Full Stack Development with MERN

Project Documentation format

1. Introduction

• **Project Title:** FreelanceFinder - Discovering opportunities, Unlocking potential

• **Team Members:** Addepalli Nikhitha - Frontend and integration Engineer
Darisi Sai Tharun - Blockchain and smart contract Developer
Vijaya Vardhini Kothapalem - QA and Technical documentation lead

2. Project Overview

• **Purpose:** To provide a decentralized, trustless environment for freelancers and clients to collaborate without third-party intermediaries

• **Features:**

- User registration (Client/Freelancer) with data stored in MongoDB
- Real-time project tracking and milestone-based escrow payments
- Profile management and reputation scoring (Report Cards).

3. Architecture

• **Frontend:** Built with React.js and Tailwind CSS, utilizing @solana/wallet-adapter for blockchain interaction

• **Backend:** The "backend" logic resides on Node.js and Express.js.

• **Database:** MongoDB stores persistent data such as user bios, project descriptions, and transaction history

4. Folder Structure

• **Client:** Contains the React frontend, including components, assets

• **Server:** Contains the Node.js backend with Mongoose models, Express routes, and controllers

5. Running the Application

• Provide commands to start the frontend and backend servers locally.
  - **Frontend:** `npm start` in the client directory.
  - **Backend:** `npm start` in the server directory.

6. API Documentation

- **POST `/api/auth/login`:** Validates a signed message and returns a JWT
- **GET `/api/projects`:** Fetches all active projects stored in MongoDB with their corresponding on-chain PDA addresses
- **POST `/api/escrow/initialize`:** Updates the database after a successful on-chain project creation transaction is detected

7. Authentication

• **Wallet-to-JWT:** Authentication is handled by verifying a user's cryptographic signature via the backend

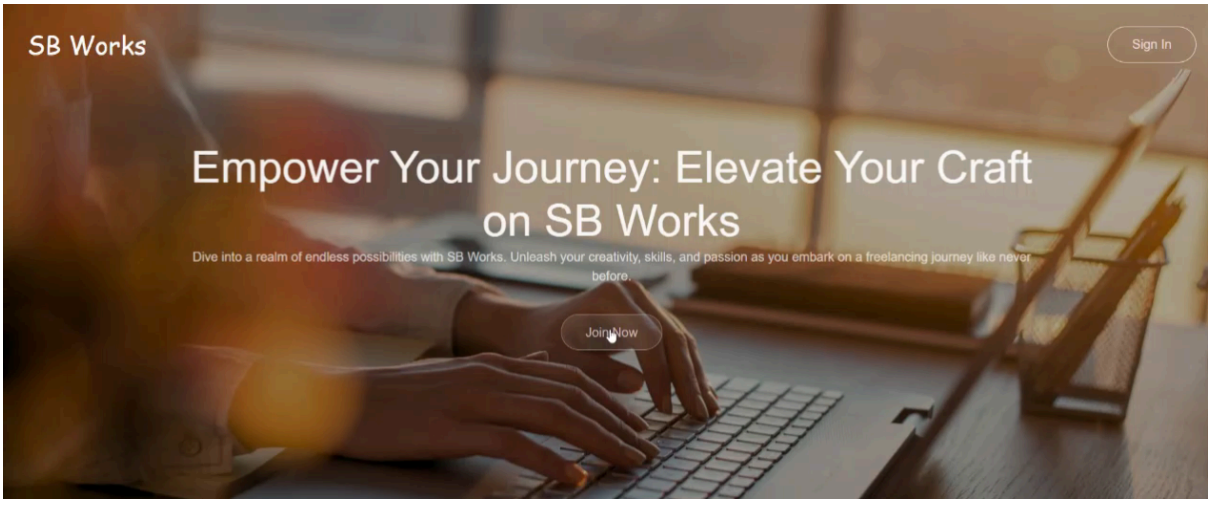
• **Authorization:** Middleware ensures that only the wallet address registered as the "Project Client" can trigger the `reviewTask` API to release funds

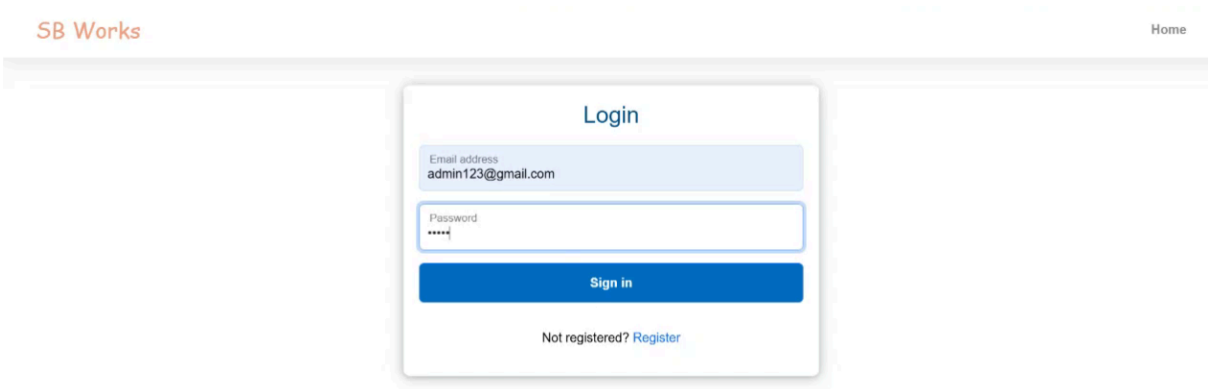
8. User Interface and Testing

• **AuthUI Showcase:** Features a clean dashboard for tracking "Report Cards" and an escrow progress bar for ongoing projects

• **Testing Strategy:** Unit tests for React components, integration tests for API routes, and Anchor's ts-mocha for validating smart contract instructions

10. Screenshots or Demo





11. Known Issues

• **Known Issues:** Transaction confirmation times may vary depending on Solana network congestion, leading to UI "pending" states

12. Future Enhancements

• **Future Enhancements:** Planning for an on-chain dispute resolution board and the ability to pay in SPL tokens (like USDC) instead of only SOL