

UE21CS351A: Database Management System

MINI PROJECT USER REQUIREMENT SPECIFICATION

Title: Co-Lancer

Prepared by

Name: Pooja Satheesh **SRN: PES2UG21CS374**

Name: Preethi M. **SRN:** PES2UG21CS396

PES UNIVERSITY, BANGALORE

Department of Computer Science and Engineering

08/10/2023

1. Abstract

The freelancer collaboration website Co-Lancer, serves as a networking platform where freelancers can collaborate on various projects. It is a one-stop shop for both freelancers and clients who require their services. Registered clients provide their project requirements, categorized by domain making it easier for freelancers to find projects aligned to their skill set. An integrated payment gateway ensures smooth transfer of compensation to freelancers once their designated project is over.

Clients can provide feedback in the form of reviews and rate the freelancers based on the work. This acts as an incentive and motivational factor for the freelancers and ensures that the clients can have the best people working on their project.

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.

2. Scope Of Project

Objectives:

Create a user-friendly platform for freelancers to connect and collaborate on projects.

Tech stack used:

ReactJS, NodeJS, ExpressJS, CSS, Tailwind, MySQL, Git (Version Control)

The key features of this project includes:

- 1) User registration and authentication
- 2) Uploading of Project
- 3) Tag Feature on Projects
- 4) Project Showcase Gallery for Freelancers
- 5) Collaboration feature for Freelancers
- 6) Chat feature for Freelancers
- 7) Review and Feedback facility
- 8) Reward system for freelancers
- 9) Payment
- 10) Progress tracking and Monthly Recap

3. Functional and Non-Functional Requirements

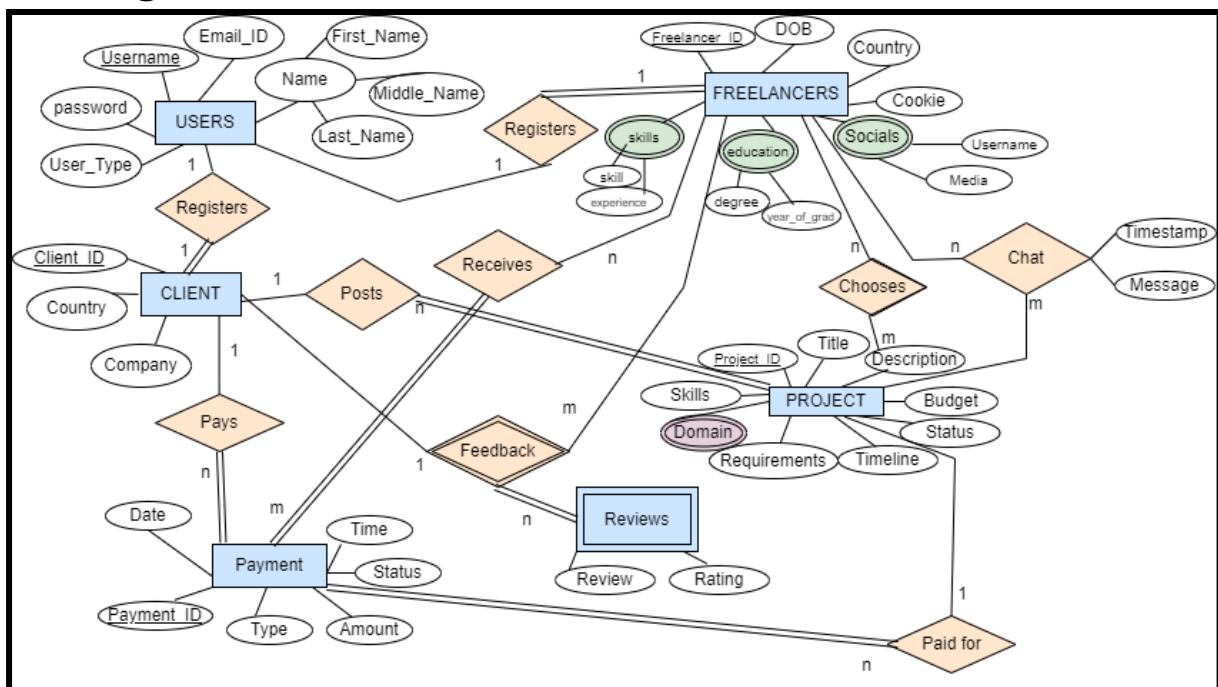
The above project has the following functional requirements

- User profile
- Freelancer can choose project
- Project showcase gallery for freelancer
- Client can upload requirements of project
- Collaboration of freelancers on projects
- Client can provide reviews on freelancer
- Verifying and categorize freelancer based on rating
- Payment options
- Chat option
- Progress tracking
- Monthly projects recap

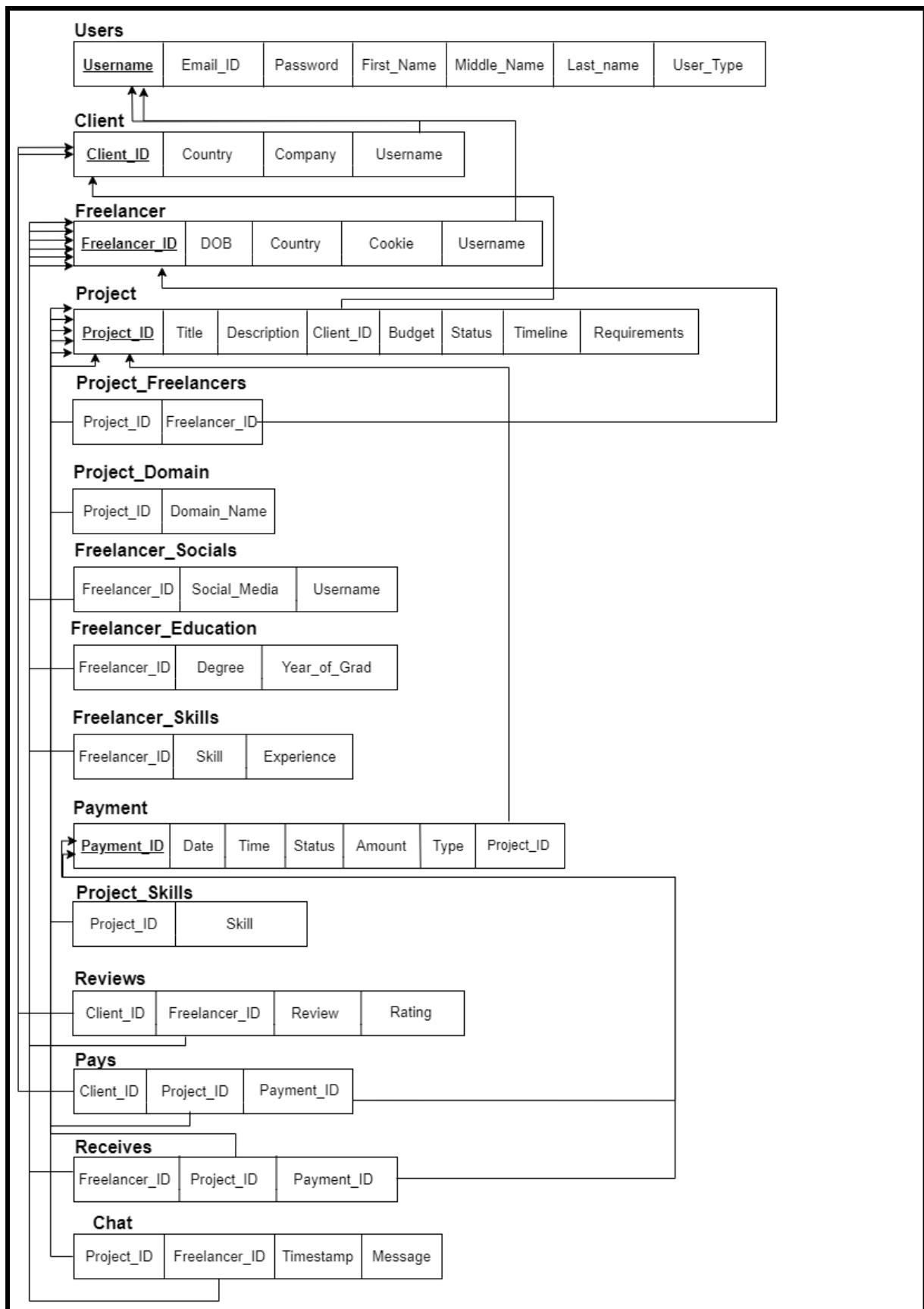
The non functional requirements of this website are:

- Performance Requirements: Concurrent multi-user support and faster retrieval of data due to usage of MySQL and tag features incorporated into the platform.
- Safety Requirements: Passwords are hashed and stored in the database. The chats of a project are deleted on completion of the project.
- Security Requirements: The chat messages are encrypted and the password must meet constraints such as length and combination of letters.
- Software Quality Attributes: Availability, Usability, Maintainability, Compatibility, Performance are provided in this platform.
- Business Rules: Users are registered as Freelancer or Client and their respective roles are well defined.

4. ER Diagram



5. Relational Schema



6. User Requirement Specification

- 1) User registration and authentication
 - Users of the website can register as Freelancer or Clients. The password provided is saved as hash in the database.
 - The databases involved are Users, Client, Freelancers, Freelancer_socials, Freelancer_education, Freelancer_skills
- 2) Uploading of Project
 - A project can be uploaded onto the site by the Clients with details such as Project title, description, requirements, and domain it belongs to.
 - The data collected here is stored in Project, Project_Domain, Project_Skills
- 3) Tag feature
 - The process of finding the apt project to work on is simplified by using a tag system. The projects are given tags of the domains they require freelancers in.
 - The data is stored in Project_Domain.
- 4) Project Showcase Gallery for Freelancers
 - Freelancers' previous and current work would be displayed on their user profile that can be viewed by all users.
 - This data is accessed from the following entities Project, Freelancer_Project
- 5) Collaboration feature for Freelancers
 - The main feature of this website is to foster easy collaboration amongst freelancers on various projects. Any project tagged #CollabNow informs freelancers that multiple freelancers can contribute.
 - The data of the freelancers working on this is saved in Project_Freelancers.
- 6) Chat feature for Freelancers
 - The above Collaboration functionality is enhanced with a chat feature built for each project with a feature to Collaborate. The data exchanged is deleted on project completion
 - The data of the chats is stored in the Chat table.
- 7) Review and Feedback facility
 - The work done by the freelancers can be rated and reviewed by clients they worked with.
 - The table used for this feature is Reviews.
- 8) Reward system for freelancers
 - A reward based system 'Cookie' is provided that represents their expertise.
 - The data is stored in the Reviews and Freelancer tables, which is used to calculate the portion of 'Cookie' provided to freelancers.
- 9) Payment
 - Freelancers are paid according to their contribution to a project. This feature ensures all the work is completed within the site.
 - The payment details are stored in Pay, Receives, Payment.
- 10) Progress tracking and Monthly Recap
 - A progress bar is used to inform Users, both freelancers and clients about the amount of work completed.
 - This is calculated from the data stored in the Project table.

7. Table Creation Screenshots:

Create Database:

```
Co-Lancer>create database co_lancer;
Query OK, 1 row affected (0.01 sec)
```

```
Co-Lancer>use co_lancer;
Database changed
Co-Lancer>show tables;
Empty set (0.02 sec)
```

Create table USERS:

```
Co-Lancer>CREATE TABLE users(username VARCHAR(30) PRIMARY KEY, email_ID VARCHAR(50) NOT NULL UNIQUE , password VARCHAR(30) NOT NULL, first_name VARCHAR(30) NOT NULL, middle_name VARCHAR(30), last_name VARCHAR(30), user_type ENUM('Client', 'Freelancer') NOT NULL);
Query OK, 0 rows affected (0.02 sec)
```

Create table CLIENT:

```
Co-Lancer>CREATE TABLE client(client_ID VARCHAR(10) PRIMARY KEY, country VARCHAR(30), company VARCHAR(30) NOT NULL, username VARCHAR(50), FOREIGN KEY (username) REFERENCES users(username));
Query OK, 0 rows affected (0.02 sec)
```

Create table FREELANCER:

```
Co-Lancer>CREATE TABLE freelancer(freelancer_ID VARCHAR(10) PRIMARY KEY, DOB DATE, country VARCHAR(30), cookies INT DEFAULT 0,username VARCHAR(50), FOREIGN KEY (username) REFERENCES users(username));
Query OK, 0 rows affected (0.02 sec)
```

Create table PROJECT:

```
Co-Lancer>CREATE TABLE project(project_ID VARCHAR(10) PRIMARY KEY, title VARCHAR(30) NOT NULL UNIQUE, description VARCHAR(100), budget INT, status ENUM('Not Assigned', 'In Progress', 'Completed') NOT NULL, timeline INT, client_ID VARCHAR(10), FOREIGN KEY (client_ID) REFERENCES client(client_ID));
Query OK, 0 rows affected (0.02 sec)
```

Create table PROJECT-FREELANCERS:

```
Co-Lancer>CREATE TABLE project_freelancers(project_ID VARCHAR(10), freelancer_id VARCHAR(10), FOREIGN KEY (project_ID) references project(project_ID), FOREIGN KEY (freelancer_ID) REFERENCES freelancer(freelancer_ID));
Query OK, 0 rows affected (0.02 sec)
```

Create table PROJECT-DOMAINS:

```
Co-Lancer>CREATE TABLE project_domains(project_ID VARCHAR(10), domain_name VARCHAR(30), FOREIGN KEY (project_ID) references project(project_ID));
Query OK, 0 rows affected (0.02 sec)
```

Create table PROJECT-SKILLS:

```
Co-Lancer>CREATE TABLE project_skills(project_ID VARCHAR(10), skill VARCHAR(30), FOREIGN KEY (project_ID) references project(project_ID));
Query OK, 0 rows affected (0.02 sec)
```

Create table FREELANCER-SOCIALS:

```
Co-Lancer>CREATE TABLE freelancer_socials(freelancer_ID VARCHAR(10), media_name VARCHAR(30), username VARCHAR(30), FOREIGN KEY (freelancer_ID) references freelancer(freelancer_ID));
Query OK, 0 rows affected (0.02 sec)
```

Create table FREELANCER-SKILLS:

```
Co-Lancer>CREATE TABLE freelancer_skills(freelancer_ID VARCHAR(10), skill VARCHAR(30), experience ENUM('Basic', 'Intermediate', 'Advanced'), FOREIGN KEY (freelancer_ID) references freelancer(freelancer_ID));
Query OK, 0 rows affected (0.02 sec)
```

Create table FREELANCER-EDUCATIONS:

```
Co-Lancer>CREATE TABLE freelancer_educations(freelancer_ID VARCHAR(10), degree VARCHAR(30), year_of_grad YEAR, FOREIGN KEY (freelancer_ID) references freelancer(freelancer_ID));
Query OK, 0 rows affected (0.02 sec)
```

Create table PAYMENT:

```
Co-Lancer>CREATE TABLE payment(payment_ID VARCHAR(10) PRIMARY KEY, project_ID VARCHAR(10), date DATE, time TIMESTAMP, status ENUM('Successful', 'Pending', 'Failed') NOT NULL, amount INT NOT NULL, type ENUM('upi', 'credit/debit card', 'netbanking') NOT NULL, FOREIGN KEY (project_ID) REFERENCES project(project_ID));
Query OK, 0 rows affected (0.01 sec)
```

Create table PAYS:

```
Co-Lancer>CREATE TABLE pays(client_ID VARCHAR(10), payment_ID VARCHAR(10), project_ID VARCHAR(10), FOREIGN KEY (client_ID) REFERENCES client(client_ID), FOREIGN KEY (payment_ID) REFERENCES payment(payment_ID), FOREIGN KEY (project_ID) REFERENCES project(project_ID));
Query OK, 0 rows affected (0.02 sec)
```

Create table RECEIVES:

```
Co-Lancer>CREATE TABLE receives(freelancer_ID VARCHAR(10), payment_ID VARCHAR(10), project_ID VARCHAR(10), FOREIGN KEY (freelancer_ID) REFERENCES freelancer(freelancer_ID), FOREIGN KEY (payment_ID) REFERENCES payment(payment_ID), FOREIGN KEY (project_ID) REFERENCES project(project_ID));
Query OK, 0 rows affected (0.02 sec)
```

Create table REVIEWS:

```
Co-Lancer>CREATE TABLE reviews(client_ID VARCHAR(10), freelancer_ID VARCHAR(10), review VARCHAR(100), rating ENUM('1', '2', '3', '4', '5', '6', '7', '8', '9', '10'), FOREIGN KEY (client_ID) REFERENCES client(client_ID), FOREIGN KEY (freelancer_ID) REFERENCES freelancer(freelancer_ID));
Query OK, 0 rows affected (0.02 sec)
```

Create table CHAT:

```
Co-Lancer>CREATE TABLE chat(project_ID VARCHAR(10), freelancer_ID VARCHAR(10), timestamp TIMESTAMP, message VARCHAR(300), FOREIGN KEY (project_ID) REFERENCES project(project_ID), FOREIGN KEY (freelancer_ID) REFERENCES freelancer(freelancer_ID));
Query OK, 0 rows affected (0.03 sec)
```

Show tables:

```
Co-Lancer>show tables;
+-----+
| Tables_in_co_lancer |
+-----+
| client
| freelancer
| freelancer_educations
| freelancer_skills
| freelancer_socials
| payment
| pays
| project
| project_domains
| project_freelancers
| project_skills
| receives
| reviews
| users
+-----+
14 rows in set (0.01 sec)
```

8. Relations and constraints, along with sample tuples:

(Actual tuple values will be inserted during the registration process from client application)

Users:

```
Co-Lancer>DESC users;
+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| username | varchar(30) | NO  | PRI  | NULL   |
| email_ID | varchar(50)  | NO  | UNI  | NULL   |
| password | varchar(30) | NO  |      | NULL   |
| first_name | varchar(30) | NO  |      | NULL   |
| middle_name | varchar(30) | YES |      | NULL   |
| last_name | varchar(30)  | YES |      | NULL   |
| user_type | enum('Client','Freelancer') | NO  |      | NULL   |
+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)

Co-Lancer>SELECT * FROM information_schema.table_constraints WHERE table_name='users';
+-----+-----+-----+-----+-----+-----+-----+
| CONSTRAINT_CATALOG | CONSTRAINT_SCHEMA | CONSTRAINT_NAME | TABLE_SCHEMA | TABLE_NAME | CONSTRAINT_TYPE | ENFORCED |
+-----+-----+-----+-----+-----+-----+-----+
| def               | co_lancer        | email_ID        | co_lancer     | users      | UNIQUE          | YES       |
| def               | co_lancer        | PRIMARY         | co_lancer     | users      | PRIMARY KEY    | YES       |
| def               | performance_schema | USER            | performance_schema | users | UNIQUE          | YES       |
+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

Co-Lancer>SELECT * FROM users;
+-----+-----+-----+-----+-----+-----+-----+
| username | email_ID      | password | first_name | middle_name | last_name | user_type |
+-----+-----+-----+-----+-----+-----+-----+
| user1    | abc@gmail.com | password1 | Preethi    | NULL       | M         | Freelancer |
| user2    | xyz@gmail.com | password2 | Pooja     | NULL       | Satheesh  | Freelancer |
| user3    | client1@gmail.com | password3 | Prajna    | A          | Shetty    | Client     |
+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

Client:

```
Co-Lancer>DESC client;
+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| client_ID | varchar(10) | NO  | PRI  | NULL   |
| country   | varchar(30)  | YES |      | NULL   |
| company   | varchar(30)  | NO  |      | NULL   |
| username  | varchar(50)  | YES | MUL  | NULL   |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

Co-Lancer>SELECT * FROM information_schema.table_constraints WHERE table_name='client';
+-----+-----+-----+-----+-----+-----+-----+
| CONSTRAINT_CATALOG | CONSTRAINT_SCHEMA | CONSTRAINT_NAME | TABLE_SCHEMA | TABLE_NAME | CONSTRAINT_TYPE | ENFORCED |
+-----+-----+-----+-----+-----+-----+-----+
| def               | co_lancer        | PRIMARY         | co_lancer     | client    | PRIMARY KEY    | YES       |
| def               | co_lancer        | client_ibfk_1  | co_lancer     | client    | FOREIGN KEY   | YES       |
+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

Co-Lancer>SELECT * FROM client;
+-----+-----+-----+-----+
| client_ID | country | company | username |
+-----+-----+-----+-----+
| CID001    | India   | Goldman Sachs | user3    |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Freelancer:

```
Co-Lancer>DESC freelancer;
+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| freelancer_ID | varchar(10) | NO   | PRI  | NULL    |
| DOB           | date      | YES  |      | NULL    |
| country       | varchar(30) | YES  |      | NULL    |
| cookies        | int       | YES  |      | 0       |
| username       | varchar(50) | YES  | MUL  | NULL    |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

Co-Lancer>SELECT * FROM information_schema.table_constraints WHERE table_name='freelancer';
+-----+-----+-----+-----+-----+-----+-----+
| CONSTRAINT_CATALOG | CONSTRAINT_SCHEMA | CONSTRAINT_NAME | TABLE_SCHEMA | TABLE_NAME | CONSTRAINT_TYPE | ENFORCED |
+-----+-----+-----+-----+-----+-----+-----+
| def               | co_lancer       | PRIMARY        | co_lancer    | freelancer  | PRIMARY KEY    | YES        |
| def               | co_lancer       | freelancer_ibfk_1 | co_lancer    | freelancer  | FOREIGN KEY    | YES        |
+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

```
Co-Lancer>SELECT * FROM freelancer;
+-----+-----+-----+-----+-----+
| freelancer_ID | DOB      | country | cookies | username |
+-----+-----+-----+-----+-----+
| FID001        | 2003-09-10 | India   | 1        | user1    |
| FID002        | 2003-07-10 | Germany | 0        | user2    |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

Freelancer-Educations:

```
Co-Lancer>DESC freelancer_educations;
+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| freelancer_ID | varchar(10) | YES  | MUL  | NULL    |
| degree         | varchar(30) | YES  |      | NULL    |
| year_of_grad  | year      | YES  |      | NULL    |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

Co-Lancer>SELECT * FROM information_schema.table_constraints WHERE table_name='freelancer_educations';
+-----+-----+-----+-----+-----+-----+
| CONSTRAINT_CATALOG | CONSTRAINT_SCHEMA | CONSTRAINT_NAME | TABLE_SCHEMA | TABLE_NAME | CONSTRAINT_TYPE | ENFORCED |
+-----+-----+-----+-----+-----+-----+-----+
| def               | co_lancer       | freelancer_educations_ibfk_1 | co_lancer    | freelancer_educations | FOREIGN KEY    | YES        |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

```
Co-Lancer>SELECT * FROM freelancer_educations;
+-----+-----+-----+
| freelancer_ID | degree  | year_of_grad |
+-----+-----+-----+
| FID001        | 10th    | 2018          |
| FID001        | 12th    | 2020          |
| FID001        | B.Tech  | 2024          |
| FID002        | 10th    | 2016          |
| FID002        | 12th    | 2018          |
| FID002        | Bsc     | 2021          |
+-----+-----+-----+
6 rows in set (0.00 sec)
```

Freelancer-Skills:

```
Co-Lancer>DESC freelancer_skills;
+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| freelancer_ID | varchar(10) | YES  | MUL  | NULL    |
| skill          | varchar(30) | YES  |      | NULL    |
| experience     | enum('Basic','Intermediate','Advanced') | YES  |      | NULL    |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

Co-Lancer>SELECT * FROM information_schema.table_constraints WHERE table_name='freelancer_skills';
+-----+-----+-----+-----+-----+-----+
| CONSTRAINT_CATALOG | CONSTRAINT_SCHEMA | CONSTRAINT_NAME | TABLE_SCHEMA | TABLE_NAME | CONSTRAINT_TYPE | ENFORCED |
+-----+-----+-----+-----+-----+-----+-----+
| def               | co_lancer       | freelancer_skills_ibfk_1 | co_lancer    | freelancer_skills | FOREIGN KEY    | YES        |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

```
Co-Lancer>SELECT * FROM freelancer_skills;
+-----+-----+-----+
| freelancer_ID | skill    | experience |
+-----+-----+-----+
| FID001        | Python   | Intermediate |
| FID001        | MERN Stack | Basic |
| FID001        | MySQL    | Advanced |
| FID002        | C++      | Basic |
| FID002        | C        | Intermediate |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

Freelancer-Socials:

```
Co-Lancer>DESC freelancer_socials;
+-----+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| freelancer_ID | varchar(10) | YES | MUL | NULL | 
| media_name    | varchar(30) | YES |      | NULL | 
| username      | varchar(30) | YES |      | NULL | 
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

Co-Lancer>SELECT * FROM information_schema.table_constraints WHERE table_name='freelancer_socials';
+-----+-----+-----+-----+-----+-----+-----+
| CONSTRAINT_CATALOG | CONSTRAINT_SCHEMA | CONSTRAINT_NAME      | TABLE_SCHEMA | TABLE_NAME      | CONSTRAINT_TYPE | ENFORCED |
+-----+-----+-----+-----+-----+-----+-----+
| def               | co_lancer        | freelancer_socials_ibfk_1 | co_lancer    | freelancer_socials | FOREIGN KEY   | YES      |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

```
Co-Lancer>SELECT * FROM freelancer_socials;
+-----+-----+-----+
| freelancer_ID | media_name | username |
+-----+-----+-----+
| FID001       | Instagram  | instauser1 |
| FID001       | Twitter    | twitteruser1 |
| FID002       | Twitter    | twitteruser2 |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

Project:

```
Co-Lancer>DESC project;
+-----+-----+-----+-----+-----+-----+
| Field     | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| project_ID | varchar(10)  | NO   | PRI | NULL    | 
| title      | varchar(30)  | NO   | UNI | NULL    | 
| description | varchar(100) | YES  |      | NULL    | 
| budget     | int           | YES  |      | NULL    | 
| status      | enum('Not Assigned','In Progress','Completed') | NO   |      | NULL    | 
| timeline    | int           | YES  |      | NULL    | 
| client_ID   | varchar(10)  | YES  | MUL | NULL    | 
| requirements | blob          | YES  |      | NULL    | 
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```

```
Co-Lancer>SELECT * FROM information_schema.table_constraints WHERE table_name='project';
+-----+-----+-----+-----+-----+-----+-----+
| CONSTRAINT_CATALOG | CONSTRAINT_SCHEMA | CONSTRAINT_NAME | TABLE_SCHEMA | TABLE_NAME | CONSTRAINT_TYPE | ENFORCED |
+-----+-----+-----+-----+-----+-----+-----+
| def               | co_lancer        | PRIMARY        | co_lancer    | project   | PRIMARY KEY   | YES      |
| def               | co_lancer        | title          | co_lancer    | project   | UNIQUE        | YES      |
| def               | co_lancer        | project_ibfk_1 | co_lancer    | project   | FOREIGN KEY  | YES      |
+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

```
Co-Lancer>select * from project;
+-----+-----+-----+-----+-----+-----+-----+
| project_ID | title      | description | budget    | status    | timeline | client_ID | requirements |
+-----+-----+-----+-----+-----+-----+-----+
| PID001     | project1  | this will be its description | 10000 | Completed | 100 | CID001 | NULL |
| PID002     | project2  | this description | 15000 | In Progress | 100 | CID001 | NULL |
+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

Project-Domains:

```
Co-Lancer>DESC project_domains;
+-----+-----+-----+-----+-----+-----+
| Field     | Type  | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| project_ID | varchar(10) | YES | MUL | NULL | 
| domain_name | varchar(30) | YES |      | NULL | 
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

Co-Lancer>SELECT * FROM information_schema.table_constraints WHERE table_name='project_domains';
+-----+-----+-----+-----+-----+-----+-----+
| CONSTRAINT_CATALOG | CONSTRAINT_SCHEMA | CONSTRAINT_NAME | TABLE_SCHEMA | TABLE_NAME | CONSTRAINT_TYPE | ENFORCED |
+-----+-----+-----+-----+-----+-----+-----+
| def               | co_lancer        | project_domains_ibfk_1 | co_lancer    | project_domains | FOREIGN KEY | YES      |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

```
Co-Lancer>SELECT * FROM project_domains;
+-----+-----+
| project_ID | domain_name |
+-----+-----+
| PID001     | WebDev      |
| PID002     | CyberSec    |
+-----+-----+
2 rows in set (0.00 sec)
```

Project-Skills:

```
Co-Lancer>DESC project_skills;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| project_ID | varchar(10) | YES | MUL | NULL    |
| skill       | varchar(30)  | YES |     | NULL    |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

Co-Lancer>SELECT * FROM information_schema.table_constraints WHERE table_name='project_skills';
+-----+-----+-----+-----+-----+-----+-----+-----+
| CONSTRAINT_CATALOG | CONSTRAINT_SCHEMA | CONSTRAINT_NAME      | TABLE_SCHEMA | TABLE_NAME      | CONSTRAINT_TYPE | ENFORCED |
+-----+-----+-----+-----+-----+-----+-----+
| def                | co_lancer        | project_skills_ibfk_1 | co_lancer    | project_skills  | FOREIGN KEY    | YES      |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

Co-Lancer>SELECT * FROM project_skills;
+-----+-----+
| project_ID | skill      |
+-----+-----+
| PID001     | Python     |
| PID001     | MERN Stack |
| PID002     | C++        |
| PID002     | MySQL      |
+-----+-----+
4 rows in set (0.00 sec)
```

Project-Freelancers:

```
Co-Lancer>DESC project_freelancers;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| project_ID | varchar(10) | YES | MUL | NULL    |
| freelancer_id | varchar(10) | YES | MUL | NULL    |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

Co-Lancer>SELECT * FROM information_schema.table_constraints WHERE table_name='project_freelancers';
+-----+-----+-----+-----+-----+-----+-----+
| CONSTRAINT_CATALOG | CONSTRAINT_SCHEMA | CONSTRAINT_NAME      | TABLE_SCHEMA | TABLE_NAME      | CONSTRAINT_TYPE | ENFORCED |
+-----+-----+-----+-----+-----+-----+-----+
| def                | co_lancer        | project_freelancers_ibfk_1 | co_lancer    | project_freelancers | FOREIGN KEY    | YES      |
| def                | co_lancer        | project_freelancers_ibfk_2 | co_lancer    | project_freelancers | FOREIGN KEY    | YES      |
+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

Co-Lancer>SELECT * FROM project_freelancers;
+-----+-----+
| project_ID | freelancer_id |
+-----+-----+
| PID001     | FID001      |
| PID002     | FID001      |
| PID002     | FID002      |
+-----+-----+
3 rows in set (0.00 sec)
```

Payment:

```
Co-Lancer>DESC payment;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| payment_ID | varchar(10)          | NO  | PRI | NULL    |
| project_ID  | varchar(10)          | YES | MUL | NULL    |
| date         | date                 | YES |     | NULL    |
| time         | timestamp            | YES |     | NULL    |
| status        | enum('Successful','Pending','Failed') | NO  |     | NULL    |
| amount        | int                  | NO  |     | NULL    |
| type          | enum('upi','credit/debit card','netbanking') | NO  |     | NULL    |
+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)

Co-Lancer>SELECT * FROM information_schema.table_constraints WHERE table_name='payment';
+-----+-----+-----+-----+-----+-----+-----+
| CONSTRAINT_CATALOG | CONSTRAINT_SCHEMA | CONSTRAINT_NAME      | TABLE_SCHEMA | TABLE_NAME      | CONSTRAINT_TYPE | ENFORCED |
+-----+-----+-----+-----+-----+-----+-----+
| def                | co_lancer        | PRIMARY             | co_lancer    | payment        | PRIMARY KEY    | YES      |
| def                | co_lancer        | payment_ibfk_1     | co_lancer    | payment        | FOREIGN KEY   | YES      |
+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

Co-Lancer>DESC pays;
Co-Lancer>SELECT * FROM payment;
+-----+-----+-----+-----+-----+-----+-----+
| payment_ID | date      | time           | status      | amount | type | project_id |
+-----+-----+-----+-----+-----+-----+-----+
| PayID001   | 2023-10-03 | 2023-10-03 07:03:30 | Successful | 10000  | upi  | NULL      |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Pays:

```
Co-Lancer>DESC pays;
+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| client_ID | varchar(10) | YES | MUL | NULL |
| payment_ID | varchar(10) | YES | MUL | NULL |
| project_ID | varchar(10) | YES | MUL | NULL |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

Co-Lancer>SELECT * FROM information_schema.table_constraints WHERE table_name='pays';
+-----+-----+-----+-----+-----+-----+-----+-----+
| CONSTRAINT_CATALOG | CONSTRAINT_SCHEMA | CONSTRAINT_NAME | TABLE_SCHEMA | TABLE_NAME | CONSTRAINT_TYPE | ENFORCED |
+-----+-----+-----+-----+-----+-----+-----+
| def | co_lancer | pays_ibfk_1 | co_lancer | pays | FOREIGN KEY | YES |
| def | co_lancer | pays_ibfk_2 | co_lancer | pays | FOREIGN KEY | YES |
| def | co_lancer | pays_ibfk_3 | co_lancer | pays | FOREIGN KEY | YES |
+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

```
Co-Lancer>SELECT * FROM pays;
+-----+-----+-----+
| client_ID | payment_ID | project_ID |
+-----+-----+-----+
| CID001 | PayID001 | PID001 |
+-----+-----+-----+
1 row in set (0.00 sec)
```

Receives:

```
Co-Lancer>DESC receives;
+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| freelancer_ID | varchar(10) | YES | MUL | NULL |
| payment_ID | varchar(10) | YES | MUL | NULL |
| project_ID | varchar(10) | YES | MUL | NULL |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

Co-Lancer>SELECT * FROM information_schema.table_constraints WHERE table_name='receives';
+-----+-----+-----+-----+-----+-----+-----+
| CONSTRAINT_CATALOG | CONSTRAINT_SCHEMA | CONSTRAINT_NAME | TABLE_SCHEMA | TABLE_NAME | CONSTRAINT_TYPE | ENFORCED |
+-----+-----+-----+-----+-----+-----+-----+
| def | co_lancer | receives_ibfk_1 | co_lancer | receives | FOREIGN KEY | YES |
| def | co_lancer | receives_ibfk_2 | co_lancer | receives | FOREIGN KEY | YES |
| def | co_lancer | receives_ibfk_3 | co_lancer | receives | FOREIGN KEY | YES |
+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

```
Co-Lancer>SELECT * FROM receives;
+-----+-----+-----+
| freelancer_ID | payment_ID | project_ID |
+-----+-----+-----+
| FID001 | PayID001 | PID001 |
+-----+-----+-----+
1 row in set (0.00 sec)
```

Reviews:

```
Co-Lancer>DESC reviews;
+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| client_ID | varchar(10) | YES | MUL | NULL |
| freelancer_ID | varchar(10) | YES | MUL | NULL |
| review | varchar(100) | YES | NO | NULL |
| rating | enum('1','2','3','4','5','6','7','8','9','10') | YES | NO | NULL |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

Co-Lancer>SELECT * FROM information_schema.table_constraints WHERE table_name='reviews';
+-----+-----+-----+-----+-----+-----+-----+
| CONSTRAINT_CATALOG | CONSTRAINT_SCHEMA | CONSTRAINT_NAME | TABLE_SCHEMA | TABLE_NAME | CONSTRAINT_TYPE | ENFORCED |
+-----+-----+-----+-----+-----+-----+-----+
| def | co_lancer | reviews_ibfk_1 | co_lancer | reviews | FOREIGN KEY | YES |
| def | co_lancer | reviews_ibfk_2 | co_lancer | reviews | FOREIGN KEY | YES |
+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

```
Co-Lancer>SELECT * FROM reviews;
+-----+-----+-----+-----+
| client_ID | freelancer_ID | review | rating |
+-----+-----+-----+-----+
| CID001 | FID001 | Amazing work, on time completion | 9 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Chat:

```
Co-Lancer>DESC chat;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| project_ID | varchar(10) | YES | MUL | NULL |
| freelancer_ID | varchar(10) | YES | MUL | NULL |
| timestamp | timestamp | YES |       | NULL |
| message | varchar(300) | YES |       | NULL |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

Co-Lancer>SELECT * FROM information_schema.table_constraints WHERE table_name='chat';
+-----+-----+-----+-----+-----+-----+-----+-----+
| CONSTRAINT_CATALOG | CONSTRAINT_SCHEMA | CONSTRAINT_NAME | TABLE_SCHEMA | TABLE_NAME | CONSTRAINT_TYPE | ENFORCED |
+-----+-----+-----+-----+-----+-----+-----+
| def      | co_lancer    | chat_ibfk_1     | co_lancer    | chat      | FOREIGN KEY | YES
| def      | co_lancer    | chat_ibfk_2     | co_lancer    | chat      | FOREIGN KEY | YES
| def      | colancer     | chat_ibfk_1     | colancer     | chat      | FOREIGN KEY | YES
| def      | colancer     | chat_ibfk_2     | colancer     | chat      | FOREIGN KEY | YES
+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

```
Co-Lancer>SELECT * FROM chat;
+-----+-----+-----+-----+
| project_ID | freelancer_ID | timestamp           | message          |
+-----+-----+-----+-----+
| PID002     | FID001        | 2023-10-07 03:14:07 | requirement 1 is completed |
| PID002     | FID002        | 2023-10-07 03:19:18 | working on requirement 3  |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

9. Work completed

- Table structures have been created on MySQL with all constraints in place.
- A basic frontend to take inputs from registration form has also been developed.
- The frontend is integrated with the backend to store the form inputs.