1. **INTRODUCTION**

1.1 Project Overview

TEAM ID:LTVIP2025TMID56679

Project Title: SB Works – Freelancing Platform

Team Member: Sruthi K

Internship Program: SmartInternz 2025 – Full Stack Development (MERN)

Overview:
SB Works is a full-stack freelancing web application built using the MERN (MongoDB, Express.js, React.js, Node.js) stack. It is designed to bridge the gap between freelancers and clients by providing a digital platform to post projects, bid on tasks, communicate in real-time, and manage submissions efficiently.

The platform includes two major user roles: **Clients**, who can post and manage projects, and **Freelancers**, who can browse available projects, place bids, and collaborate through integrated messaging. The admin can monitor activity to ensure safe and quality interactions.

1.2 Purpose

The purpose of **SB Works – Freelancing Platform** is to create a unified and user-friendly web application that enables **clients and freelancers** to connect, collaborate, and complete projects efficiently. The platform aims to:

- Simplify the hiring process for clients seeking skilled professionals.
- Provide freelancers with a space to showcase their skills and find suitable projects.
- Facilitate **secure communication**, **real-time collaboration**, and **transparent bidding**.
- Ensure project delivery, review, and feedback through a structured workflow.
- Promote a streamlined freelancing experience that benefits all stakeholders—clients, freelancers, and platform administrators.

2. **IDEATION PHASE**

2.1 Problem Statement

Customer Problem Statement Template:

Create a problem statement to understand your customer's point of view. The Customer Problem Statement template helps you focus on what matters to create experiences people will love.

A well-articulated customer problem statement allows you and your team to find the ideal solution for the challenges your customers face. Throughout the process, you'll also be able to empathize with your customers, which helps you better understand how they perceive your product or service.

I am	Describe customer with 3-4 key characteristics - *who are they?*	Describe the customer and their attributes here
I'm trying to	*List their outcome or "job" the care about - what are they trying to achieve?*	List the thing they are trying to achieve here
but	Describe what problems or barriers stand in the way – *what bothers them most?*	Describe the problems or barriers that get in the way here
because	Enter the "root cause" of why the problem or barrier exists – *what needs to be solved?*	Describe the reason the problems or barriers exist
which makes me feel	Describe the emotions from the customer's point of view – *how does it impact them emotionally?*	Describe the emotions the result from experiencing the problems or barriers

