

SB Works – Freelancing Application MERN

1 Introduction

SB Works is a full-stack **Freelancing Web Application** developed using the **MERN Stack** (**MongoDB**, **Express.js**, **React.js**, **Node.js**). The platform provides an online marketplace where **Clients** can post projects and **Freelancers** can bid, communicate, and submit completed work. The application includes **real-time chat**, **role-based dashboards**, and **modern glowing animated UI**.

2 Objectives of the Project

- To connect freelancers and clients on a single platform
- To allow secure project bidding and management
- To enable real-time communication using Socket.IO
- To implement a visually attractive UI using animations & glowing effects
- To demonstrate full-stack development using MERN

3 System Architecture

Architecture Pattern

- Client–Server Architecture
- REST API + WebSockets

Components

- Frontend – React.js
- Backend – Node.js + Express.js
- Database – MongoDB Atlas
- Real-time Communication – Socket.IO

4 Technology Stack

Frontend

- React.js
- HTML5
- CSS3 (Glowing animations, gradients, transitions)
- JavaScript (ES6)
- Axios
- React Router DOM

Backend

- Node.js
- Express.js
- Socket.IO
- bcrypt (password encryption)
- dotenv (environment variables)

Database

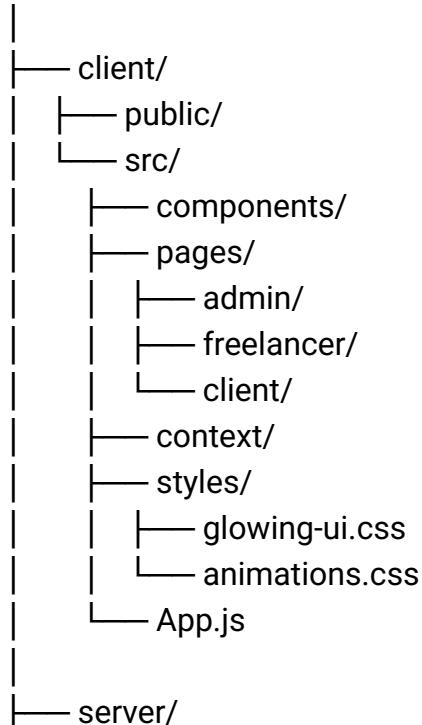
- MongoDB Atlas
- Mongoose ODM

Tools

- VS Code
- MongoDB Compass
- Postman
- Git & GitHub

5 Project Folder Structure

SB-Works-Freelancing-Application-MERN/



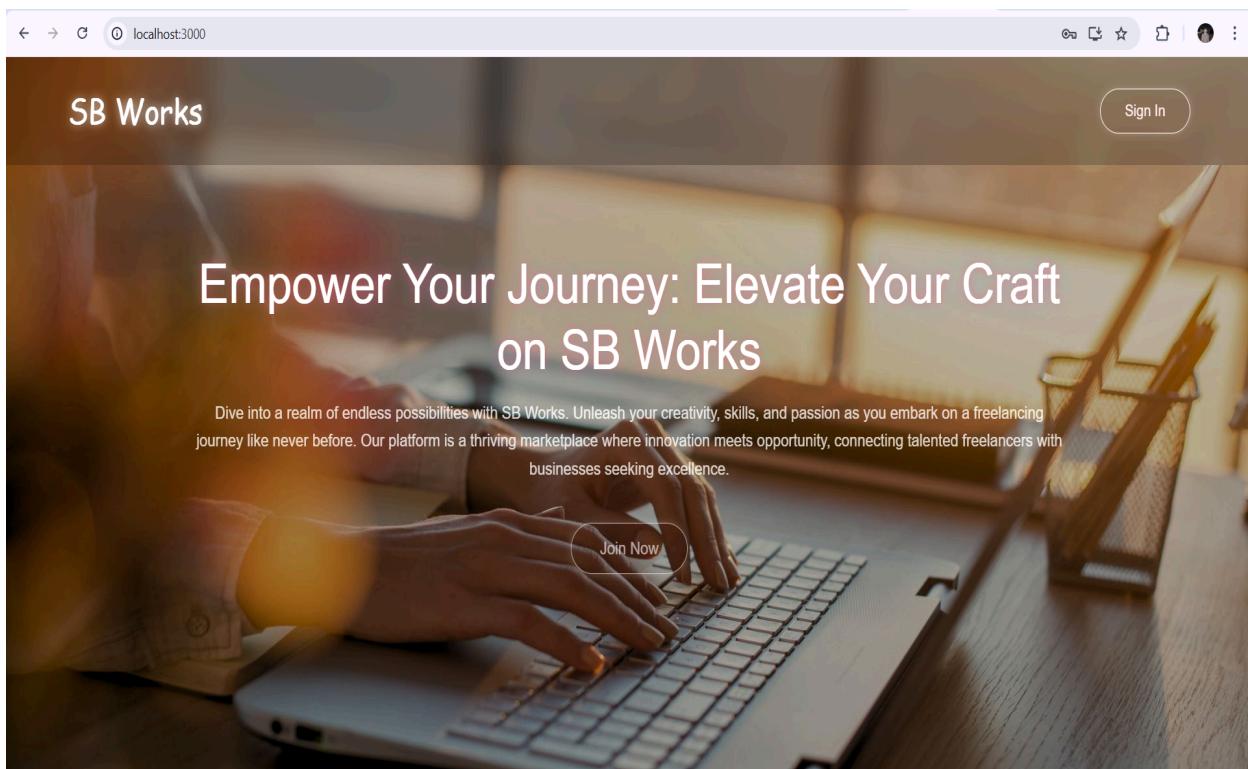
```
|   └── index.js  
|   └── Schema.js  
|   └── SocketHandler.js  
|   └── .env  
|   └── package.json  
  
└── screenshots/  
    └── README.md
```

