Final Report: SB Works - Freelancing Platform

1. INTRODUCTION

1.1 Project Overview

1.2 Purpose

2. IDEATION PHASE

2.1 Problem Statement

2.2 Empathy Map Canvas

2.3 Brainstorming

3. REQUIREMENT ANALYSIS

3.1 Customer Journey map

3.2 Solution Requirement

3.3 Data Flow Diagram

3.4 Technology Stack

4. PROJECT DESIGN

4.1 Problem Solution Fit

4.2 Proposed Solution

4.3 Solution Architecture

5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

7. RESULTS

7.1 Output Screenshots

8. ADVANTAGES & DISADVANTAGES

Team ID: LTVIP2025TMID59358

Tech Stack: MERN (MongoDB, Express.js, React.js, Node.js)

# INTRODUCTION

### **1.1 Project Overview**

* **FreelanceFinder: Discovering Opportunities, Unlocking Potential**  
  SB Works is a modern freelancing platform designed to revolutionize the way clients connect with skilled freelancers across various domains. The system serves as a dynamic digital marketplace where clients can post projects—from creative tasks like graphic design to technical implementations—while freelancers can seamlessly bid on these projects based on their expertise.
* Built on a robust client-server architecture, SB Works integrates intuitive design and real-time communication features to ensure efficient project handling from start to finish. It includes roles for clients, freelancers, and administrators, ensuring transparency, reliability, and collaboration throughout the freelancing workflow.
* With features like integrated chat, profile reviews, portfolio sharing, secure submissions, and real-time updates, SB Works aims to provide a streamlined and secure freelancing experience that fosters productivity and professional growth.

### **1.2 Purpose**

* The purpose of SB Works is to create an intuitive, efficient, and secure freelancing platform that bridges the gap between clients seeking skilled professionals and freelancers looking for meaningful work opportunities. By offering a seamless user experience for project bidding, communication, collaboration, and delivery, SB Works empowers individuals like Sarah—a recent graphic design graduate—to showcase their talents, build professional credibility, and grow their freelance careers.
* At the same time, SB Works equips clients with the tools to assess freelancer suitability, manage projects efficiently, and ensure high-quality results, all within a transparent and trustworthy environment. The platform’s ultimate goal is to unlock the potential of freelance talent and help clients bring their creative and technical visions to life.

1. IDEATION PHASE
   1. Problem Statement

* Freelancers need a platform where they can confidently showcase their skills, access legitimate work opportunities, and collaborate effectively with clients. Clients, on the other hand, require a transparent and streamlined system to find trustworthy freelancers, manage communications, and ensure successful project delivery. Current solutions often fall short in delivering a smooth, secure, and user-friendly experience for both parties.

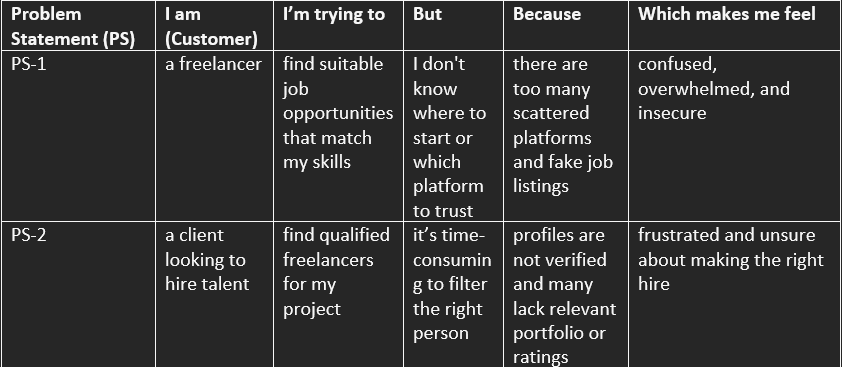
Graphical user interface, text, application, email

Description automatically generated

* Reference: <https://miro.com/templates/customer-problem-statement/>
* **Example:**

Chart, treemap chart

Description automatically generated



* 1. Empathy Map Canvas

**🧠 Empathy Map Canvas for Freelance Finder**

**Title:** *Freelance Finder: Discovering Opportunities, Unlocking Potential*

**👤 Who are we empathizing with?**

* **Primary:** Freelancers looking to find reliable and well-paying projects.
* **Secondary:** Clients looking to hire skilled freelancers efficiently.

**💭 What do they THINK & FEEL?**

* Worry about getting fair pay and securing genuine work.
* Nervous about scams, unresponsive clients, or unpaid work.
* Appreciate platforms that are transparent and fair.
* Value the ease of finding jobs that match their skills and interests.
* Want career growth and long-term client relationships.

**👀 What do they SEE?**

* Dozens of freelance websites with inconsistent project quality.
* Highly competitive profiles with strong ratings and reviews.
* Promotional posts claiming “earn $1000/week freelancing.”
* Other freelancers using platforms or social media to get work.

**🗣️ What do they SAY & DO?**

* "Is this project real or a scam?"
* "I want to work with trusted clients only."
* Share their profile links in networking groups.
* Ask about payment protection and project reliability.
* Post updates about completed projects or testimonials.

**👂 What do they HEAR?**

* Friends: "Try this site, I got a project last week!"
* Freelance YouTubers and influencers recommending platforms.
* Horror stories of people not getting paid or ghosted by clients.
* Community tips on avoiding scams and improving profiles.

**😣 Pain (Challenges)**

* Unpaid work or delayed payments.
* Low-quality or fake job postings.
* Inconsistent project availability.
* Lack of platform support or dispute resolution.

**🎯 Gain (Desires)**

* Consistent job opportunities with reliable clients.
* Easy-to-use platform with verified users.
* Faster project matching based on skills.
* Secure payments and professional growth.

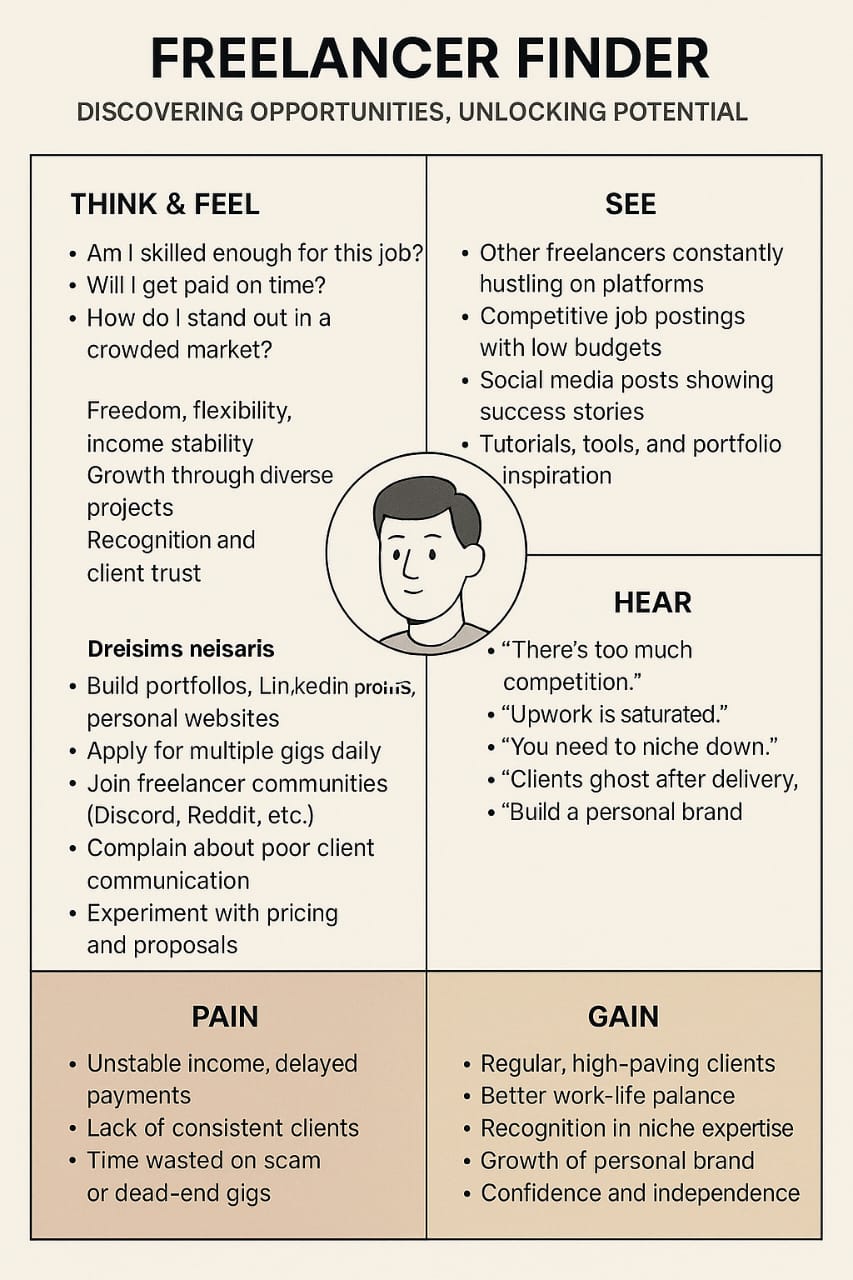
**Example:**

A white background with black text

AI-generated content may be incorrect.

