

Project Support: Freelancing Platform

Overview

SB Works is a full-stack MERN build. It uses MongoDB, Express.js, React.js, and Node.js. The site acts as a freelancing hub for clients and freelancers. Clients can post projects. Freelancers can send offers, chat through the site, and handle deals well. An admin panel allows for easing and help.

App Roles & Duties

Freelancer Duties

Freelancers must submit completed work through the platform. They need to follow all client requisite and platform rules. Clear and prompt with clients is essential. Freelancers should manage time well and meet all deadlines. Respectful conduct are expected always. Their work must be error-free and of high quality. Transfer should match the agreed project standards.

Client Duty

Clients must provide full project descriptions and clear requisite. They should answer freelancer messages without delay. Timely responses help keep work on track. Clients must give all needed input for task completion. Payments should be made promptly after work is delivered. Giving helpful feedback improves the project cycle. Clients should also rate freelancers honestly based on performance.

Admin Duties

Admins are in charge of keeping the platform secure. They must protect data unity and user data. Admins enforce platform rules for all users. They handle disputes and resolve issues fairly. Admins offer technical support when needed by users. They also work to improve system speed and parts. Regular updates help keep the platform running smoothly.

Backend Building Plan

Project Start

The backend is built around a Node.js setup using Express.js as the web frame. It starts with project folder setup and module adding. The backend uses Express.js for server setup and Mongoose for handling database work. More tools like dotenv, cors, and body-parser handle setting values, cross-site access, and request reading.

Database Setup

MongoDB is the main database, either kept local or through MongoDB Atlas. The database has these groups:

- **Users** – keeps basic account data like roles (freelancer, client, admin)
- **Freelancers** – keeps more data for freelancer profiles, like skills and skills
- **Projects** – has details about client-made projects, like budgets and needs
- **Diligence** – keeps freelancer offers with rates and work links
- **Chats** – handles talk history between users about projects

Server Setup

An Express.js server is set up to handle all HTTP request work. Middleware is used for request reading and cross-site handling. The build team splits routes based on their use. Routes include user, project, offer, chat, and freelancer.

API Route Meanings

The system sorts API ends by roles and uses for better order.

- **User Routes** handle sign up, login, and profile care
- **Project Routes** handle project posting, listing, and getting
- **Offer Routes** handle freelancer offers and client reviews
- **Chat Routes** help user talk within projects

Data Shaping

Mongoose forms set the shape for each data thing. The Freelancer form adds to the User form with more traits. Each model supports full CRUD actions to handle kept data.

User Check

The backend uses check using JWT (JSON Web Tokens) for safe access. Middleware keeps user times safe. It checks tokens and stops entry to allowed users only.

Project Life Handling

Clients can make and handle project lists. Freelancers can look at open projects, apply using set offers, and get feedback from clients. This cycle helps openness and strong bidding.

Talk & Teamwork

A safe, project-based chat part allows clear talk between clients and freelancers. Help for feedback swaps and possible file adds makes teamwork better.

Admin Work

The admin panel gives system tools. These include account control, project watching, and entry to deal logs. Admins ensure the site runs smoothly and enforce the rules.

Database Building Plan

The server links to a MongoDB setup using a set URI for data keeping. The database has groups that keep user accounts, project data, offers, freelancer profiles, and messages. Mongoose handles all database work, making sure of shaped asks and form keeping.

Frontend Building Steps

Start and Setup

The frontend is made using React.js. The project starts by setting up the cover shape and adding the needed libraries. These include packs for routing, HTTP asks, state care, and looks.

User Look and Design

Makers built repeat-use parts to keep a clean and steady user look. These parts include sailing bars, forms, buttons, and project cards. The layout works well on screens of all sizes. It ensures a smooth user go through on any device.

Routing and Moving

React Router handles client-side moving, letting smooth switches between views. It allows easy moves to pages like login, dashboard, and project lists.

Frontend-Backend Joining

The frontend talks with the backend through API calls. Axios handles HTTP asks, like getting and sending data to and from the backend. React's state hooks allow quick updates. Check tokens help keep routes safe and control time entry.

Setting Setup

A setting setup file (.env) is made to safely keep secret changing things like:

- Server port
- MongoDB URI
- JWT secret key

Putting Live Plan

Frontend Putting Live

The build team puts the React app on cloud hosting sites like Vercel or Netlify. They build the app for live use and upload it using linked stores.

Backend Putting Live

The team puts the backend live using a cloud site like Render or Heroku. The live step starts with uploading all server files. Then the team sets up build and start commands. The team stores setting values securely in the provider's panel.

Optional Upgrades

Future upgrades can enhance the platform in many ways. Adding payment options will make transactions easier. A live chat feature using socket-based messages can help with faster talks. An alert system, either by email or in-app, will keep users informed. A review and rating system can help clients judge freelancer work. Ranking freelancers based on past tasks will build trust. These steps can improve user go through and site growth.

Conclusion

SB Works is a growing, split-up MERN stack freelancing site. It has main parts like project listing, bidding, talk, check, and admin control. With solid deployment and upkeep, the app can grow. Makers can add new parts and improve its scale.