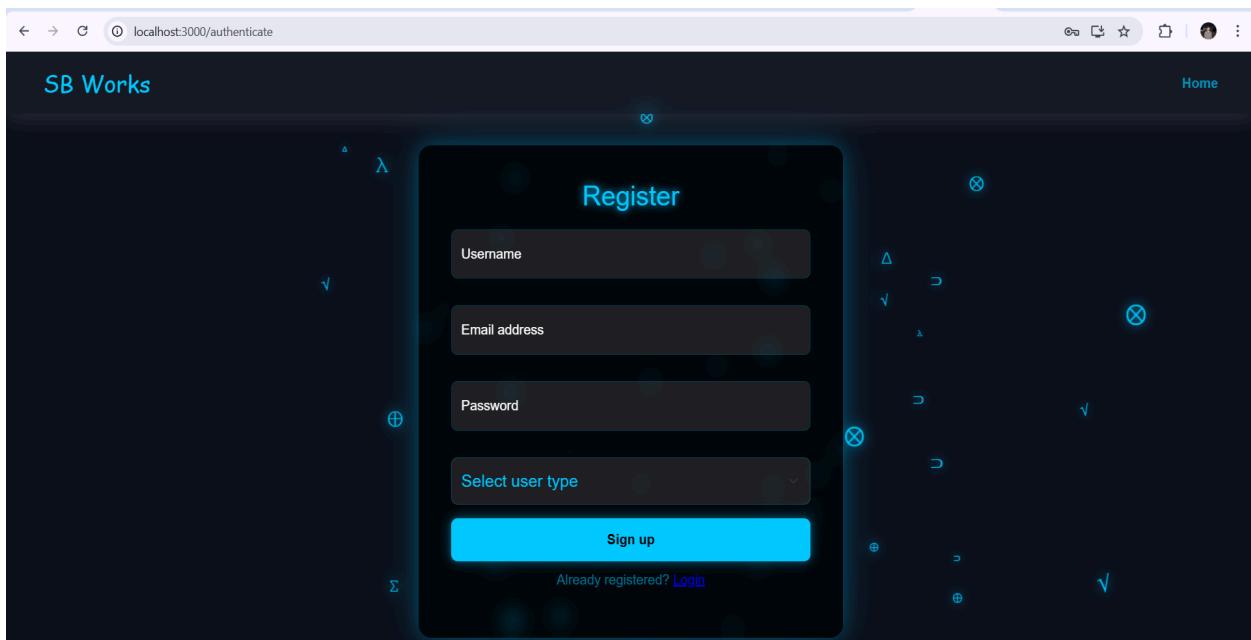
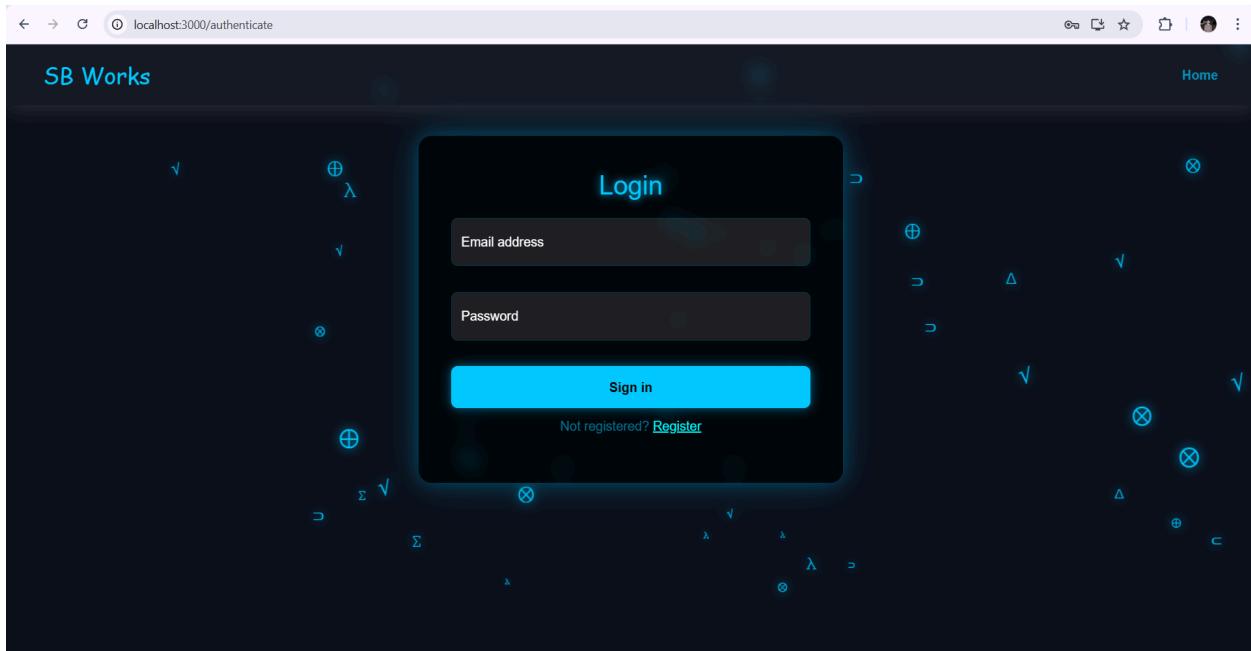
6 User Roles & Functionalities

👤 Authentication

- User registration & login
- Role-based access (Client / Freelancer)
- Encrypted passwords using bcrypt

outputs:





Client Module

- Create and post projects
- View all freelancer applications
- Accept or reject applications
- Assign project to freelancer
- Monitor project progress

outputs:

The screenshot shows the 'Applications' section of the SB Works platform. It displays two entries for a project named 'Tale app'.
1. The first entry shows the project details: 'Tale app', 'A multimedia platform for story writing and listening', 'Skills' (frontend, backend, business logic), and 'Budget - ₹ 1000'.
2. The second entry shows the same details but includes a 'Proposal' section with 'Skills' (python) and 'Proposed Budget - ₹ 0'. It features a green 'Approve' button and an orange 'Decline' button.
3. The status of the second entry is indicated as 'Status: Rejected'.

The screenshot shows the 'My projects' section of the SB Works client interface. It lists two projects:
1. A project with 'Budget - ₹ 0' and 'Status - Available', created on 'Tue Dec 23 2025 11:56:43'.
2. A project named 'Tale app' with 'Budget - ₹ 1000' and 'A multimedia platform for story writing and listening', with 'Status - Assigned', created on 'Tue Dec 23 2025 11:18:39'.
A red bracket on the left side groups the first two items, while a red bracket on the right side groups the last two items.

localhost:3000/new-project

SB Works

Post New Project

Project title

Description

Budget (in ₹)

Required skills (comma separated)

Submit

localhost:3000/client-project/694a2d377d18854e88ca36a8

SB Works

Tale app
A multimedia platform for story writing and listening

Required skills: frontend, backend, business logic

Budget: ₹ 1000

Submission
No submissions yet!!

