



**Department of Computer Science and Engineering  
BENGALURU, KARNATAKA, INDIA.  
B. TECH. (CSE)**

**V SEMESTER**

**Aug. – Dec. 2023**

**UE21CS341A – SOFTWARE  
ENGINEERING PROJECT REPORT  
ON  
*CoLancer*  
*Freelancer Collaboration Website***

**SUBMITTED BY**

- 1) PES2UG21CS374 – POOJA SATHEESH**
- 2) PES2UG21CS396 – PREETHI M.**

## TABLE OF CONTENTS

| Sl. No. | Topic                               | Page No |
|---------|-------------------------------------|---------|
| 1.      | Proposal of Project                 | 2       |
| 2.      | Software Requirements Specification | 3       |
| 3.      | Project Plan                        | 28      |
| 4.      | Design Diagrams                     | 36      |
| 5.      | Test Cases                          | 43      |
| 6.      | Screenshots of Output               | 53      |

# **1. Proposal of Project**

Freelancers, despite their diverse skill sets, faced challenges in finding suitable projects and collaborating effectively. The identified issues prompted the development of a comprehensive platform aimed at not only addressing the existing gaps but also introducing innovative features to enhance the overall freelance experience.

- Some platforms lacked effective communication channels, which was introduced in CoLancer in the form of a chat feature.
- To assess a freelancer's expertise, a client can view the freelancer's skill sets and reviews
- A freelancer is rated by the client based on their work and reviews posted by clients help build the freelancer's profile
- To ease the communication between clients and freelancers, a client can post the requirements document directly on the website and freelancers can personally send the finished project to the client

All the above measures help set up a good and reliant communication channel and make the process of assigning and building a project easier.

## 2. Software Requirement Specification

### Table of Contents

| Sl. No. | Topic                                     | Pg. No. |
|---------|---|---------|
| 1.      | Introduction                              | 3       |
| 1.1     | Purpose                                   | 3       |
| 1.2     | Intended Audience and Reading Suggestions | 3       |
| 1.3     | Product Scope                             | 3       |
| 1.4     | References                                | 4       |
| 2.      | Overall Description                       | 4       |
| 2.1     | Product Perspective                       | 4       |
| 2.2     | Product Functions                         | 4       |
| 2.3     | User Classes and Characteristics          | 5       |
| 2.4     | Operating Environment                     | 6       |
| 2.5     | Design and Implementation Constraints     | 6       |
| 2.6     | Assumptions and Dependencies              | 7       |
| 3       | External Interface Requirements           | 7       |
| 3.1     | User Interfaces                           | 7       |
| 3.2     | Software Interfaces                       | 12      |
| 3.3     | Communications Interfaces                 | 13      |
| 4       | Analysis Models                           | 13      |
| 5       | System Features                           | 14      |
| 5.1     | Create user profile                       | 14      |

|      |   |    |
|------|---|----|
| 5.2  | Freelancer Chooses Project                        | 14 |
| 5.3  | Project Showcase Gallery for Freelancer           | 15 |
| 5.4  | Upload Requirements of Project                    | 15 |
| 5.5  | Collaboration of Freelancers on Projects          | 16 |
| 5.6  | Review and Feedback on Freelancer                 | 16 |
| 5.7  | Verify and Categorise Freelancers Based on Rating | 17 |
| 5.8  | Payment Options                                   | 17 |
| 5.9  | Chat Option                                       | 18 |
| 5.10 | Progress Tracking                                 | 18 |
| 5.11 | Monthly Projects Recap                            | 19 |
| 6    | Other Nonfunctional Requirements                  | 19 |
| 6.1  | Performance Requirements                          | 19 |
| 6.2  | Safety Requirements                               | 19 |
| 6.3  | Security Requirements                             | 19 |
| 6.4  | Software Quality Attributes                       | 19 |
| 6.5  | Business Rules                                    | 20 |
| 7    | Other Requirements                                | 20 |
|      | APPENDIX A: Glossary                              | 21 |
|      | APPENDIX B: Field Layouts                         | 22 |
|      | APPENDIX C: Requirement Traceability Matrix       | 24 |

# **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 organisations 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 organisation 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 organisation details and publish projects. Freelancers can register with their personal details along with their skill set, they can view the projects categorised 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

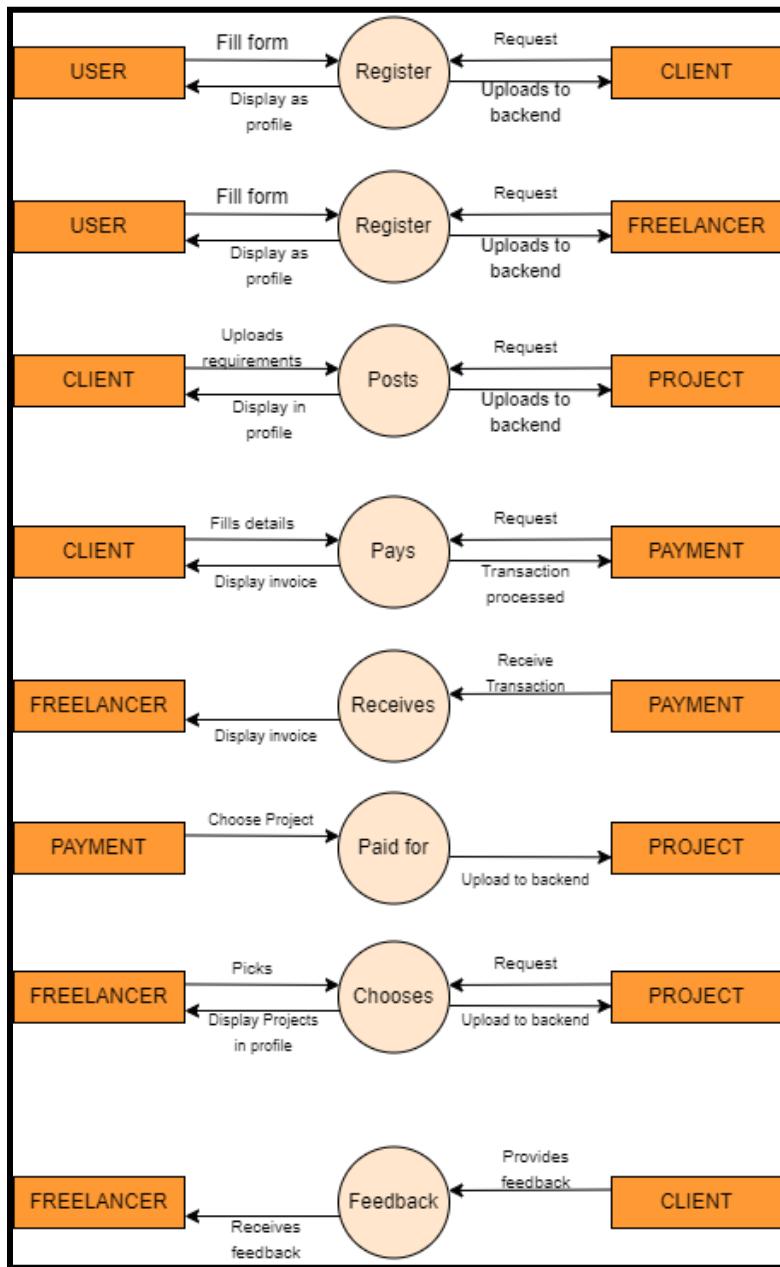
This product can work independently on any OS and any device with a modern browser support. It does not require any installation and does no modification to the host device. It does not require any additional permissions to be granted for its working . Co-Lancer is an independent standalone product and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

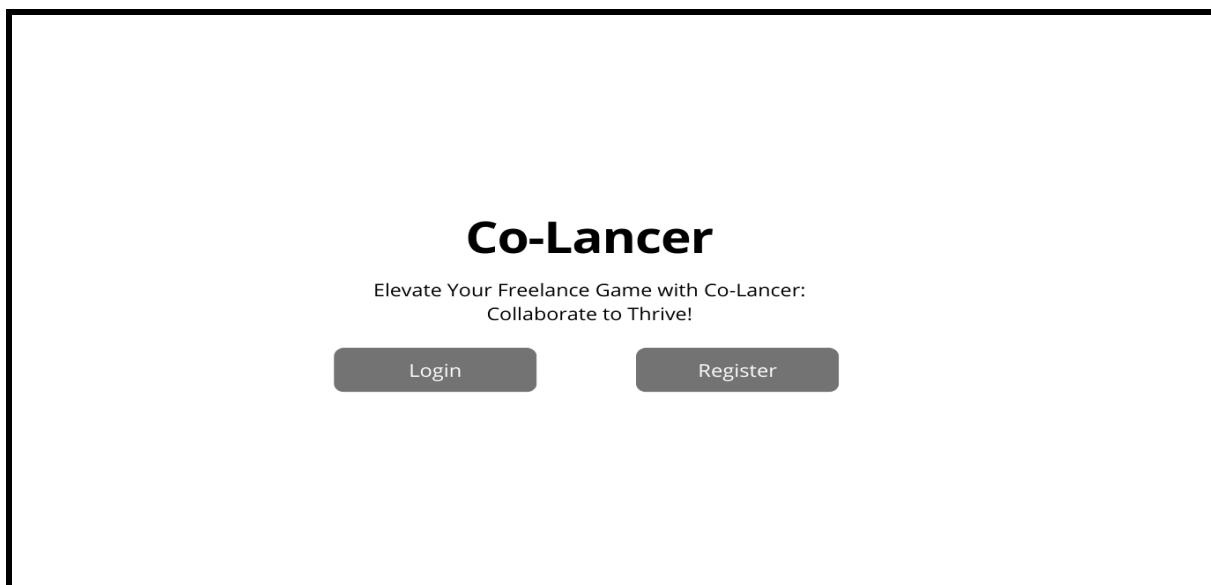
MySQL Version: 8.0

ExpressJS Version: 4.18.2

## **3. External Interface Requirements**

### **3.1 User Interfaces**

#### **1. User Home Page:**



> Both client and freelancer are considered as a user.

## 2. User Registration View:

# Register

|           |  |
|-----------|--|
| USER NAME |  |
| PASSWORD  |  |
| NAME      |  |
| EMAIL ID  |  |

Are you a  Freelancer?  Client?

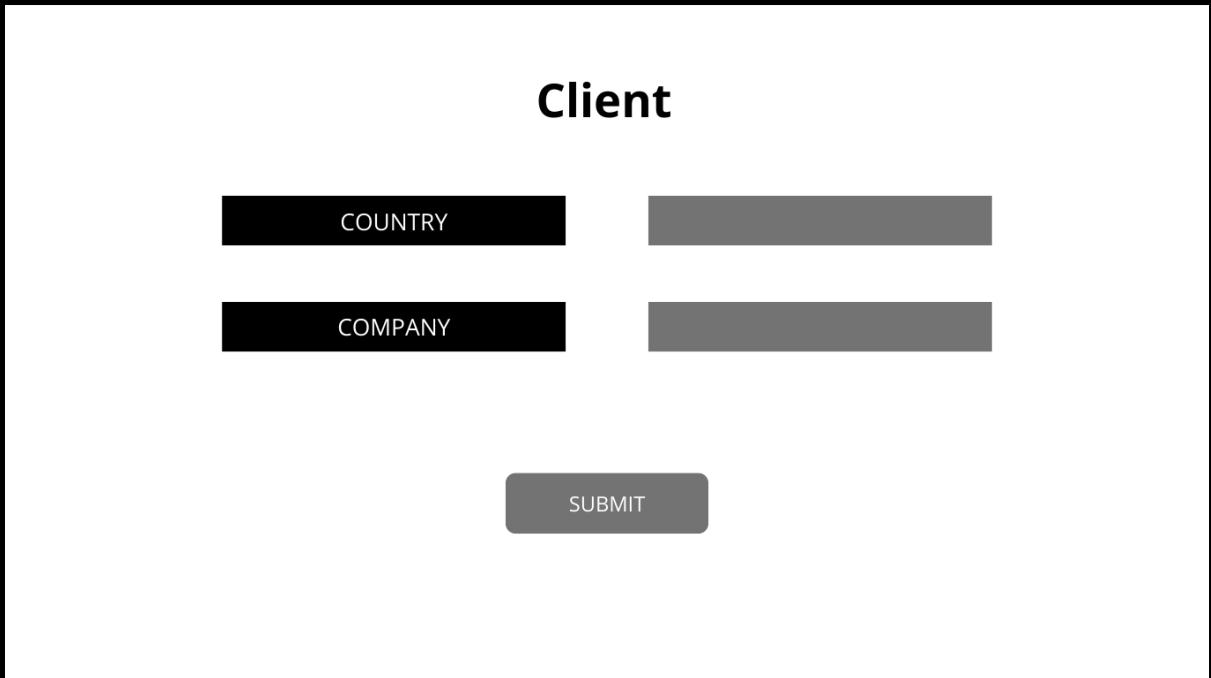
## 3. Freelancer Register View:

# Freelancer

|                 |                    |               |
|-----------------|--------------------|---------------|
| DATE OF BIRTH   |                    |               |
| EDUCATION       | Degree             | Year of Grad. |
| Social Profiles | Add Education      |               |
| COUNTRY         | Media              | Username      |
| SKILL SET       | Add Social Profile |               |
|                 | Skill              | Experience    |
|                 | Add Skill          |               |

SUBMIT

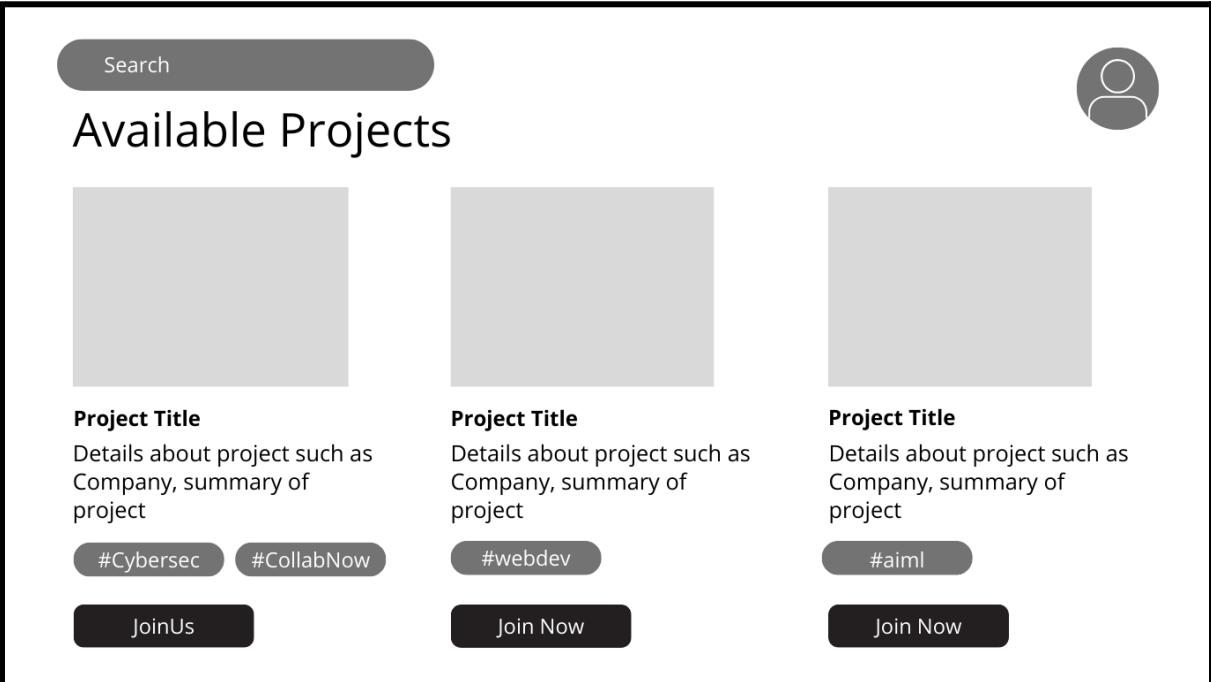
#### 4. Client Register View:



The form is titled "Client". It contains two sets of input fields: "COUNTRY" and "COMPANY", each with a black input field and a grey placeholder field. Below these is a "SUBMIT" button.

| Client  |  |
|---------|--|
| COUNTRY |  |
| COMPANY |  |
| SUBMIT  |  |

#### 5. Freelancer Project View:

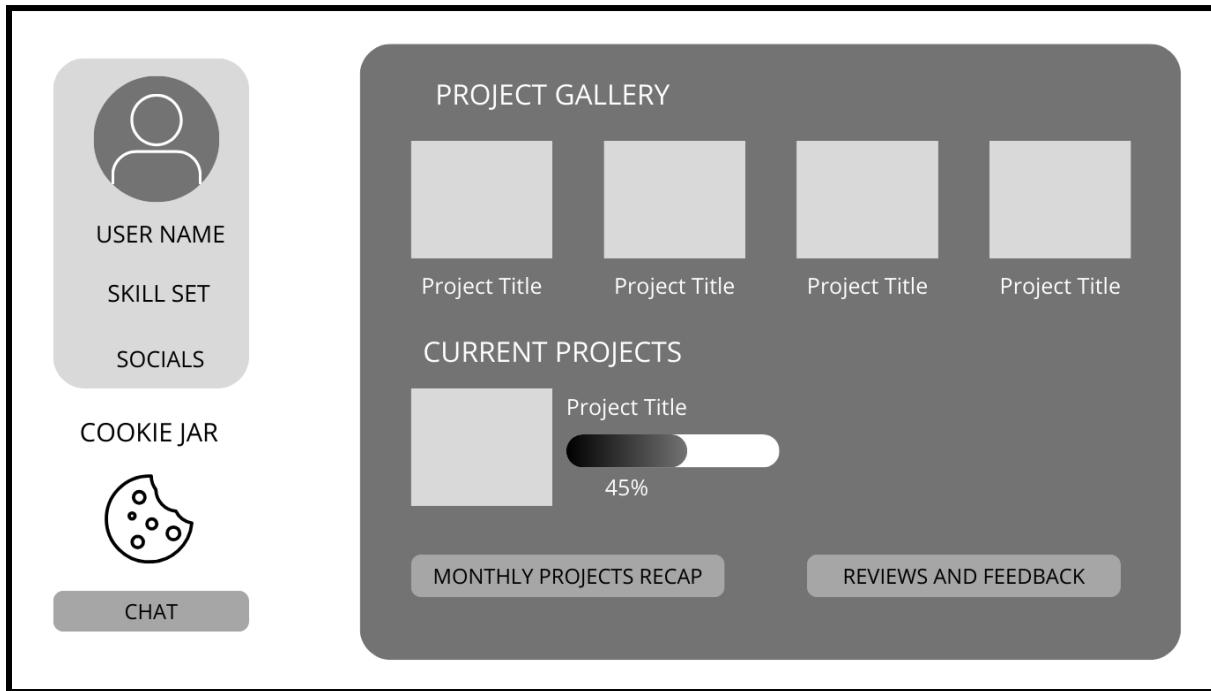


The interface shows a search bar at the top left and a user profile icon at the top right. Below is a section titled "Available Projects" with three project cards. Each card has a placeholder image, a title, a description, hashtags, and a "Join Us" button.

| Project Title   | Project Title   | Project Title   |
|---|---|---|
| Details about project such as Company, summary of project | Details about project such as Company, summary of project | Details about project such as Company, summary of project |
| #Cybersec #CollabNow                                      | #webdev   | #aiml   |
| Join Us   | Join Now  | Join Now  |

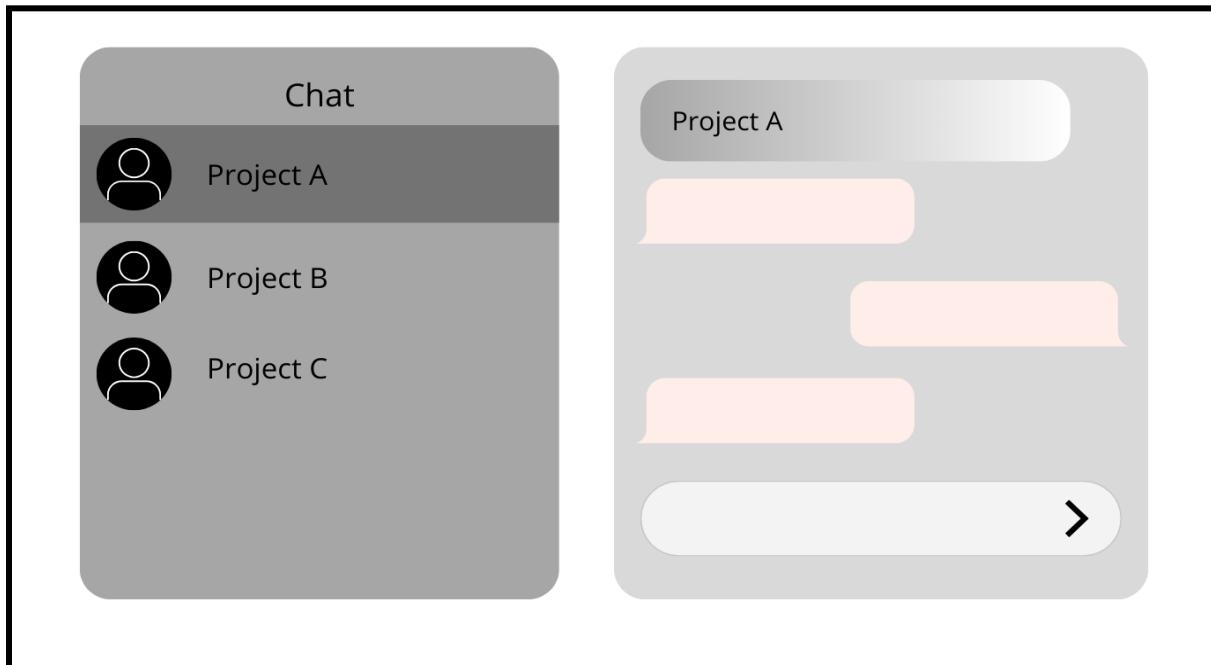
> View available only for freelancers, each freelancer's view differs based on their domains and skill set.

## 6. Freelancer Profile View:



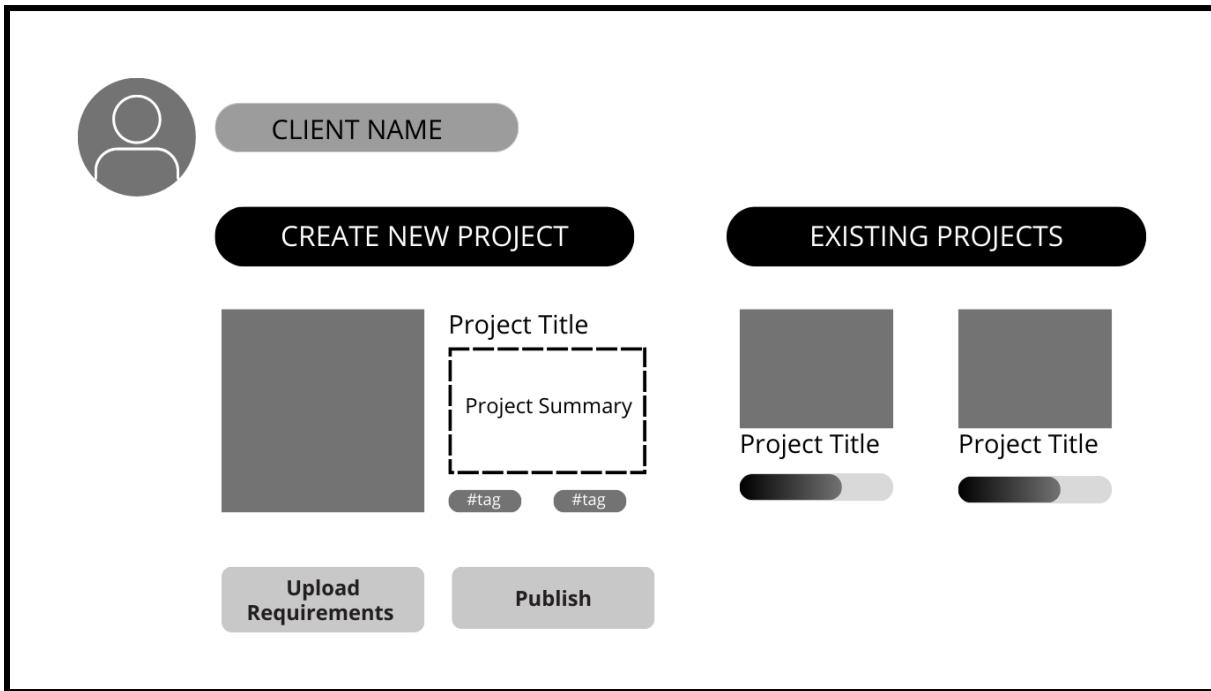
> View available for freelancers as well as clients.

## 7. Chat View:



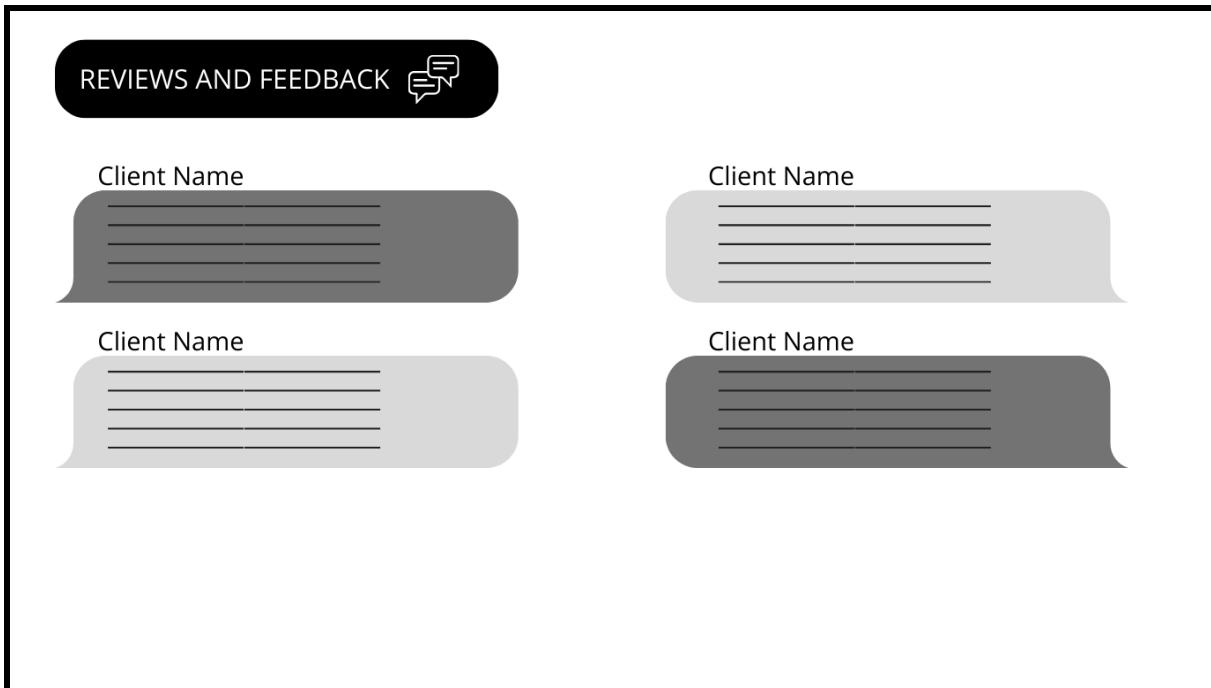
> View available for freelancers working on a project tagged with #CollabNow

## 8. Client Project View:



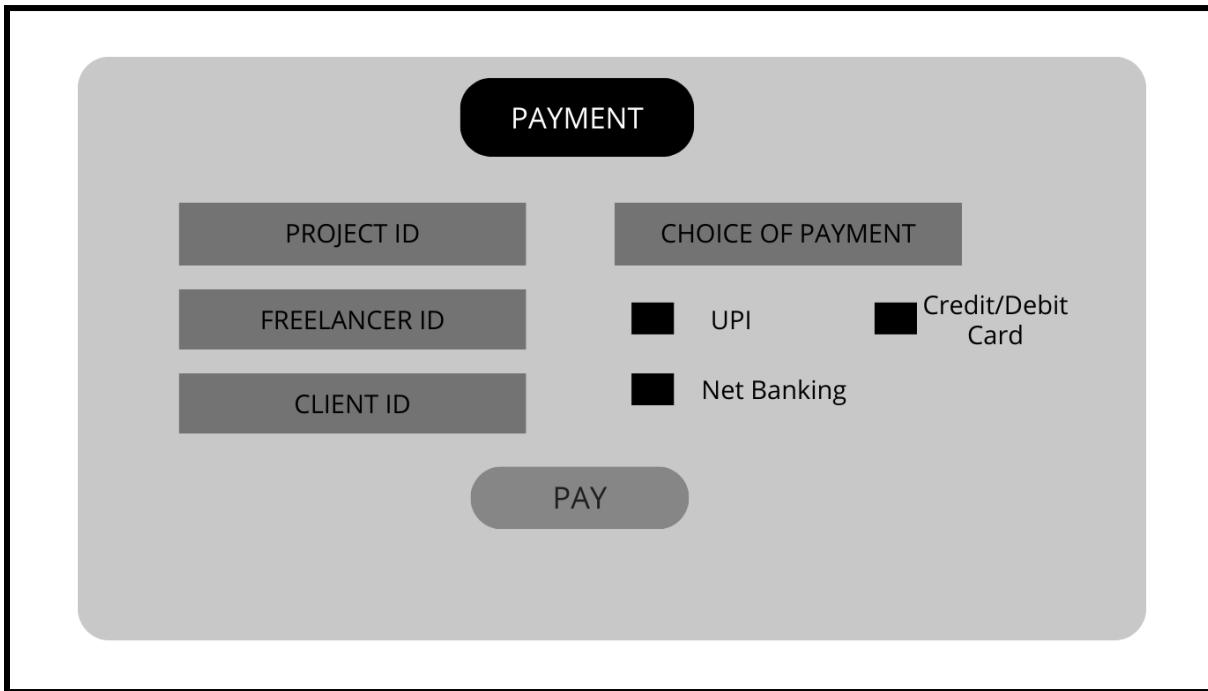
> View available for each client, a client cannot see another client's project view.

## 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:

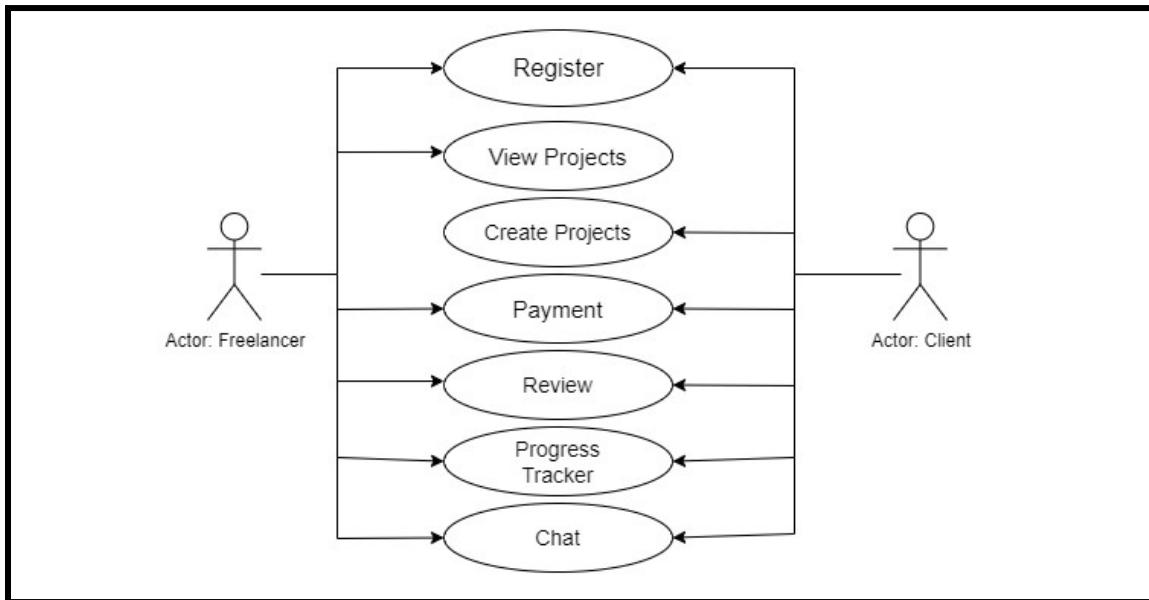
- A hosting provider like Heroku, Vercel is used to deploy the frontend and backend.
- Deployment pipelines to deploy the frontend and backend code to the hosting platform.

### 3.3 Communications Interfaces

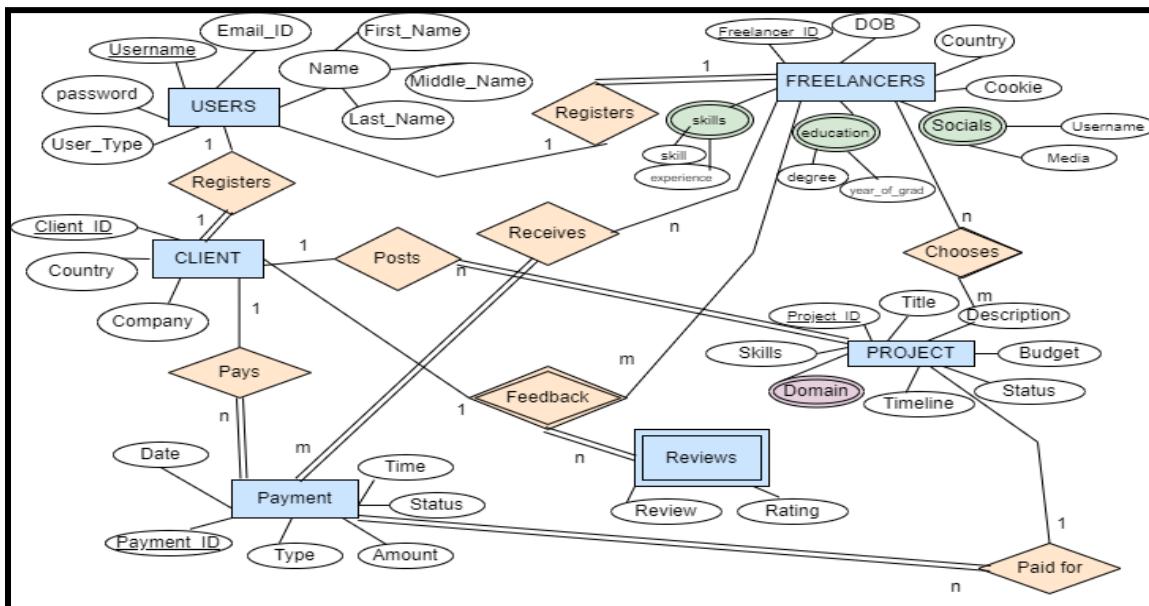
- HTTPS protocol will be used to ensure security and authentication of actions like payment .
- Fetch API library in NodeJS will be used to connect the frontend to backend database.
- Secure password storage and transfer by using hashed password in database.

## 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’ metre present in their page which displays their experience level.

## 5.7 Verify and Categorise 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 categorise 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 categorise 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<br>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 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<br>Year of graduation | 30<br>4      | alphanumeric year (YYYY)                            | -                                  | Can be multi-valued (multiple educational qualifications) |
| Skill set<br>Skill Experience            | 30<br>-      | string<br>ENUM('basic', 'intermediate', 'advanced') | -                                  | Can be multi-valued                                       |
| Social Profile<br>media_name<br>username | 30<br>30     | string<br>alphanumeric                              | Social media name and the username | 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 basic aim of the site.           | High   |  |  |  |  |
| 6.  | Review and Feedback on Freelancer                 | Freelancers are motivated by reviews and clients judge the freelancers based on the reviews.                         | Medium |  |  |  |  |
| 7.  | Verify and Categorise Freelancers Based on Rating | Clients must be assured of the freelancers expertise and a rating system based on their work fulfils 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    |  |  |  |  |

### **3. PROJECT PLANNING**

#### **Lifecycle Method**

The **incremental model** is used as the life cycle method for the Co-Lancer project.

The incremental model is beneficial for a freelancer collaboration website project because it allows for the early delivery of core features, enables quick response to changes in the dynamic freelance industry, facilitates continuous feedback from stakeholders, mitigates overall project risks, supports parallel development by different teams or freelancers, enhances client engagement through tangible progress, and streamlines testing and debugging processes. This approach ensures a flexible and adaptive development process, aligning the evolving project with user expectations and market needs.

#### **Features of incremental model**

**Early Delivery of Core Features:** With the incremental model, you can deliver essential features in the early stages of development. This allows users (freelancers and clients) to start using and benefiting from the system sooner rather than having to wait until the entire project is complete.

**Quick Response to Changes:** The freelance industry and collaboration requirements can evolve rapidly. The incremental model allows for flexibility and adaptability, making it easier to respond to changing market needs, user feedback, or emerging trends during the development process.

**Continuous Feedback:** Each increment is a deliverable version of the product. This allows stakeholders to provide feedback on the implemented features early in the development process. This iterative feedback loop can help ensure that the final product aligns closely with user expectations and requirements.

**Risk Management:** By breaking the project into smaller increments, the overall risk is reduced. If issues arise in one increment, they can be addressed before moving on to the next.

**Parallel Development:** Different parts of the system can be developed concurrently by different teams or individuals. This parallel development can accelerate the overall development process and is particularly useful in a collaborative environment where multiple freelancers may be contributing to various aspects of the project.

**Client Engagement:** Clients or end-users can see tangible progress with each increment, which can enhance client satisfaction and engagement. It also helps in building trust between the development team and clients as they can witness the project evolving in real-time.

**Easier Testing and Debugging:** Testing and debugging can be performed incrementally, making it easier to identify and fix issues in specific parts of the system. This approach streamlines the testing process and ensures a more reliable final product.

## **Iterations:**

### **Iteration 1:**

#### **Planning and initial requirements gathering:**

Define the high level overview of required features and objectives. Identify all the functional and non-functional features and formulate a SRS document that can benefit all stakeholders.

### **Iteration 2:**

#### **Design Phase:**

Design the ER diagram, convert ER Diagram to Relational Schema. Create Data Flow Diagrams (DFDs) along with Use Case diagrams and Class diagrams.

### **Iteration 3:**

#### **Database formation:**

Create a database schema based on the relational schema and grant necessary permissions to user roles based on the principle of Least Access Privileges.

### **Iteration 4:**

#### **Start project implementation**

- Create a user registration page along with the client and freelancer registrations. Also build the Login page.
- Include password hashing and security constraints and validation of all form inputs to avoid sql injections
- Do unit testing to ensure it works as expected

### **Iteration 5:**

#### **Create, View and Join Projects**

- Build the create project feature for clients and create the view and join project features for the freelancers.
- Search feature to search by domain of projects.
- Build a recap of completed and in progress projects and their monthly earnings.
- Unit testing

### **Iteration 6:**

#### **Payment Portal**

A payment portal for clients to pay compensation to the freelancer when finalising the project submitted by the freelancer.

## **Iteration 7:**

### **Chat feature**

- Option for freelancers to chat with other freelancers in near real-time when working on a collaborated project.
- Chat should be deleted once the project is submitted.
- Unit test all possibilities.

## **Iteration 8:**

### **Refinements and system testing:**

Make any refinements in UI, usability and features,

System testing for validation of all features.

## **Iteration 9:**

Add any additional features if required

Get user feedback and make changes.

## Tools to use in the entire lifecycle

### **Design:**

**Tool:** Canva, DrawIO

**Use:** Wireframe of user interfaces and design of ER diagrams and other design diagrams

### **Development**

**Tool:** Visual Studio Code, NeoVim or any IDE

**Use:** To write source code and perform any basic debugging and testing.

### **Version Control and Collaboration:**

**Tool:** Git and Github

**Use:** Seamless collaboration and versioning of codebase

### **Testing:**

**Tool:** Postman

**Use:** Test backend API calls through customised request frames.

### **Deployment:**

**Services:** Vercel, AWS RDS

**Use:** Database hosting and website deployment

## Deliverables categorised as reuse/build components

### **Reuse Components:**

#### **1. User Login**

- Categorising: Reuse component
- Justification: The User Login component is used across various websites aiming to personalise their site.

#### **2. Chat Feature**

- Categorising: Reuse component
- Justification: A chat feature is an integral part of multi user websites that require user interactions.

#### **3. Review Feature**

- Categorising: Reuse component
- Justification: Review feature is commonly used across various projects that provide services or products.

## **Build Components:**

### **1. Profile Pages**

- Categorising: Build Component
- Justification: The profile pages may vary across websites according to the services provided.

### **2. Monthly Recap**

- Categorising: Build Component
- Justification: A monthly recap feature is not frequently used and differs in terms of what is displayed and considered.

### **3. Payment Feature**

- Categorising: Build Component
- Justification: Payment gateway is specific to the project and API used.

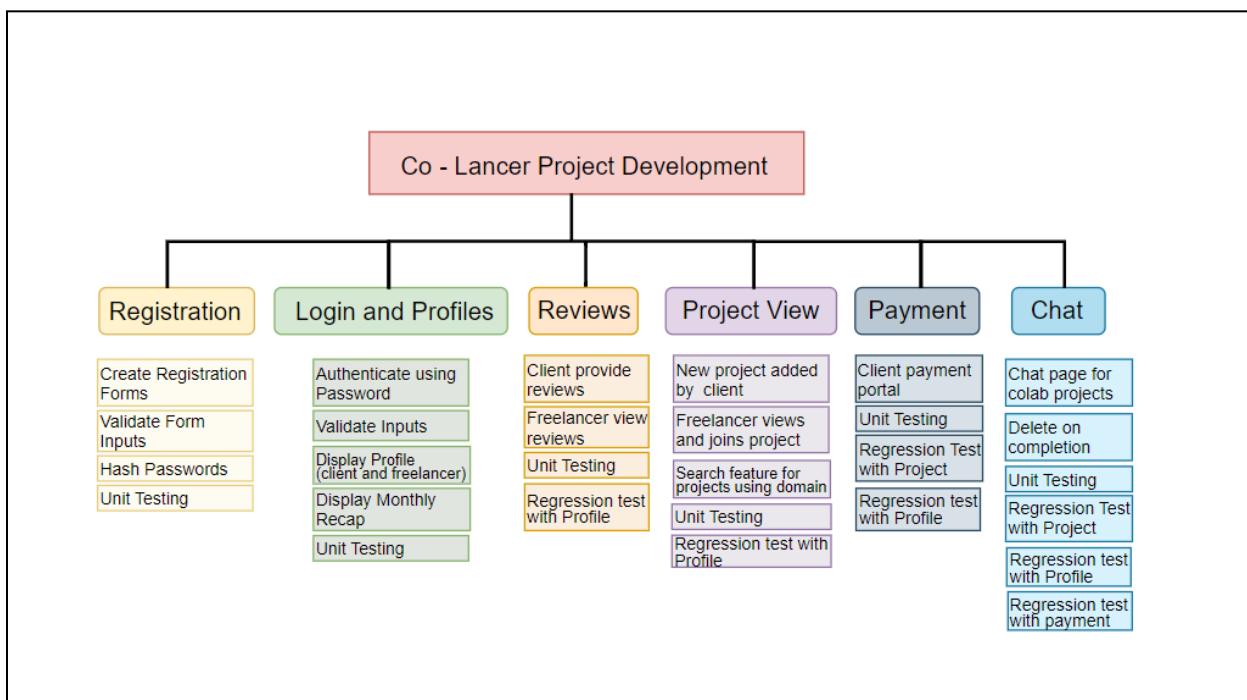
### **4. Project Display**

- Categorising: Build Component
- Justification: This feature is unique to this project and requires customised components

### **5. Project Creation**

- Categorising: Build Component
- Justification: Project creation page is limited to only a few applications. Hence this component requires additional features that have to be built.

## **WBS for functionalities**



## **Rough Estimate of effort to accomplish each task in person months**

### **1. Initial Planning and SRS formulation**

Participants: Pooja, Preethi

Time allocated: 2 weeks

Total: 1 Person Month

### **2. Design and architecture**

Participants: Pooja, Preethi

Time allocated: 1 Week

Total: 0.5 Person Month

### **3. Database schema creation and access rights**

Participants: Preethi

Time allocated: 1 week

Total: 0.25 Person Month

### **4. Registration**

Participants: Preethi

Time allocated: 2 weeks

Total: 0.5 Person Month

### **5. Login and Profile**

Participants: Preethi

Time allocated: 3 weeks

Total: 0.75 Person Months

### **6. Project View with Search feature**

Participants: Preethi

Time allocated: 2 weeks

Total: 0.5 Person Months

### **7. Create Project**

Participants: Preethi

Time allocated: 2 weeks

Total: 0.5 Person Months

## **8. Monthly Recap**

Participants: Pooja

Time allocated: 1 week

Total: 0.25 Person Months

## **9. Reviews**

Participants: Preethi

Time allocated: 1 week

Total: 0.25 Person Months

## **10. Payment**

Participants: Pooja

Time allocated: 1 week

Total: 0.25 Person Months

## **11. Chat**

Participants: Pooja

Time allocated: 2 weeks

Total: 0.5 Person Months

## **12. Frontend**

Participants: Pooja

Time allocated: 3 weeks

Total: 0.75 Person Months

## **13. Deploy**

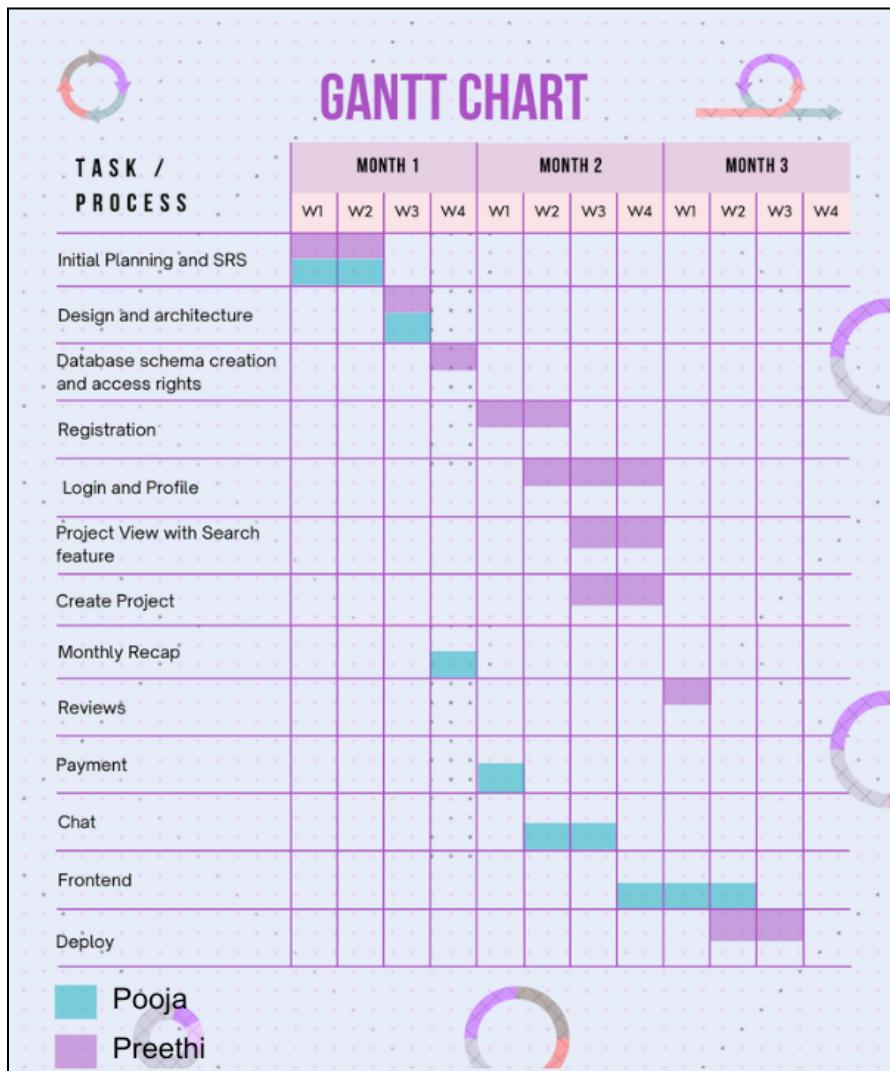
Participants: Preethi

Time allocated: 2 weeks

Total: 0.25 Person Months

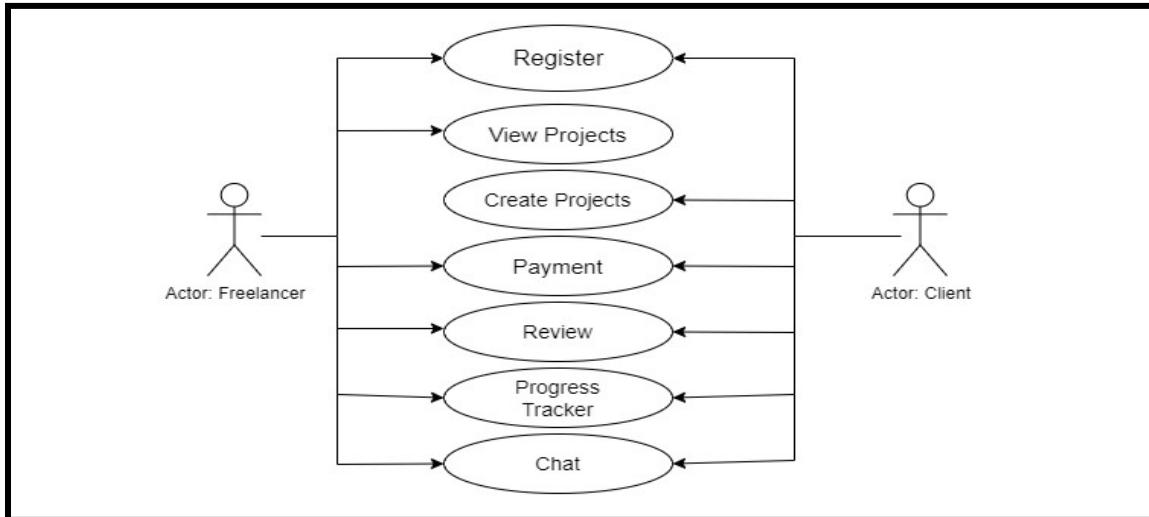
**TOTAL ESTIMATE OF EFFORT: 6.25 Person Months**

## Gantt Chart

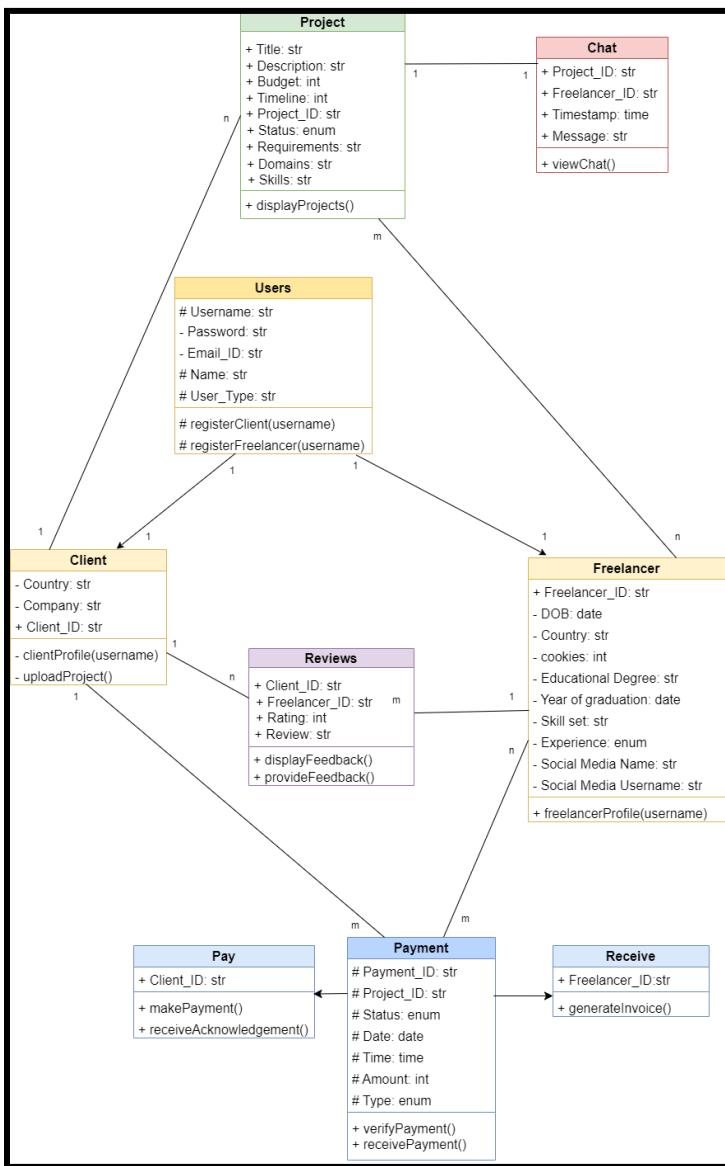


## 4. DESIGN DOCUMENT

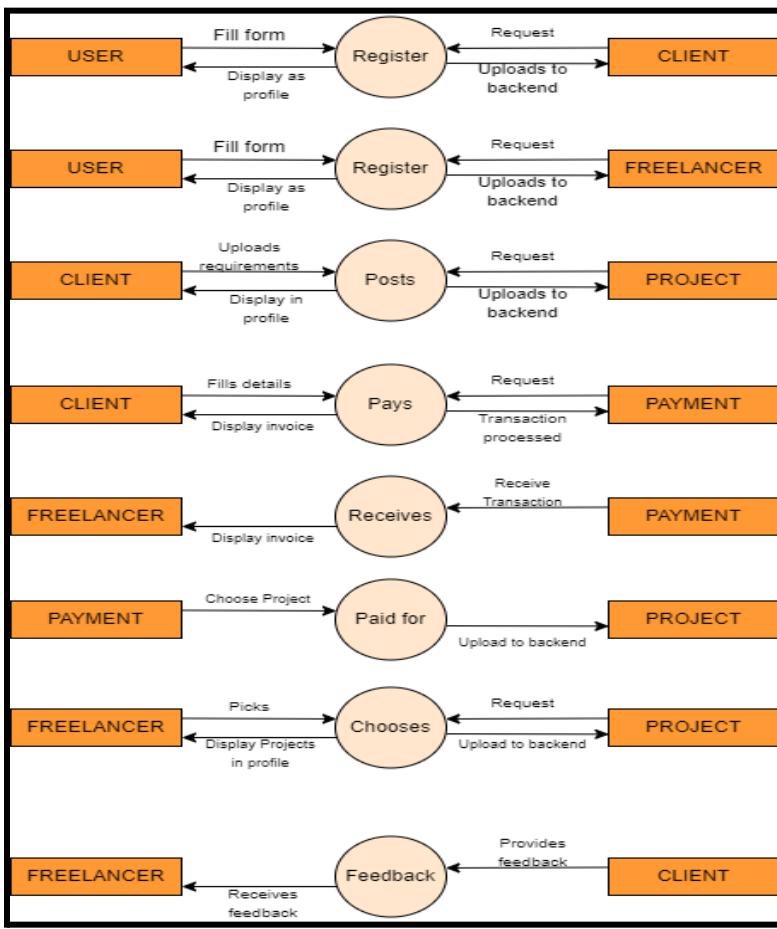
### 1. USE CASE DIAGRAM



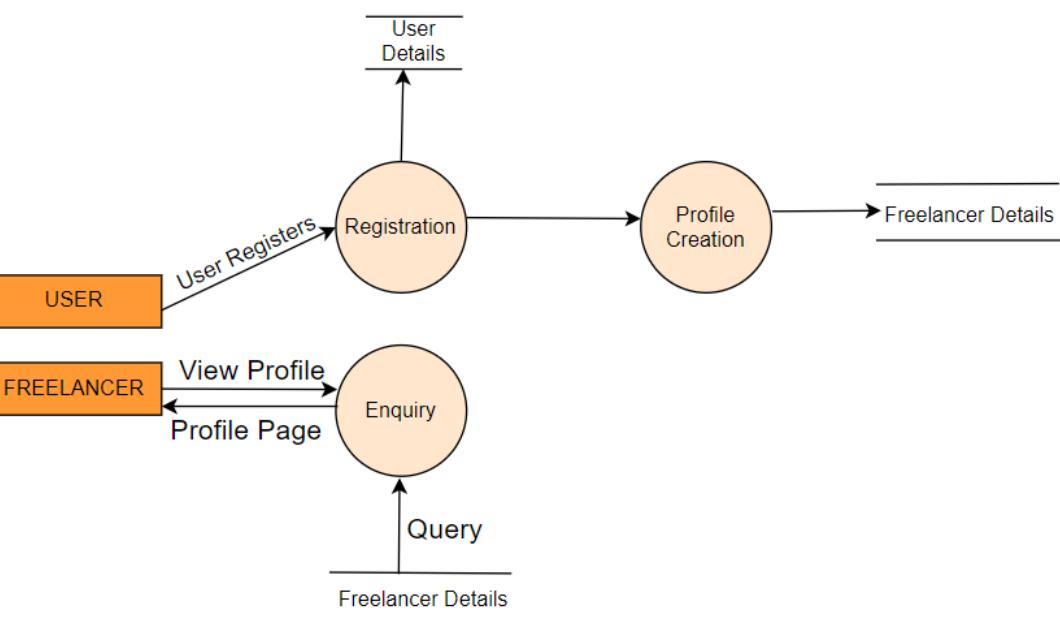
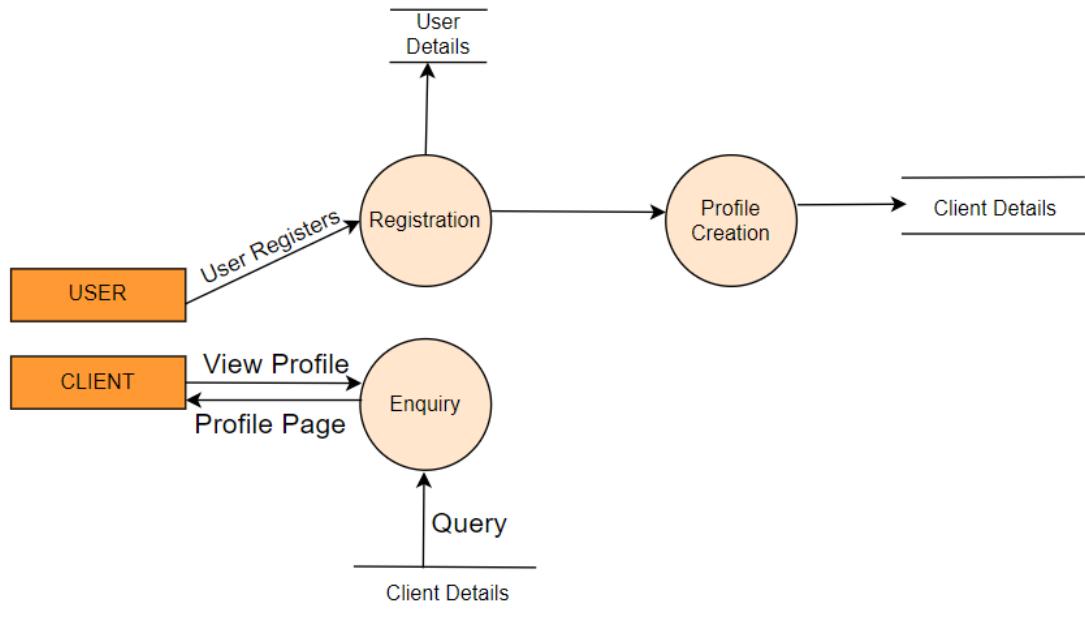
## 2. CLASS DIAGRAM:

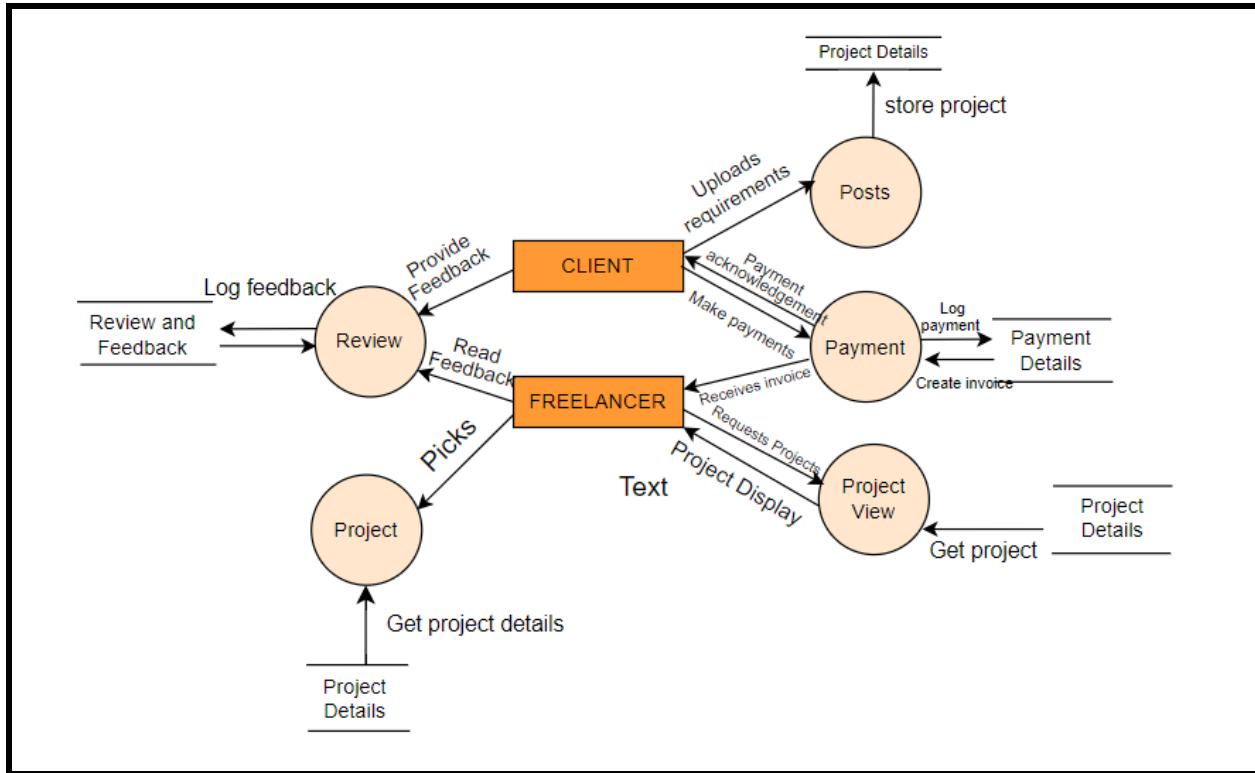


### 3. TOP LEVEL DATA FLOW DIAGRAM



#### 4. FIRST LEVEL DFD





## 5. ARCHITECTURAL STYLE INTEGRATION

**Client-server** architecture will be used in Co-Lancer.

### Features of client-server architecture:

**Centralised Data Management:** In this freelancer collaboration website, data of user profiles, project data, etc are collected. Centralised data storage on the server simplifies data management such that users can access consistent data securely. Enforcing data integrity, backup and security is made easier.

**Security:** The website deals with sensitive information, such as project details, financial transactions, and personal details. A server centralises security measures, through which implementation of authentication, authorization, encryption, and other security protocols occur.

**Collaboration and Real-Time Features:** The website offers features like messaging and project updates. The server facilitates real-time communication and collaboration by acting as a central hub where clients exchange data. This ensures synchronisation of messages and updates.

**Consistency and Reliability:** A server based architecture maintains consistency of user experience. Since clients access the same server, all users access the same data and features.

**Maintenance and Updates:** It is easier to update, fix bugs and add new features. A client-server architecture makes deployments easier. It allows for modification of the server independently without modifying the clients. It can ensure that all users have access to the latest version of the platform, reducing the need for users to update their individual platforms.

**Cross-Platform Compatibility:** Freelancers can use various devices and operating systems to access the website . A client-server architecture allows the users to have a consistent experience across any platform as the backend is accessed by the server so no specific client system platform is needed for it, making the website more accessible.

**Data Backup and Recovery:** A centralised server can be routinely backed up which makes it easier to recover the data in case of loss or system failures.

## 5. Test Cases

| Test Case ID | Name of Module    | Test Case Description         | Precondition      | Test Steps  | Test Data  | Expected Results           | Actual Result              | Test Result |
|--------------|-------------------|-------------------------------|-------------------|---|--|----------------------------|----------------------------|-------------|
| UT_01        | User Registration | To test registration of users | Access to browser | 1. Navigate to <a href="https://co-lancer-frontend.vercel.app/">https://co-lancer-frontend.vercel.app/</a><br>2. Choose register option | Name: Ajay Mathew<br>Email: ajay@gmail.com<br>Username: user10<br>Password: user10P@ss<br>Freelancer | Username or email is taken | Username or email is taken | PASS        |
| UT_02        | User Registration | To test registration of users | Access to browser | 1. Navigate to <a href="https://co-lancer-frontend.vercel.app/">https://co-lancer-frontend.vercel.app/</a><br>2. Choose register option | Name: Shaun S<br>Email: shaun@@g mail.com<br>Username: user11<br>Password: user11P@ss<br>Freelancer  | Invalid inputs             | Invalid inputs             | PASS        |
| UT_03        | User Registration | To test registration of users | Access to browser | 1. Navigate to <a href="https://co-lancer-frontend.vercel.app/">https://co-lancer-frontend.vercel.app/</a>                              | Name: Kiran M  | Successfully registered    | Successfull y registered   | PASS        |

|       |                         |                                   |  |  |  |  |  |      |
|-------|-------------------------|-----------------------------------|--|--|--|--|--|------|
|       |                         |                                   |  | 2. Choose register option  | Email:<br>kiran@gmail.com<br><br>Username: user11<br><br>Password: user11P@ss<br><br>Client  |  |  |      |
| UT_04 | Freelancer Registration | Registration page for freelancers | Access to browser and successful user registration | 1.Navigate to site and finish registration<br><br>2. Choose freelancer | DOB: 23/08/2003<br><br>Country: India<br><br>Education: 10th 2021<br>12th 2023<br>Btech 2027<br><br>Skills: c basic<br>html basic<br>Javascript intermediate<br><br>Socials: insta<br>insta_user | Registration successful, you will now be required to select a profile picture. | Registration successful, you will now be required to select a profile picture. | PASS |
| UT_05 | Freelancer Registration | Registration page for freelancers | Access to browser and successful user registration | 1.Navigate to site and finish registration<br><br>2. Choose freelancer | DOB: 26/08/2003<br><br>Country: India1<br><br>Education: 10th 2021<br>12th 2023<br>Btech 2027<br><br>Skills: c basic<br>java advance<br>Socials: insta<br>insta_user                             | Invalid country or username  | Registration successful, you will now be required to select a profile picture. | FAIL |

|       |                         |                                   |  |  |  |   |   |      |
|-------|-------------------------|-----------------------------------|--|--|--|---|---|------|
| UT_06 | Freelancer Registration | Registration page for freelancers | Access to browser and successful user registration | 1.Navigate to site and finish registration<br><br>2. Choose freelancer | Upload a image of size 3MB                 | Max size is 1MB   | File uploaded successfully . You will now be redirected to the home page!       | FAIL |
| UT_07 | Client Registration     | Registration page for client      | Access to browser and successful user registration | 1.Navigate to site and finish registration<br><br>2.Choose client      | Country: India<br><br>Company: Arista      | Registration successfully,you will now be required to select a profile picture. | Registration successfully,you will now be required to select a profile picture. | PASS |
| UT_08 | Client Registration     | Registration page for client      | Access to browser and successful user registration | 1.Navigate to site and finish registration<br><br>2.Choose client      | Choose a pdf file                          | Only png/jpg/jpeg allowed   | File uploaded successfully . You will now be redirected to the home page!       | FAIL |
| UT_09 | Client Registration     | Registration page for client      | Access to browser and successful user registration | 1.Navigate to site and finish registration<br><br>2.Choose client      | Choose a jpg file of size 1.6 kb           | File uploaded successfully. You will now be redirected to home page!            | File uploaded successfully . You will now be redirected to home page!           | PASS |
| UT_10 | Log In                  | Login for users                   | Access to browser and successful registration      | 1.Navigate to site<br><br>2. Choose Login                              | Username: user3<br><br>Password: user3P@ss | “Successfull y Logged In”   | “Successfull y Logged in”   | PASS |
| UT_11 | Log In                  | Login for users                   | Access to browser and successful registration      | 1.Navigate to site<br><br>2. Choose Login                              | Username: user5<br><br>Password: userPass  | “Username not found”  | “Username not found”  | PASS |

|       |                      |   |  |  |                                       |   |   |      |
|-------|----------------------|---|--|--|---------------------------------------|---|---|------|
| UT_12 | Log In               | Login for users                                       | Access to browser and successful registration        | 1.Navigate to site<br>2. Choose Login  | Username: user3<br>Password: userPass | “Wrong Password”  | “Wrong Password”                                | PASS |
| UT_13 | Freelancer Profile   | Profile display for freelancers                       | Access to browser, successful registration and login | 1.Navigate to site<br>2.Successful login   | NA                                    | Successful display of profile                           | Profile displayed successfully                  | PASS |
| UT_14 | Client Profile       | Profile display for clients                           | Access to browser, successful registration and login | 1.Navigate to site<br>2.Successful login   | NA                                    | Successful display of profile                           | Profile displayed successfully                  | PASS |
| UT_15 | Client Profile       | Profile display for clients                           | Access to browser, successful registration and login | 1.Navigate to site<br>2.Successful login   | NA                                    | Successful display of profile                           | Unable to connect to backend                    | FAIL |
| UT_16 | Reviews and Feedback | Freelancers and clients can view the reviews provided | Access to browser and successful login of freelancer | 1.Navigate to site and successful login<br>2. Move to freelancer profile and click on Reviews and Feedback | NA                                    | Successful display of all previous reviews and feedback | All reviews and feedback successfully displayed | PASS |
| UT_17 | Reviews and Feedback | Freelancers and clients can view the reviews provided | Access to browser and successful login of client     | 1.Navigate to site and successful login<br>2. Move to freelancer profile and click on Reviews and Feedback | NA                                    | Successful display of all previous reviews and feedback | Unable to retrieve the freelancer reviews       | FAIL |

|       |                               |  |  |  |   |   |   |      |
|-------|-------------------------------|--|--|--|---|---|---|------|
| UT_18 | Project Handoff by freelancer | Freelancers can provide github link for the client to access the completed project | Access to browser and successful login of freelancer | 1.Navigate to site and successful login<br>2. Move to freelancer profile and click on Handoff Project  | Link: <a href="https://github.com/user3/title3">https://github.com/user3/title3</a> | “Successful handoff of project”                         | “Couldn't handoff the project”                  | FAIL |
| UT_19 | Project Handoff by freelancer | Freelancers can provide github link for the client to access the completed project | Access to browser and successful login of freelancer | 1.Navigate to site and successful login<br>2. Move to freelancer profile and click on Handoff Project  | Link: <a href="https://github.com/user7/title6">https://github.com/user7/title6</a> | “Successful handoff of project”                         | “Couldn't handoff the project”                  | FAIL |
| UT_20 | Project Handoff by freelancer | Freelancers can provide github link for the client to access the completed project | Access to browser and successful login of freelancer | 1.Navigate to site and successful login<br>2. Move to freelancer profile and click on Handoff Project  | Link: <a href="https://github.com/user3/title3">https://github.com/user3/title3</a> | “Successful handoff of project”                         | “Successful handoff of project”                 | PASS |
| UT_21 | Explore Projects              | Freelancers can view a wide range of available projects                            | Access to browser and successful login of freelancer | 1.Navigate to site and successful login<br>2. Move to freelancer profile and click on Explore projects | Search: web dev   | Displays all available projects in the specified domain | Successfully displays all projects in “web dev” | PASS |
| UT_22 | Explore Projects              | Freelancers can view a wide range of available projects                            | Access to browser and successful login of freelancer | 1.Navigate to site and successful login<br>2. Move to freelancer profile and click on Explore projects | Search: ai/ml   | Displays all available projects in the specified domain | Couldn't retrieve data from backend             | FAIL |

|       |                          |  |   |  |  |   |                                     |      |
|-------|--------------------------|--|---|--|--|---|-------------------------------------|------|
| UT_23 | Explore Projects         | Freelancers can view a wide range of available projects  | Access to browser and successful login of freelancer  | 1.Navigate to site and successful login<br>2. Move to freelancer profile and click on Explore projects | Search: data science                       | Displays all available projects in the specified domain | Couldn't retrieve data from backend | FAIL |
| UT_24 | Joining Project          | Freelancers can join projects aligned to their skill set | Access to browser and successful login of freelancer  | 1.Navigate to site and successful login<br>2. Move to freelancer profile and click on Explore projects | Click "Join Now" on project "Title3"       | Successfully joined Title 3                             | Successfully joined Title 3         | PASS |
| UT_25 | Joining Project          | Freelancers can join projects aligned to their skill set | Access to browser and successful login of freelancer  | 1.Navigate to site and successful login<br>2. Move to freelancer profile and click on Explore projects | Click "Join Now" on project "Title4"       | Successfully joined Title 4                             | Successfully joined Title 4         | PASS |
| UT_26 | Project View             | Freelancers can view                                     | Access to browser and successful login of freelancer  | 1.Navigate to site and successful login<br>2. Move to explore projects and click on title              | Click on project Title3 to see the details | Successfully display the respective project details     | Project Details not displayed       | FAIL |
| UT_27 | Project View             | Freelancers can view                                     | Access to browser and successful login of freelancer  | 1.Navigate to site and successful login<br>2. Move to explore projects and click on title              | Click on project Title4 to see the details | Successfully display the respective project details     | Project Details displayed           | PASS |
| UT_28 | View project requirement | Freelancers can view the project requirements            | Access to browser and successful login of freelancer. | 1.Navigate to site and successful login  | Click on View PDF for Title3               | The project requirements is                             | PDF displayed                       | PASS |

|       |                          |   |   |  |                                 |  |   |      |
|-------|--------------------------|---|---|--|---------------------------------|--|---|------|
|       |                          |   | The chosen project must be accessible.  | 2. Choose project to view and click on View PDF  |                                 | displayed as a PDF   |   |      |
| UT_29 | View project requirement | Freelancers can view the project requirements | Access to browser and successful login of freelancer.<br>The chosen project must be accessible. | 1.Navigate to site and successful login<br><br>2. Choose project to view and click on View PDF | Click on View PDF for Title4    | The project requirements is displayed as a PDF   | PDF displayed   | PASS |
| UT_30 | Monthly Recap            | The monthly recap of a freelancer             | Access to browser and successful login of freelancer  | 1.Navigate to site and successful login<br><br>2.Chose the monthly recap option                | Select the monthly recap button | A pie displaying the completed and in progress projects.<br><br>The number of projects worked on in that month and the amount earned | The pie chart was displayed.<br><br>The amount earned was not displayed           | FAIL |
| UT_31 | Monthly Recap            | The monthly recap of a freelancer             | Access to browser and successful login of freelancer  | 1.Navigate to site and successful login<br><br>2.Chose the monthly recap option                | Select the monthly recap button | A pie displaying the completed and in progress projects.<br><br>The number of projects worked on in that month and the amount earned | The pie chart, total projects in a month and amount earned were rightly displayed | PASS |

|       |                  |  |   |  |   |  |  |      |
|-------|------------------|--|---|--|---|--|--|------|
| UT_32 | Chat feature     | Chat feature for freelancers working on the same project                               | <p>Access to browser and successful login of freelancer.</p> <p>The freelancer must be working on a collaborated project for this feature to be enabled</p> | <p>1.Navigate to site and successful login</p> <p>2.Choose the chat option</p>             | New message: Hello, this is user2                   | The old messages must appear, and the new message must display on the chat when sent | The old messages weren't displayed                             | FAIL |
| UT_33 | Chat feature     | Chat feature for freelancers working on the same project                               | <p>Access to browser and successful login of freelancer.</p> <p>The freelancer must be working on a collaborated project for this feature to be enabled</p> | <p>1.Navigate to site and successful login</p> <p>2.Choose the chat option</p>             | New message: Hello, this is user3                   | The old messages must appear, and the new message must display on the chat when sent | The old messages were loaded and the new message was displayed | PASS |
| UT_34 | Chat feature     | Chat feature for freelancers working on the same project                               | <p>Access to browser and successful login of freelancer.</p> <p>The freelancer must be working on a collaborated project for this feature to be enabled</p> | <p>1.Navigate to site and successful login</p> <p>2.Choose the chat option</p>             | New message: Welcome to the project                 | The old messages must appear, and the new message must display on the chat when sent | The old messages were loaded and the new message was displayed | PASS |
| UT_35 | Finalize Project | This feature for the clients is to accept the completed project and pay the freelancer | <p>Access to browser and successful login of client.</p> <p>The freelancer must have handed the project.</p>  | <p>1.Navigate to site and successful login of client</p> <p>2.Navigate to profile page</p> | <p>Click the finalize project button for Title3</p> | <p>“Project marked as complete”</p> <p>Page is redirected to payments</p>            | <p>“Error”</p>   | FAIL |

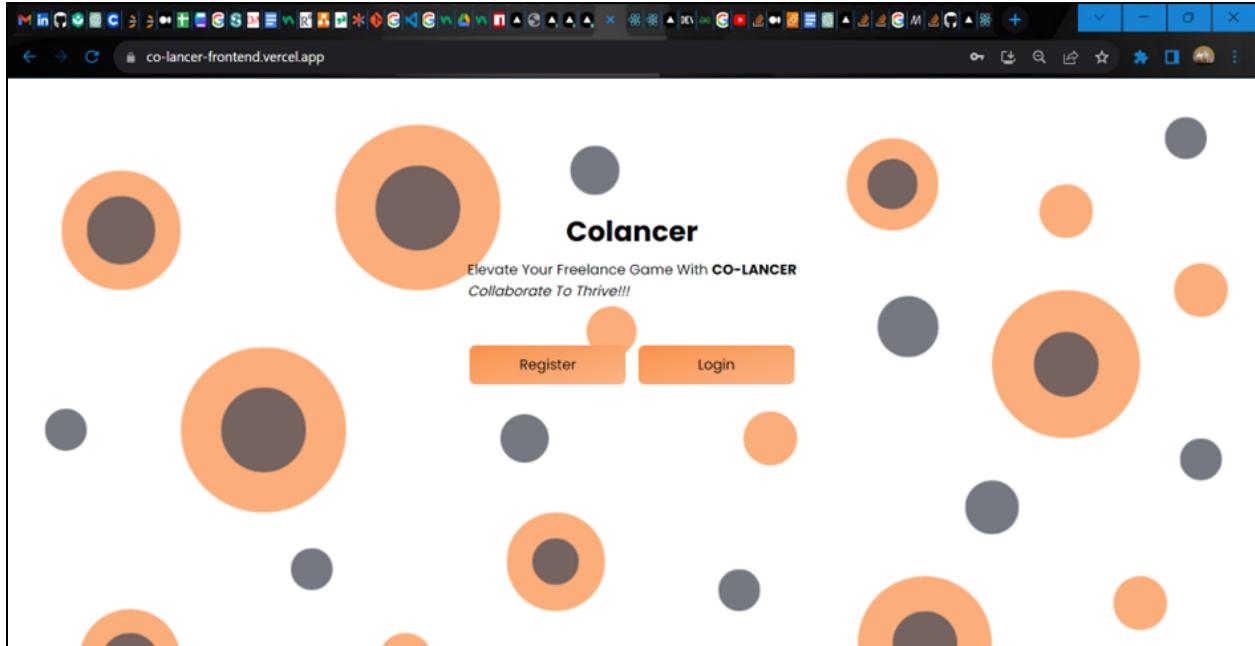
|       |                     |  |   |   |  |  |   |      |
|-------|---------------------|--|---|---|--|--|---|------|
| UT_36 | Finalize Project    | This feature for the clients is to accept the completed project and pay the freelancer               | Access to browser and successful login of client<br><br>The freelancer must have handed the project.  | 1.Navigate to site and successful login of client<br><br>2.Navigate to profile page | Click the finalize project button for Title4 | “Project marked as complete”<br><br>Page is redirected to payments | “Project marked as complete”<br><br>Failed to redirect page                         | FAIL |
| UT_37 | Finalize Project    | This feature for the clients is to accept the completed project and pay the freelancer               | Access to browser and successful login of client.<br><br>The freelancer must have handed the project. | 1.Navigate to site and successful login of client<br><br>2.Navigate to profile page | Click the finalize project button for Tile2  | “Project marked as complete”<br><br>Page is redirected to payments | “Project marked as complete”<br><br>Page is successfully redirected to payments     | PASS |
| UT_38 | Return For Revision | Unsatisfied clients can send back the completed project for revisions                                | Access to browser and successful login of client<br><br>The freelancer must have handed the project   | 1.Navigate to site and successful login of client<br><br>2.Navigate to profile page |  |  |   |      |
| UT_39 | Feedback            | Clients can provide reviews and feedbacks to freelancers which will reflect on the freelancer’s page | Access to browser and successful login of client<br><br>The freelancer must have handed the project   | 1.Navigate to site and successful login of client<br><br>2.Navigate to profile page | “Good work! Cookie: 5”                       | The comment must be reflected in the freelancer’s profile          | The comment is reflected in the freelancer’s profile and the cookie is also updated | PASS |
| UT_40 | Feedback            | Clients can provide reviews and feedbacks to freelancers which will reflect on the                   | Access to browser and successful login of client<br><br>The freelancer must have                      | 1.Navigate to site and successful login of client<br><br>2.Navigate to profile page | “Clean codes...”                             | The comment must be reflected in the freelancer’s profile          | The comment is reflected in the freelancer’s profile.                               | PASS |

|       |                    |                                 |  |   |   |  |   |      |
|-------|--------------------|---------------------------------|--|---|---|--|---|------|
|       |                    | freelancer's page               | handed the project                               |   |   |  |   |      |
| UT_41 | Create New Project | Clients can upload new projects | Access to browser and successful login of client | 1.Navigate to site and successful login of client<br><br>2.Navigate to profile page | Title: Tile3<br><br>Description: Description 3<br><br>Domains: web dev<br><br>Skills: MERN stack<br><br>Budget: 10000<br><br>Timeline: 60 days<br><br>Collaboration: No           | The project must be accessible to freelancers  | The project is reflected in explore projects page   | PASS |
| UT_42 | Create New Project | Clients can upload new projects | Access to browser and successful login of client | 1.Navigate to site and successful login of client<br><br>2.Navigate to profile page | Title: Tile4<br><br>Description: Description 4<br><br>Domains: cyber sec<br><br>Skills: linux, python, cpp<br><br>Budget: 8000<br><br>Timeline: 10 days<br><br>Collaboration: Yes | The project must be accessible to freelancers as a collaborative project.<br><br>The chat feature must be enabled for this project | The project is reflected in explore projects page as a collaborated project and the chat is enabled | PASS |

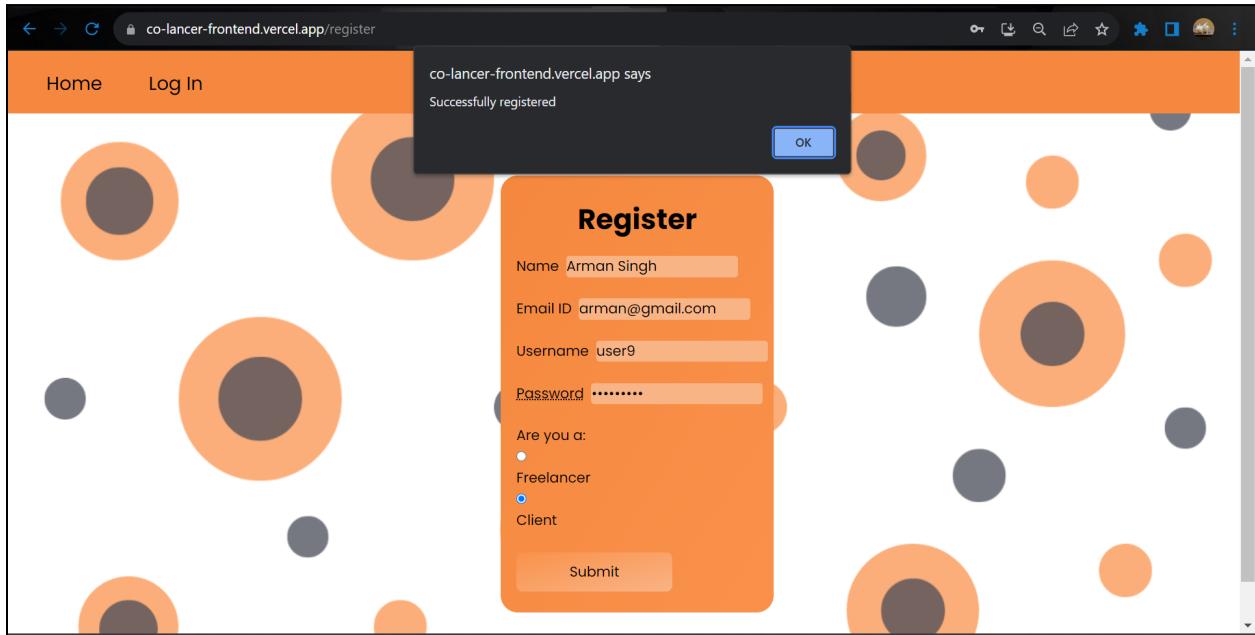
|       |          |  |  |   |  |                      |                      |      |
|-------|----------|--|--|---|--|----------------------|----------------------|------|
|       |          |  |  |   |  |                      |                      |      |
| UT_43 | Payments | Client pays the freelancer for a completed project | Access to browser and successful login of client.<br><br>The project must be marked complete | 1.Navigate to site and successful login of client<br><br>2.Navigate to profile page | Payment Mode: credit/debit card<br><br>Amount: 50000 | “Payment successful” | “Payment Failed”     | FAIL |
| UT_43 | Payments | Client pays the freelancer for a completed project | Access to browser and successful login of client.<br><br>The project must be marked complete | 1.Navigate to site and successful login of client<br><br>2.Navigate to profile page | Payment Mode: credit/debit card<br><br>Amount: 50000 | “Payment successful” | “Payment Successful” | PASS |

## 6. Screenshot of Output

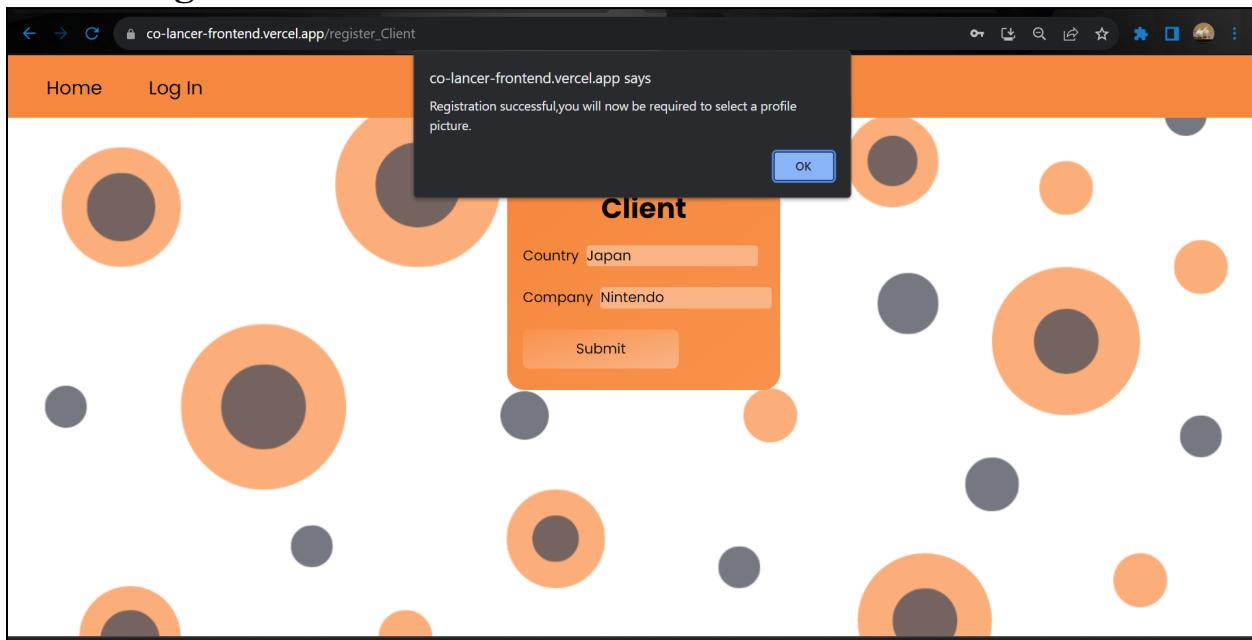
### Home Page:



### Registration:

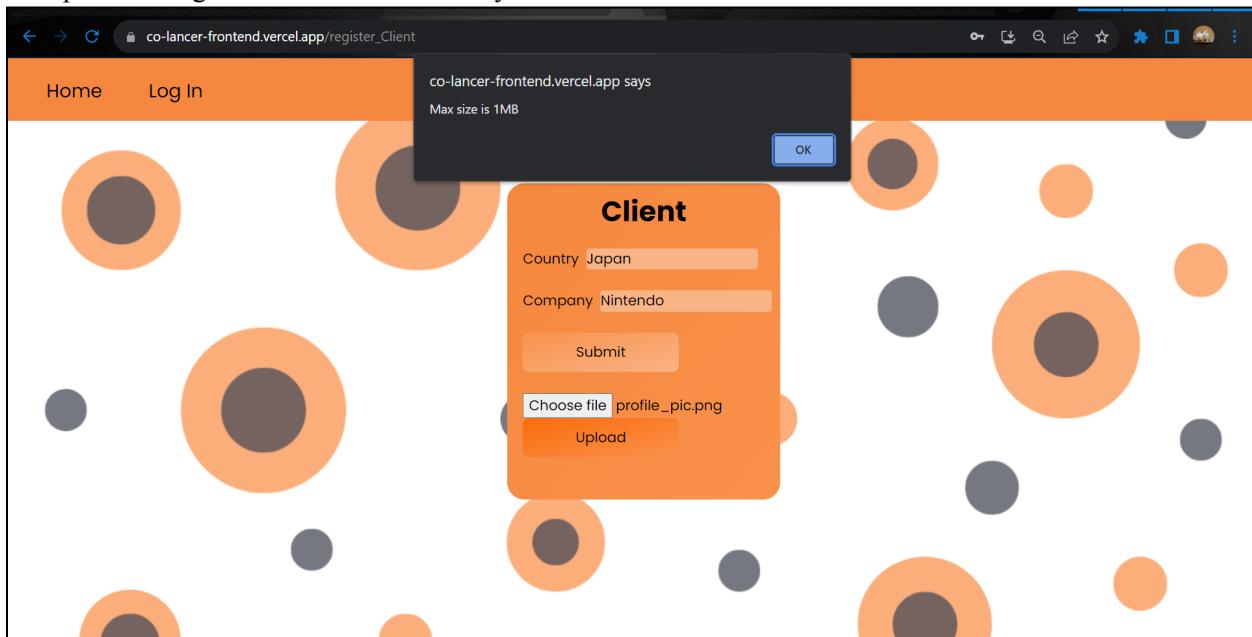


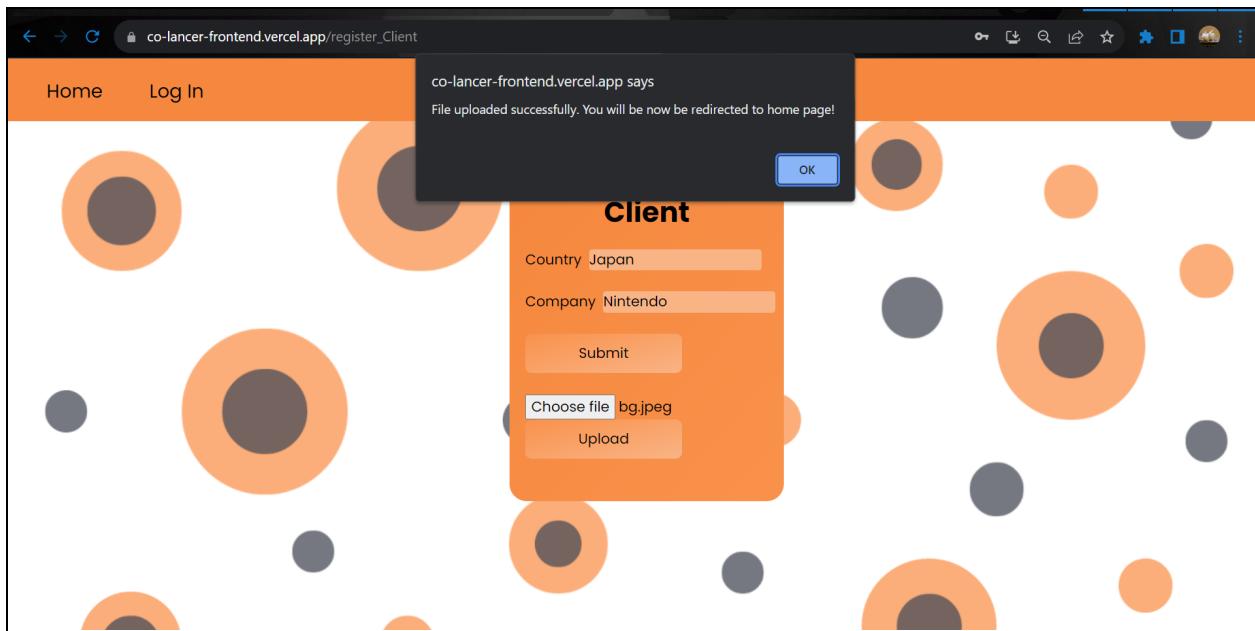
## Client Registration



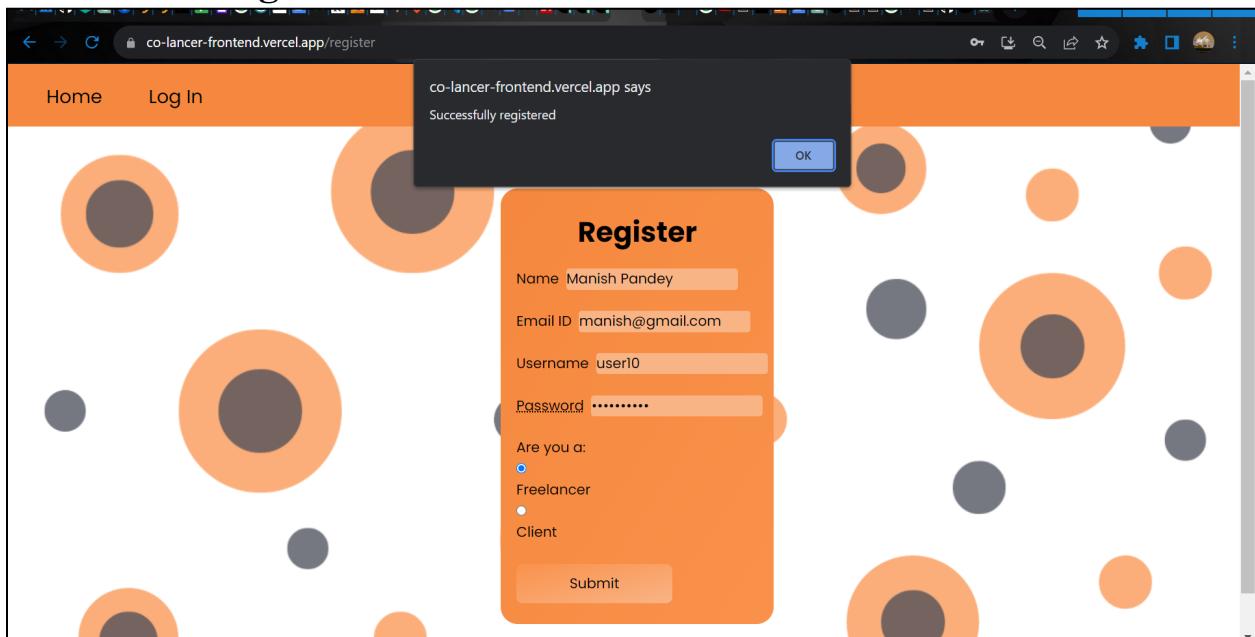
## Upload Profile

Max profile image size is 1MB for security reasons





## Freelancer Registration



## Freelancer

Date of Birth 11-06-2003

Country Russia

Education degree

|       |      |                                       |
|-------|------|---------------------------------------|
| 10th  | 2021 | <input type="button" value="Delete"/> |
| 12th  | 2023 | <input type="button" value="Delete"/> |
| btech | 2027 | <input type="button" value="Delete"/> |

Skill Set skill

|            |              |                                       |
|------------|--------------|---------------------------------------|
| c          | Basic        | <input type="button" value="Delete"/> |
| html       | Intermediate | <input type="button" value="Delete"/> |
| JavaScript | Advanced     | <input type="button" value="Delete"/> |

