# FreelanceFinder: Discovering opportunities unlocking potential

## INTRODUCTION

FreelanceFinder is a dynamic freelance marketplace connecting clients with skilled freelancers across various domains. With intuitive search, secure payments, and efficient project management, the platform streamlines the freelance hiring process. Admins efficiently manage users and transactions through a robust admin dashboard.  
  
Key Features:  
- Simple Client-Freelancer Collaboration  
- Secure Payment Process  
- Project Listing and Bidding  
- Freelancer Portfolio Management  
- Real-Time Notifications  
- Admin Dashboard for User & Project Oversight

## SCENARIO EXAMPLE:

Finding a Web Developer  
  
1. Project Posting:  
Alex, a startup founder, posts a web development project on FreelanceFinder. He specifies the skills required, budget, and timeline.  
  
2. Effortless Freelancer Discovery:  
Freelancers browse the new project listing. Emma, a skilled React developer, applies with a proposal highlighting her experience.  
  
3. Secure Hiring:  
Alex reviews Emma’s profile, reviews, and portfolio. Satisfied, he hires Emma directly through the platform.  
  
4. Seamless Payment & Project Tracking:  
Payments are secured using FreelanceFinder’s integrated payment gateway. The project’s progress and milestones are tracked from the dashboard.  
  
5. Successful Completion:  
After completion, Alex leaves a 5-star review, and Emma’s earnings are transferred securely to her account.

## TECHNICAL ARCHITECTURE

Frontend:  
React.js | Redux | Tailwind CSS  
Features: Authentication, Project Listings, Freelancer Profiles, Messaging, Admin Dashboard  
  
Backend:  
Node.js | Express.js | JWT Authentication | RESTful APIs  
APIs: Users, Projects, Payments, Reviews, Admin Management  
  
Database:  
MongoDB + Mongoose  
Collections: Users, Freelancers, Projects, Proposals, Payments, Reviews

## ER DIAGRAM:

Entities:  
- User: Clients and freelancers  
- Projects: Details of job postings  
- Proposals: Freelancer applications to projects  
- Payments: Secure transactions between clients and freelancers  
- Reviews: Feedback and ratings system

## FEATURES

Client Features:  
- Post Projects  
- Search Freelancers  
- Hire & Pay Securely  
- Manage Projects & Track Progress  
- Review Freelancers  
  
Freelancer Features:  
- Create Portfolio  
- Browse & Bid on Projects  
- Manage Proposals  
- Receive Payments  
- Build Reputation with Reviews  
  
Admin Features:  
- Manage Users (Clients/Freelancers)  
- Manage Projects & Proposals  
- Handle Complaints/Disputes  
- Analyze Platform Metrics

## PREREQUISITES

- Node.js & npm: Server-side runtime and package manager  
- MongoDB & Compass/Atlas: Database  
- Express.js: REST API framework  
- React.js: Frontend library  
- Git & GitHub: Version control  
- Mongoose: MongoDB ODM

## PROJECT STRUCTURE

/client - React frontend  
 /components  
 /pages  
 /redux  
 /services  
  
/server - Node.js backend  
 /controllers  
 /models  
 /routes  
 /middleware

## PROJECT SETUP

Clone the repository:  
git clone https://github.com/your-username/FreelanceFinder-MERN.git  
  
Backend setup:  
cd server  
npm install  
npm run dev  
  
Frontend setup:  
cd client  
npm install  
npm start  
  
Visit: http://localhost:3000

## DATABASE DEVELOPMENT

Schemas:  
- UserSchema: Name, email, password, role (client/freelancer), bio, profilePic  
- ProjectSchema: Title, description, skills required, budget, status  
- ProposalSchema: ProjectId, freelancerId, proposal text, bid amount  
- PaymentSchema: Amount, payment status, related project & users  
- ReviewSchema: Rating, comment, userId, freelancerId

## BACKEND DEVELOPMENT

- Setup Express server and define routes  
- JWT-based authentication  
- RESTful API endpoints for users, projects, proposals, and payments  
- Admin-only routes with role-based access control  
- Error handling middleware

## FRONTEND DEVELOPMENT

- React components for Login, Register, Home, Project List, Freelancer Profiles, etc.  
- Redux Toolkit for state management  
- Axios for API calls  
- Protected routes for authenticated access  
- Responsive UI with Tailwind CSS

## USER & ADMIN FLOW

Client Flow:  
- Register/Login  
- Post projects  
- Browse freelancers & hire  
- Manage projects and payments  
  
Freelancer Flow:  
- Register/Login  
- Setup profile  
- Browse and apply for projects  
- Get paid after work completion  
  
Admin Flow:  
- Login to Admin Dashboard  
- Manage users & projects  
- Handle payments & disputes

## REFERENCES

- React Docs: https://reactjs.org/docs/getting-started.html  
- Node.js Docs: https://nodejs.org/en/docs  
- MongoDB Docs: https://www.mongodb.com/docs/  
- Express Docs: https://expressjs.com/en/starter/installing.ht