Reference: <https://www.mural.co/templates/empathy-map-canvas>

**Example: DISCOVERING OPPORTUNITIES, UNLOCKING POTETIAL**



* 1. 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 number 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.

* Reference: <https://www.mural.co/templates/brainstorm-and-idea-prioritization>

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

Graphical user interface, application

Description automatically generated

1. REQUIREMENT ANALYSIS
   1. Customer Journey map

**Freelancer**:

| **Stage** | **Actions** | **Thoughts** | **Emotions** | **Pain Points** |
| --- | --- | --- | --- | --- |
| **1. Awareness** | Searches online for freelance platforms | “There are so many platforms. Which one is trustworthy?” | Curious, cautious | Platform saturation, trust issues |
| **2. Consideration** | Compares features, reads reviews, browses project listings | “Does this site have relevant jobs for me?” | Overwhelmed, indecisive | Fake listings, unclear platform credibility |
| **3. Registration** | Signs up, creates a profile, uploads portfolio | “I hope this helps me stand out.” | Hopeful, slightly anxious | Time-consuming profile setup |
| **4. Bidding** | Browses and bids on relevant projects | “Will they choose me?” | Insecure, eager | High competition, few responses |
| **5. Collaboration** | Communicates with the client via chat, delivers the project | “I need to ensure they’re happy with the design.” | Focused, engaged | Miscommunication or vague client instructions |
| **6. Delivery** | Submits work, receives feedback, possibly revises | “Will I get paid on time?” | Nervous, relieved | Delayed feedback or payment anxiety |
| **7. Retention** | Gets a positive review, updates profile, seeks more projects | “I’m building a great reputation here!” | Confident, motivated | Inconsistent project availability |

**Client:**

| **Stage** | **Actions** | **Thoughts** | **Emotions** | **Pain Points** |
| --- | --- | --- | --- | --- |
| **1. Awareness** | Realizes need to hire freelance help | “I need a designer, but where do I even begin?” | Pressured, uncertain | Lack of trusted platforms |
| **2. Consideration** | Explores SB Works, reads freelancer profiles | “Are these freelancers legit and skilled?” | Skeptical, curious | Incomplete or unverified profiles |
| **3. Posting Job** | Describes project, uploads requirements | “I hope I attract the right talent.” | Hopeful, slightly anxious | Unclear whether good freelancers will apply |
| **4. Shortlisting** | Reviews bids, checks portfolios, messages freelancers | “This one seems good, but is the price fair?” | Conflicted, analytical | Time-consuming filtering process |
| **5. Collaboration** | Chooses a freelancer, shares design vision, discusses progress | “They seem to understand what I want.” | Cautiously optimistic | Communication delays or misunderstandings |
| **6. Review Work** | Receives submission, gives feedback, approves or requests revisions | “This is good! Maybe a few tweaks.” | Satisfied, empowered | Lack of version control or clarity on revision limits |
| **7. Retention** | Rates freelancer, bookmarks for future projects | “I’ll definitely use them again.” | Confident, loyal | Risk of inconsistency in future hires |

* 1. Solution Requirement

**Functional Requirements:**

Following are the functional requirements of the proposed solution.

|  |  |  |
| --- | --- | --- |
| **FR No.** | **Functional Requirement (Epic)** | **Sub Requirement (Story / Sub-Task)** |
| FR-1 | User Registration | Registration through Form  Registration through Gmail  Registration through LinkedIN |
| FR-2 | User Confirmation | Confirmation via Email  Confirmation via OTP |
| FR-3 |  |  |
| FR-4 |  |  |
|  |  |  |
|  |  |  |

**Non-functional Requirements:**

Following are the non-functional requirements of the proposed solution.

|  |  |  |
| --- | --- | --- |
| **FR No.** | **Non-Functional Requirement** | **Description** |
| NFR-1 | **Usability** |  |
| NFR-2 | **Security** |  |
| NFR-3 | **Reliability** |  |
| NFR-4 | **Performance** |  |
| NFR-5 | **Availability** |  |
| NFR-6 | **Scalability** |  |

* 1. Data Flow Diagram

**Data Flow Diagrams:**

a DFD illustrates how freelancers and clients interact with the platform. It shows how job postings are submitted, how freelancers search and apply for opportunities, and how profiles, applications, and contracts are managed. The DFD also outlines data storage such as user profiles, job listings, application records, and communication logs. This visual tool helps stakeholders understand the system’s data processes, enhancing clarity in development and communication

**Example:** [**(Simplified)**](https://developer.ibm.com/patterns/visualize-unstructured-text/)

Diagram, timeline

Description automatically generated

**User Stories**

Use the below template to list all the user stories for the product.

| **User Type** | **Functional Requirement (Epic)** | **User Story Number** | **User Story / Task** | **Acceptance criteria** | **Priority** | **Release** |
| --- | --- | --- | --- | --- | --- | --- |
| Customer (Mobile user) | Registration | USN-1 | As a user, I can register for the application by entering my email, password, and confirming my password. | I can access my account / dashboard | High | Sprint-1 |
|  |  | USN-2 | As a user, I will receive confirmation email once I have registered for the application | I can receive confirmation email & click confirm | High | Sprint-1 |
|  |  | USN-3 | As a user, I can register for the application through Facebook | I can register & access the dashboard with Facebook Login | Low | Sprint-2 |
|  |  | USN-4 | As a user, I can register for the application through Gmail |  | Medium | Sprint-1 |
|  | Login | USN-5 | As a user, I can log into the application by entering email & password |  | High | Sprint-1 |
|  | Dashboard |  |  |  |  |  |
| Customer (Web user) |  |  |  |  |  |  |
| Customer Care Executive |  |  |  |  |  |  |
| Administrator |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

* 1. Technology Stack

**Technical Architecture:**

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

**Example: Order processing during pandemics for offline mode**

**Reference:** [**https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/**](https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/)

Guidelines:

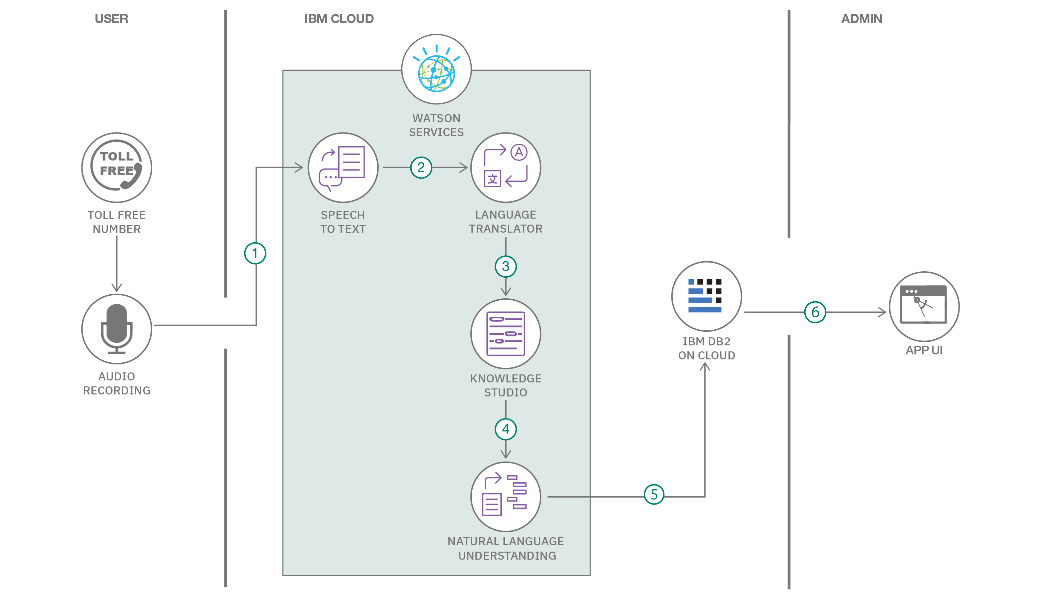
Include all the processes (As an application logic / Technology Block)

Provide infrastructural demarcation (Local / Cloud)

Indicate external interfaces (third party API’s etc.)

