**Full Stack Development with MERN Project Documentation format**

# 1. Introduction

* **Project Title:** FreelanceFinder
* **Team Members:**

1. P Yashwitha (Project Developer)
2. Lingubattina Rupa
3. Polamreddy Mounika

# 2. Project Overview

* **Purpose:** Freelance Finder is a job portal designed to connect freelancers with potential clients. The goal is to create a seamless platform where freelancers can showcase their skills, upload resumes, apply for jobs, and communicate with clients.
* **Features:**

** User registration/login with image upload**

** Role-based dashboards for clients and freelancers**

** Job listing and filtering (by category and location)**

** Job application tracking and Job Posting**

** Real-time messaging between clients and freelancers**

** Profile editing with resume, company logo, phone, skills, and experience updates**

** Responsive UI for all screen sizes**

# 3. Architecture

* **Frontend:** Built with **React.js**, using functional components, useState, useEffect, and react-router-dom. TailwindCSS and Bootstrap used for styling. Responsive layouts are implemented using CSS grid/flexbox.
* **Backend:** Developed using **Node.js** and **Express.js**. Handles authentication, user management, file uploads (using Multer & Cloudinary), and job posting logic.
* **Database:** **MongoDB Atlas** is used to store users, jobs, and chat messages. Mongoose models define schemas for employees, jobs, and applications.

# 4. Setup Instructions

* **Prerequisites:** Node.js, MongoDB Atlas account , Render (for deployment)
* **Installation:**

1. Git clone https://github.com/yashwithatutorials/freelancer\_finder\_client.git
2. cd freelancer\_finder\_client
3. npm install
4. npm run dev
5. for server git clone https://github.com/yashwithatutorials/freelancer\_finder\_server.git
6. cd freelancer\_finder\_server
7. node index.js

# 5. Folder Structure

* **Client:** Describe the structure of the React frontend.
* **Server:** Explain the organization of the Node.js backend.

# 6. Running the Application

• Commands to start the frontend and backend servers locally.

**Frontend:** npm run dev in the client directory.

**Backend:** node index.js in the server directory.

# 7. API Documentation

| **Method** | **Endpoint** | **Description** |
| --- | --- | --- |
| POST | /signup | Register user with image upload |
| POST | /login | Authenticate user |
| GET | /api/jobs | Fetch jobs with filters |
| POST | /api/employees/update | Update user profile |
| POST | /api/chat/send | Send message |

# 8. Authentication

#  Passwords are hashed using bcrypt

#  JWT tokens stored in browser local storage after login

#  Middleware validates tokens on protected routes

# 

1. **User Interface**

 Signup/login with profile image upload

 Dashboard with job listings

 View Profile page to edit resume, skills, experience

 Responsive layout on desktop and mobile

1. **Testing**

 Manual testing performed for:

* + Signup/Login
  + File uploads
  + API integration

 Postman used for backend endpoint validation

1. **Demo**

**Demo**: https://freelancer-finder-client.vercel.app

**12.Known Issues**

 Resume and logo may not update instantly without refreshing

 Real-time chat works only after both users have logged in at least once

 No pagination for job list (may slow down with many jobs)

# 13. Future Enhancements

 Add pagination and infinite scroll for job listings

 Integrate payment feature for clients

 Add video upload support in profiles

 Improve real-time chat with WebSocket