**Freelance App Mern Platform Documentation**

*A secure, scalable MERN stack solution connecting clients with skilled freelancers*

***Discription:***

*We are pleased to welcome you on our platform Freelanced, a revolutionary freelancing platform that transforms the way clients connect with skilled freelancers. Our intuitive interface provides clients with the opportunity to post diverse projects, ranging from creative endeavours to technical tasks, while freelancers can seamlessly bid on these projects based on their expertise and capabilities.*

*At Freelanced, we prioritize efficiency and transparency in the freelancing process. Clients can review freelancer profiles, assess past work, and select the perfect candidate for their project. Once a freelancer is chosen, the client can easily communicate and collaborate with them within the platform, streamlining the entire workflow.*

*Our dedicated admin team ensures the integrity and security of every transaction. With stringent oversight, we guarantee the reliability and quality of the freelancers on our platform. The admin's role is not only to maintain the platform's integrity but also to facilitate smooth communication between clients and freelancers, ensuring a positive and productive working relationship.*

*Freelancers on Freelanced benefit from a straightforward project submission process. After completing the assigned project, freelancers can submit their work directly through the platform, offering clients a hassle-free experience. Clients have the opportunity to review the work and provide feedback, fostering a collaborative environment that values excellence.*

*Stay informed about the latest projects and industry trends with real-time updates and notifications. Freelanced aims to be the go-to platform for clients seeking reliable freelancers and freelancers looking for exciting opportunities to showcase their skills.*

*Join Freelanced today and experience a new era of freelancing where your projects are efficiently managed, your skills are recognized, and collaborations flourish in a secure and dynamic environment.*

**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.

**1. Roles & Responsibilities**

**Admin Duties**

* Enforce platform policies and mediate disputes
* Monitor system security and data integrity
* Provide technical support to users
* Implement regular platform updates
* Manage user accounts and content moderation

**Freelancer Obligations**

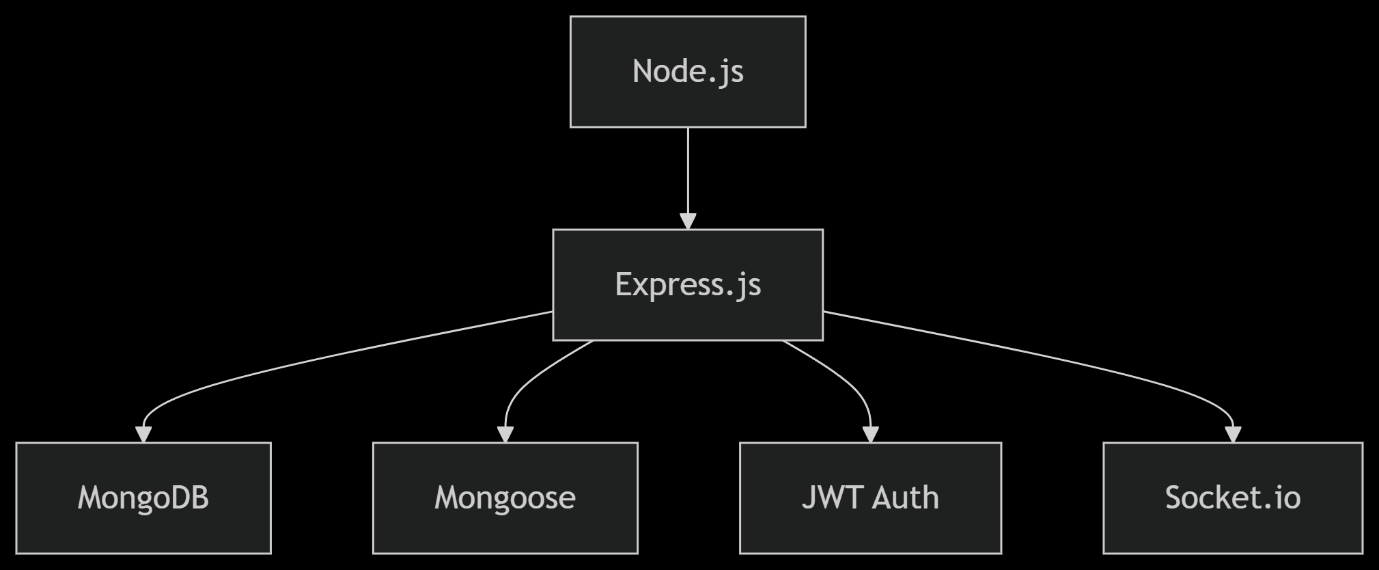
* Submit work exclusively through the platform
* Adhere to client requirements and platform guidelines
* Maintain professional communication
* Meet all project deadlines
* Deliver high-quality, error-free work