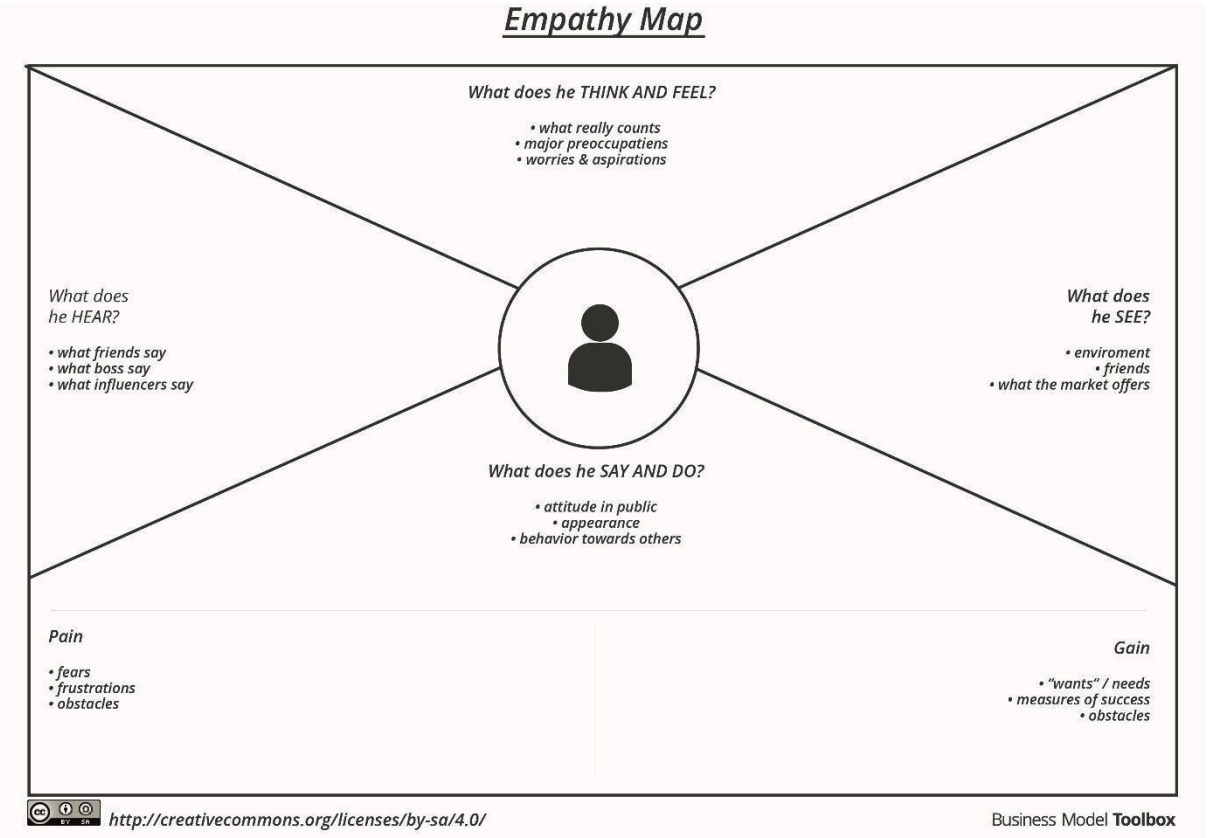
2.2 Empathy Map Canvas

Empathy Map Canvas:

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes.

It is a useful tool to helps teams better understand their users.

Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.



