Technical Specifications

**Project Name:** Freelance Finder  
**Technology Stack:** MERN (MongoDB, Express.js, React, Node.js)  
**Database:** MongoDB Atlas (Cloud-based)  
**Version:** 1.0.0

Table of Contents

1. Database Schema
2. Installation Guide
3. Authentication System
4. User Stories
5. API Endpoints
6. Future Roadmap

1. Database Schema

Collections Structure

Users Collection

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| userId | String | Unique identifier |
| username | String | User display name |
| email | String | Unique email address |
| password | String | Hashed password |
| userType | String | 'client'/'freelancer'/'admin' |
| createdAt | Date | Registration timestamp |

Projects Collection

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| projectId | String | Unique identifier |
| clientId | String | Reference to client |
| title | String | Project name |
| budget | Number | In USD |
| status | String | Open/In Progress/Completed |

Applications Collection

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| applicationId | String | Unique identifier |
| projectId | String | Reference to project |
| freelancerId | String | Reference to freelancer |
| bidAmount | Number | Proposed amount |
| status | String | Pending/Accepted/Rejected |

2. Installation Guide

Prerequisites

* Node.js v16+
* MongoDB Atlas account
* Git

Setup Instructions

1. Clone repository:

bash

Copy

Download

git clone https://github.com/yourrepo/freelance-finder.git

1. Install dependencies:

bash

Copy

Download

cd server && npm install

cd ../client && npm install

1. Configure environment variables:

text

Copy

Download

MONGO\_URI=your\_mongodb\_connection\_string

JWT\_SECRET=your\_jwt\_secret

1. Start application:

bash

Copy

Download

*# Backend*

npm run server

*# Frontend*

npm run client

3. Authentication System

Sequence Diagram

text

Copy

Download

[User] -> [Frontend]: Enters credentials

[Frontend] -> [Backend]: POST /api/auth/login

[Backend] -> [Database]: Verify user

[Database] -> [Backend]: Return user data

[Backend] -> [Frontend]: Return JWT token

Security Features

* BCrypt password hashing
* JWT token expiration (24h)
* Email verification
* Rate limiting

4. User Stories

| **User Type** | **Feature** | **Acceptance Criteria** |
| --- | --- | --- |
| Client | Post Projects | Can specify budget, timeline, skills |
| Freelancer | Browse Jobs | Filter by skills, budget range |
| Both | Real-time Chat | Messages persist in database |

5. API Reference

Authentication Endpoints

* POST /api/auth/register - User registration
* POST /api/auth/login - User login
* GET /api/auth/verify - Email verification

Project Endpoints

* POST /api/projects - Create new project
* GET /api/projects - List all projects
* GET /api/projects/:id - Get project details

6. Future Roadmap

Q3 2024

* Payment gateway integration
* Mobile responsive design

Q4 2024

* AI-based job matching
* Video interview feature