Chat with the Freelancer

hi
12-23 - 08:18:20

hello
12-23 - 08:49:57

okok
12-23 - 08:41:56

Enter something... Send

Freelancer Module

- View all available projects
- Filter projects by skills
- Submit bids with proposal & budget
- View application status
- Submit completed project work
- Chat with client in real time

outputs:

The screenshot shows the SB Works application interface. At the top, there's a navigation bar with links for Dashboard, All Projects, My Projects, Applications, and Logout. Below the navigation, a project card for "Tale app" is displayed. The project description is "A multimedia platform for story writing and listening". It lists "Required Skills" as frontend, backend, and business logic. A budget of ₹ 1000 is mentioned. To the right of the project card is a "Chat with Client" section showing a conversation between the user and a client. The messages are: "hi" (12/23/2025, 1:48:20 PM), "hello" (12/23/2025, 2:10:57 PM), and "okok" (12/23/2025, 2:11:56 PM). There's also a text input field "Type a message..." and a "Send" button.

The screenshot shows the "All Projects" page of the SB Works application. On the left, there's a sidebar titled "Filters" with a "Skills" section containing checkboxes for python, machine learning, frontend, backend, and business logic. The main area displays three project cards. The first project is "AutoAttend" with a budget of ₹ 500, described as "Face detect and auto attendance". It shows 1 bid at ₹ 450 (avg bid). The second project is "Budget ₹ 0" with 0 bids at ₹ 0 (avg bid). The third project is "Tale app" with a budget of ₹ 1000, described as "A multimedia platform for story writing and listening". It shows 0 bids at ₹ 0 (avg bid). The date "Tue Dec 23 2025 14:53:22" is visible above the AutoAttend card.

localhost:3000/myApplications

SB Works

Dashboard All Projects My Projects Applications Logout

AutoAttend
Face detect and auto attendance

Skills
`python` `machine learning`

Budget - ₹ 500

Proposal
u have less time

Skills
`python`

Proposed Budget - ₹ 450
Status: PENDING

Tale app
A multimedia platform for story writing and listening

Skills
`frontend` `backend` `business logic`

Budget - ₹ 1000

Proposal
Skills

Proposed Budget - ₹ 0
Status: PENDING

Tale app
A multimedia platform for story writing and listening

Skills

Proposal
Skills

localhost:3000/freelancer

SB Works

Dashboard All Projects My Projects Applications Logout

Current Projects
1

Completed Projects
0

Applications
4

Funds
₹ 0

My Skills
`python`

Description
A interpreter language.

Update Profile

The screenshot shows the 'My projects' section of the SB Works application. At the top, there is a navigation bar with links for Dashboard, All Projects, My Projects, Applications, and Logout. Below the navigation bar, the title 'My projects' is displayed. A button labeled 'Choose project status' is visible. The main content area contains a card for a project named 'Tale app'. The card includes the budget ('Budget - ₹ 1000'), a description ('A multimedia platform for story writing and listening'), and the status ('Status - Assigned'). To the right of the card, the date 'Tue Dec 23 2025 11:18:39 GMT+0530 (India Standard Time)' is shown.

🛡 Admin Module

- View all users
- View all applications
- Monitor system activity

outputs:

The screenshot shows the 'All Applications' section of the SB Works application. At the top, there is a navigation bar with links for Home, All Users, Projects, Applications, and Logout. Below the navigation bar, the title 'All Applications' is displayed. The main content area lists two applications: 'AutoAttend' and 'Tale app'. The 'AutoAttend' application card includes details such as 'Face detect and auto attendance', 'Skills (python, machine learning)', 'Budget - ₹ 500', 'Client: srijith', 'Client Id: 694a5f4eb340a8b6c290fc01', and 'Client email: sri@gmail.com'. The 'Tale app' application card includes details such as 'A multimedia platform for story writing and listening', 'Skills (frontend, backend, business logic)', 'Budget - ₹ 1000', 'Client: Adhi', and 'Client Id: 6949090ff79b59357e2e82ba'. To the right of each application card, there is a 'Proposal' section with details like 'u have less time', 'Skills (python)', 'Proposed Budget - ₹ 450', 'Freelancer: Adhi', 'Freelancer Id: 6949096bf79b59357e2e82c7', 'Freelancer email: adhi1@gmail.com', and 'Status: PENDING'.

localhost:3000/admin-projects

SB Works

Home All Users Projects Applications Logout

Filters

Skills

- python
- machine learning
- frontend
- backend
- business logic

All projects

AutoAttend

Budget ₹ 500
Client name: srijith
Client email: sri@gmail.com
Face detect and auto attendance

python machine learning

1 bids ₹ 450 (avg bid)

Status - Available

The Dec 23 2025 14:51:22 GMT+0530 (India Standard Time)

Budget ₹ 0
Client name: Adhi

The Dec 23 2025 11:58:43 GMT+0530 (India Standard Time)

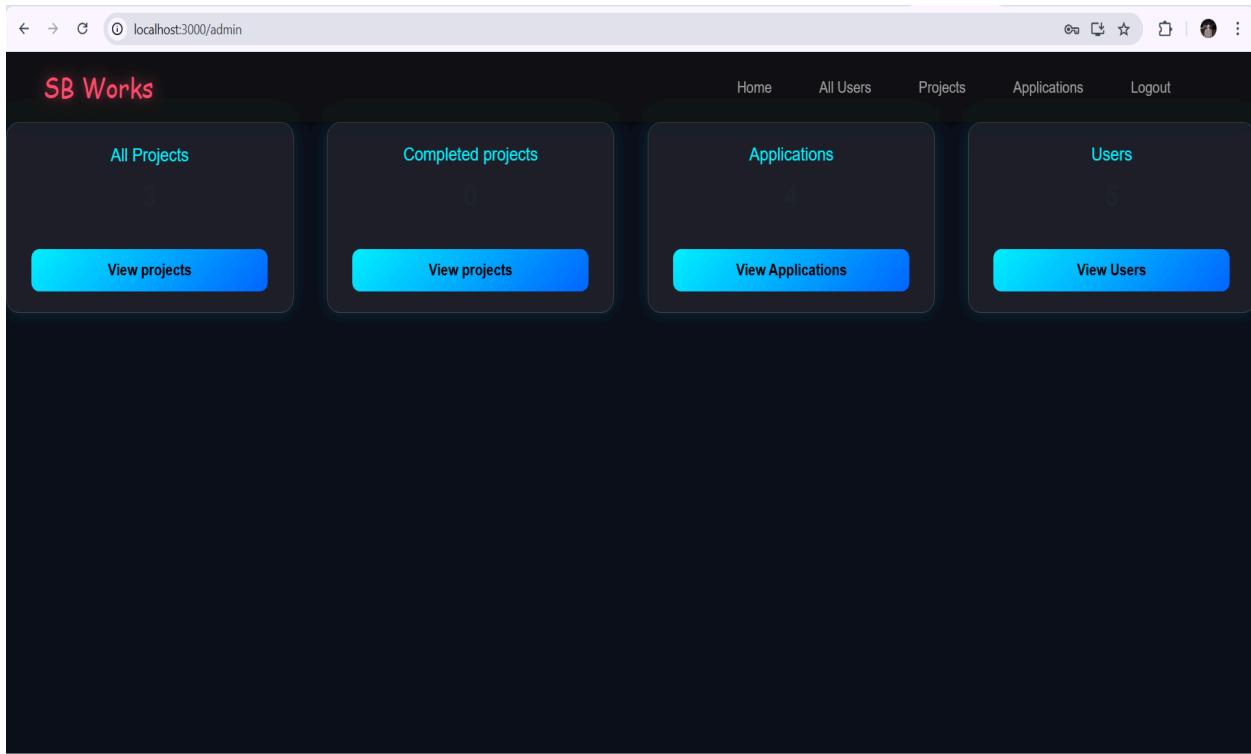
localhost:3000/all-users

SB Works

Home All Users Projects Applications Logout

All Users

User Id	Username	Email	User Role
6949090179b59357e2e82ba	Adhi	adhi@gmail.com	client
6949096b79b59357e2e82c7	Adhi	adhi1@gmail.com	freelancer
694a51e6b340a8b6c296fbcd	Adhi	adhi2@gmail.com	Admin
694a5234b340a8b6c296fb6	Adhi	adhi3@gmail.com	admin
694a5f4eb340a8b6c296fc1	srijith	sri@gmail.com	client



7 User Interface Design

UI Highlights

- Digital glowing gradients
- Glassmorphism panels
- Animated hover cards
- Neon skill badges
- Smooth page transitions
- Custom glowing scrollbars

CSS Features Used

- @keyframes animations
- Gradient backgrounds
- box-shadow glow effects
- Hover transitions
- Backdrop blur
- Responsive layouts

8 Real-Time Chat System

Technology Used

- Socket.IO

Chat Features

- Project-based chat rooms
- Instant message delivery
- MongoDB chat persistence
- Sender-based message UI
- Time-stamped messages

9 Backend API Overview

Example Endpoints

Method	Endpoint	Description
GET	/fetch-projects	Get all projects
GET	/fetch-project/:id	Single project
POST	/make-bid	Submit bid
POST	/submit-project	Submit project
GET	/fetch-chats/:id	Fetch chats
GET	/fetch-users	Admin users



Environment Variables

PORt=5000

MONGO_URI=mongodb+srv://username:password@cluster.mongodb.net/freelancer_ap
p

Installation & Execution

Backend

cd	server
npm	install
npm run dev	

Frontend

cd	client
npm	install
npm start	

MongoDB

- MongoDB Atlas (cloud)
- Or local MongoDB using mongod

1 Database Design

Collections

- Users
- Projects
- Applications
- Chats

Relationships

- One client → many projects
- One project → many applications
- One project → one assigned freelancer
- One project → one chat room

Security Measures

- Password hashing (bcrypt)
- Environment variables (.env)
- Role-based access control
- CORS enabled
- Secure MongoDB Atlas access

Testing

- Manual UI testing
- API testing using Postman
- Socket testing using multiple browsers

Limitations

- No payment gateway integration
- No email notifications
- Admin privileges limited to viewing
- No file uploads (only links)

11 ER DIAGRAM – Freelancing Application

◆ Entities & Relationships

1 User

User

_id (PK)
username
email
password
usertype (Client/Freelancer)
skills[]

2 Project

Project

_id (PK)
title
description
budget
skills[]

status
clientId (FK → User)
freelancerId (FK → User)
bids[]
submission
submissionAccepted

3 Application (Bid)

Application

_id (PK)
projectId (FK → Project)
clientId (FK → User)
freelancerId (FK → User)
proposal
bidAmount
estimatedTime
status
freelancerSkills[]

4 Chat

Chat

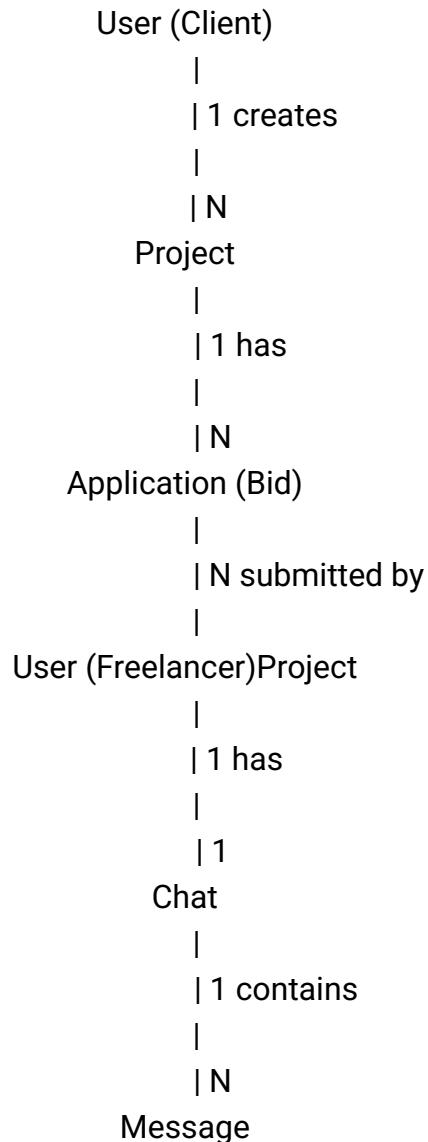
_id (PK)
projectId (FK → Project)
messages[]

5 Message

Message

senderId (FK → User)
text
time

12 ER Relationship Diagram (Textual)



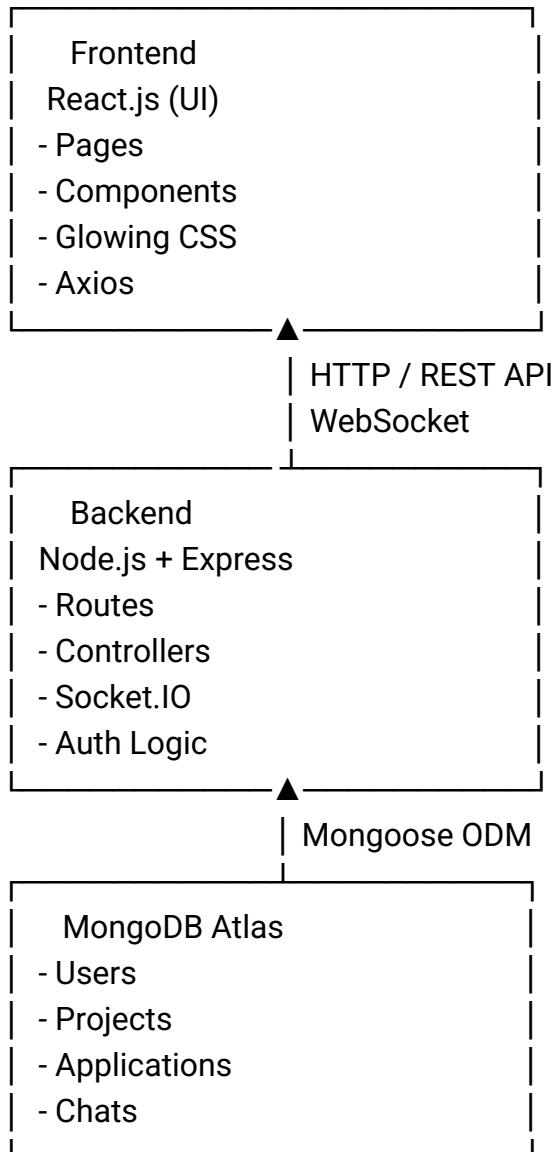
ER Relationship Summary

Entity	Relationship
User → Project	One Client can create many Projects
Project → Application	One Project can have many Applications

Freelancer → Application	One Freelancer can submit many Applications
Project → Chat	Each Project has one Chat room
Chat → Message	One Chat has many Messages

13 SYSTEM ARCHITECTURE DIAGRAM

◆ High-Level Architecture



Detailed Architecture Flow

1 Client / Freelancer (Browser)

- React UI
- Animated glowing UI
- Form validation
- Sends API requests using Axios



2 Express Server (Node.js)

- Handles REST APIs
- Business logic
- Socket.IO for chat
- Authentication & authorization

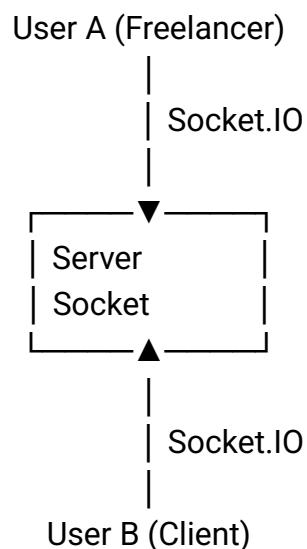


3 MongoDB Database

- Stores users, projects, applications
- Stores chat messages persistently
- Schema validation using Mongoose



Real-Time Chat Architecture



- ✓ Messages stored in MongoDB
- ✓ Project-based chat rooms
- ✓ Instant updates

Security Architecture

- Password hashing using **bcrypt**
- Environment variables (.env)
- MongoDB Atlas authentication
- Role-based UI rendering
- CORS protection

14 Future Enhancements

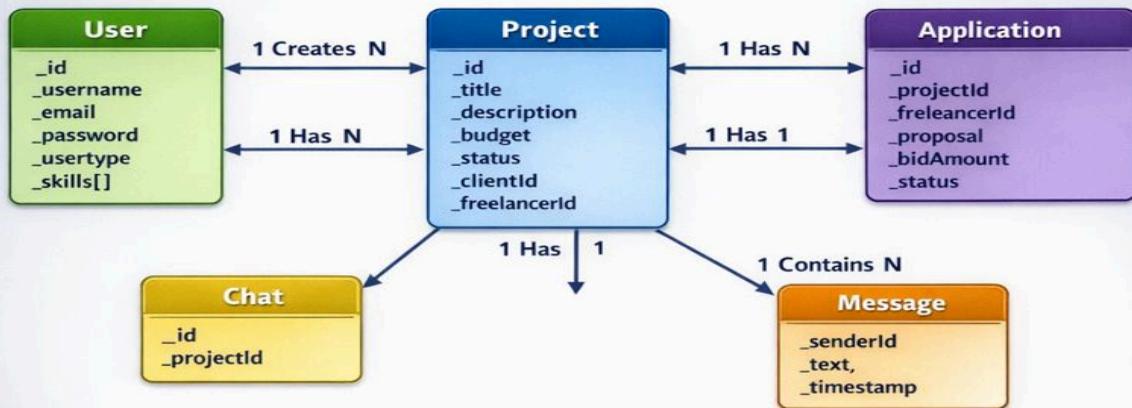
- Payment system (Stripe/Razorpay)
- Profile ratings & reviews
- File uploads
- Email notifications
- Admin analytics dashboard
- JWT-based authentication

15 Conclusion

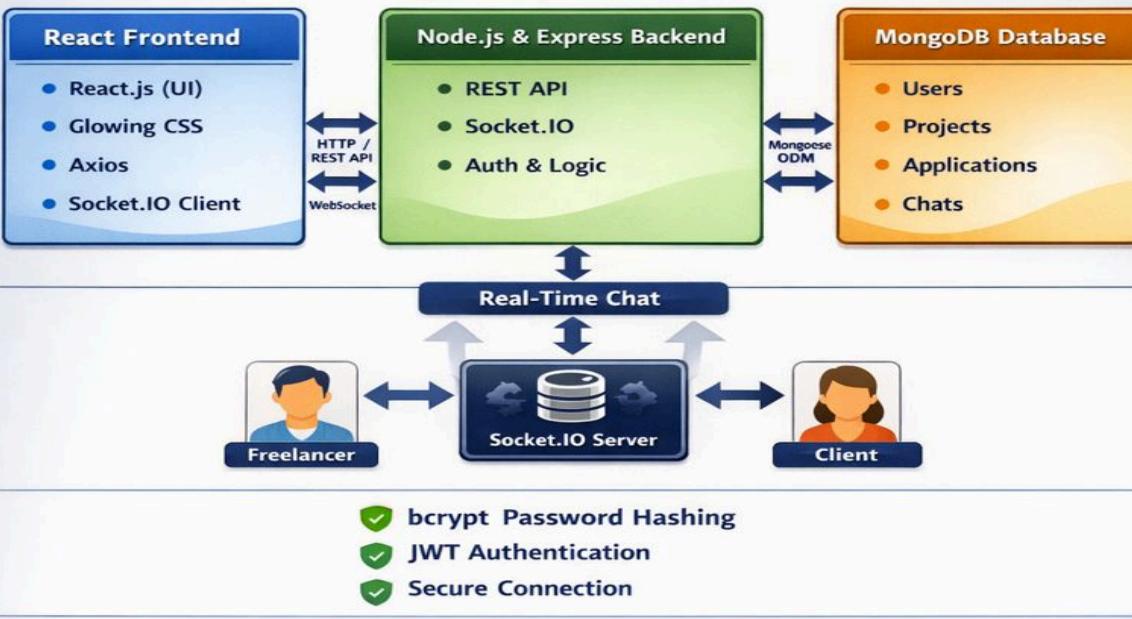
SB Works is a MERN-based freelancing application that allows clients and freelancers to connect on a single platform. Clients can post projects and review applications, while freelancers can apply, bid, and communicate in real time. The system uses a secure backend, MongoDB for data storage, and Socket.IO for live chat. The application is easy to use, visually attractive, and suitable for learning, academic projects, and practical demonstrations.

- Built using the MERN stack
- Supports Client and Freelancer roles
- Enables project posting, bidding, and submissions
- Real-time chat using Socket.IO
- Secure and responsive design

ER DIAGRAM – Freelancing Application



SYSTEM ARCHITECTURE – MERN Stack



Developed By

Angothu Adhisheshu

GitHub: <https://github.com/Adhisheshu1210/>