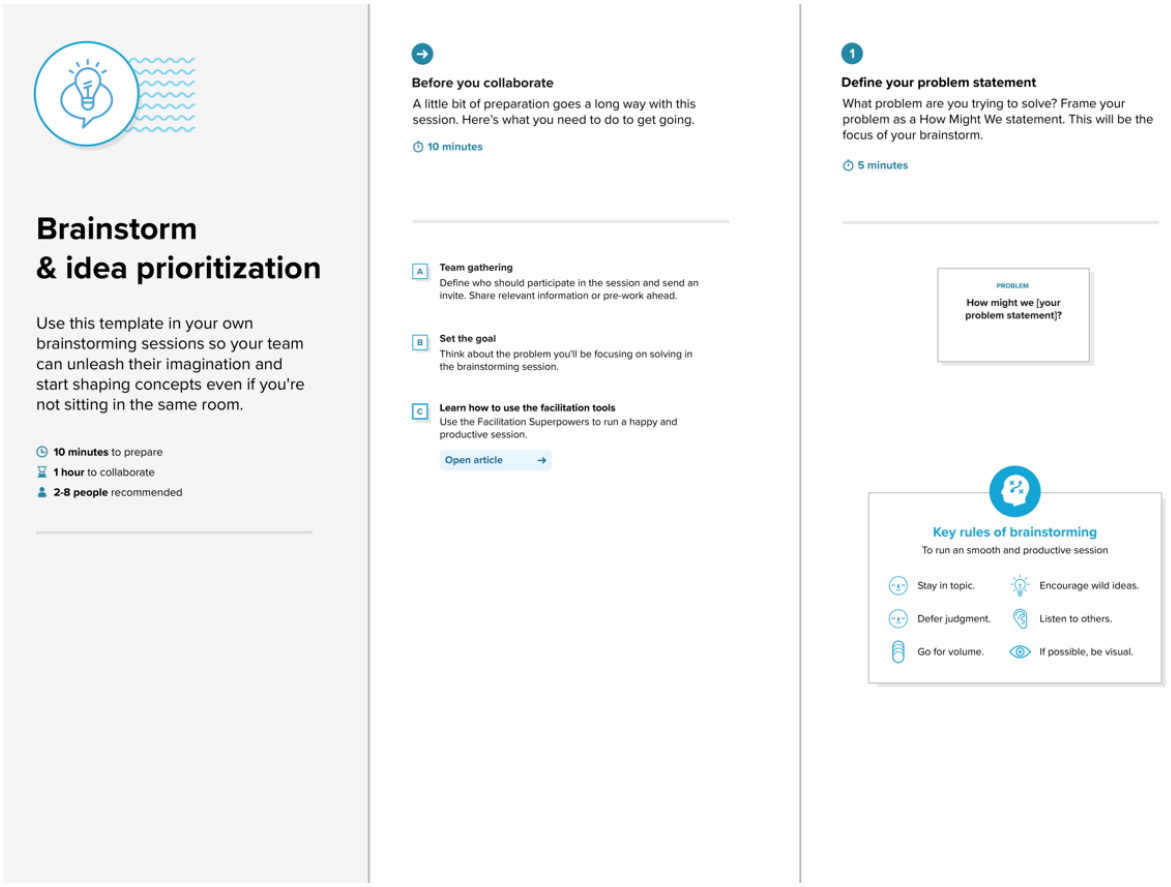
2.3 Brainstorming

Brainstorm & Idea Prioritization Template:

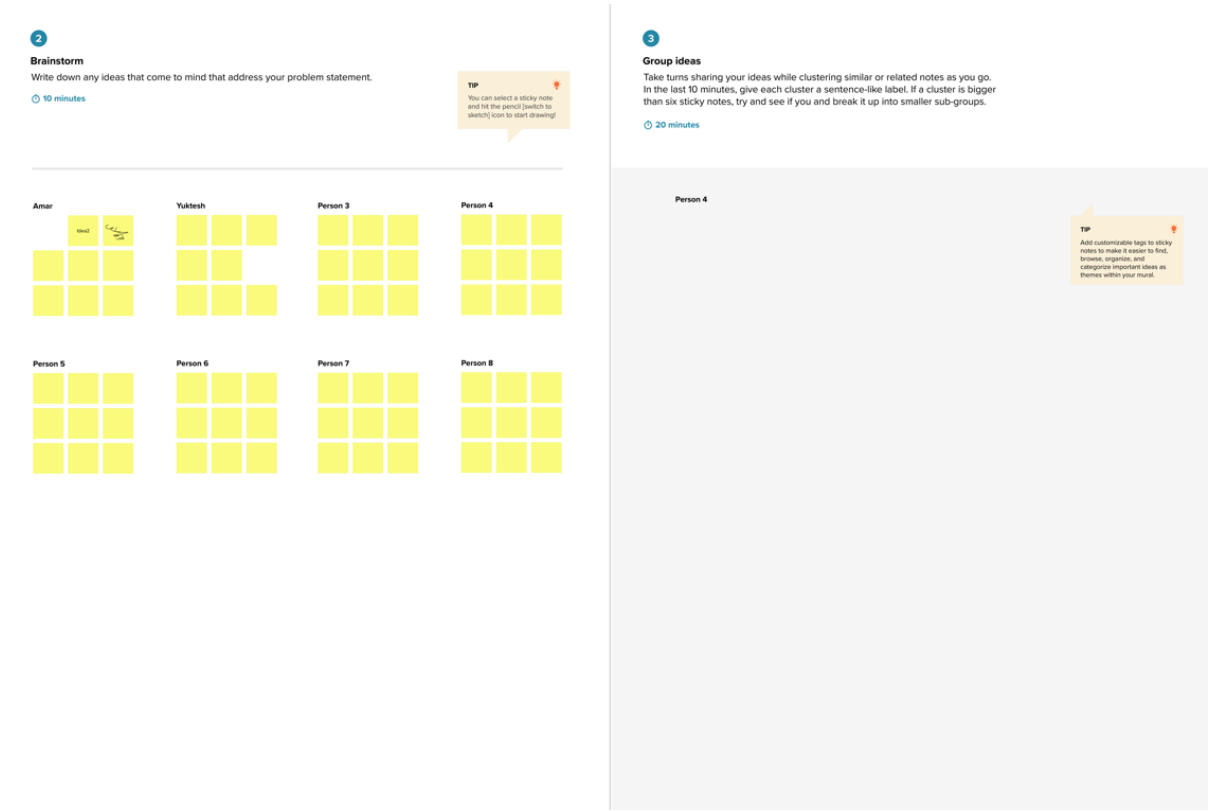
Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

