Software Requirements Specification

for

Co-Lancer

Version 1.0 approved

Prepared by

PES2UG21CS374- POOJA SATHEESH

PES2UG21CS396 – PREETHI M.

PES UNIVERSITY

06/10/2023

Table of Contents

1. Introduction

1.1 Purpose

This Software Requirements Specification document details the requirements and objectives of the Co-Lancer platform, which facilitates collaboration of freelancers on various projects.

The purpose of the document is to provide a detailed and unambiguous description of the product. The objectives include

- Detail the functionalities of the product
- Elaborate working of the software
- Describe the coupling between various components and modules
- Interaction between the users and software

The document is structured to understand the goals and working of the product better. This document is designed for developers who build this product, the customers who use this product and professors.

1.2 Intended Audience

This document is essentially aimed towards the following stakeholders of the product

- Current developers
- Testers and maintainers
- Future developers working on the product
- Professors reviewing the document
- Freelancers aspiring to collaborate on new projects
- Clients from various organizations who want to publish their project requirements.

1.3 Product Scope

This project is a website based on freelancer collaboration. It facilitates the interaction between freelancers and clients and helps them achieve their goals seamlessly through our product. This is achieved by facilitating

- Clients representing an organization or individual business to register on the website and post projects.
- Freelancers to register and give their preferences and skill set and view project recommendations based on their profile.
- Multiple freelancers to collaborate and work on the same project.
- An integrated payment gateway that ensures smooth transfer of compensation to freelancers once their designated project is over.
- Clients to review and rate the freelancer's works.

The primary objective of this website is to offer freelancers an accessible portal to involve themselves in various projects allowing them to thrive and earn within their respective domain. It aims to simplify the process for clients to easily find and hire freelancers.

1.4 References

1. https://react.dev/learn
2. https://nodejs.org/en
3. https://www.mysql.com/
4. https://expressjs.com/
5. https://tailwindcss.com/

2. Overall Description

2.1 Product Perspective

Co-Lancer is a new, self-contained product. Co-Lancer will act as an interface between budding freelancers looking for new projects and clients who wish to have various projects implemented.

The lack of an application catering to the needs of only freelancers lead to the development of this project. The requirement of such an application is what drives this project. The resulting website aims to please all its stakeholders, principally the freelancers and the clients.

Co-Lancer can have two types of users, clients and freelancers. Clients can register with their organization details and publish projects. Freelancers can register with their personal details along with their skill set, they can view the projects categorized by the skills required for its completion and pick up any project. Projects needing multiple freelancers are tagged with *#CollabNow* and these allow collaboration between multiple freelancers. Freelancers get paid by the clients and this happens across a payment gateway which ensures a smooth transaction. Clients can use the review feature of Co-Lancer to provide feedback, reviews and rating to the freelancer. These ratings will affect the freelancer's credibility score and give them *'cookies'* accordingly.

The product has 2 main views:

1. Freelancer view
2. Client view

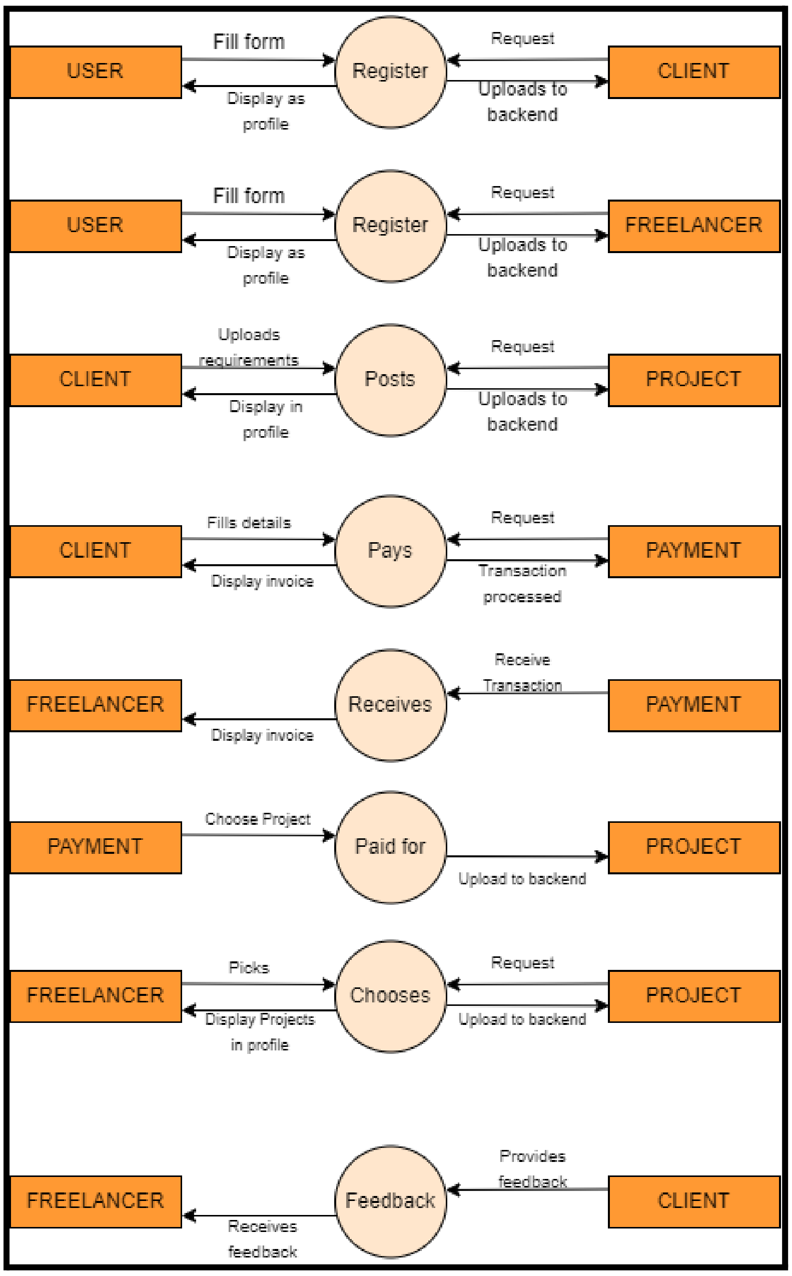
This product can work independently on any OS and any device with a modern browser support. It does not require any installation and does no modification to the host device. It does not require any additional permissions to be granted for its working . Co-Lancer is an independent standalone product and does not interact with any external components.

2.2 Product Functions

Co-Lancer will implement the following functionalities:

1. Create user profiles for freelancers and clients
2. Upload and showcase projects domain-wise
3. Choose and collaborate on a project
4. Feedback and review
5. Payment option
6. Progress tracking for each project
7. Monthly projects recap for freelancers
8. Chat option
9. Provide freelancer with *Cookies* which enhances the profile and credibility of freelancer

Top Level Data Flow Diagram (DFD)



2.3 User Classes and Characteristics

Co-Lancer has two main user classes:

1. Freelancer
   Freelancers can view and work on projects. Based on the project difficulty and individual contribution, they get paid by the clients. Freelancer gets reviews and ratings from clients.

2. Client
   Client uploads requirements and publishes projects under various domains. These will be picked up and completed by freelancers. Clients will pay freelancers and give feedback and reviews and rate their performance.