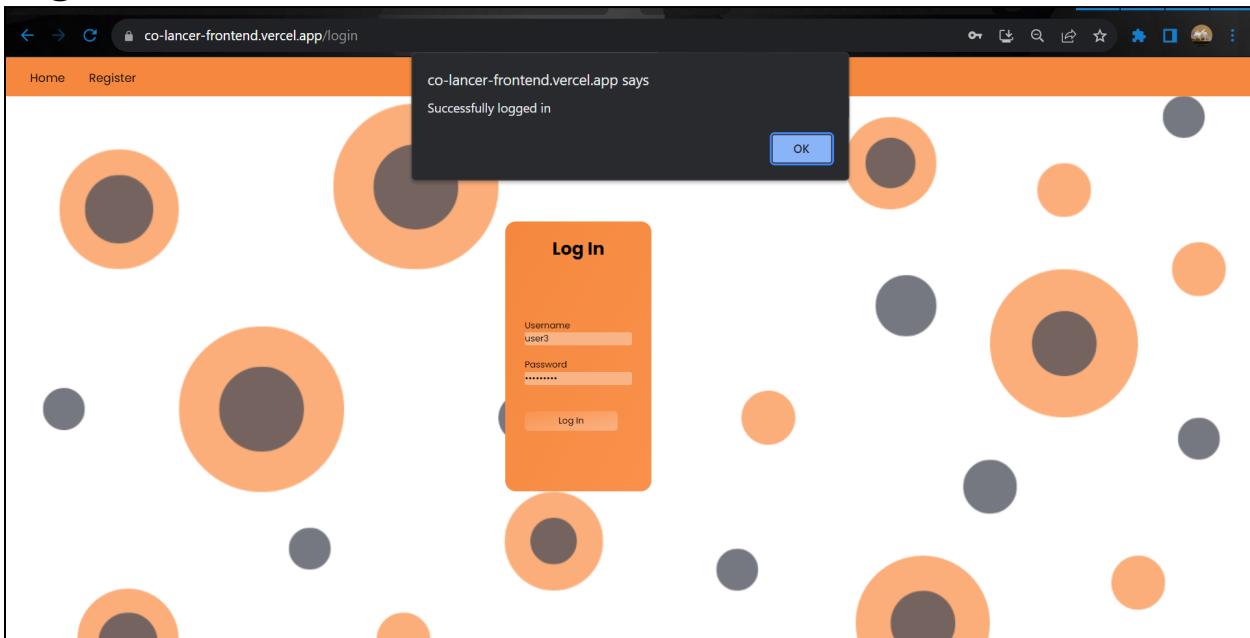
Social Profile media

userhandle

|      |     |                                       |
|------|-----|---------------------------------------|
| meta | abc | <input type="button" value="Delete"/> |
|------|-----|---------------------------------------|

bg.jpeg

## Login:



## Freelancer Profile:

A screenshot of a web browser window titled "co-lancer-frontend.vercel.app/Freelancer\_profile". The top navigation bar includes links for "Profile", "Explore", and "Log Out". On the left side, there is a sidebar for user "user3" showing their skill set (hadoop - Basic, aws - Basic, spark - Intermediate), social accounts (Instagram), and a "Cookie Jar" section with the number 17. Below the sidebar is a "Chat" button. The main content area is a large orange box containing sections for "Completed Projects" and "Current Projects". The "Completed Projects" section lists five items: "Title2", "Description2"; "Title4", "Description4"; "Title5", "Description5". The "Current Projects" section lists one item: "Title3", "Description3", with a "Handoff Project" button. At the bottom of the main content area are buttons for "Monthly Projects Recap", "Reviews and Feedback", and "Explore Projects". The background of the browser window features a pattern of orange and gray circles.

## Monthly Recap:

The screenshot shows the 'Monthly Recap' section of a web application. At the top, there are navigation links for 'Profile' and 'Explore' on the left and 'Log Out' on the right. Below this, there are two main sections: 'Monthly Recap' and 'Monthly Review November'.  
**Monthly Recap:** This section features a pie chart titled 'Projects Completed' with a legend indicating 'completed' (purple) and 'in-progress' (green). The chart shows approximately 75% completed and 25% in-progress.  
**Monthly Review November:** This section displays the following statistics:

- No. of projects: 4
- Monthly Earnings: 3000

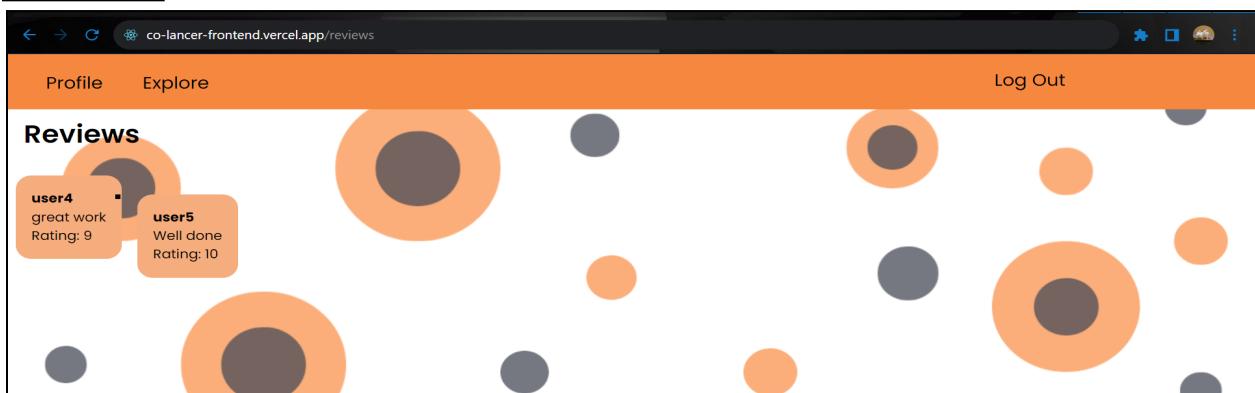
## Chat

The screenshot shows the 'Open Chat' interface. On the left, there is a sidebar with a 'Groups' section containing 'PID1' and 'PID5'. The main area is titled 'Chats' and shows a conversation between three users:

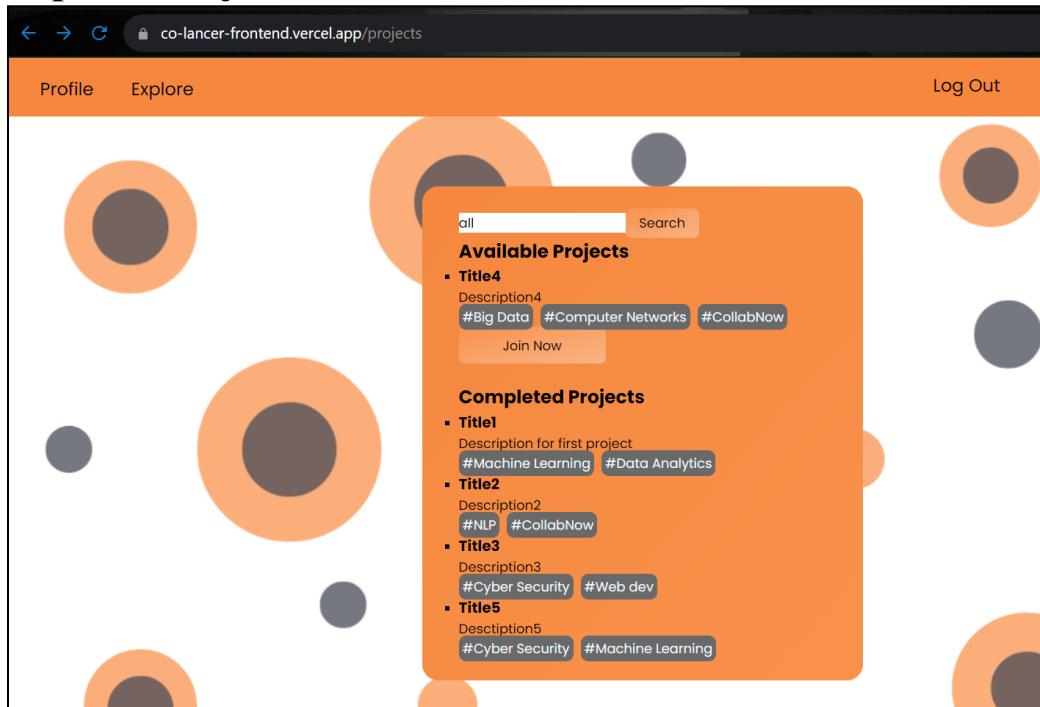
- user3:** hello this is user3  
2023-11-18T13:29:53.000Z
- user2:** user2  
hey this is user 2  
2023-11-18T13:30:16.000Z
- You:** hey everyone user1 here  
2023-11-18T13:30:47.000Z

A message input field at the bottom left contains 'Write message', and a green 'SEND' button is on the right.

## Reviews:



## Explore Projects:



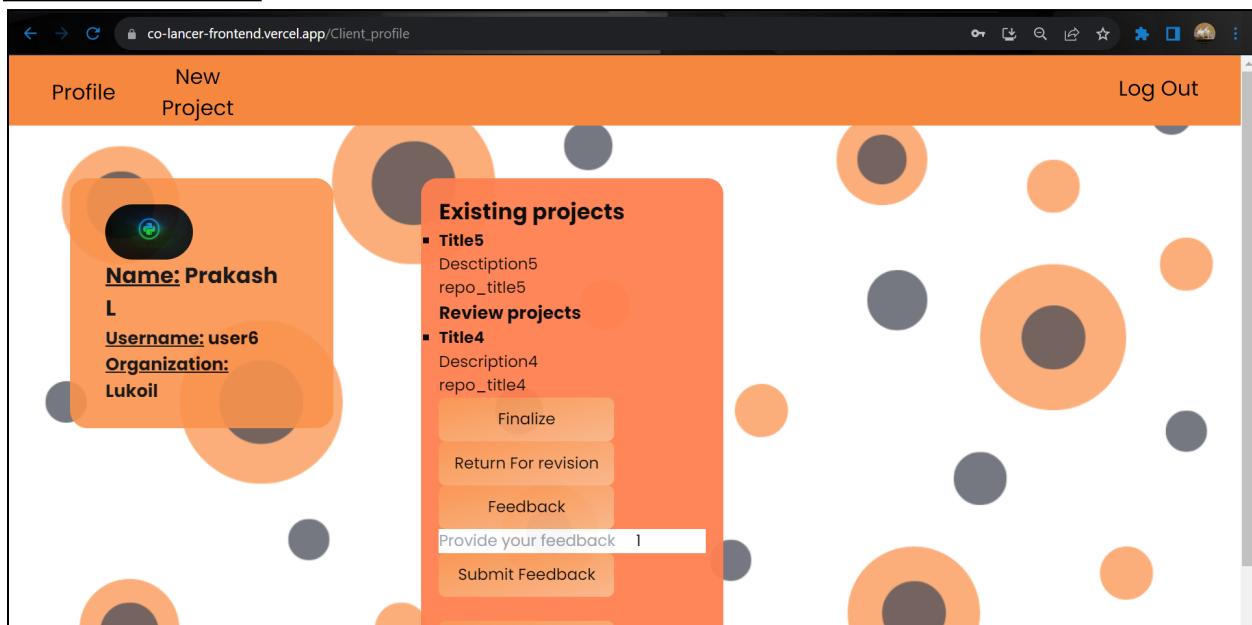
## Project Details:

A screenshot of a web browser displaying a project details page. The URL in the address bar is `co-lancer-frontend.vercel.app/project_details`. The page has a white background with several orange and grey circular icons scattered around. A central orange rectangular box contains the project information. At the top of this box is the title **Title6**, followed by a [View PDF](#) button. Below the title is the section **Description** with the text "Description 6". Under **Budget**, it says "2000". Under **Timeline**, it says "10". Under **Domains**, there is a list: "Cyber Security", "NLP", and "Machine Learning". Under **Skills**, there is a list: "c", "python", and "hadoop". Under **Freelancers**, there is a list: "FIDO", "FIDI", and "FID2".

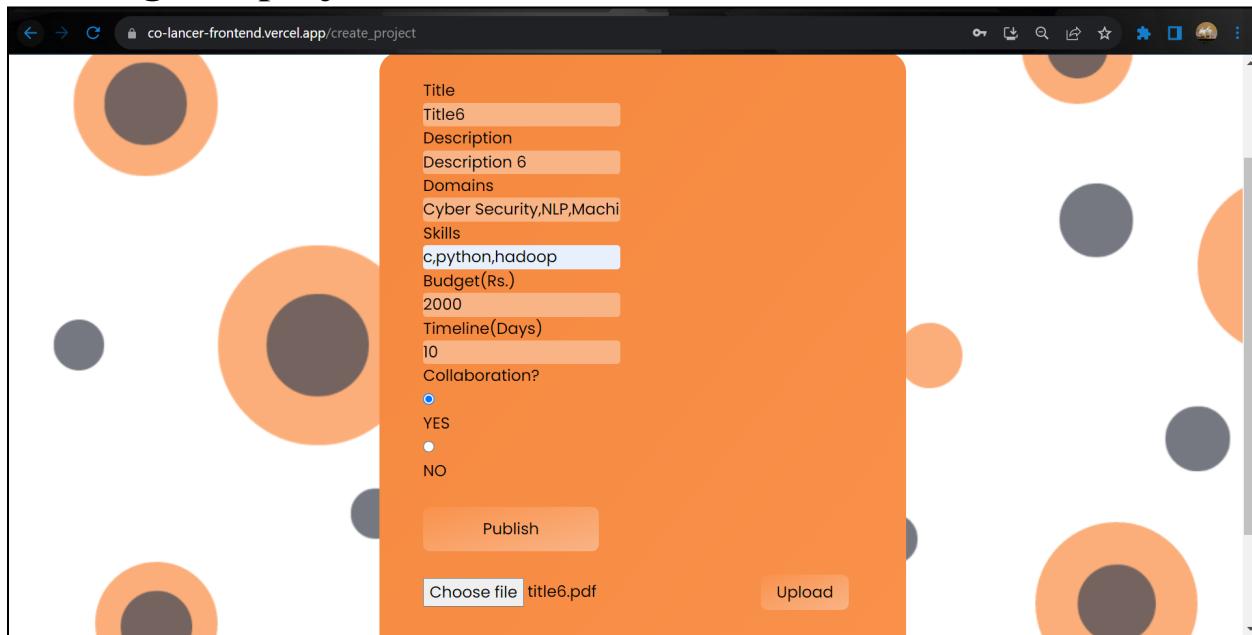
## PDF Display:

A screenshot of a web browser displaying a PDF document titled "PDF Display". The URL in the address bar is `co-lancer-frontend.vercel.app/project_pdf`. The PDF content is visible in the center of the screen. At the top left of the PDF area, there is a list of table of contents items: "Abstract", "ER diagram", "Relational Schema", "User requirement specification", "Table creation screenshot", "Relations and tuples screenshot", and "work completed". The rest of the PDF page is blank white space.

## Client Profile:



## Creating new project:



## Payment:

