Project Documentation

Introduction

Project Title:

FreelanceHub – MERN Stack Freelancing Application

Team Members:

- **Ritvik Talwar** – Full Stack Developer (Frontend, Backend, Database Integration)

2. Overview Project

Purpose:

The purpose of this project is to develop a web-based freelancing platform where clients can post job requirements and freelancers can browse and apply for available jobs. The application aims to simplify the hiring process by providing a centralized system for job posting, application management, and user authentication using modern web technologies.

Features:

- User registration and login (Client & Freelancer roles)
- Secure authentication using JWT
- Clients can post freelancing jobs
- Freelancers can view and apply for jobs
- Dashboard for managing jobs and applications
- MongoDB database for persistent storage

3. Architecture

Frontend:

The frontend is built using **React.js**, following a component-based architecture. React Router is

used for navigation between pages such as Home, Login, Register, and Job Listings. Axios is used to communicate with backend APIs.

Backend:

The backend is developed using **Node.js** with **Express.js** framework. It handles API requests, user authentication, job management, and application logic. RESTful APIs are implemented to ensure smooth communication with the frontend.

Database:

MongoDB is used as the NoSQL database. Data is stored in collections such as Users, Jobs, and Applications. Mongoose ODM is used for schema definition and database interaction.

4. Setup Instructions

Prerequisites:

- Node.js (LTS version)
- npm (Node Package Manager)
- MongoDB Atlas account
- Git
- Code Editor (VS Code recommended)

Installation:

1. Clone the repository:

```
git clone https://github.com/prabhakaran339/Freelancing-Application-MERN.git
```

2. Navigate to backend directory and install dependencies:

```
cd api
npm install
```

3. Create a `.env` file in the `api` folder and configure environment variables:

```
PORT=5000
MONGO_URI=your_mongodb_connection_string
JWT_SECRET=your_secret_key
```

4. Navigate to frontend directory and install dependencies:

```
cd ../client
npm install
```

5. Folder Structure

Client (Frontend):

```
client/
├── src/
│   ├── components/
│   ├── pages/
│   ├── services/
│   └── App.js
├── package.json
```

The client folder contains all React components, pages, and API service files.

Server (Backend):

```
api/
├── models/
├── routes/
├── controllers/
├── config/
├── server.js
```

The server folder contains models for MongoDB, route definitions, controllers for business logic, and configuration files.

6. Running the Application

Backend:

```
cd api
npm start
```

Frontend:

```
cd client
npm start
```

The application runs locally at:

`http://localhost:3000`

7. API Documentation

Authentication APIs

- **POST**`/api/auth/register`
  - Registers a new user
- **POST**`/api/auth/login`
  - Authenticates user and returns JWT

Job APIs

- **POST**`/api/jobs/create`
  - Create a new job (Client)
- **GET**`/api/jobs`
  - Fetch all jobs

Application APIs

- **POST**`/api/apply`
  - Apply for a job (Freelancer)

8. Authentication

Authentication is implemented using **JSON Web Tokens (JWT)**.

- User credentials are verified during login
- A JWT token is generated and sent to the client
- Protected routes require a valid token for access
- Role-based authorization ensures correct access control

9. User Interface

The user interface is clean and responsive, built using React components.
Key pages include:

- Home Page
- Login Page

- Registration Page
- Job Listing Page

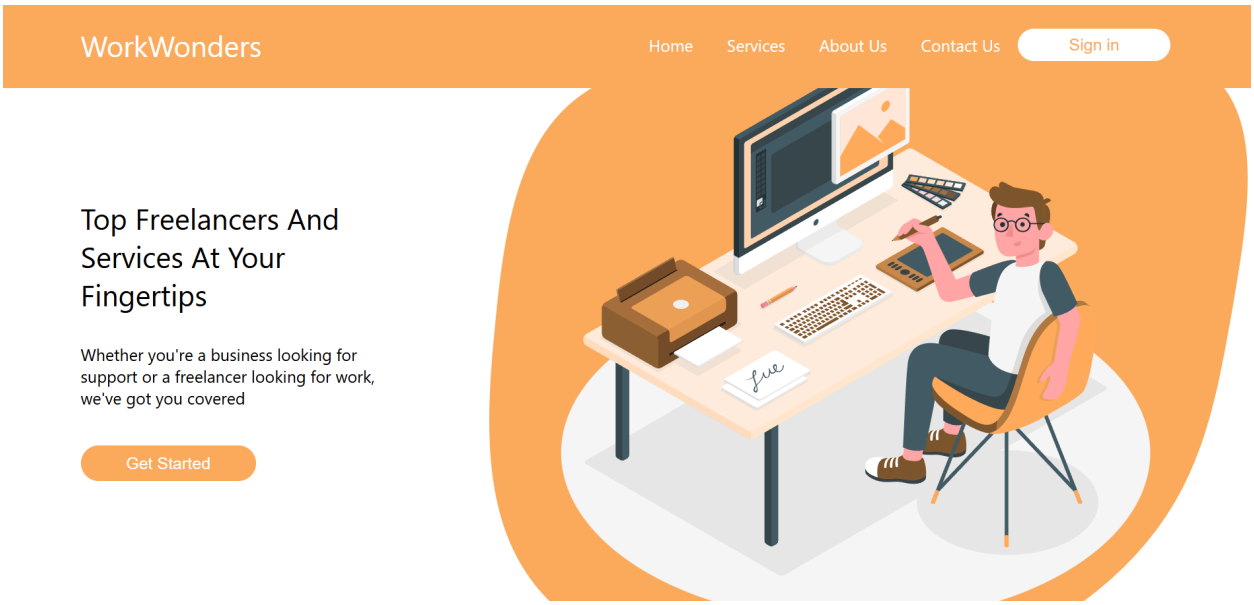
10. Testing

Basic manual testing was performed to verify:

- User registration and login
- Job creation and listing
- API responses using Postman
- Database updates in MongoDB Atlas

11. Screenshots or Demo

- Screenshots of running application UI
- MongoDB Atlas collections
- Backend server running in terminal



12. Known Issues

- Some UI elements may require refinement
- Real-time chat feature may not function if socket server is not running
- Limited validation on input fields

13. Future Enhancements

- Real-time chat between clients and freelancers
- Payment gateway integration
- Advanced job filtering and search
- Rating and review system
- Improved UI/UX design