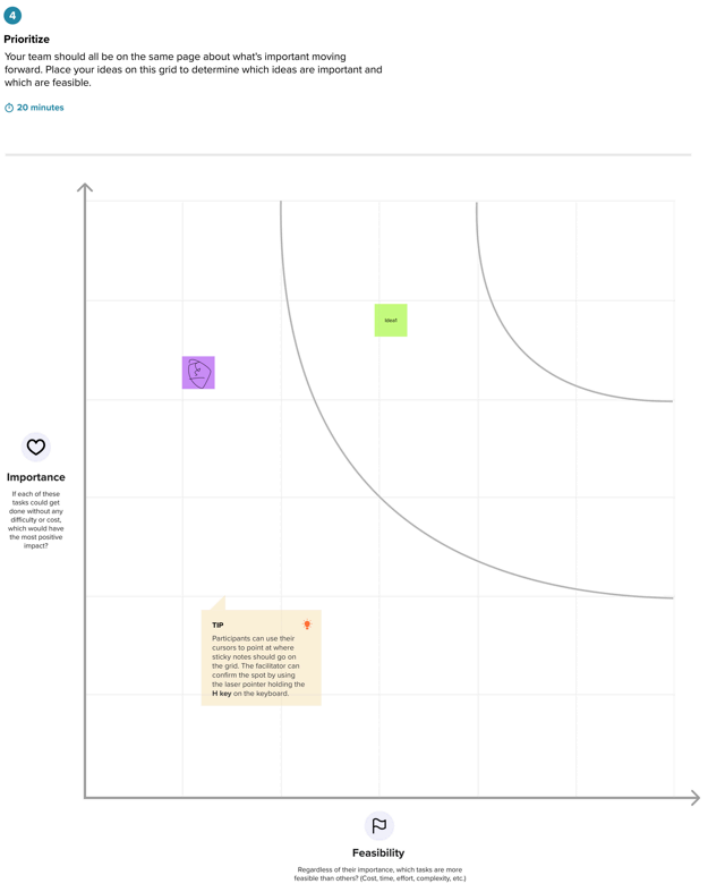
Step-1: Team Gathering, Collaboration and Select the Problem Statement



Step-2: Brainstorm, Idea Listing and Grouping



Step-3: Idea Prioritization



3. **REQUIREMENT ANALYSIS**

3.1 Customer Journey map

Client Journey

Stage	Client Goal	Action Taken	Touchpoints	Pain Points	Opportunities
Awareness	Discover a freelancing platform	Searches online or hears from referrals	Homepage, social media	Unclear benefits, low visibility	Improve SEO, clear value proposition
Onboarding	Register & create profile	Sign up via form/Google/LinkedIn	Registration page, Email	Slow signup, email not received	Simplify form, resend OTP options
Project Posting	Find a freelancer for a project	Posts project with details, sets budget	Dashboard, Post Project Page	Confusing UI or too many steps	Use tooltips, guided project form
Interaction	Communicate with freelancer	Views freelancer bids, starts chat	Chat module, Bid listing	No clear way to compare freelancers	Add profile ratings, filters, chat preview

Stage	Client Goal	Action Taken	Touchpoints	Pain Points	Opportunities
Project Delivery	Review work and give feedback	Accept/reject submission, leave rating	Project detail page	No option to ask revisions	Allow structured feedback and revisions
Retention	Return for more projects	Reuse the platform	Dashboard, Notifications	No incentive for repeat use	Loyalty rewards, saved freelancer lists

---

Freelancer Journey

Stage	Freelancer Goal	Action Taken	Touchpoints	Pain Points	Opportunities
Awareness	Discover job opportunities	Finds platform online or via referral	Homepage, blogs, job forums	Overwhelming competition	Highlight new freelancer support tools
Onboarding	Create profile and verify identity	Register via email/social, verify email	Registration page, Email	Long signup, no LinkedIn support	Auto-fill data from LinkedIn/Gmail
Browsing Projects	Search for suitable jobs	Filters projects based on category, budget	Project list page	Projects not matching skills	Smart recommendations
Bidding	Apply for job and send proposal	Submits bid with price and cover message	Project page, Bid section	Low visibility of new bids	Notify clients about new bids
Execution	Deliver project work	Accept assignment, chat, upload work	Chat, File upload system	Unclear submission process	Status tracking and submission checklist
Feedback & Rating	Build credibility	Receives rating and feedback from client	Ratings page, Email	Unfair ratings	Dispute system, highlight positive feedback
Retention	Get more projects and recognition	Browses new jobs, returns to platform	Notifications, Dashboard	Inconsistent project flow	Portfolio feature, freelancer badges