2.4 Operating Environment

- **End-User Devices:**
  Users can access the website from various devices, including desktop computers, laptops, tablets, and smartphones. The website would be designed to be responsive and accessible across different screen sizes and devices.
- **Web Browsers:**
  The website will be accessible through any modern web browsers such as Google Chrome, Mozilla Firefox, Apple Safari, Microsoft Edge and others that can support ReactJS. Co-Lancer ensures browser-compatibility and is optimized for faster rendering.
- **Operating System:**
  Users may have different operating systems on their devices, such as Windows, macOS, Linux, Android, or iOS. Co-Lancer is not OS dependent.
- **Internet Connection:**
  Users can access the website over the internet, so a stable internet connection is essential for the website to load and function properly.
- **Hardware:**
  No hardware specifications required for Co-Lancer

2.5 Design and Implementation Constraints

1. **Security Considerations:**
   - Robust security measures are implemented to protect user data (such as password, personal details) and payment information.
   - User authentication and authorization mechanisms are secured to prevent unauthorized access to accounts and confidential data.
   - Best practices are implemented to prevent fraud, scams, or misuse of the platform.
2. **Technology Stack and Tools:**
   - Constraints with respect to choice of technology stack, programming language and development tools depend on existing expertise of the development team and assigned budget.
   - Compatibility constraints exist with specific databases, APIs, or third-party services that are integrated with the platform.
3. **Data Storage and Retrieval:**
   - Constraints related to database design and optimization for efficient data storage and retrieval.
4. **Integration with Third-Party Services:**
   - Constraints related to integration of external services like payment gateways or communication tools.
5. **Development Timeline:**
   - Constraints related to project deadlines that require adherence to a strict schedule.

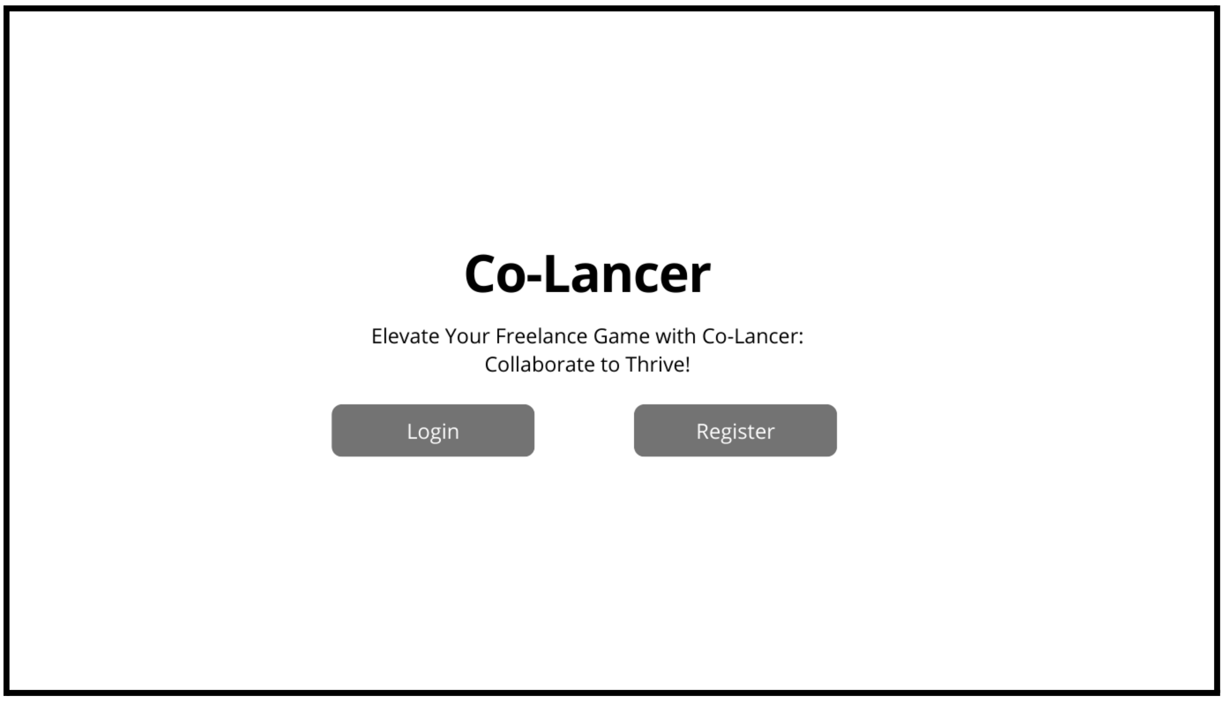
2.6 Assumptions and Dependencies

- **Assumptions**
  User is using a modern web browser like Google Chrome, Mozilla Firefox, Apple Safari, Microsoft Edge which supports ReactJS
- **Dependencies**
  ReactJS Version: 18.2.0
  NodeJS Version: 18.18.0
  MySQL Version: 8.0
  ExpressJS Version: 4.18.2