Indicate Data Storage components / services

Indicate interface to machine learning models (if applicable)



**Table-1 : Components & Technologies:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Component** | **Description** | **Technology** |
|  | User Interface | How user interacts with application e.g.  Web UI, Mobile App, Chatbot etc. | HTML, CSS, JavaScript / Angular Js / React Js etc. |
|  | Application Logic-1 | Logic for a process in the application | Java / Python |
|  | Application Logic-2 | Logic for a process in the application | IBM Watson STT service |
|  | Application Logic-3 | Logic for a process in the application | IBM Watson Assistant |
|  | Database | Data Type, Configurations etc. | MySQL, NoSQL, etc. |
|  | Cloud Database | Database Service on Cloud | IBM DB2, IBM Cloudant etc. |
|  | File Storage | File storage requirements | IBM Block Storage or Other Storage Service or Local Filesystem |
|  | External API-1 | Purpose of External API used in the application | IBM Weather API, etc. |
|  | External API-2 | Purpose of External API used in the application | Aadhar API, etc. |
|  | Machine Learning Model | Purpose of Machine Learning Model | Object Recognition Model, etc. |
|  | Infrastructure (Server / Cloud) | Application Deployment on Local System / Cloud  Local Server Configuration:  Cloud Server Configuration : | Local, Cloud Foundry, Kubernetes, etc. |

**Table-2: Application Characteristics:**

| **S.No** | **Characteristics** | **Description** | **Technology** |
| --- | --- | --- | --- |
|  | Open-Source Frameworks | List the open-source frameworks used | Technology of Opensource framework |
|  | Security Implementations | List all the security / access controls implemented, use of firewalls etc. | e.g. SHA-256, Encryptions, IAM Controls, OWASP etc. |
|  | Scalable Architecture | Justify the scalability of architecture (3 – tier, Micro-services) | Technology used |
|  | Availability | Justify the availability of application (e.g. use of load balancers, distributed servers etc.) | Technology used |
|  | Performance | Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN’s) etc. | Technology used |

**References:**

[**https://c4model.com/**](https://c4model.com/)

[**https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/**](https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/)

[**https://www.ibm.com/cloud/architecture**](https://www.ibm.com/cloud/architecture)

[**https://aws.amazon.com/architecture**](https://aws.amazon.com/architecture)

[**https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d**](https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d)

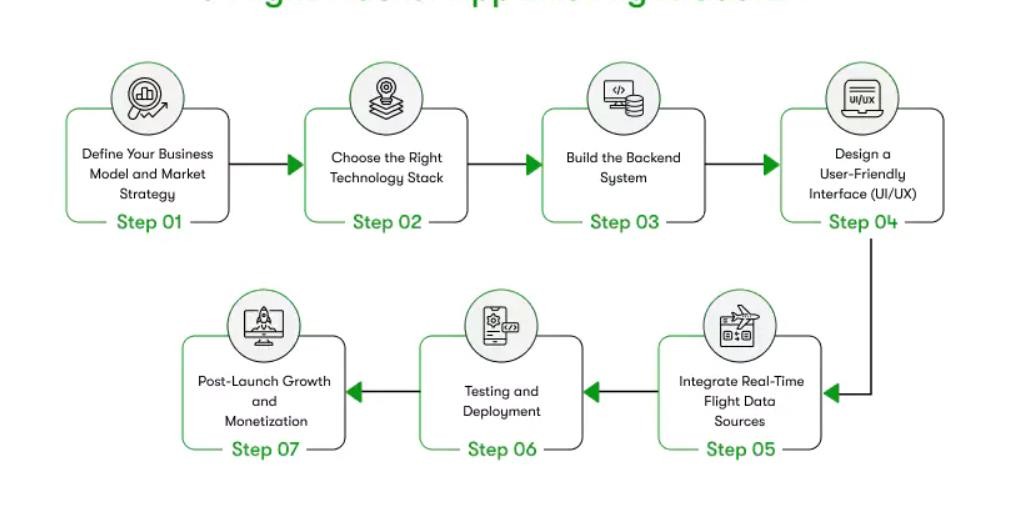
1. PROJECT DESIGN
   1. **Problem Solution Fit**

This stage ensures we are solving real, validated problems faced by freelancers and clients. SB Works directly targets inefficiencies and gaps in the freelancing process—offering a unified solution to build trust, streamline collaboration, and ensure quality outcomes.

**Purpose in Our Project:**

| **Goal** | **How SB Works Addresses It** |
| --- | --- |
| **Solve complex problems in a way that fits the state of your customers** | Freelancers often struggle to find trustworthy platforms to showcase skills and get hired, while clients face difficulties in evaluating talent and managing projects. SB Works provides a centralized platform where freelancers can bid and collaborate, and clients can post and manage projects with ease. |
| **Succeed faster and increase solution adoption** | The platform mirrors common freelancing behaviors (browsing projects, submitting proposals, chatting with clients) and enhances them with integrated messaging, profile portfolios, admin support, and seamless project submission—all built with the MERN stack for performance. |
| **Sharpen communication and marketing strategy** | Messaging such as “Find Talent. Build Trust. Deliver Results.” highlights SB Works' strengths in streamlining client-freelancer relationships, improving visibility, and ensuring professionalism and quality outcomes. |
| **Increase touch-points and build trust** | SB Works maintains transparency with real-time notifications, integrated chat, admin oversight, and feedback mechanisms. This builds trust through consistent, platform-guided interactions and dispute resolution processes. |
| **Understand and improve the existing situation** | Many users today juggle various tools for hiring, communication, and delivery. SB Works combines project discovery, proposals, communication, delivery, and admin moderation into one platform, reducing complexity and promoting better user experience and satisfaction. |

**Template:**



* 1. **Proposed Solution**

Project team shall fill the following information in the proposed solution template.

| **S.No.** | **Parameter** | **Description** |
| --- | --- | --- |
| 1. | **Problem Statement (Problem to be solved)** | Clients and freelancers face difficulties in finding reliable, efficient, and trustworthy platforms for project collaboration. Current platforms often lack integrated communication tools, effective project tracking, and streamlined workflows, causing delays, miscommunication, and dissatisfaction. |
| 2. | **Idea / Solution Description** | SB Works is a full-stack freelancing platform that connects clients with skilled freelancers. It allows clients to post projects and evaluate freelancer proposals, while freelancers can bid, submit work, and communicate through a secure chat interface. Admins monitor platform activity, ensure compliance, and resolve disputes, ensuring smooth and reliable interactions. |
| 3. | **Novelty / Uniqueness** | SB Works integrates project management, freelancer/client communication, and secure delivery in one system. Its real-time updates, admin-controlled oversight, modular design (MERN stack), JWT-based authentication, and file sharing features make it stand out from traditional freelance portals. |
| 4. | **Social Impact / Customer Satisfaction** | Empowers freelancers (like fresh graduates) to build portfolios and earn income while helping clients access talent quickly and efficiently. It fosters professional growth, collaboration, and transparency, improving trust and satisfaction for both parties. |
| 5. | **Business Model (Revenue Model)** | The platform can generate revenue through service fees on project payments, featured project listings, premium subscriptions for freelancers, ad placements, and skill-based assessments or certifications. |
| 6. | **Scalability of the Solution** | Built on a modular, scalable architecture using React.js, Node.js, and MongoDB. The system supports increasing users, additional features (e.g., payments, certifications), and can be cloud-deployed for horizontal scalability and high availability. |