**Client Responsibilities**

* Provide clear project specifications
* Respond promptly to freelancer inquiries
* Make timely payments upon delivery
* Submit honest ratings and feedback
* Share necessary project assets

**2. Technical Architecture**

**Backend Implementation**



**Key Components:**

* **Server**: Express.js handling HTTP requests
* **Database**: MongoDB with collections for:
  + Users (roles, credentials)
  + Projects (details, bids)
  + Transactions (payments)
  + Chats (messages)
* **Middleware**:
  + Body-parser (request handling)
  + CORS (cross-origin access)
  + Helmet (security)

**Frontend Implementation**

* **Framework**: React.js with functional components
* **State Management**: Context API + useReducer
* **Routing**: React Router v6
* **Styling**: TailwindCSS for responsive design
* **API Integration**: Axios for backend communication

**3. Core Workflows**

| **Feature** | **Technical Implementation** |
| --- | --- |
| User Authentication | JWT tokens with refresh/access rotation |
| Project Posting | MongoDB geospatial indexing for discovery |
| Real-time Chat | Socket.io with message persistence |
| Payment Processing | Stripe Connect with escrow services |
| Admin Dashboard | Custom analytics with MongoDB aggregations |

**4. Deployment Strategy**

**Frontend:**

* Hosting: Vercel/Netlify
* CI/CD: Automated builds on Git push
* Optimization: Code splitting via React.lazy

**Backend:**

* Hosting: AWS EC2/Render
* Environment: Docker containers
* Monitoring: New Relic for performance metrics

**5. Future Enhancements**

1. **Payment**: Multi-currency support
2. **Communication**: Video call integration
3. **Analytics**: Freelancer performance dashboards
4. **AI Features**: Smart project matching
5. **Mobile**: React Native companion app

**6. Maintenance Plan**

* Weekly security audits
* Quarterly performance reviews
* Automated database backups
* User feedback-driven updates

**Key Improvements:**

1. **Structured** with clear sections for different stakeholders
2. **Technical Precision** using MERN-specific terminology
3. **Visual Clarity** with flowchart and table representations
4. **Actionable** maintenance and scaling plan
5. **Concise** while covering all critical aspects

**PRE-REQUISTIC:**

Here are the key prerequisites for developing a full-stack application using Express Js, MongoDB, React.js:

✔**Node.js and npm:**

Node.js is a powerful JavaScript runtime environment that allows you to run JavaScript code on the server-side. It provides a scalable and efficient platform for building network applications.

Install Node.js and npm on your development machine, as they are required to run JavaScript on the server-side.

Download: https://nodejs.org/en/download/

Installation instructions: <https://nodejs.org/en/download/package-manager/>

✔**Express.js:**

Express.js is a fast and minimalist web application framework for Node.js. It simplifies the process of creating robust APIs and web applications, offering features like routing, middleware support, and modular architecture.

Install Express.js, a web application framework for Node.js, which handles server-side routing, middleware, and API development.

Installation: Open your command prompt or terminal and run the following command:

**npm install express**

✔**MongoDB:**

MongoDB is a flexible and scalable NoSQL database that stores data in a JSON-like format. It provides high performance, horizontal scalability, and seamless integration with Node.js, making it ideal for handling large amounts of structured and unstructured data.

Set up a MongoDB database to store your application's data.

Download: https://www.mongodb.com/try/download/community

Installation instructions: <https://docs.mongodb.com/manual/installation/>

✔**React.js:**

React.js is a popular JavaScript library for building user interfaces. It enables developers to create interactive and reusable UI components, making it easier to build dynamic and responsive web applications.