3. External Interface Requirements

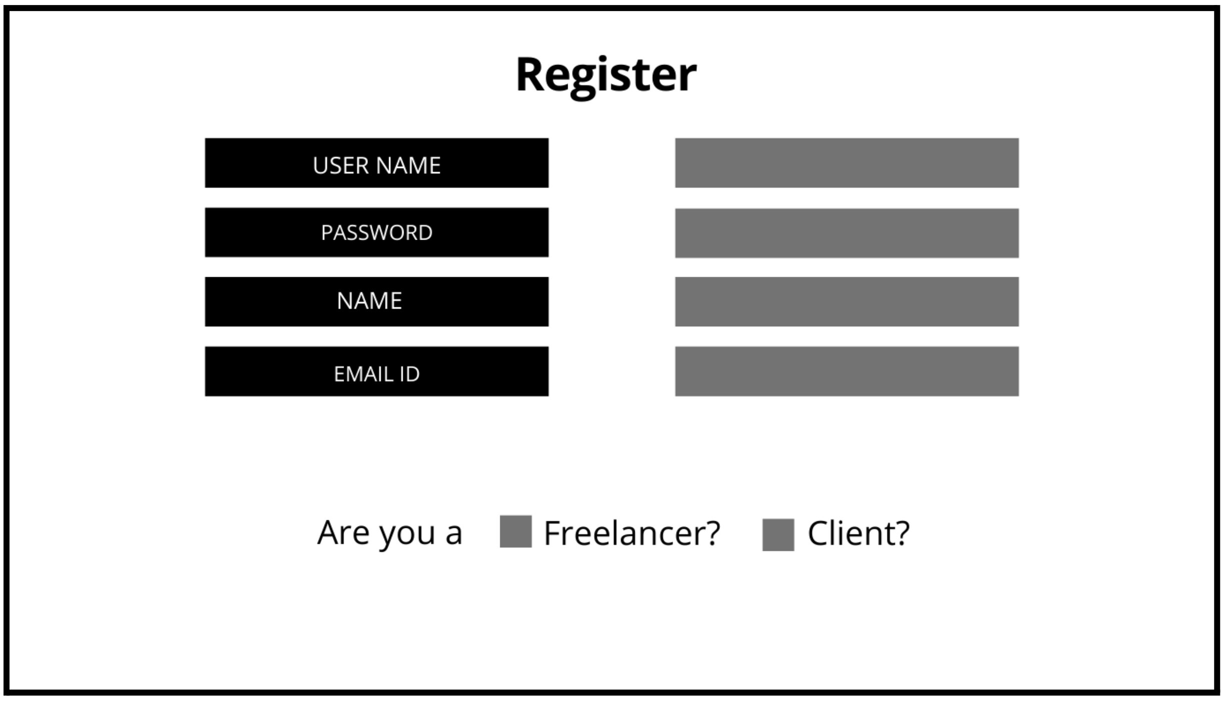
3.1 User Interfaces

1. User Home Page:

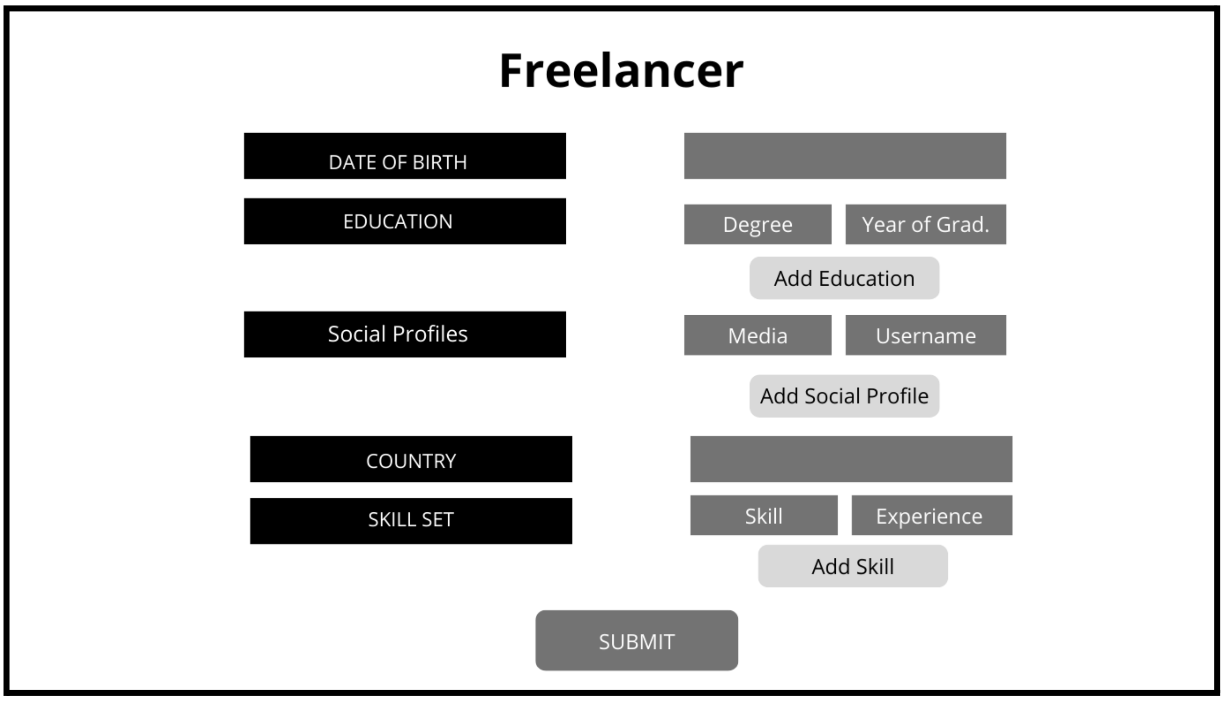

Co-Lancer
Elevate Your Freelance Game with Co-Lancer: Collaborate to Thrive!
Login
Register

> Both client and freelancer are considered as a user.

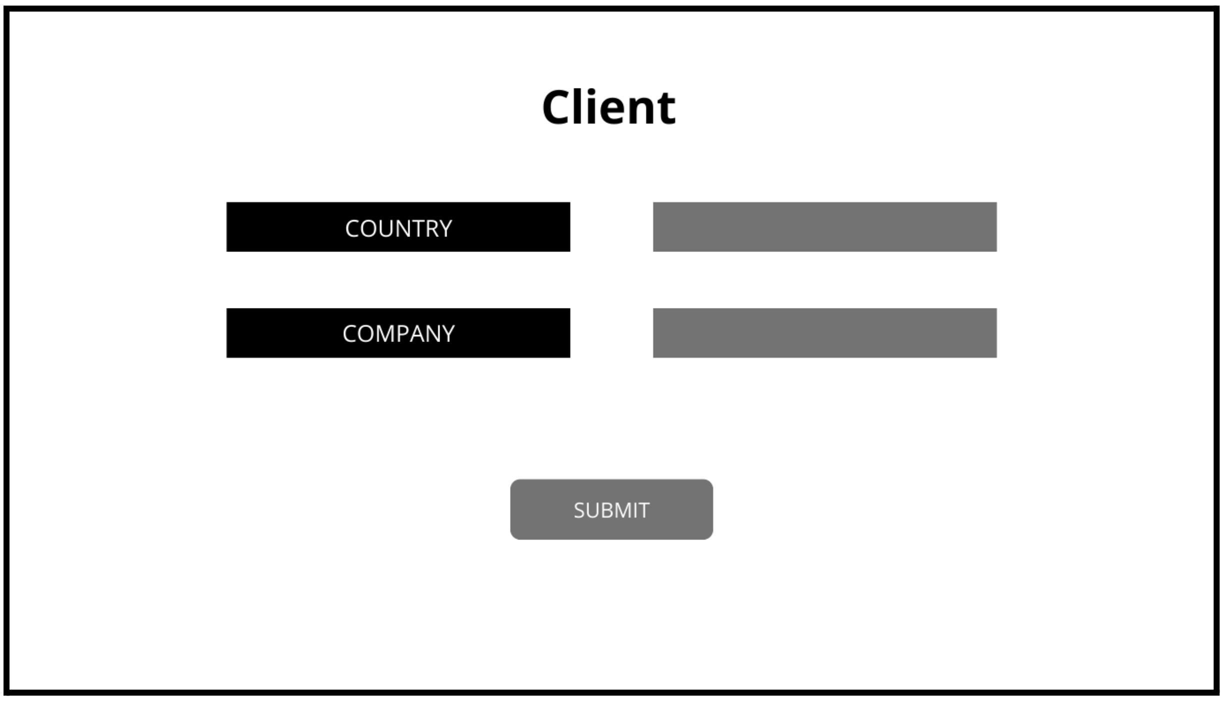
2. User Registration View:



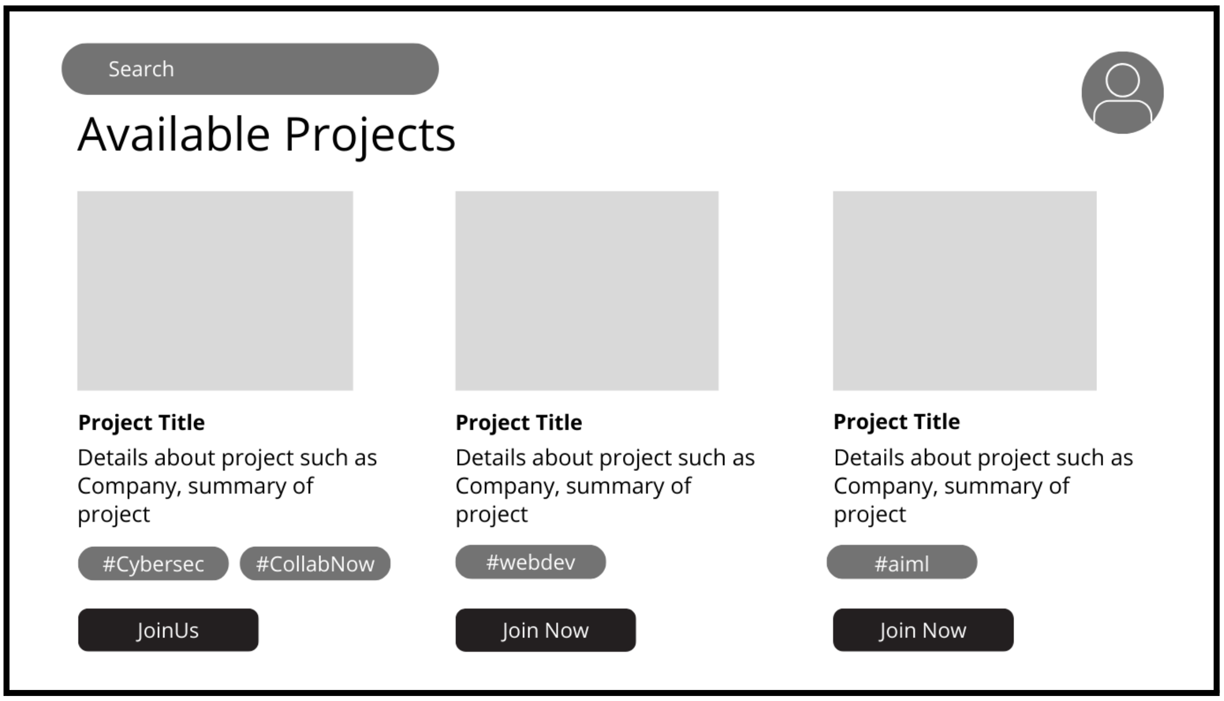
3. Freelancer Register View:



4. Client Register View:

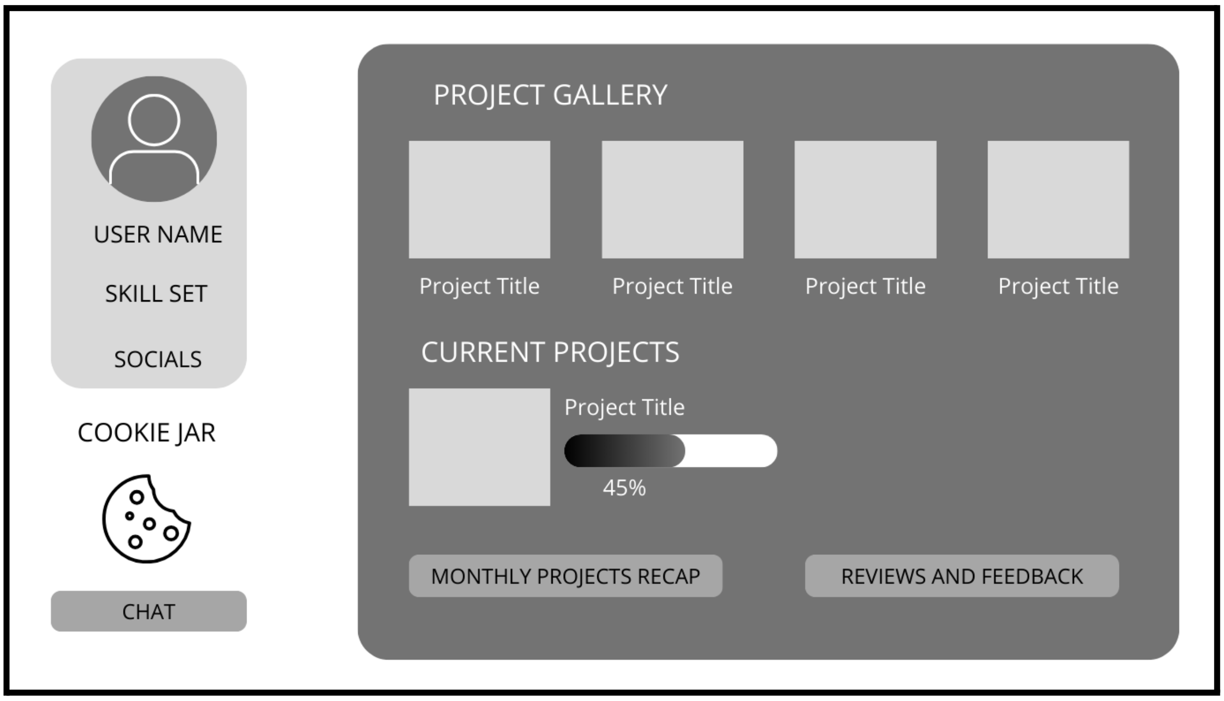


5. Freelancer Project View:


Search
Available Projects
Project Title
Details about project such as Company, summary of project
#Cybersec
#CollabNow
JoinUs
Project Title
Details about project such as Company, summary of project
#webdev
Join Now
Project Title
Details about project such as Company, summary of project
#aiml
Join Now

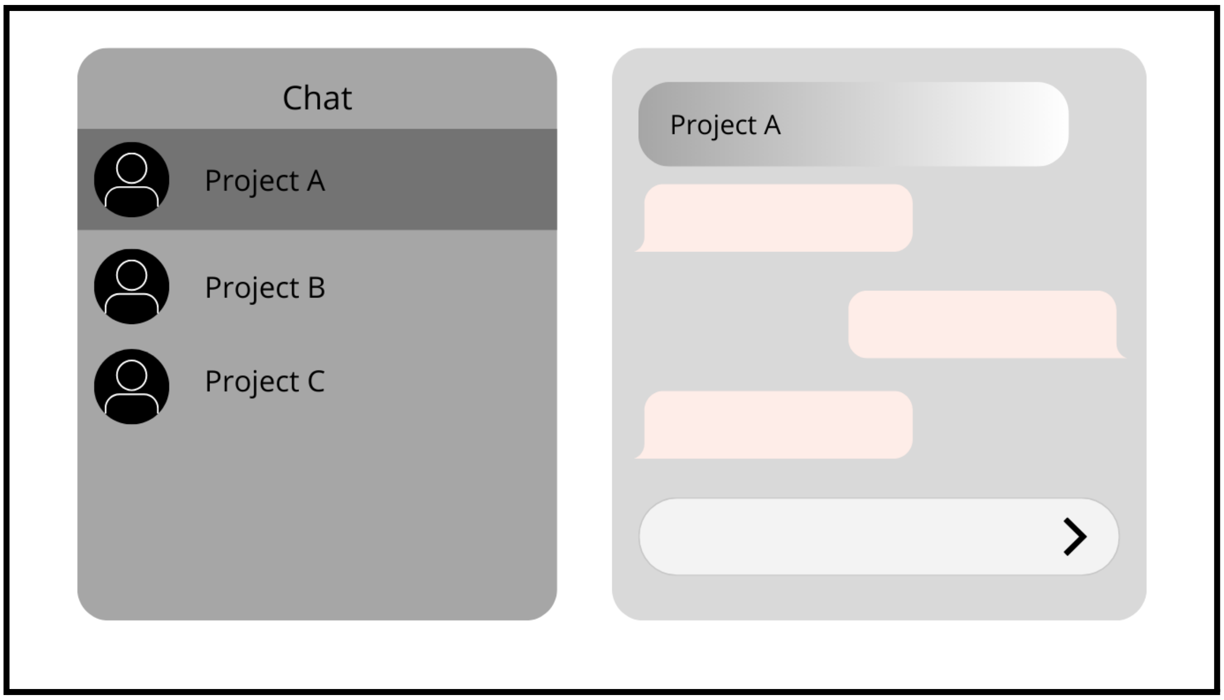
> View available only for freelancers, each freelancer’s view differs based on their domains and skill set.

6. Freelancer Profile View:



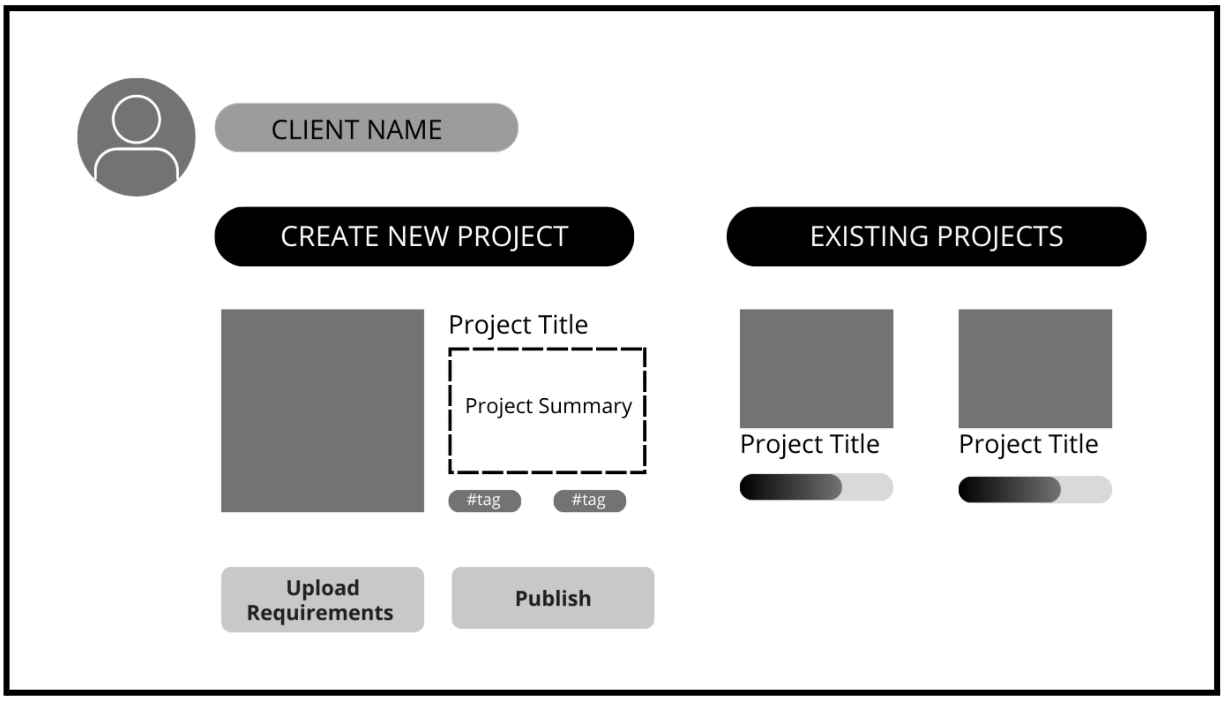
> View available for freelancers as well as clients.

7. Chat View:



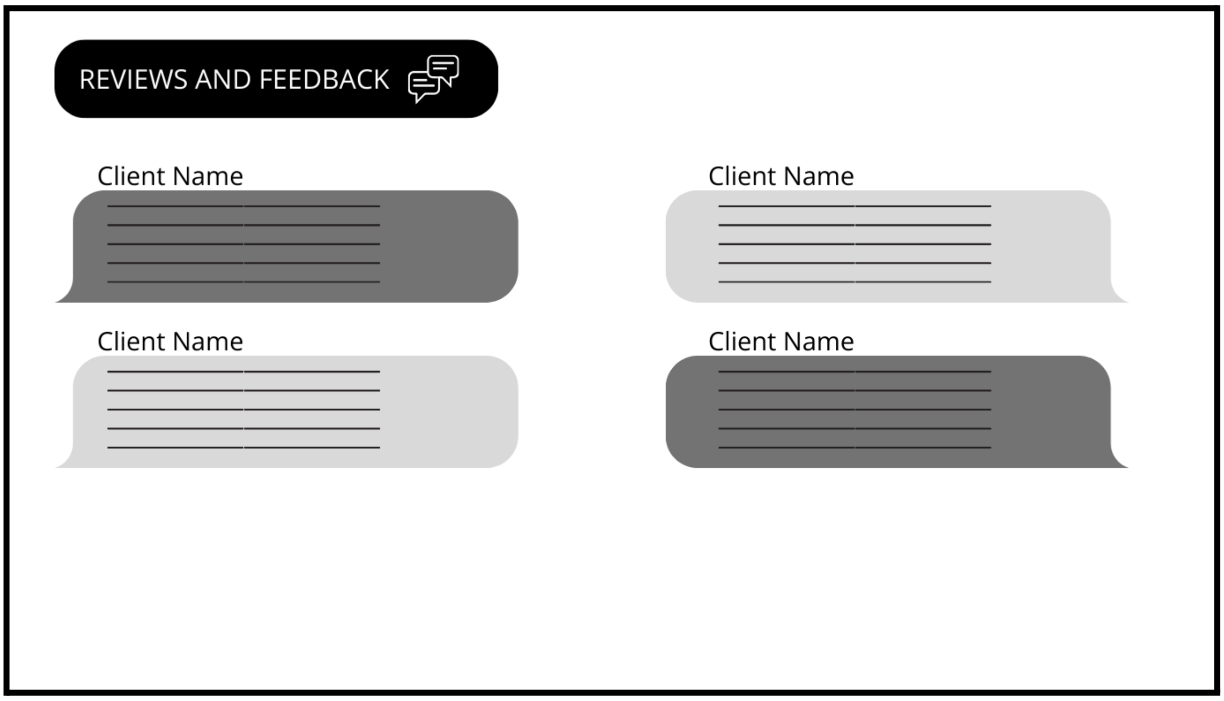
> View available for freelancers working on a project tagged with #CollabNow

8. Client Project View:



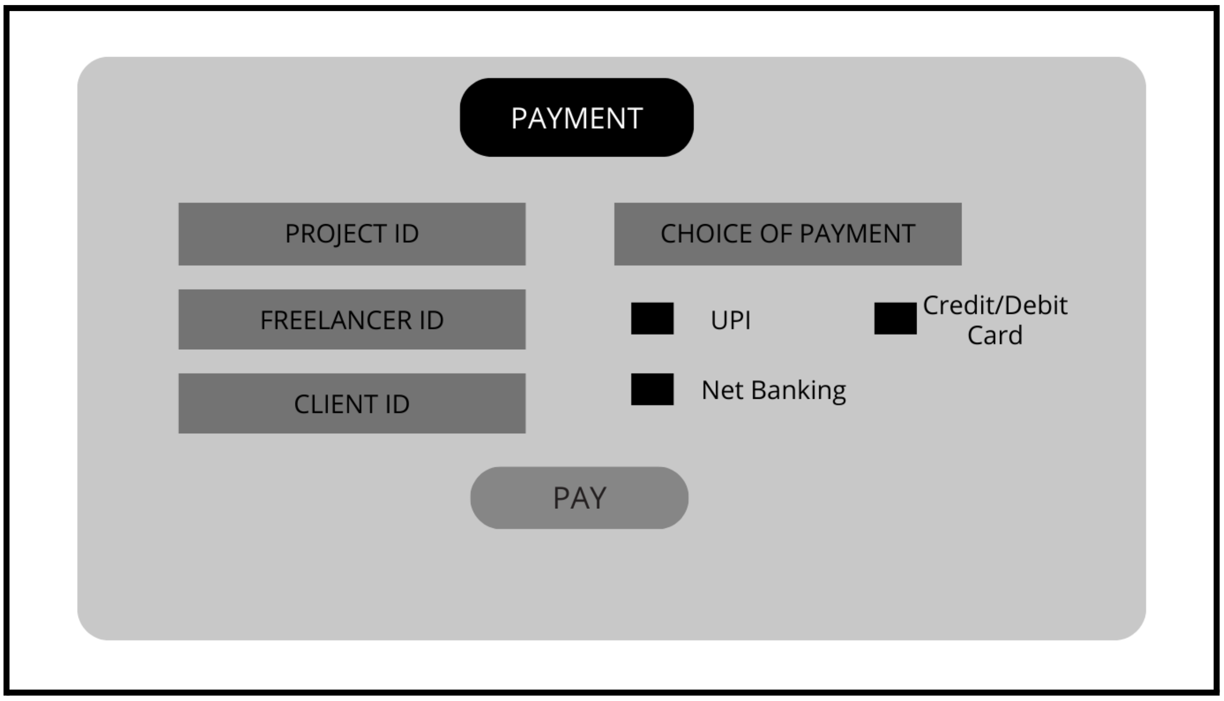
> View available for each client, a client cannot see another client's project view.

9. Review and Feedback View:



> Feedback view available for freelancers and clients.

10. Payment View:



> Payment view available for client to pay.

3.2 Software Interfaces

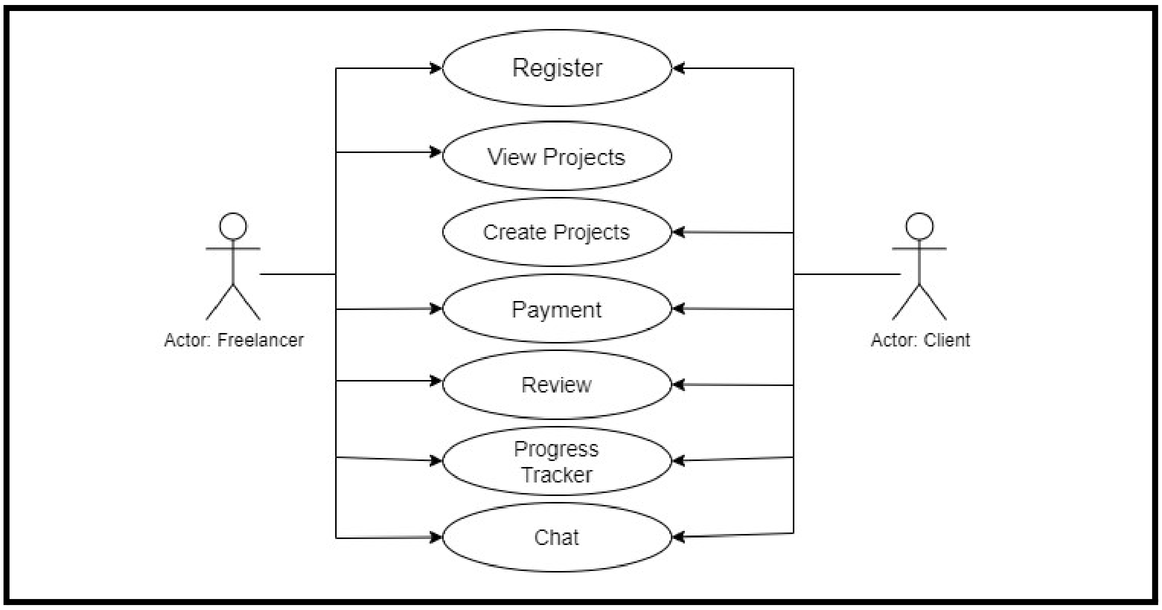
1. **Frontend:**
    - React.js: The frontend framework used to build the user interface.
    - Tailwind CSS: Used to styling the webpages.
2. **Backend Server:**
    - Node.js: A JavaScript runtime for server-side development.
    - Fetch API: NodeJS library for making HTTP requests to backend servers.
    - Express.js: Handle HTTP requests from frontend using API
3. **Database Interface:**
    - MySQL Database: The relational database management system for storing data.
    - MySQL Node.js Driver: A Node.js package for connecting to and interacting with the MySQL database from the backend.
    - Follows a three tier application architecture.
4. **API Endpoints:**
    - API endpoints in the backend server handle CRUD operations on the MySQL database.
    - These endpoints will be designed to receive HTTP requests (GET, POST, PUT, DELETE) from React frontend and perform the corresponding database operations.
5. **Deployment and Hosting:**
    - A hosting provider like Heroku, Vercel is used to deploy the frontend and backend.
    - Deployment pipelines to deploy the frontend and backend code to the hosting platform.

3.3 Communications Interfaces

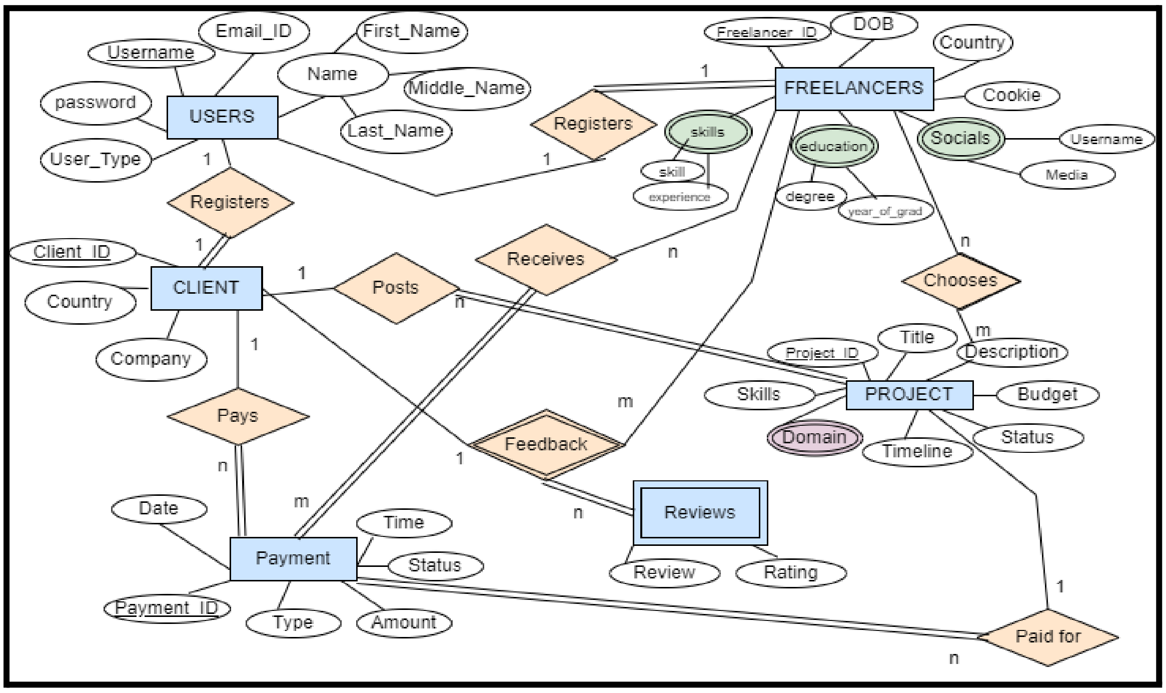
- HTTPS protocol will be used to ensure security and authentication of actions like payment .
- Fetch API library in NodeJS will be used to connect the frontend to backend database.
- Secure password storage and transfer by using hashed password in database.

4. Analysis Model

Use-case Diagram



ER Diagram



5. System Features

5.1 Create User Profile

5.1.1 Description and Priority

It allows users to provide their details and create a user profile which uniquely identifies them, this profile is enough to access all necessary information and functions.
This feature is of high priority as the user classes are decided based on this feature and this decides the views and functionalities of the user.

5.1.2 Stimulus/Response Sequences

Stimulus: User clicks on ‘Register’ button
Response: Display of registration page where the user can fill in the details
Stimulus: User clicks on ‘Create Profile’ button
Response: Details get submitted and the user profile is created

5.1.3 Functional Requirements

- **Registration Page:** Provide a registration page accessed by clicking 'Register.'
- **User Profile Creation:** Collect details like Name, Email, Username, password from clients and freelancers. Freelancers should provide additional details namely Education and Skill. The Clients must provide details about the Organization they work for. Ensure that the usernames are unique and passwords are stored in hashed manner. The users submit the details by clicking on a ‘Submit’ button.
- **Error Handling:** Error messages to be displayed on wrong entries.
- **User Profile Page:** A profile page is created for both freelancers and clients offering different functionalities. Freelancer’s profile must display the projects completed by them and a rating of their work. Clients will be able to access the feedback page for the freelancer. The client page must display their details along with the projects offered by them.
- **Data Validation:** The input values must be validated, username for their uniqueness, passwords must follow their constraints.
- **User Class Determination:** The users are classified as freelancers or clients.

5.2 Freelancer Chooses Project

5.2.1 Description and Priority

Freelancers can view the various projects matching with their skill set open for contribution and choose a project. If the project is tagged for collaboration then they can choose to collaborate with other freelancers working on the project.
This feature is of high priority as it forms the core of the website.

5.2.2 Stimulus/Response Sequences

Stimulus: User clicks on ‘Join Now’ button inside the project details page
Response: The selected project gets assigned to the freelancer and shows up in the ‘Current Projects’ tab of the freelancer.

5.2.3 Functional Requirements

- **Project Listings:** A list of projects must be displayed that match the skills of a freelancer. The details displayed must include the project title, description, required skills, project requirements and collaboration status. A ‘JoinNow’ button allows freelancers to work on the project.

- **Project Details Page:** A project page consists of detailed information such as requirements, the number of freelancers working on it, the proposed timeline.
- **Freelancer Selection:** On clicking ‘JoinNow’ the project is assigned to the respective freelancer and it is reflected on the ‘Current Project’ tab of the freelancer’s profile.
- **Collaboration Tagging:** Projects can be tagged for collaboration which signals the freelancers that they can collaborate on that project.

5.3 Project Showcase Gallery for Freelancer

5.3.1 Description and Priority

Displays all the projects that are in-progress or completed by the specific freelancer.
This feature is of medium priority as it gives a better understanding of the freelancer however the website can function even without this feature.

5.3.2 Stimulus/Response Sequences

Stimulus: User clicks on ‘Show Projects’ button in a freelancer’s details page
Response: The projects that the freelancer has completed or is currently working on is displayed.

5.3.3 Functional Requirements

- **Project Display:** Projects marked as complete or work in progress must be displayed along with project details such as name, description and completion status.

5.4 Upload Requirements of Project

5.4.1 Description and Priority

The client can upload the project details along with the domain and skills required based on the client’s technology choices for the implementation.
This feature is of high priority as the whole website is based on this core functionality of a freelancer collaboration website .

5.4.2 Stimulus/Response Sequences

Stimulus: Client clicks on the ‘Create New Project’ button on their profile page.
Response: A upload page is loaded which has a form to fill in the project details.
Stimulus: Client clicks on ‘Upload Project’ button on the form page
Response: The project details are stored in the backend and a new project is created globally and is accessible to all users.

5.4.3 Functional Requirements

- **Project Upload Page:** The ‘Create New Project’ button leads to an upload page where the client fills in the project details. A ‘Publish’ button on the page is used to upload the project along with its details onto the website.
- **Project Details Page:** This page is aimed to ensure that the users viewing the page get clarity on the project. The client is provided with a form to fill in details such as title, description, timeline, tags, requirements.
- **Project Requirements:** ‘Upload Requirements’ button allows the Client to upload the requirements as a document.
- **Backend Storage:** The project details are stored in tables in MySQL.
- **Globally Accessible:** The project data is accessible to all the users of Co-Lancer.

5.5 Collaboration of Freelancers on Projects

5.5.1 Description and Priority

A freelancer can choose a *'#CollabNow'* tagged project to start collaborating. All the freelancers involved with that project can have a discussion using the chat feature (as referenced in 5.9). Once the project is completed the freelancers will receive their payment based on their contribution to the project.

This feature is of high priority as it allows multiple freelancers to work on a project which is a core feature of a freelancer collaboration website .

5.5.2 Stimulus/Response Sequences

Stimulus: Freelancer clicks on 'JoinUs' button on the project tagged with *#CollabNow*
Response: Project gets added to the freelancer's profile and the freelancer gets added as a collaborator in the selected project.

5.5.3 Functional Requirements

- **Project Collaboration:** Projects tagged "#CollabNow" are open for collaboration and accessible to all freelancers.
- **Collaborator Management:** 'JoinUs' button present on the project adds the freelancer as collaborator on the project and the project is displayed on their profile.
- **Discussion Feature:** A chat feature (as referenced in 5.9) is enabled for freelancers to communicate with each other.
- **Payment Distribution:** The payment of freelancers is based on their respective contribution to the project which is decided based on LOC.

5.6 Review and Feedback on Freelancer

5.6.1 Description and Priority

This feature allows a client to give feedback and reviews for a freelancer based on a project implementation. They can also give ratings to the freelancer which will improve their credibility score accordingly and give them *'cookies'* which can be used to verify and display their experience level.

This feature is of medium priority and improves the overall experience of using the website but does not affect the actual goal of the website.

5.6.2 Stimulus/Response Sequences

Stimulus: Client clicks on the 'Feedback' button in the home page.
Response: A feedback form opens with necessary fields.
Stimulus: Client clicks on the 'Submit Feedback' button.
Response: The form details are stored in the backend and it gets reflected on the reviews tab of the freelancer.

5.6.3 Functional Requirements

- **Feedback Form:** A "Review and Feedback" button present on the freelancers profile leads to a feedback form. Only the clients can submit feedback and view reviews whereas a freelancer can only view their reviews and feedback.
- **Submission of Feedback:** Clients submit feedback and reviews using a "Submit" button. This data is stored in the backend database.
- **Rating System:** A rating system where clients rate freelancers based on their performance. This contributes to the 'Cookie' meter present in their page which displays their experience level.

5.7 Verify and Categorize Freelancers Based on Rating

5.7.1 Description and Priority

This feature enables the clients to give a rating to the freelancer based on the freelancer's performance on a project. This rating is used to categorize the freelancer based on their experience level and client satisfaction and they are provided with pieces of cookie as they contribute which indicates the same in their profile in the form of completeness of cookies. This feature is of medium priority and it provides a better user experience.

5.7.2 Stimulus/Response Sequences

Stimulus: Client clicks on the ‘Feedback’ button in the home page.
Response: A feedback form opens with a rating field and other necessary fields
Stimulus: Client clicks on the ‘Submit Feedback’ button.
Response: The form details are stored in the backend and the rating value updates the credibility score of the freelancer

5.7.3 Functional Requirements

- **Rating in Feedback Form:** Allow clients to rate freelancers on a scale of 1 to 10. When clients submit feedback with a rating it is stored in the backend.
- **Categorization:** The rating will be used to update the freelancer's credibility score and categorize freelancers based on their experience level and client satisfaction.
  Experience levels will be categorised as: novice, intermediate and expert.
- **Display of Cookies:** Provide freelancers with "cookies" as they receive positive ratings. Display the ‘cookie jar’ on the freelancer's profile to indicate their experience level and client satisfaction.

5.8 Payment Options

5.8.1 Description and Priority

Allows a smooth transaction of payment to the freelancers from the client. Multiple payment modes will be available and the transaction status will reflect on both client side and freelancer side in the form of a receipt . In case of multiple freelancers working on a project the payment is split based on LOC (Lines Of Code) and transferred to them separately .
This feature is of high priority as it is necessary to complete the functionality of a freelancer collaboration website.

5.8.2 Stimulus/Response Sequences

Stimulus: Client clicks on the ‘Go to Payment’ button on the project page which gets enabled only when the project is fully completed.
Response: A form taking in the necessary details is loaded.
Stimulus: Client fills the form and clicks on the ‘Pay’ button.
Response: The backend is updated and it completes the transaction.

5.8.3 Functional Requirements

- **Payment Modes:**Multiple payment modes available for clients to choose when making payments to freelancers. Options that will be included are credit/debit card, UPI, netbanking.
- **Transaction Status:** To ensure atomicity and integrity in the transaction there will be three status options for a transaction which will be ‘complete’, ‘pending’ and ‘failed’. It will be ensured that the transaction status is reflected on both the client and freelancer sides with a receipt and confirmation of the transaction.

- **Payment Splitting:** For projects where multiple freelancers collaborate, the payment is split based on Lines Of Code (LOC) contributed by each freelancer. The transfer will be made to each freelancer separately.
- **Payment Initiation:** The "Go to Payment" button on the project page will be enabled only when the project is fully completed. When clicked on it, it will load a form to collect necessary payment details and complete the transaction.
- **Payment Confirmation:** After the client fills the form and clicks "Pay" , update the backend to complete the transaction and send confirmation to both freelancer and client.