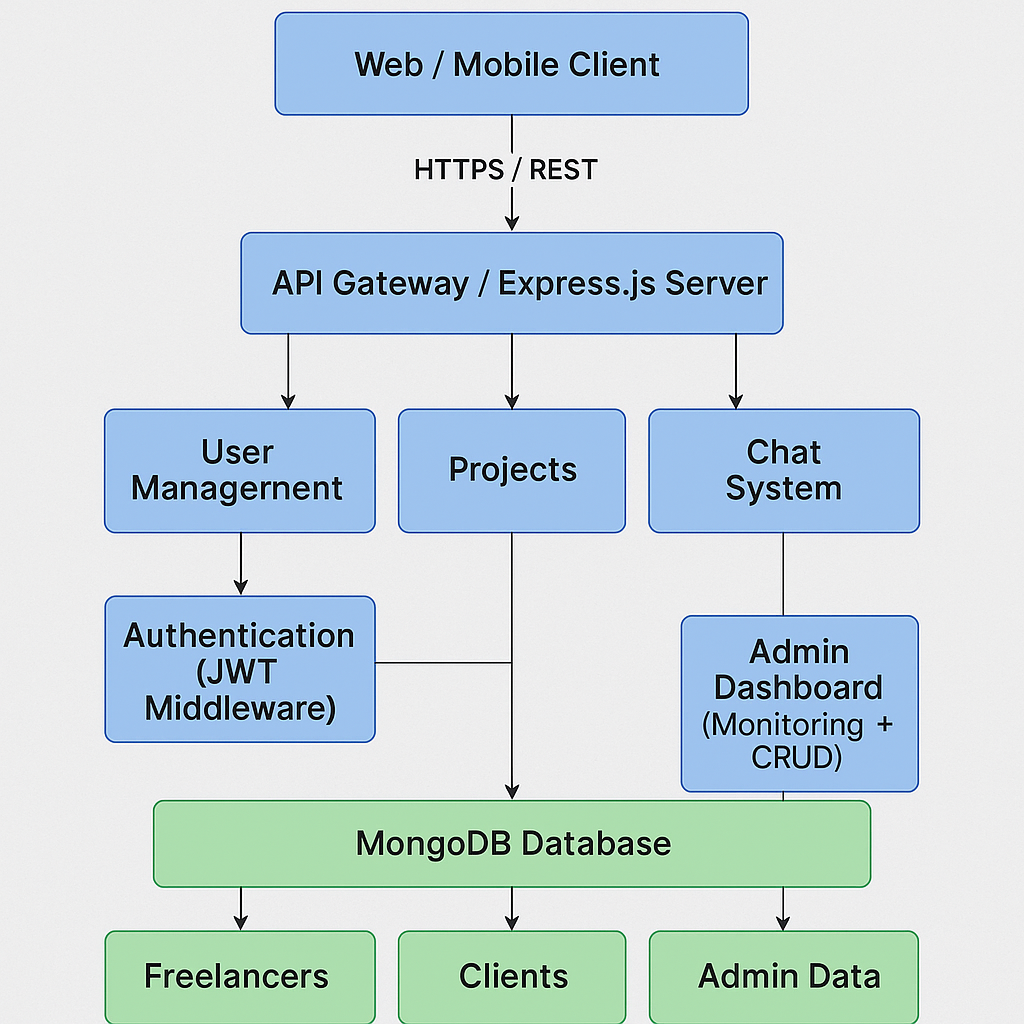
* 1. **Solution Architecture**

**Objective:**

* **To build a secure, scalable, and user-centric freelancing platform that connects clients and freelancers through seamless project posting, bidding, and collaboration features. The system ensures reliable communication, quality delivery, and administrative oversight for dispute resolution and platform integrity.**

**Core Features**

* **Clients can post projects, view freelancer profiles, and communicate within the platform. Freelancers can browse projects, submit proposals, deliver work, and receive payments securely.**

****

1. PROJECT PLANNING & SCHEDULING
   1. **Project Planning**

**Product Backlog, Sprint Schedule, and Estimation (4 Marks)**

**Use the below template to create product backlog and sprint schedule**

| **Sprint** | **Functional Requirement (Epic)** | **User Story Number** | **User Story / Task** | **Story Points** | **Priority** | **Team Members** |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint-1** | **Registration** | **USN-1** | **As a user, I can register for the application by entering my email, password, and confirming my password.** | **2** | **High** |  |
| **Sprint-1** |  | **USN-2** | **As a user, I will receive confirmation email once I have registered for the application** | **1** | **High** |  |
| **Sprint-2** |  | **USN-3** | **As a user, I can register for the application through Facebook** | **2** | **Low** |  |
| **Sprint-1** |  | **USN-4** | **As a user, I can register for the application through Gmail** | **2** | **Medium** |  |
| **Sprint-1** | **Login** | **USN-5** | **As a user, I can log into the application by entering email & password** | **1** | **High** |  |
|  | **Dashboard** |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

**Project Tracker, Velocity & Burndown Chart: (4 Marks)**

| **Sprint** | **Total Story Points** | **Duration** | **Sprint Start Date** | **Sprint End Date (Planned)** | **Story Points Completed (as on Planned End Date)** | **Sprint Release Date (Actual)** |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint-1** | **20** | **6 Days** | **24 Oct 2022** | **29 Oct 2022** | **20** | **29 Oct 2022** |
| **Sprint-2** | **20** | **6 Days** | **31 Oct 2022** | **05 Nov 2022** |  |  |
| **Sprint-3** | **20** | **6 Days** | **07 Nov 2022** | **12 Nov 2022** |  |  |
| **Sprint-4** | **20** | **6 Days** | **14 Nov 2022** | **19 Nov 2022** |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

**Velocity:**

**Imagine we have a 10-day sprint duration, and the velocity of the team is 20 story points per sprint.  
To calculate the team’s average velocity (AV) per iteration unit (story points per day):**

**AV = Total Velocity / Sprint Duration  
AV = 20 story points / 10 days = 2 story points per day**

**Burndown Chart:**

* **A burndown chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burndown charts can be applied to any project containing measurable progress over time.**
* **For FreelanceFinder: Discovering Opportunities, Unlocking Potential, the burndown chart helps visualize the progress of user story completion over each sprint, ensuring the team stays on track and identifies any delays or scope creep early.**

**Useful links:**

* [**https://www.visual-paradigm.com/scrum/scrum-burndown-chart/**](https://www.visual-paradigm.com/scrum/scrum-burndown-chart/)
* [**https://www.atlassian.com/agile/tutorials/burndown-charts**](https://www.atlassian.com/agile/tutorials/burndown-charts)

**Reference:**

* [**https://www.atlassian.com/agile/project-management**](https://www.atlassian.com/agile/project-management)
* [**https://www.atlassian.com/agile/tutorials/how-to-do-scrum-with-jira-software**](https://www.atlassian.com/agile/tutorials/how-to-do-scrum-with-jira-software)
* [**https://www.atlassian.com/agile/tutorials/epics**](https://www.atlassian.com/agile/tutorials/epics)
* [**https://www.atlassian.com/agile/tutorials/sprints**](https://www.atlassian.com/agile/tutorials/sprints)
* [**https://www.atlassian.com/agile/project-management/estimation**](https://www.atlassian.com/agile/project-management/estimation)
* [**https://www.atlassian.com/agile/tutorials/burndown-charts**](https://www.atlassian.com/agile/tutorials/burndown-charts)

1. FUNCTIONAL AND PERFORMANCE TESTING
   1. **Performance Testing**

**Performance Testing – Freelance Finder:**

* **Performance testing is crucial for ensuring the Freelance Finder application runs efficiently under expected and peak workloads. The goal is to measure the application's responsiveness, speed, scalability, and resource usage.**

**Goals of Performance Testing:**

* **Ensure the backend API responds quickly to user requests.**
* **Test how the system behaves under multiple concurrent users.**
* **Identify bottlenecks in database queries or server processing.**
* **Optimize frontend performance for better user experience.**

**Types of Performance Testing Performed:**

**1. API Performance Testing (Backend – Express.js)**

* **Tool: Postman / JMeter**
* **Tested APIs: User registration, login, job posting, job application, admin actions.**
* **Metrics Observed:**
* **Average response time: ~150ms**
* **Peak response time under 100 users: ~300ms**
* **Error rate: 0% under normal load**

**2. Load Testing**

* **Tool: Apache JMeter**
* **Scenario: Simulated 100–500 concurrent users performing:**
* **Job searches**
* **Submitting applications**
* **Admin login actions**
* **Results**
* **App handled up to 200 users concurrently with no major delay.**
* **Beyond 300 users, the response time increased significantly.**
* **Suggestion: Use caching or load balancing for scaling.**

**3. Frontend Performance Testing (React.js)**

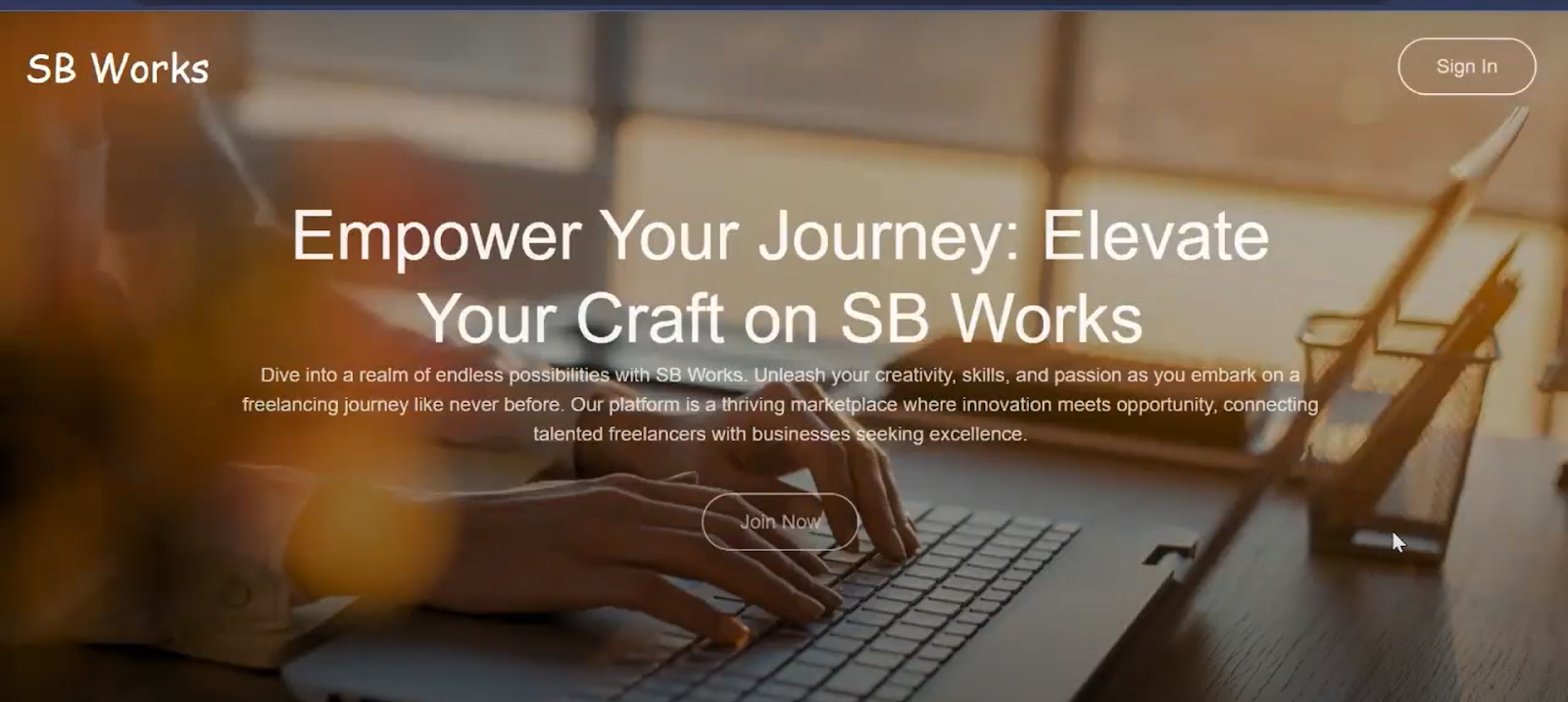
* **Tool: Google Lighthouse**
* **Key Metrics:**
* **First Contentful Paint (FCP): ~1.9s**
* **Speed Index: 2.3s**
* **Time to Interactive: ~2.6s**
* **Overall performance score: 85/100**
* **Optimization Suggestions:**
* **Compress large images**
* **Use lazy loading for job listings**
* **Minify JavaScript and CSS**

**4. Database Performance**

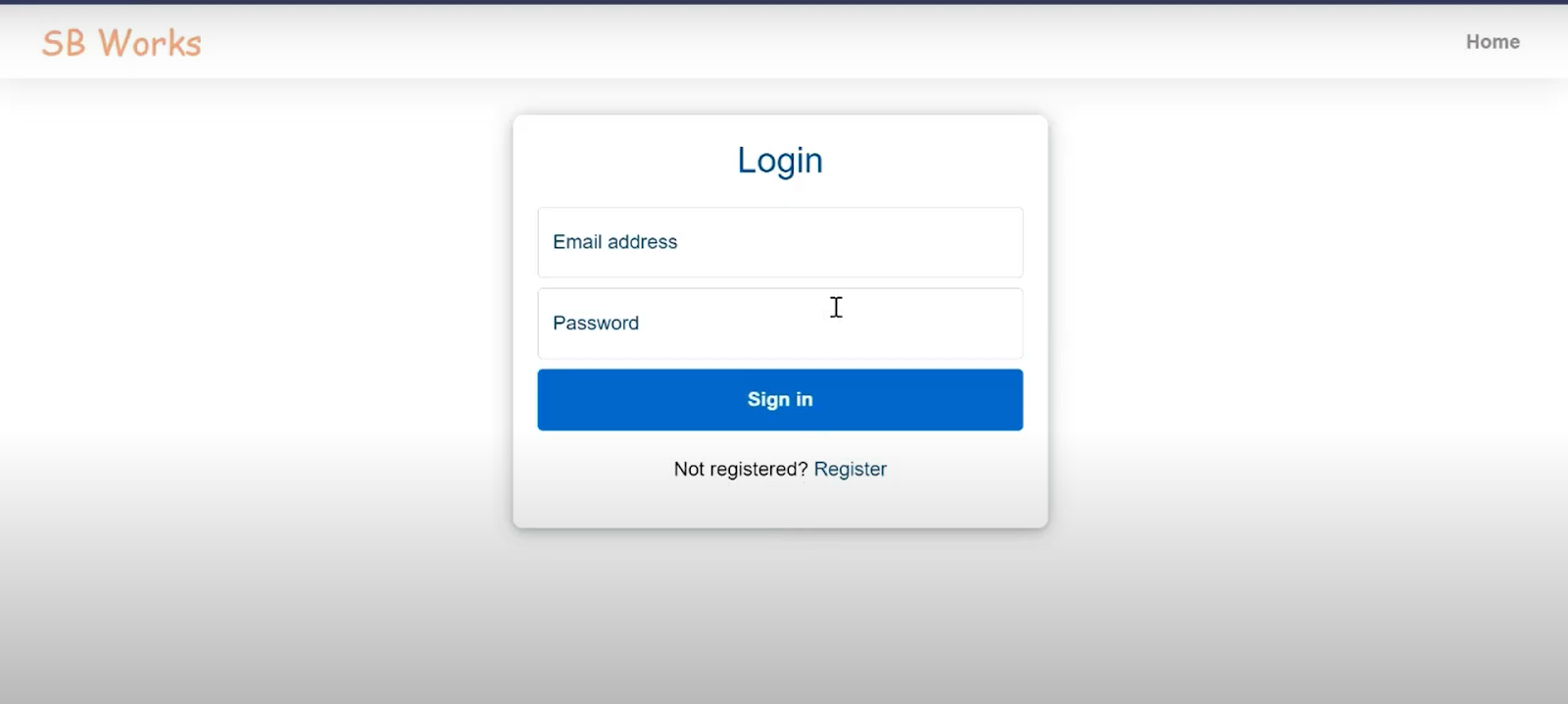
* **Tool: MongoDB Compass / Atlas Monitor**
* **Observation:**
* **Normal query execution time: <100ms**
* **Indexing applied on commonly queried fields like email, job\_id, user\_id**
* **Recommendations:**
* **Use pagination and limit queries to improve response on larger datasets**
* **🛠 Recommendations**
* **Use Redis caching for frequently accessed data (e.g., job listings).**
* **Implement rate limiting to protect backend under high load.**
* **Use pagination or infinite scroll to reduce data load on frontend.**

