend development.

**MongoDB**: Flexible NoSQL database for storing app data.

**React.js:** Frontend development framework.

**Git:** Version control for code collaboration.

**Code Editor:** Recommended: Visual Studio Code.

**Installation Guides:**

* [**Node.js**](https://nodejs.org/en/download/)
* [**MongoDB**](https://www.mongodb.com/try/download/community)
* [**React.js**](https://reactjs.org/docs/getting-started.html)
* [**Git**](https://git-scm.com/downloads)

**Roles and Responsibilities**

**Freelancer:**

1.Create a profile and update skills, portfolio, and availability.

2.Browse projects and submit proposals with a bid amount.

3.Communicate with clients through the chat system.

4.Submit project deliverables and request payment.

5.Rate and review clients after project completion.

**Client:**

1.Create an account and post projects with detailed requirements.

2.Review freelancer proposals and profiles to hire suitable candidates.

3.Communicate project details through chat.

4.Release payments upon project completion.

5.Provide feedback for freelancers**.**

**Admin:**

1.Manage user accounts (clients and freelancers).

2.Moderate project postings and proposals.

3.Oversee payment transactions to prevent fraud.

4.Monitor platform activity and resolve disputes.

# Project Flow

## Milestone 1: Project Setup

**Frontend Setup:**

* Install React.js.
* Add Material-UI and Bootstrap for styling.

**Backend Setup:**

* Create an Express.js server.
* Connect to MongoDB using Mongoose.

## Milestone 2: Backend Development

1.Define schemas for Users, Projects, Proposals, and Payments.

2.Set up API routes for CRUD operations.

3.Implement JWT authentication for secure login.

## Milestone 3: Frontend Development

1. Design and develop pages for User Registration, Dashboard, and Project Details.
2. Integrate APIs using Axios for dynamic data rendering.
3. Implement real-time chat using Socket.io.

# How to Run the Project

**1.Clone the Repository:**

git clone https://github.com/your-repo/freelancing-app.git cd freelancing-app

**2.Install Dependencies:** # Frontend cd frontend npm install

# Backend cd ../backend npm install

**3.Run the Application:** # Frontend npm start

# Backend npm run server

**4.Access the application at** [**http://localhost:3000**](http://localhost:3000/)**.**

**Demo and Code Repository**

* Demo: https://drive.google.com/drive/folders/1xqfonwsNR9kZQodLwb5EEcpBa1hzjksp
* Repository: https://github.com/Raghavprasad007/Freelancing-Application-Mern.git

This documentation provides a comprehensive guide for setting up, running, and understanding the freelancing application.