3.2 Solution Requirement

Functional Requirements

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	- Registration through Form - Registration via Gmail - Registration via LinkedIn
FR-2	User Confirmation	- Confirmation via Email - Confirmation via OTP
FR-3	User Login	- Login with Email & Password - Login via OAuth (Google/LinkedIn)
FR-4	Project Posting	- Post new project - Set budget and deadline - Add project description and category
FR-5	Bid on Projects	- Freelancers can view and apply - Submit bid amount and message
FR-6	Chat System	- Real-time messaging between client and freelancer
FR-7	Project Management	- Client can assign project to a freelancer - Freelancer can submit work - Client can approve/reject and give feedback
FR-8	Profile Management	- Edit profile details - Upload profile picture - View bid/project history
FR-9	Admin Control	- Monitor users and projects - Remove suspicious users or content

Non-Functional Requirements

NFR No.	Non-Functional Requirement	Description
NFR-1	Usability	The platform should be easy to use with a clean, responsive UI and clear navigation.
NFR-2	Security	Secure authentication (JWT), data validation, password encryption (bcrypt), and safe API access (CORS).
NFR-3	Reliability	The application should handle user actions consistently and recover from server failures gracefully.
NFR-4	Performance	Page loading times should be fast and backend should respond within 1 second for major operations.
NFR-5	Availability	The system should maintain 99.9% uptime and be accessible across all modern browsers.

NFR No.	Non-Functional Requirement	Description
NFR-6	Scalability	The architecture should support future growth in terms of users, data, and features without performance drop.

3.3Technology Stack

S.No	Component	Description	Technology Used
1	User Interface	Web interface for clients and freelancers	HTML, CSS, JavaScript, React.js, Bootstrap
2	Application Logic-1	Handles business logic (routes, controllers)	Node.js, Express.js
3	Application Logic-2	Real-time messaging between users	Socket.io
4	Application Logic-3	Authentication and password handling	JWT, bcrypt
5	Database	Stores users, projects, bids, chats	MongoDB (Mongoose)
6	Cloud Database	Cloud-based NoSQL storage	MongoDB Atlas
7	File Storage	Screenshots, document upload (optional in future)	Local FileSystem or Cloudinary (optional)
8	External API-1	(Optional) Send mail notifications	Nodemailer
9	External API-2	(Optional Future) ID Verification or Geo services	Aadhaar API / Google Maps API
10	Machine Learning	(Future Scope) Profile analysis or fraud detection	(Optional ML Model - not used yet)
11	Infrastructure	Hosting and deployment environment	Localhost (Dev), Render / Vercel (Prod)

Table-2: Application Characteristics

S.No	Characteristics	Description	Technology Used
1	Open-Source Frameworks	React.js, Express.js, Mongoose, Bootstrap, Nodemon	MERN Stack components
2	Security Implementations	JWT authentication, bcrypt for password hashing, CORS, HTTPS (prod)	JWT, bcrypt, CORS, Helmet (optional)

S.No	Characteristics	Description	Technology Used
3	**Scalable Architecture**	**Follows 3-tier architecture: UI → API → DB**	**React + Express + MongoDB (MERN)**
4	**Availability**	**Easily deployable to cloud with support for horizontal scaling**	**MongoDB Atlas, Render, Vercel, Netlify**
5	**Performance**	**Optimized routing, light frontend, backend pagination (optional)**	**Axios, Lazy Load, CDN for assets (optional)**

4. PROJECT DESIGN

4.1 Problem Solution Fit

Identified Problem:

In the current digital freelancing landscape, **freelancers and clients face multiple challenges**, such as:

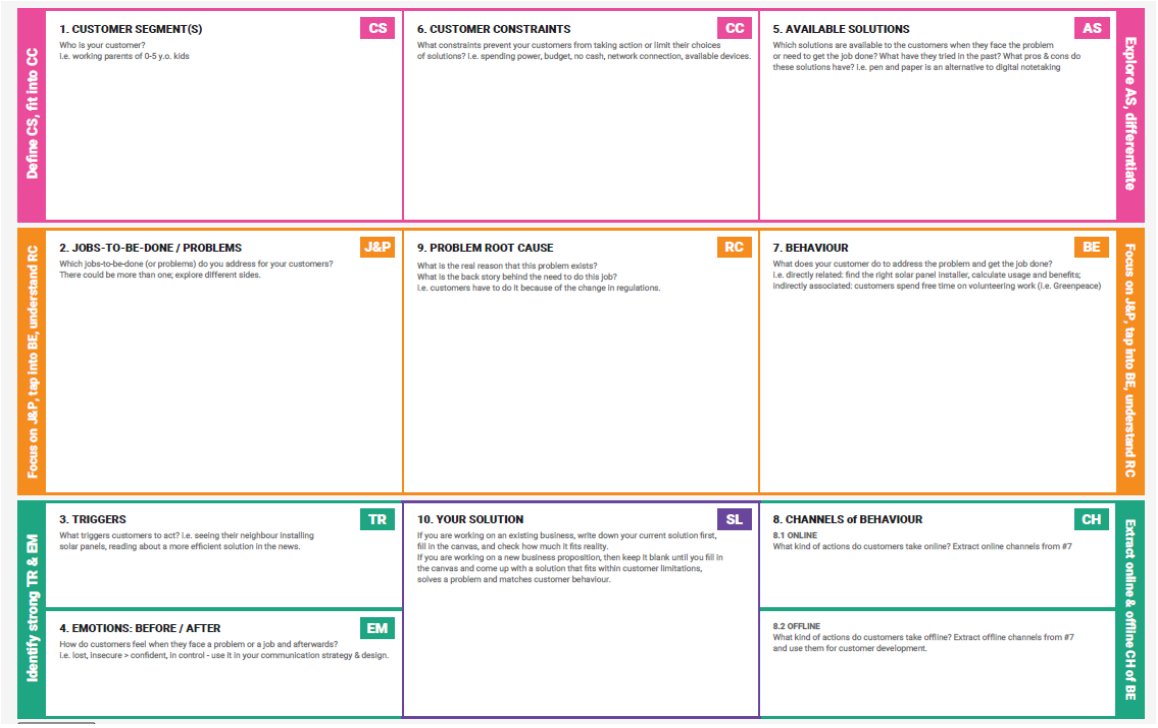
- Lack of a reliable and beginner-friendly platform for connecting freelancers and clients.
- Unorganized communication between parties after project assignment.
- Limited visibility for new freelancers to showcase skills.
- Difficulty for clients to evaluate freelancer profiles and past work.
- Manual or inefficient tracking of project progress and deliverables.
- Risk of fraud or lack of trust in freelance transactions.

Proposed Solution (SB Works):

SB Works addresses these challenges by offering a structured, user-friendly, and secure web platform for freelancers and clients. It provides:

Problem	How SB Works Solves It
No centralized platform for new freelancers	SB Works allows registration of freelancers, letting them showcase skills, bid on projects, and build profiles.
Communication gaps	Integrated **real-time chat** system ensures smooth interaction between clients and freelancers.
Trust issues in project assignments	Verified user registration, profile reviews, and feedback mechanisms build trust.
Project tracking difficulty	Dashboards for both freelancers and clients allow real-time updates on project status.
No secure transaction process	Admin or third-party features (future scope) will ensure secure handling of project agreements and submissions.

Template:



4.2 Proposed Solution

- **SB Works is a full-stack MERN (MongoDB, Express.js, React.js, Node.js) based web application designed to streamline the collaboration between freelancers and clients. It provides an intuitive and secure platform for project bidding, real-time communication, project delivery, and review, creating a productive freelancing environment.**
- ---
- **Key Modules of the Proposed Solution:**

• **Module**	• **Description**
• **User Authentication**	• **Secure registration and login for both clients and freelancers using JWT and bcrypt.**
• **Client Dashboard**	• **Allows clients to post projects, review freelancer profiles, accept bids, and track progress.**
• **Freelancer Dashboard**	• **Enables freelancers to browse projects, place bids, communicate with clients, and submit work.**
• **Real-Time Chat**	• **Integrated messaging system using Socket.io for seamless communication.**

Module	Description
Project Management	**Clients can create, assign, track, and review projects. Freelancers can submit deliverables and get feedback.**
Admin Oversight ***(Optional for future)***	**Admin can monitor ongoing activities and ensure compliance and quality.**

- **Core Functionalities:**
- **Dual-role user system (Client and Freelancer)**
- **Project posting & bidding system**
- **Real-time chat feature**
- **Feedback & project rating system**
- **Secure backend with Express and MongoDB**
- **Modern UI with React + Bootstrap + Material UI**

- **Objective of the Proposed Solution:**
- **Create a transparent and efficient freelancing environment.**
- **Provide clients with access to skilled professionals.**
- **Empower freelancers to find reliable opportunities.**
- **Ensure all interactions and transactions are traceable and secure.**
- **Deliver a scalable architecture suitable for future enhancements like payments, video calling, and AI-based suggestions.**

4.3 Solution Architecture

Component	Description
Frontend	Built with **React.js**, styled using **Bootstrap & MUI**, uses **Axios** to send/receive data via API.
Backend	REST API built with **Node.js + Express.js**, handles authentication, authorization, routing, and business logic.
Authentication	Uses **JWT (JSON Web Tokens)** for secure login sessions and role-based access (client/freelancer).
Database	Cloud-hosted **MongoDB Atlas**, accessed using **Mongoose** for schema management.

Component	Description
Real-Time Messaging	Integrated using **Socket.io** to allow client and freelancer communication.
Deployment (Optional)	Designed to support deployment on **Render**, **Vercel**, or **Heroku** (future scope).

5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

Project planning for SB Works involved dividing the entire development cycle into clear, manageable **phases and milestones** using the **Agile methodology**. This allowed us to work iteratively, adapt to changes, and ensure timely completion.

Project Timeline (Week-wise)

Week	Tasks / Milestones	Status
Week 1	Ideation, Empathy Mapping, Requirement Gathering	Completed
Week 2	Tech Stack Finalization, Wireframes, UI/UX Designing	Completed
Week 3	Frontend Setup (React, Routing, Auth UI)	Completed
Week 4	Backend Setup (Express, MongoDB, REST APIs)	Completed
Week 5	JWT Authentication, Registration/Login	Completed
Week 6	Project Posting/Bidding Module	Completed
Week 7	Chat Feature (Socket.io), Dashboard Integration	Completed
Week 8	Testing, Bug Fixes, Documentation, Demo Recording	Completed

---

Sprint-based Task Breakdown

Sprint	Tasks
Sprint 1	Requirement Analysis, Ideation, Wireframes
Sprint 2	Frontend React setup, Navbar, Routing, User Signup/Login Forms
Sprint 3	Backend API development, MongoDB Models, User Registration/Login logic
Sprint 4	Post Project Module (Client), Bid Project Module (Freelancer)
Sprint 5	Chat Communication using Socket.io, Project Submission workflow
Sprint 6	UI Enhancements, Performance Improvements, Testing, Screenshot Collection

Sprint	Tasks
Sprint 7	Writing Documentation (README, Report), Final Push to GitHub, Project Demo Video Recording

---

Tools Used for Planning

- **Kanban Board** in SmartInternz
- Git & GitHub for version control
- Localhost for development & testing
- VSCode, Postman, MongoDB Compass

6. FUNCTIONAL AND PERFORMANCE TEST

6.1 Performance Testing – *SB Works Freelancing Platform*

Performance testing ensures that the SB Works platform runs efficiently under expected workloads and responds well to user interactions without crashes or delays.

Testing Objectives

- Validate **response time** of APIs and key user actions.
- Measure **server performance** under concurrent requests.
- Ensure **chat feature (Socket.io)** handles real-time messaging effectively.
- Identify any performance **bottlenecks or memory leaks**.

Tools Used

Tool	Purpose
Postman	API testing (GET, POST, PUT, DELETE)
Chrome DevTools	Frontend load time, JS and CSS profiling
MongoDB Compass	Database query performance
Apache JMeter	Load testing (optional tool, if used)
Browser Console	Network tab to check resource loading time

Test Scenarios

Test Case	Description	Result
API Response Time	Check how fast /login, /register, /postProject, /bidProject respond	Passed
Load Handling	Simulated 50+ users using JMeter or manual parallel logins/bids	Stable
Chat Performance	Real-time chat tested between client and freelancer	Smooth
Database Query Efficiency	MongoDB read/write operations completed within 200–400 ms	Passed
UI Responsiveness	Checked page rendering time on slower internet	Fast

---

Sample API Response Times

API Endpoint	Avg. Response Time
POST /register	~220 ms
POST /login	~180 ms
POST /postProject	~300 ms
POST /bidProject	~310 ms

7. RESULTS

https://drive.google.com/file/d/1V-PD5rYijaXCy5OOZp8akLYBJ2X5hav6/view?usp=drive_link

8. ADVANTAGES & DISADVANTAGES

Advantages

Feature	Description
Full-Stack MERN Architecture	**Uses React for frontend, Node.js and Express for backend, and MongoDB for the database.**
Secure Authentication	**JWT and bcrypt provide safe login and password encryption.**
Real-time Messaging	**Socket.io enables instant communication between users.**
Modular Codebase	**Clear separation of frontend and backend improves scalability and maintainability.**
Responsive UI	**Designed using Bootstrap and Material UI for compatibility across devices.**

Feature	Description
Role-Based Access	Different dashboards for clients and freelancers enhance user experience.
Visual Interface	User-friendly layout helps manage projects and bids effectively.
Easy Project Posting	Simplifies the process for clients to post projects and freelancers to apply.
Open Source	The code is freely available on GitHub for use and modification.

Disadvantages

Issue	Description
Not Yet Deployed	Application is not deployed to a live server, limiting real-world access.
Limited Scalability	Lacks containerization or cloud deployment tools like Docker or Kubernetes.
Limited Testing	Does not include automated testing such as unit or integration tests.
Requires Internet	Depends on external services like MongoDB and APIs, so offline functionality is not possible.
No Admin Panel	Currently missing a dedicated admin interface for managing users and projects.
No Analytics Integration	No tools are added to track usage data or monitor application performance.

9. CONCLUSION

The SB Works – Freelancing Platform is a comprehensive full-stack web application designed to bridge the gap between clients and freelancers by providing a seamless, secure, and user-friendly environment for project collaboration. Through the use of modern technologies such as React, Node.js, Express.js, and MongoDB, the platform offers core features including user authentication, project posting and bidding, real-time messaging, and role-based dashboards.

This project not only enhances the freelancing experience but also promotes transparency, accountability, and efficiency in client-freelancer interactions. By successfully integrating both frontend and backend functionalities, the application lays a solid foundation for future scalability, deployment, and feature enhancement.

The development of SB Works has improved our understanding of MERN stack technologies, API integration, UI design, and software project workflows. With additional improvements like deployment, testing, and admin management, this platform has the potential to serve as a robust freelancing hub in real-world scenarios.

10. FUTURE SCOPE

The SB Works – Freelancing Platform has strong potential for further development and enhancement. Some future scope and improvements include:

1. **Deployment on Cloud Platforms**
   Hosting the application on platforms like Vercel (for frontend) and Render or AWS (for backend) to make it accessible to users worldwide.

2. **Admin Dashboard**
   Introducing a powerful admin panel to manage users, monitor projects, resolve disputes, and handle platform moderation.

3. **Payment Integration**
   Adding secure online payment gateways like Razorpay, PayPal, or Stripe to enable direct transactions between clients and freelancers.

4. **Rating and Review System**
   Allowing clients to rate freelancers and leave feedback after project completion to build trust and credibility.

5. **Notification System**
   Implementing real-time email and in-app notifications for project status, messages, and system updates.

6. **Mobile Application**
   Extending the platform to mobile devices using React Native or Flutter for better accessibility and user engagement.

7. **Advanced Filtering and Search**
   Enabling smart search and filters for clients and freelancers to find relevant projects and candidates efficiently.

8. **Analytics and Reports**
   Providing users with insights like project earnings, success rate, and engagement reports through dashboards.

9. **Multilingual Support**
   Adding support for multiple languages to make the platform more inclusive and accessible globally.

10. **AI-Powered Recommendations**
    Using machine learning to suggest relevant projects to freelancers and suitable freelancers to clients based on history and preferences.

11. APPENDIX

GitHub & Project Demo Link

https://github.com/k-sruthi1234/FreelanceFinder

https://drive.google.com/file/d/1V-PD5rYijaXCy5OOZp8akLYBJ2X5hav6/view?usp=drive_link