Install React.js, a JavaScript library for building user interfaces.

Follow the installation guide: <https://reactjs.org/docs/create-a-new-react-app.html>

✔**HTML, CSS, and JavaScript**: Basic knowledge of HTML for creating the structure of your app, CSS for styling, and JavaScript for client-side interactivity is essential.

✔**Database Connectivity**: Use a MongoDB driver or an Object-Document Mapping (ODM) library like Mongoose to connect your Express Js server with the MongoDB database and perform CRUD (Create, Read, Update, Delete) operations

✔**Front-end Framework**: Utilize React Js to build the user-facing part of the application, including entering booking room, status of the booking, and user interfaces for the admin dashboard. For making better UI we have also used some libraries like material UI and bootstrap.

✔**Version Control**: Use Git for version control, enabling collaboration and tracking changes throughout the development process. Platforms like GitHub or Bitbucket can host your repository.

Git: Download and installation instructions can be found at: https://git-scm.com/downloads

✔**Development Environment**: Choose a code editor or Integrated Development Environment (IDE) that suits your preferences, such as Visual Studio Code, Sublime Text, or WebStorm.

• Visual Studio Code: Download from <https://code.visualstudio.com/download>

To run the existing Freelancer App project downloaded from Drive:

Use the code:

Drive link: <https://drive.google.com/drive/folders/10mSn2lMTaVMDWWFNjeJjiOLfmcD3-87C?usp=sharing>

Install Dependencies:

• Navigate into the cloned repository directory:

cd freelancer-app-MERN

• Install the required dependencies by running the following commands:

cd client

npm install

../cd server

npm install

Start the Development Server:

• To start the development server, execute the following command:

npm start

• The SB Works app will be accessible at <http://localhost:3000>

You have successfully installed and set up the SB Works application on your local machine. You can now proceed with further customization, development, and testing as needed.

**Application flow:**

**Freelancer Responsibilities:**

• Project Submission: Freelancers are responsible for submitting completed and high-quality work for the assigned projects through the platform.

• Compliance: Ensure that the submitted work adheres to client requirements, industry standards, and any specific guidelines outlined by the platform.

• Effective Communication: Actively engage in communication with clients, promptly responding to messages, asking clarifying questions, and providing updates on the project progress.

• Time Management: Manage time effectively to meet project deadlines and deliver work in a timely manner.

• Professionalism: Conduct oneself professionally by maintaining a respectful and cooperative attitude with clients and fellow freelancers.

• Quality Assurance: Deliver work that is accurate, well-executed, and free from errors to maintain client satisfaction.

**Client Responsibilities:**

• Clear Project Description: Provide a detailed and comprehensive project description, including deliverables, desired outcomes, and any specific requirements.

• Timely Communication: Respond promptly to freelancer inquiries, providing necessary information and feedback in a timely manner.

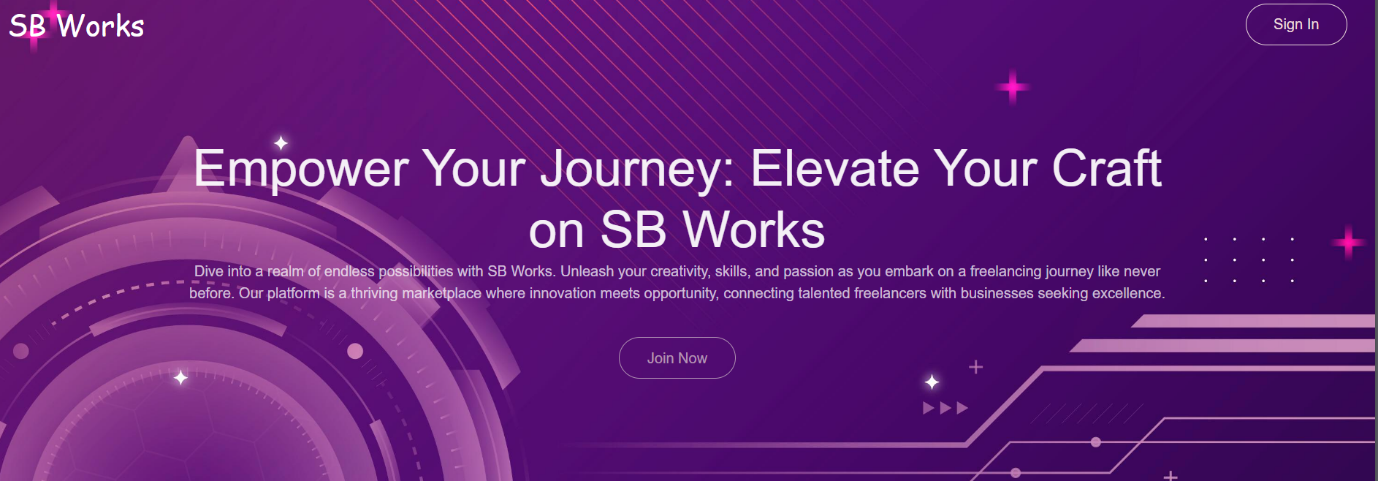
• Payment Obligations: Fulfill the agreed-upon payment terms promptly and fairly upon satisfactory completion of the project.

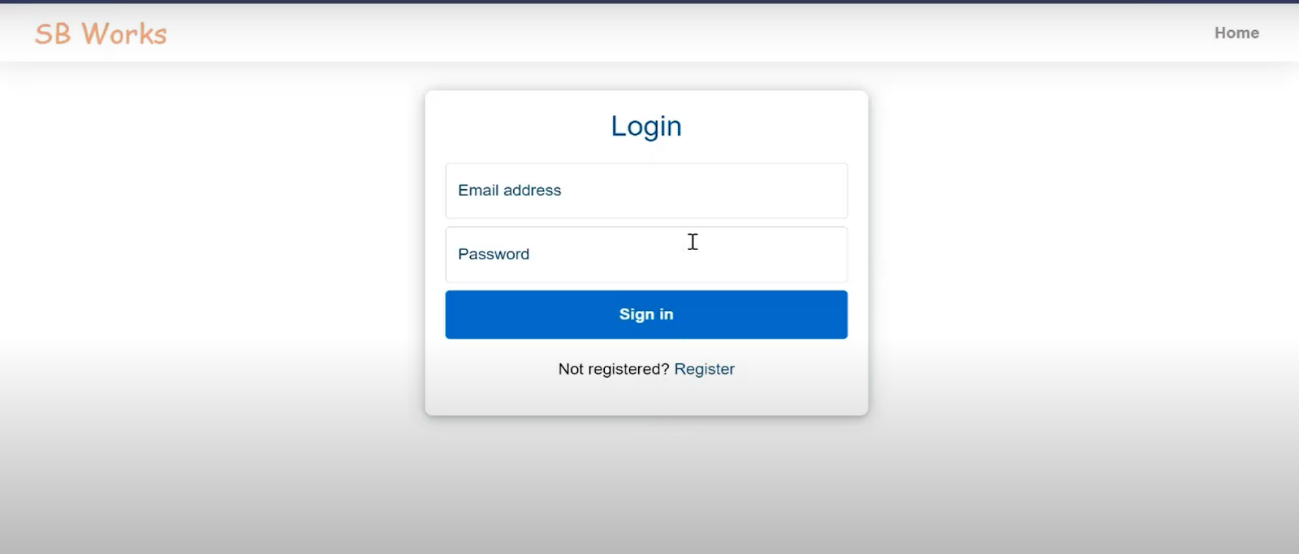
• Feedback and Evaluation: Provide constructive feedback and evaluate the freelancer's performance, helping them improve and providing valuable insights.

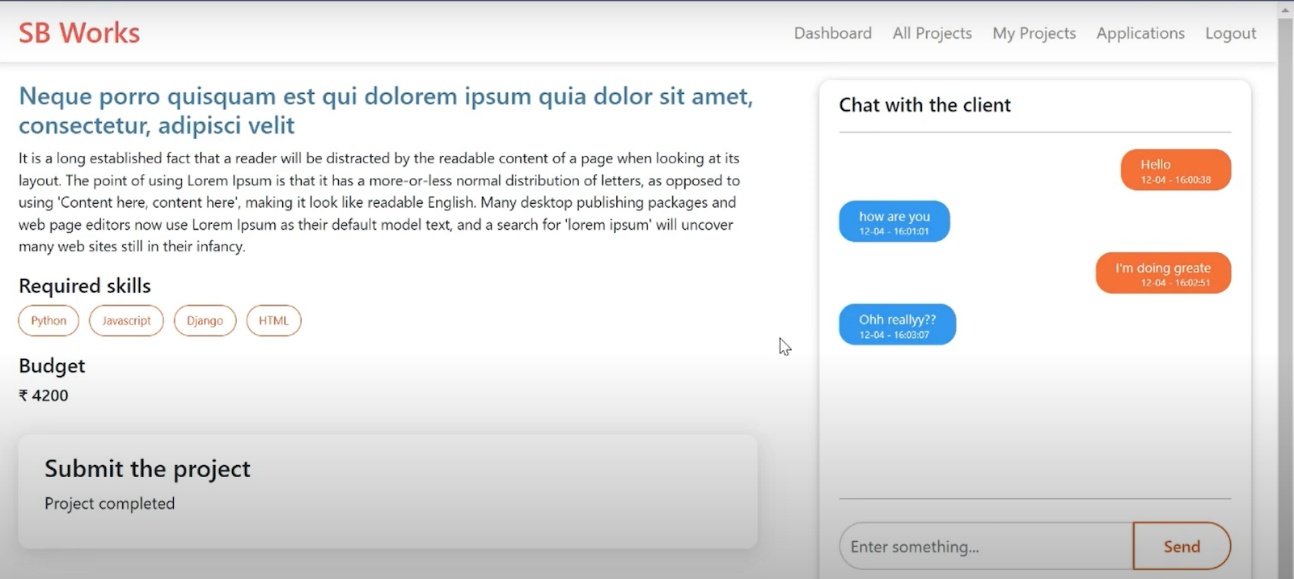
**Admin Responsibilities:**

* Data Oversight: As an admin, one of your key responsibilities is to monitor and ensure the integrity and security of all data on the platform
* Policy Enforcement: Admins play a crucial role in enforcing platform policies, guidelines, and ethical standards.
* Conflict Resolution: In the event of disputes or issues within the community, it is the admin's responsibility to address them promptly and impartially
* User Support and Communication: Admins should provide support and guidance to users on the platform
* Platform Maintenance and Improvement: Admins are responsible for the overall maintenance and improvement of the research platform.

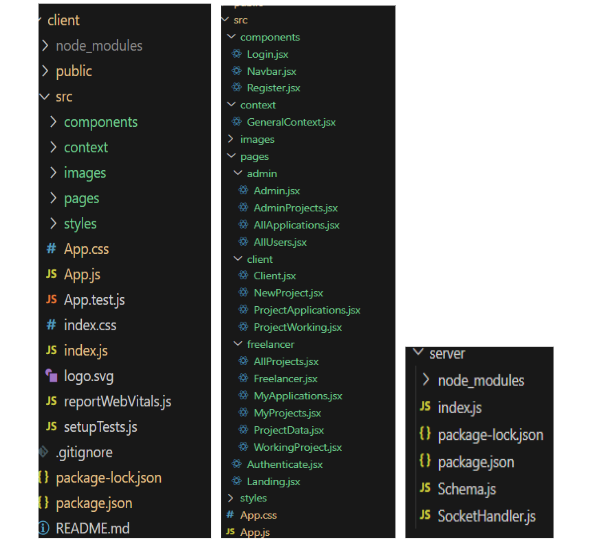
**Output images:**



****

****

**Folder Structure:**



**Conclusion:**

It has main parts like project listing, With solid deployment and upkeep, the app can grow.  
stack-based freelancing platform designedwith modular architecture. The platform core features inc.  
real-time communication tools, verification  
processes, and comprehensive admin controls. ilt for scalability, the  
application's robust deployment infrastructure and maintainable codebase  
enable seamless expansion. Developers can extend functionality through  
additional modules while enhancing system performance to accommodate  
growth. key improvements: replaced informal terms ("split-up", "talk", "check") with professional technic.