5.9 Chat Option

5.9.1 Description and Priority

Provides an interface to communicate between multiple freelancers and clients.
This is a medium priority feature and it enhances the user experience and makes communication easier.

5.9.2 Stimulus/Response Sequences

Stimulus: User clicks on the 'Chat' button
Response: A conversation box opens and allows in-browser chat functionality

5.9.3 Functional Requirements

- **Chat Interface:** A chat interface for communication between multiple freelancers collaborating on a project will be provided.
- **Access Trigger:** A "Chat" button that users can click to initiate a chat.
- **In-Browser Chat:** Clicking the "Chat" button will open a conversation box with in-browser chat functionality. It allows users to send and receive messages within the chat interface and supports real-time or near-real-time message delivery.

5.10 Progress Tracking

5.10.1 Description and Priority

It allows the freelancers and clients to track the project's progress
This is a medium priority feature, the goal of the website can be satisfied even without this feature but it allows periodic evaluation of the project completion by both freelancers and clients

5.10.2 Stimulus/Response Sequences

Stimulus: User clicks on the 'View Progress' button in the project page.
Response: The progress of the project is displayed

5.10.3 Functional Requirements

- **Progress Display:** The progress is displayed as a bar in the freelancer's profile under the respective projects.
- **User Interaction:** Helps track the project completion in a clear and understandable format.

5.11 Monthly Projects Recap

5.11.1 Description and Priority

It provides a recap of all the projects, reviews and payments for a freelancer on a monthly basis.

It is a low priority feature, it does not add any improvement to the website as such but provides a better overview of the freelancer's project history .

5.11.2 Stimulus/Response Sequences

Stimulus: User clicks on the 'View Project Recap' button.

Response: The project recap page is loaded.

5.11.3 Functional Requirements

- **Recap Page:** 'Monthly Project Recap' leads to a recap page that displays all projects, payments, reviews on a monthly basis.

6. Other Non-functional Requirements

6.1 Performance Requirements

- Concurrent multi-user support
- Component based rendering and virtual DOM evokes faster loading of pages
- Usage of MySQL database provides faster retrieval of data
- Easier searching of projects using the search filter and domain tags

6.2 Safety Requirements

- Password hash stored in the backend instead of the actual password for user authentication to ensure privacy and safety.
- Delete chat history once the project is complete.

6.3 Security Requirements

- Encrypted chat messages.
- Passwords must meet constraints such as length and combination of alphanumeric and special characters to improve security.

6.4 Software Quality Attributes

- Availability: The website will be available to users concurrently accessing it without any server crashes.
- Usability: User friendly interface which allows intuitive navigation and easy to understand components.
- Maintainability: Clear and concise documentation of the product and modularized code will make the product maintainable. Github will be used for version control making it easier to rollback and document the changes.
- Compatibility: The product can be compatible with any device and any operating system.
- Performance: ReactJS ensures better performance due to its dynamic and component based page rendering.

6.5 Business Rules

- An individual must register as either Freelancer or Client (on behalf of a Company)
- Only a user registered as client can create new project
- Review and Feedback features are limited to a Client.
- Users registered as Freelancers are given the privilege of collaborating on projects.

7. Other Requirements

No other requirements were identified.

Appendix A: Glossary

1.	Coupling	The interdependence between the modules in the project
2.	Stakeholders	Individuals that play a part in the project building process
3.	Cookies	Proposed rating system that is used in this project
4.	OS	Operating System is the software component that manages and controls hardware
5.	Backend	Backend involves processes responsible for data manipulation
6.	LOC	Lines of Code is a measure of how many lines of code a developer has coded in a particular time slot
7.	Three Tier Architecture	Architecture where the application client does not make database calls and only acts as an interface to connect to the application server which in turn connects to the database.
8.	CRUD	Create, Read, Update, Delete It is the four basic operations that should be possible in persistent storage.

Appendix B: Field Layouts

Information needed for registering an user:

Field Name	Length (Max)	Data Type	Description	Constraints
Username	30	alphanumeric	Username of user	mandatory and unique
Email_ID	50	alphanumeric	Email id of user	mandatory and unique
Password	30	alphanumeric	Strong password of user stored as secure hash	mandatory
Name	90	string	Full name of user	mandatory
User_type	-	ENUM (‘client’,’freelancer’)	If user is freelancer of client	-

Additional Information needed for freelancer registration (after registering as an user)

Field Name	Length (Max)	Data Type	Description	Constraints
DOB	10	date (YYYY-MM-DD)	Date of birth	-
Country	30	string	Nationality of freelancer	-
Educational Degree Year of graduation	30 4	alphanumeric year (YYYY)	-	Can be multi-valued (multiple educational qualifications)
Skill set Skill Experience	30 -	string ENUM(‘basic’, ‘intermediate’, ‘advanced’)	-	Can be multi-valued
Social Profile media_name username	30 30	string alphanumeric	Social media name and the username in that media	Can be multi-valued

Additional Information needed for client registration (after registering as an user)

Field Name	Length (Max)	Data Type	Description	Constraints
Country	30	string	Nationality of client	-
Company	30	string	Company represented by client	mandatory

Information for payment:

Field Name	Length(Max)	Data Type	Description	Constraints
Project ID	10	alphanumeric	Project id for which payment is to be done	mandatory
Client ID	10	alphanumeric	Client's own id for payment verification	mandatory

Information for creating new project (done by client)

Field Name	Length (Max)	Data Type	Description	Constraints
Title	30	string	Title for project	mandatory and unique
Description	100	string	Brief description about the project	-
Budget	-	int	Client's budget estimation	-
Timeline	-	int	Client's timeline requirements	-
Skills	30	string	Skills required for the project	can be multi-valued
Domains	30	string	Domains in which the project belongs	can be multi-valued

Information for submitting reviews:

Field Name	Length (Max)	Data Type	Description	Constraints
Client_id	10	alphanumeric	Unique id given to client	mandatory
Freelancer_id	10	alphanumeric	Unique id given to freelancer	mandatory
Review	100	string	Review about freelancer	-
Rating	-	ENUM(1-10)	Rating given to freelancer	-

APPENDIX C: Requirements Traceability Matrix

ID	Requirement Description	Business Need	Priority	WBS	Specification	Design	Test Cases
1.	Create user profile	User needs to create a profile on the website	High				
2.	Freelancer chooses project	Freelancer has to pick up a project based on skill set	High				
3.	Project Showcase Gallery for Freelancer	Freelancer is selected based on the projects done therefore included in freelancer profile	Medium				
4.	Upload Requirements of Project	Every project needs the requirements and functionalities that must be implemented. Hence an option to upload the requirements of the project is given.	High				
5.	Collaboration of Freelancers on Projects	Complex projects and diverse projects require a variety of freelancers. This is the	High				

		basic aim of the site.					
6.	Review and Feedback on Freelancer	Freelancers are motivated by reviews and clients judge the freelancers based on the reviews.	Medium				
7.	Verify and Categorize Freelancers Based on Rating	Clients must be assured of the freelancers expertise and a rating system based on their work fulfills that need.	Medium				
8.	Payment Options	The freelancers must be paid for the work they do. This feature makes the payment process easier.	High				
9.	Chat Option	The freelancers collaborating on a project need to be in touch constantly. This need is fulfilled by a chat feature.	Medium				
10.	Progress Tracking	Freelancers need to keep track of the progress of their project. This can be achieved by a progress tracker.	Medium				
11.	Monthly Projects Recap	Freelancers can keep track of their monthly contribution and payments received.	Low				