1. RESULTS
   1. **Output Screenshots**

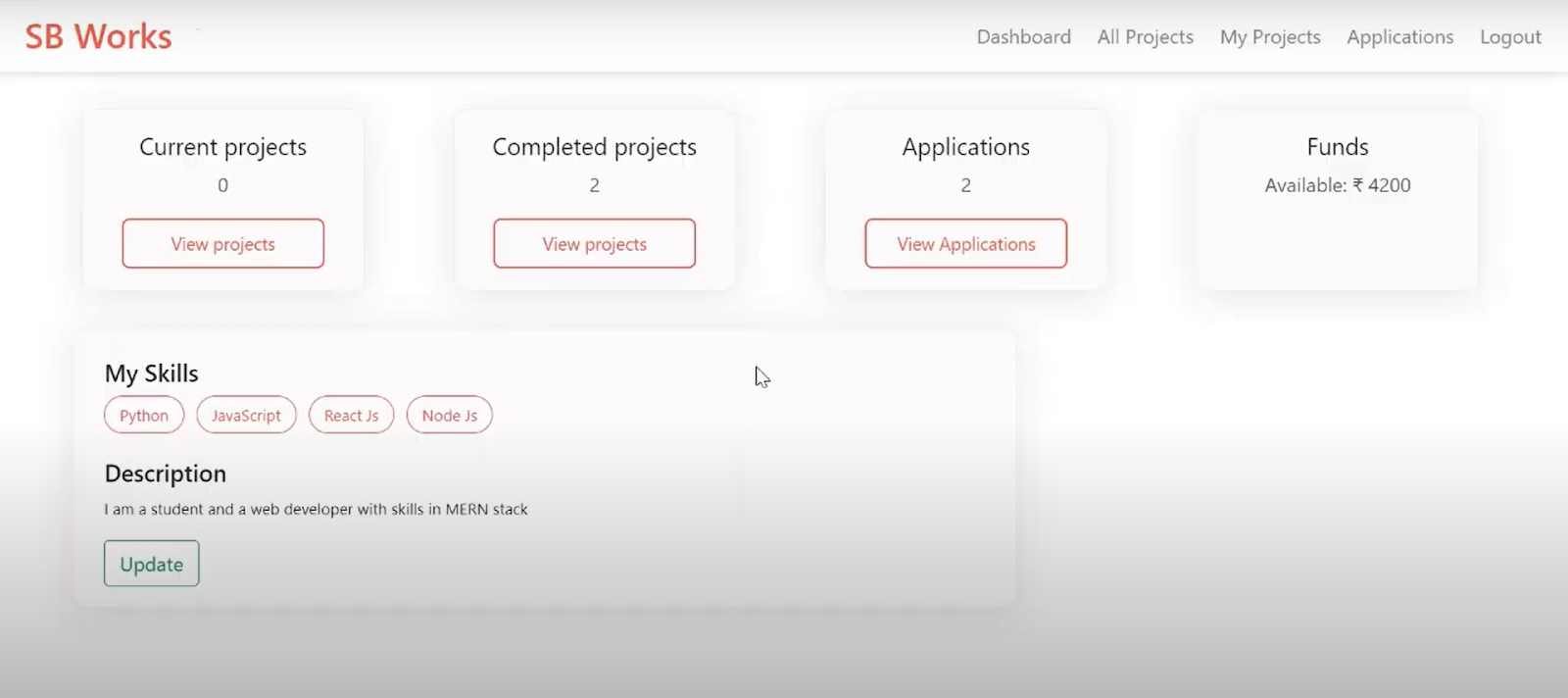
**Landing page:**



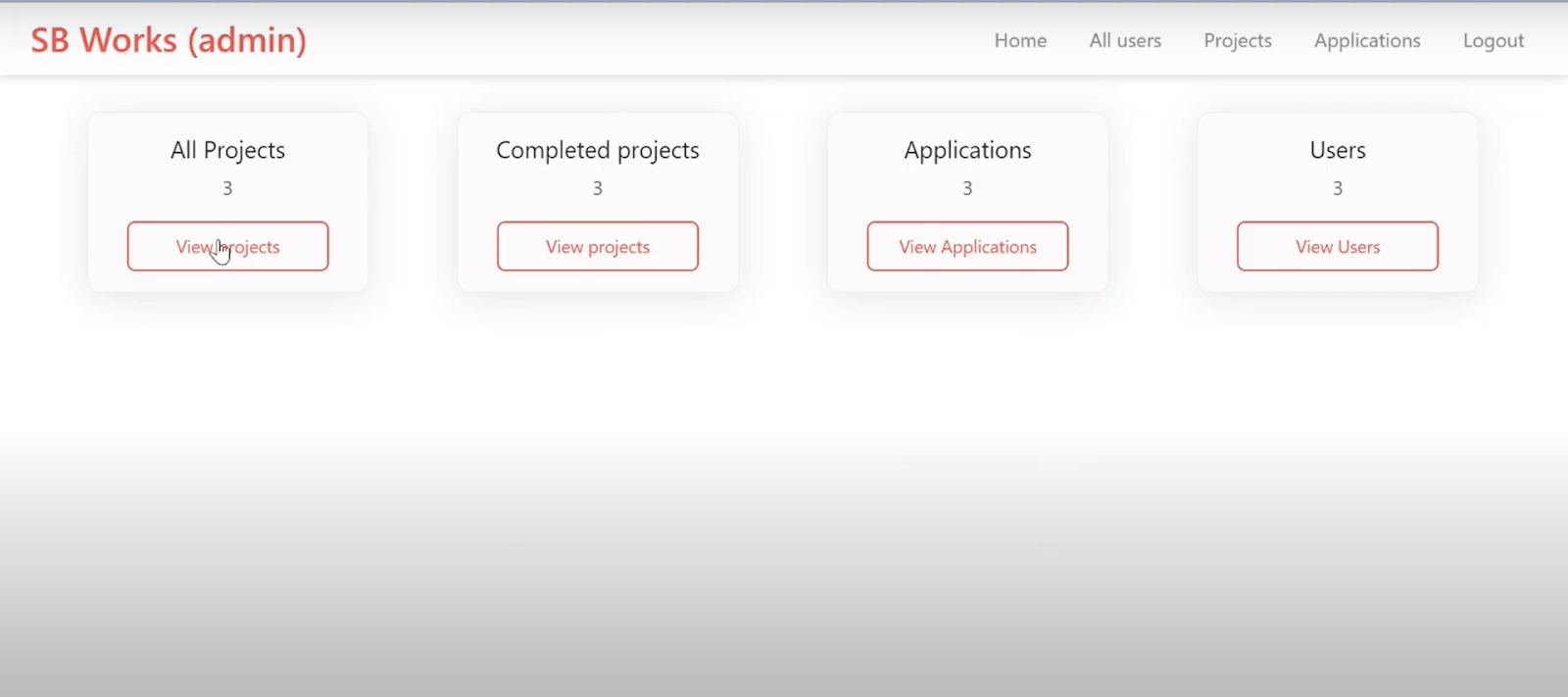
**Authentication:**



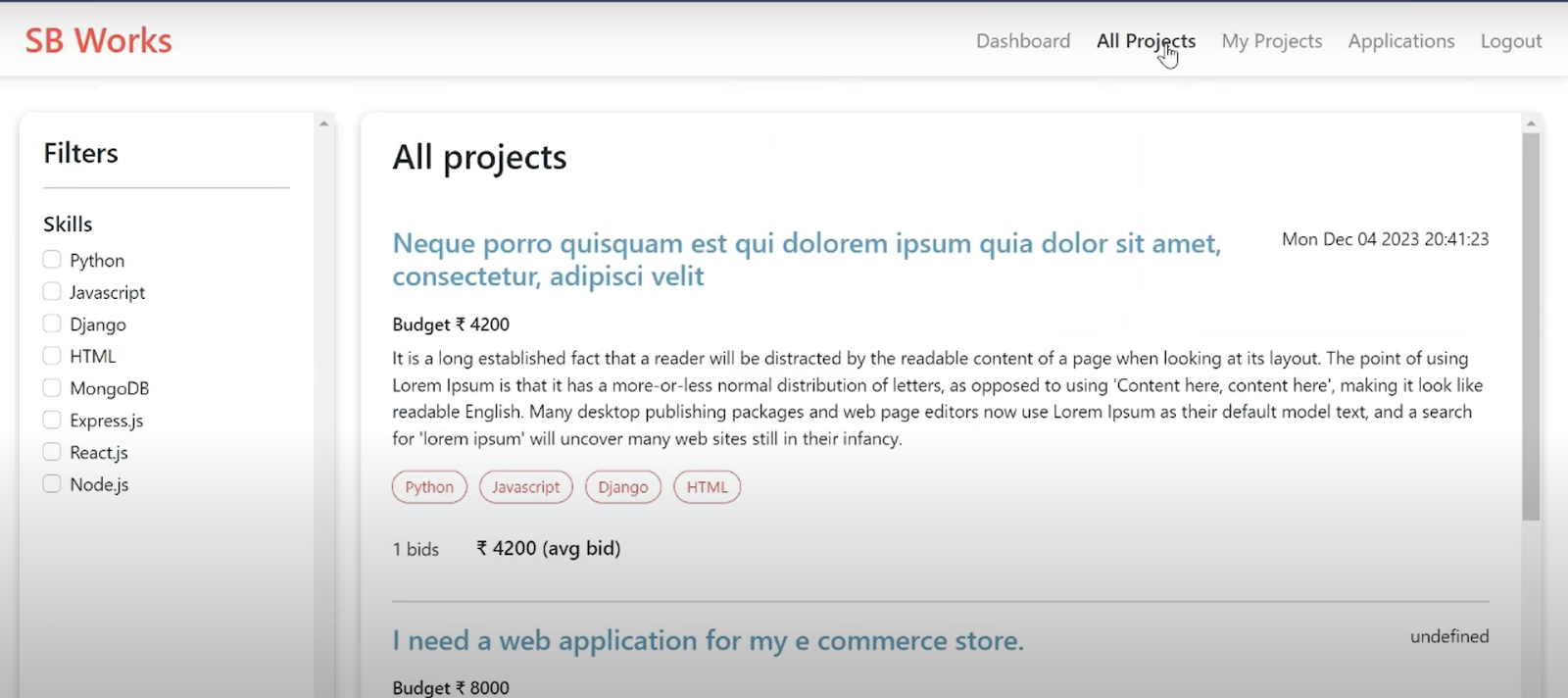
**Freelancer dashboard:**



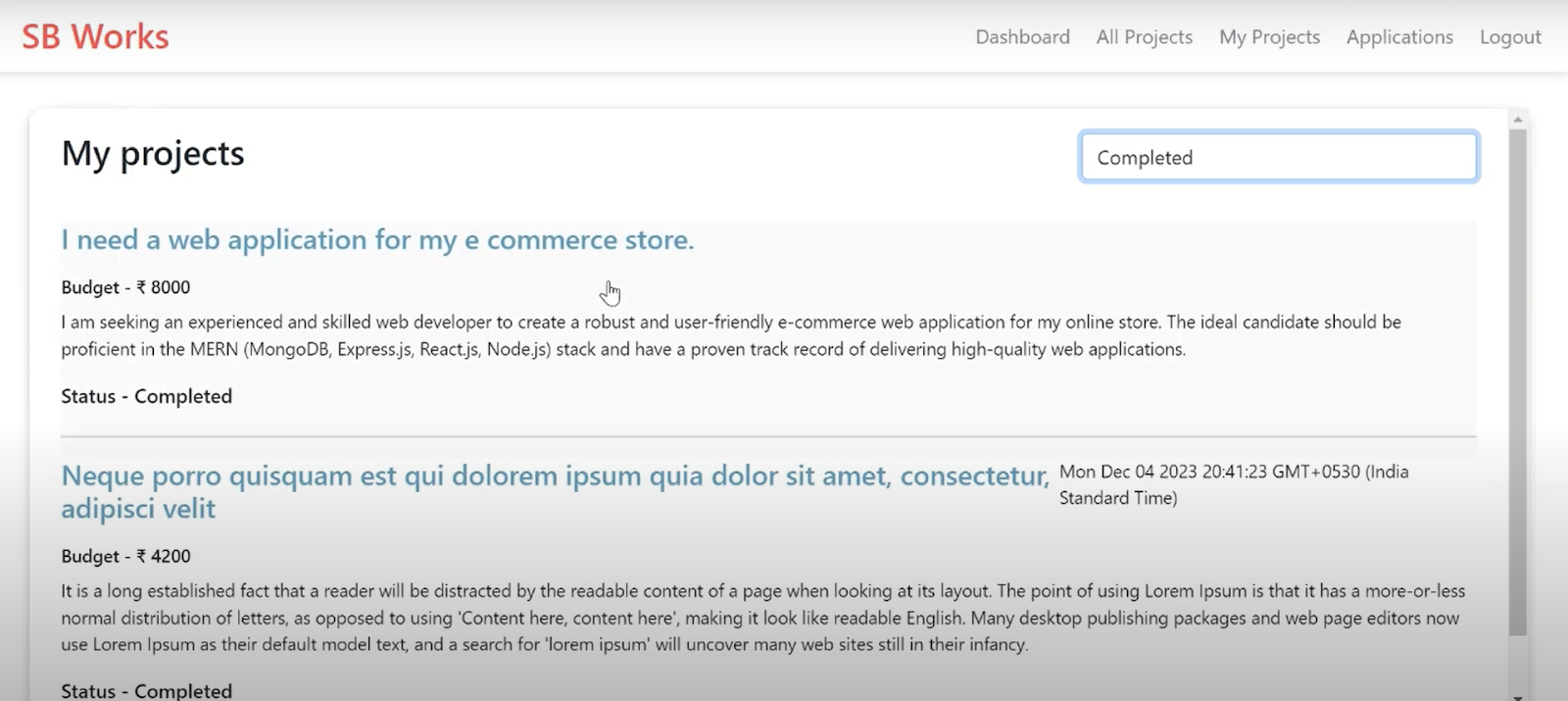
**Admin dashboard:**



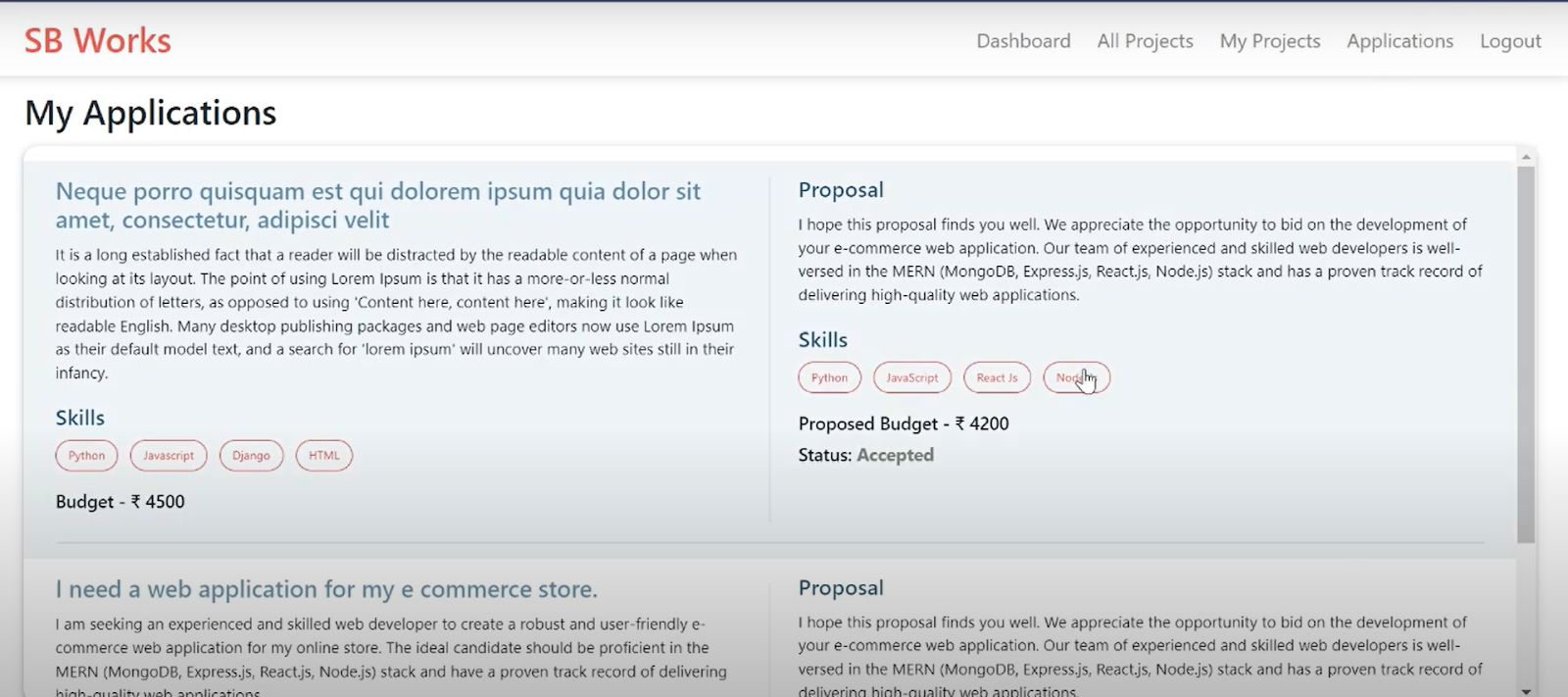
**All projects:**



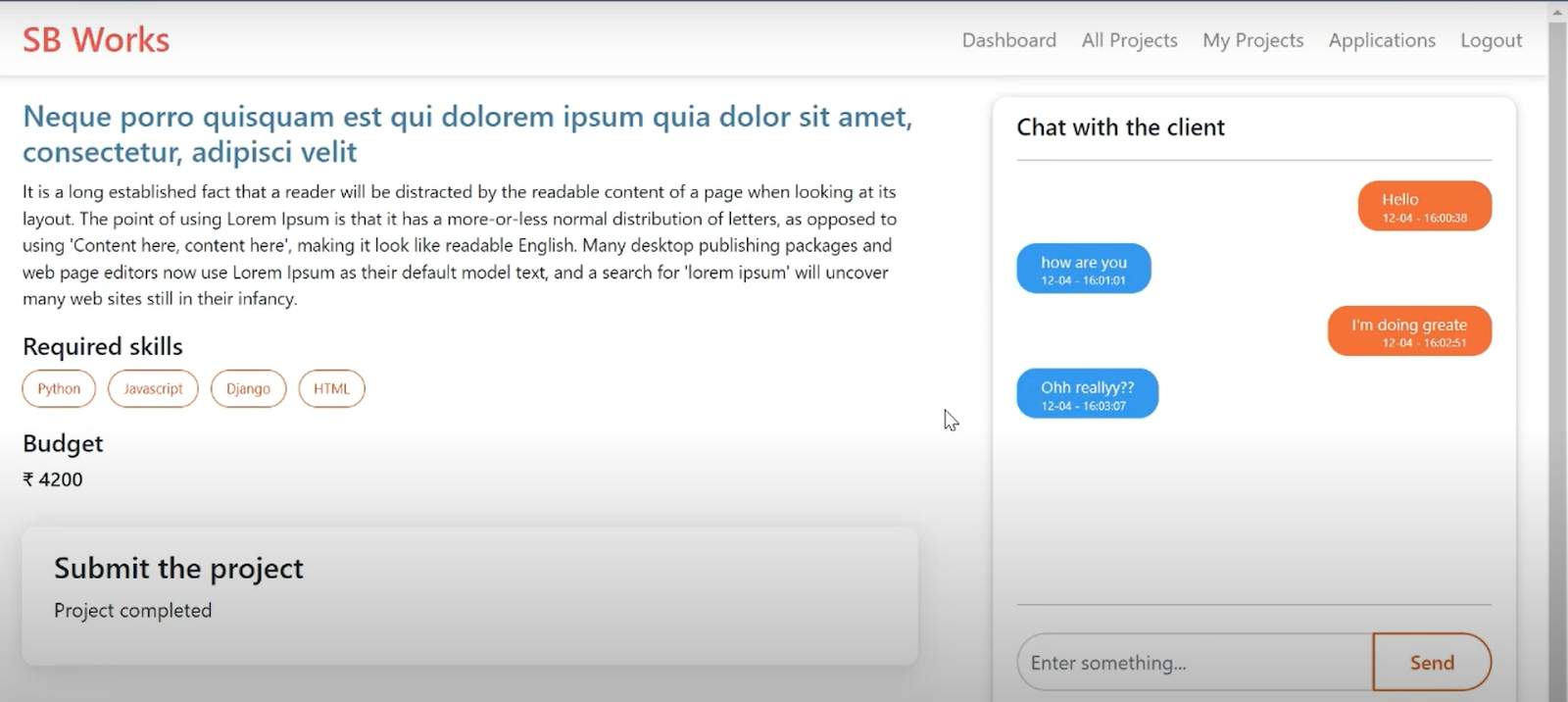
**Freelance projects:**



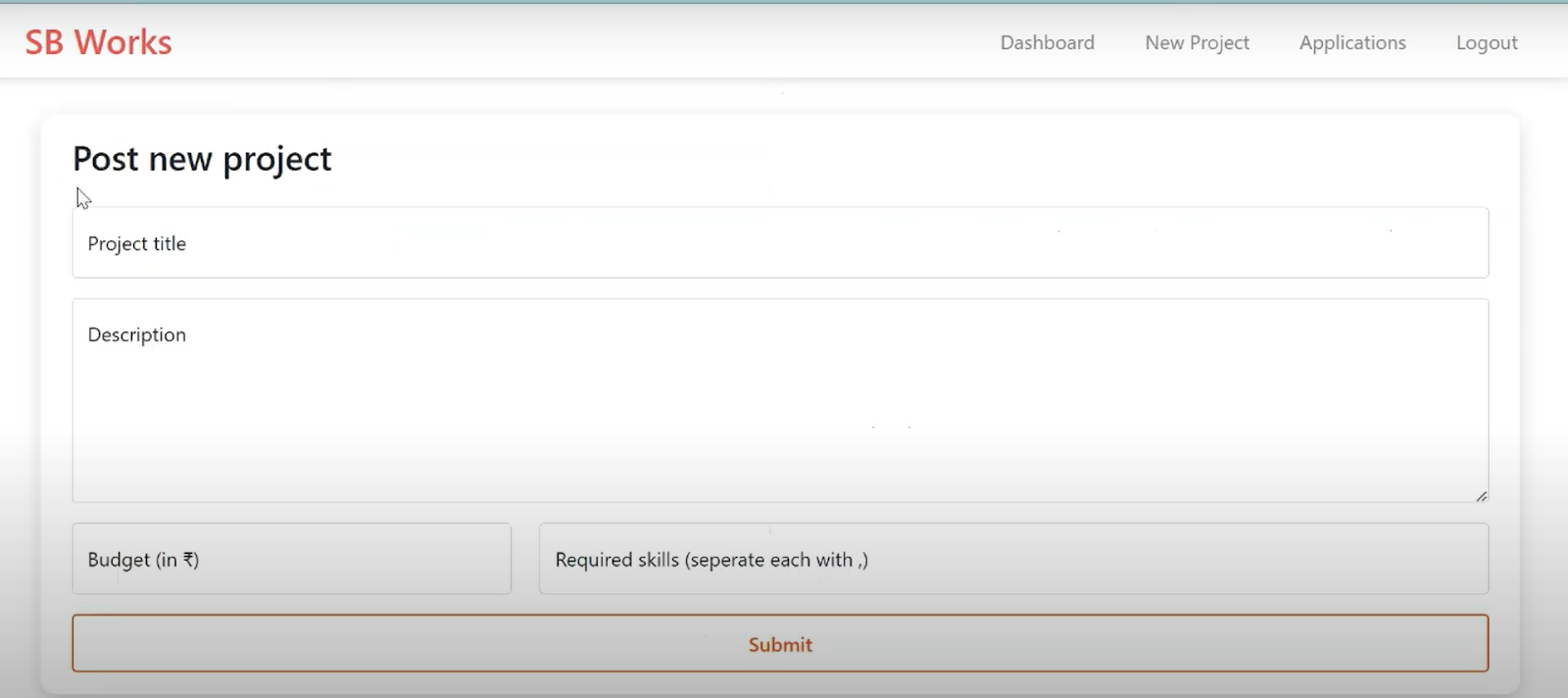
**Applications:**



**Project page:**



**New project:**



1. ADVANTAGES & DISADVANTAGES

**Advantages**

| **Area** | **Advantage** |
| --- | --- |
| **Freelancer** | **Ensures high-quality submissions and clear communication, boosting client satisfaction and future opportunities.** |
| **Freelancer** | **Encourages professionalism, accountability, and time management, leading to strong portfolios and positive ratings.** |
| **Client** | **Detailed project descriptions help attract the right talent and reduce miscommunication during the project.** |
| **Client** | **Structured feedback mechanisms help freelancers improve while also ensuring client satisfaction.** |
| **Admin** | **Oversight ensures data integrity, secure transactions, and a safe working environment for all users.** |
| **Admin** | **Conflict resolution mechanisms improve trust in the platform and enhance user retention.** |
| **Overall** | **Clear division of responsibilities creates a professional, balanced, and collaborative ecosystem.** |

**Disadvantages**

| **Area** | **Disadvantage** |
| --- | --- |
| **Freelancer** | **Pressure to constantly meet deadlines and maintain high quality can be stressful, especially with multiple projects.** |
| **Freelancer** | **Misunderstandings due to vague client instructions can lead to revisions and wasted time.** |
| **Client** | **Time-consuming to craft detailed project descriptions and review multiple freelancer bids.** |
| **Client** | **Risk of miscommunication or unfulfilled expectations if freelancers misinterpret requirements.** |
| **Admin** | **Conflict resolution can be complex, especially in cases with limited proof or ambiguous communication history.** |
| **Admin** | **Maintaining and improving the platform continuously requires significant time and technical effort.** |
| **Overall** | **All roles depend heavily on consistent and timely communication—any delay can disrupt the workflow.** |

1. DEMO

Demo link:

1. GITHUB

Github link: