Mini project report on

Co - Lancer

Freelancer Collaboration Website

Submitted in partial fulfilment of the requirements for the award of degree of

Bachelor of Technology

in

Computer Science & Engineering

UE21CS351 – DBMS Project

Submitted by:

Pooja Satheesh	**PES2UG21CS374**
Preethi M	**PES2UG21CS396**

Under the guidance of

Prof. Shilpa S

Assistant Professor

Designation

PES University

AUG - DEC 2023

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

FACULTY OF ENGINEERING

PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)

Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, India

PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)

Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, India

CERTIFICATE

This is to certify that the mini project entitled

Co - Lancer

is a bonafide work carried out by

Pooja Satheesh	**PES2UG21CS374**
Preethi M	**PES2UG21CS396**

In partial fulfilment for the completion of fifth semester DBMS Project (UE21CS351) in the Program of Study -Bachelor of Technology in Computer Science and Engineering under rules and regulations of PES University, Bengaluru during the period AUG. 2023 – DEC. 2023. It is certified that all corrections / suggestions indicated for internal assessment have been incorporated in the report. The project has been approved as it satisfies the 5th semester academic requirements in respect of project work.

Signature

Prof. Shilpa S

Assistant Professor

DECLARATION

We hereby declare that the DBMS Project entitled **Co - Lancer** has been carried out by us under the guidance of **Prof. Shilpa S, Assistant Professor** and submitted in partial fulfilment of the course requirements for the award of degree of **Bachelor of Technology** in **Computer Science and Engineering** of **PES University, Bengaluru** during the academic semester AUG – DEC 2023.

Pooja Satheesh **PES2UG21CS374**

Preethi M **PES2UG21CS396**

ACKNOWLEDGEMENT

We would like to express our gratitude to Prof. Shilpa S, Department of Computer Science and Engineering, PES University, for her continuous guidance, assistance, and encouragement throughout the development of this UE21CS351 - DBMS Project.

We take this opportunity to thank Dr. Sandesh B J C, Professor, Chairperson, Department of Computer Science and Engineering, PES University, for all the knowledge and support we have received from the department.

We are deeply grateful to Dr. M. R. Doreswamy, Chancellor, PES University, Prof. Jawahar Doreswamy, Pro Chancellor – PES University, Dr. Suryaprasad J, Vice-Chancellor, PES University for providing to us various opportunities and enlightenment every step of the way.

Finally, this DBMS Project could not have been completed without the continual support and encouragement we have received from our families and friends.

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.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

Freelancers can connect and work together on projects on the Co-Lancer website. It gives clients a place to publish projects under several categories. Freelancers that show interest can choose to work on projects that fit their skill set. When working on a project that needs collaboration, the chat option is activated.

The website seeks to provide access to projects and to other freelancers. Reviews and cookies (a point system) are two incentives that push freelancers to take on additional work. After a project is finished, the client can review it and verify its completion. The client is then provided with a payment choice.

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.

The following project was developed using ReactJS for frontend, NodeJS for backend and MySQL as the database.

The following document provides a detailed explanation of the backend table creations and queries employed for retrieving and updating the collected and stored data.

2. PROBLEM DEFINITION

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.

3. ER MODEL

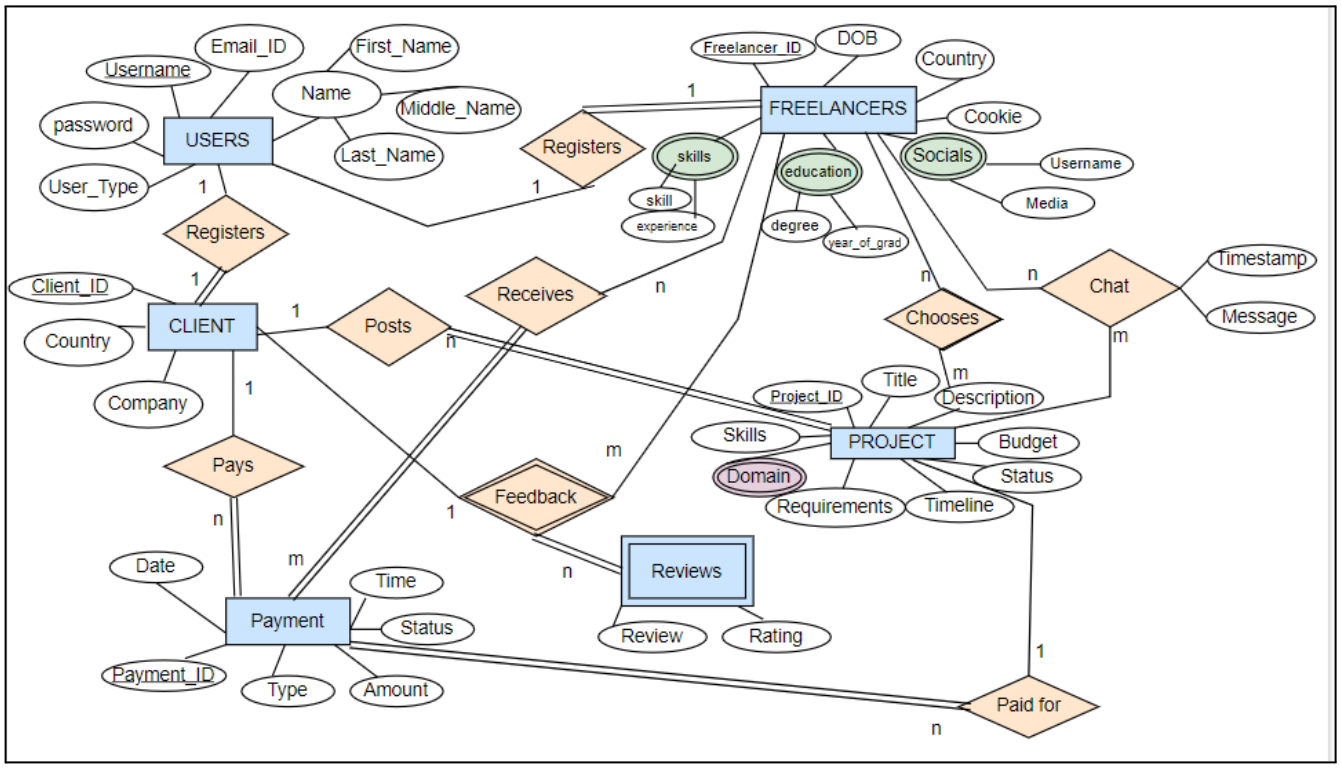

Fig. 3.1

4. ER TO RELATIONAL MAPPING

4.1 STEPS OF ALGORITHM FOR CHOSEN PROBLEM

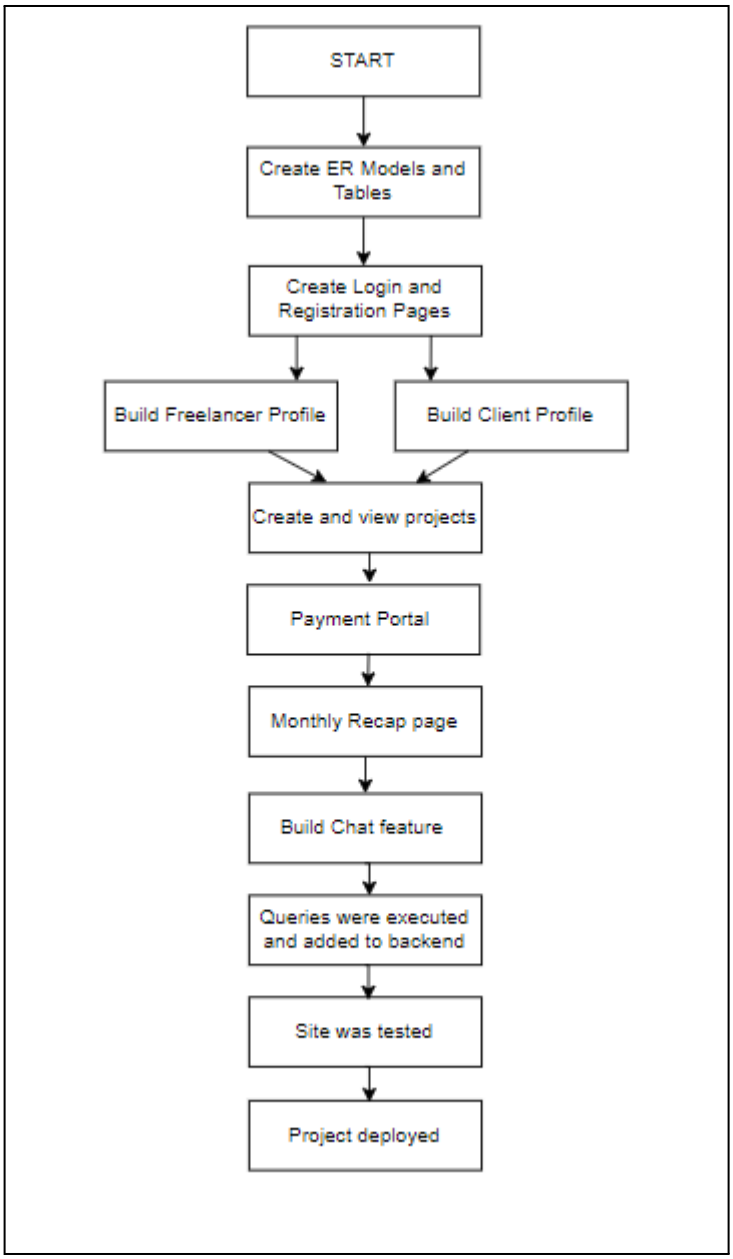

Fig. 4.1

4.2 COMPLETE DIAGRAM OF RELATIONAL MAPPING

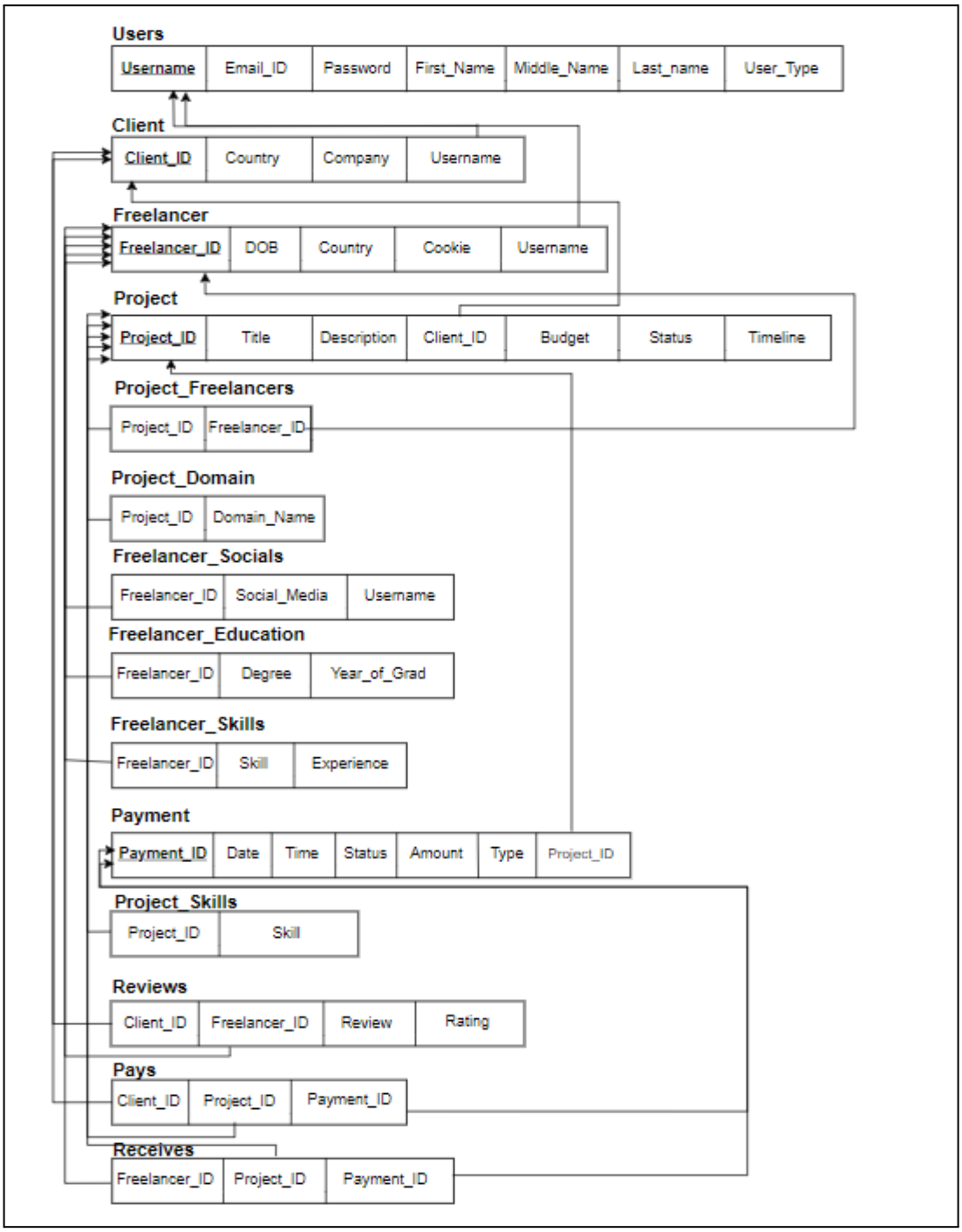

Fig. 4.2

5. DDL STATEMENTS

STATEMENTS WITH SCREEN SHOTS OF THE TABLE CREATION

1. users

```
Co-Lancer>CREATE TABLE users(username VARCHAR(30) PRIMARY KEY, email_ID VARCHAR(50) NOT NULL UNIQUE , password VARCHAR(100) 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 (0.04 sec)
```

2. client

```
Co-Lancer>CREATE TABLE client(client_ID VARCHAR(10) PRIMARY KEY,country VARCHAR(30), company VARCHAR(30) NOT NULL, username VARCHAR(5
0), FOREIGN KEY (username) REFERENCES users(username) ON DELETE CASCADE);
Query OK, 0 rows affected (0.02 sec)
```

3. freelancer

```
Co-Lancer>CREATE TABLE freelancer(freelancer_ID VARCHAR(10) PRIMARY KEY, DOB DATE, country VARCHAR(30), cookies INT DEFAULT 0,usernam
e VARCHAR(50), FOREIGN KEY (username) REFERENCES users(username) ON DELETE CASCADE);
Query OK, 0 rows affected (0.01 sec)
```

4. project

```
Co-Lancer>CREATE TABLE project(project_ID VARCHAR(10) PRIMARY KEY, title VARCHAR(30) NOT NULL UNIQUE, description VARCHAR(60) NOT NUL
L, budget INT NOT NULL, status ENUM('Not Assigned', 'In Progress', 'Completed') NOT NULL, timeline INT NOT NULL, client_username VARC
HAR(50), FOREIGN KEY (client_username) REFERENCES users(username) ON DELETE CASCADE, colab ENUM('YES', 'NO') NOT NULL, url VARCHAR(60
), start_date DATE);
Query OK, 0 rows affected (0.01 sec)
```

5. project_domains

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

6. project_skills

```
Co-Lancer>CREATE TABLE project_skills(project_ID VARCHAR(10), skill VARCHAR(30), FOREIGN KEY (project_ID) references project(project_
ID) ON DELETE CASCADE);
Query OK, 0 rows affected (0.01 sec)
```

7. project_freelancers

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

8. freelancer_socials

```
Co-Lancer>CREATE TABLE freelancer_socials(freelancer_ID VARCHAR(10), media VARCHAR(30), userhandle VARCHAR(30), FOREIGN KEY (freelanc
er_ID) references freelancer(freelancer_ID) ON DELETE CASCADE);
Query OK, 0 rows affected (0.04 sec)
```

9. freelancer_educations

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

10. freelancer_skills

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

11. payment

```
Co-Lancer>CREATE TABLE payment(payment_ID VARCHAR(10) PRIMARY KEY, project_ID VARCHAR(10), date DATE, time TIMESTAMP, status ENUM('Su
ccessful', '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) ON DELETE CASCADE);
Query OK, 0 rows affected (0.01 sec)
```

12. 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) ON DELETE CASCADE,FOREIGN KEY (payment_ID) REFERENCES payment(payment_ID) ON DELETE CASCADE,FOREIGN KEY (project_I
D) REFERENCES project(project_ID) ON DELETE CASCADE);
Query OK, 0 rows affected (0.02 sec)
```

13. 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) ON DELETE CASCADE,FOREIGN KEY (payment_ID) REFERENCES payment(payment_ID) ON DELETE CASCADE,FO
REIGN KEY (project_ID) REFERENCES project(project_ID) ON DELETE CASCADE);
Query OK, 0 rows affected (0.02 sec)
```

14. reviews

```
Co-Lancer>CREATE TABLE reviews(client_username VARCHAR(30), freelancer_ID VARCHAR(10), review VARCHAR(100), rating ENUM('1','2','3','
4','5','6','7','8','9','10'), FOREIGN KEY (client_username) REFERENCES users(username) ON DELETE CASCADE, FOREIGN KEY (freelancer_ID)
 REFERENCES freelancer(freelancer_ID) ON DELETE CASCADE);
Query OK, 0 rows affected (0.02 sec)
```

15. 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) ON DELETE CASCADE, FOREIGN KEY (freelancer_ID) REFERENCES freelancer(freelancer_ID) ON D
ELETE CASCADE);
Query OK, 0 rows affected (0.02 sec)
```

16. counter

```
Co-Lancer>CREATE TABLE counter (client_count INT, freelancer_count INT, payment_count INT, project_count INT);
Query OK, 0 rows affected (0.01 sec)
```

17. project_pdf

```
Co-Lancer>CREATE TABLE project_pdf (title VARCHAR(30), data LONGBLOB, contentType VARCHAR(255), FOREIGN KEY (title) REFERENCES projec
t(title) ON DELETE CASCADE);
Query OK, 0 rows affected (0.01 sec)
```

18. image

```
Co-Lancer>CREATE TABLE image (username VARCHAR(30), data LONGBLOB, contentType VARCHAR(255), FOREIGN KEY (username) REFERENCES users(
username) ON DELETE CASCADE);
Query OK, 0 rows affected (0.01 sec)
```

19. user_client

```
Co-Lancer>CREATE VIEW user_client AS SELECT first_name, middle_name, last_name FROM users;
Query OK, 0 rows affected (0.01 sec)
```

Tables:

```
Co-Lancer>show tables;
+--------------------------+
| Tables_in_delete_colancer |
+--------------------------+
| chat                     |
| client                   |
| counter                  |
| freelancer               |
| freelancer_educations    |
| freelancer_skills        |
| freelancer_socials       |
| image                    |
| payment                  |
| pays                     |
| project                  |
| project_domains          |
| project_freelancers      |
| project_pdf              |
| project_skills           |
| receives                 |
| reviews                  |
| user_client              |
| users                    |
+--------------------------+
19 rows in set (0.01 sec)
```

Table 5.1

Database security:

1. CREATE USER 'colancer_admin'@'localhost' IDENTIFIED BY 'P@ss@colancer_admin'
2. CREATE USER 'colancer_freelancer'@'localhost' IDENTIFIED BY 'P@ss@freelancer';
3. CREATE USER 'colancer_client'@'localhost' IDENTIFIED BY 'P@ss@client';
4. CREATE USER 'colancer_user'@'localhost' IDENTIFIED BY 'P@ss@user'

```
Co-Lancer>CREATE USER 'colancer_client'@'%' IDENTIFIED BY 'P@ss@client';
Query OK, 0 rows affected (0.02 sec)

Co-Lancer>CREATE USER 'colancer_user'@'%' IDENTIFIED BY 'P@ss@user';
Query OK, 0 rows affected (0.00 sec)

Co-Lancer>CREATE USER 'colancer_freelancer'@'%' IDENTIFIED BY 'P@ss@freelancer';
Query OK, 0 rows affected (0.01 sec)
```

```
Co-Lancer>CREATE USER 'colancer_admin'@'%' IDENTIFIED BY 'P@ss@colancer_admin';
Query OK, 0 rows affected (0.00 sec)
```

Access Rights:

```
Co-Lancer>show grants for 'colancer_user'@'%';
+-----------------------------------------------------------------------------------------------+
| Grants for colancer_user@%                                                                    |
+-----------------------------------------------------------------------------------------------+
| GRANT USAGE ON *.* TO `colancer_user`@`%`                                                     |
| GRANT SELECT ON `co_lancer`.`freelancer` TO `colancer_user`@`%`                               |
| GRANT SELECT, INSERT ON `co_lancer`.`image` TO `colancer_user`@`%`                            |
| GRANT SELECT ON `co_lancer`.`payment` TO `colancer_user`@`%`                                  |
| GRANT SELECT ON `co_lancer`.`project_freelancers` TO `colancer_user`@`%`                      |
| GRANT SELECT ON `co_lancer`.`project_pdf` TO `colancer_user`@`%`                              |
| GRANT SELECT ON `co_lancer`.`project` TO `colancer_user`@`%`                                  |
| GRANT SELECT ON `co_lancer`.`reviews` TO `colancer_user`@`%`                                  |
| GRANT SELECT, INSERT ON `co_lancer`.`users` TO `colancer_user`@`%`                            |
| GRANT EXECUTE ON PROCEDURE `co_lancer`.`makefreelancerprofile` TO `colancer_user`@`%`         |
+-----------------------------------------------------------------------------------------------+
10 rows in set (0.00 sec)
```

Table 5.2

```
Co-Lancer>show grants for 'colancer_client'@'%';
+-----------------------------------------------------------------------------------------------+
| Grants for colancer_client@%                                                                  |
+-----------------------------------------------------------------------------------------------+
| GRANT USAGE ON *.* TO `colancer_client`@`%`                                                   |
| GRANT SELECT, INSERT ON `co_lancer`.`client` TO `colancer_client`@`%`                         |
| GRANT SELECT, UPDATE ON `co_lancer`.`counter` TO `colancer_client`@`%`                        |
| GRANT INSERT ON `co_lancer`.`payment` TO `colancer_client`@`%`                                |
| GRANT SELECT, INSERT ON `co_lancer`.`project_domains` TO `colancer_client`@`%`                |
| GRANT SELECT, INSERT ON `co_lancer`.`project_freelancers` TO `colancer_client`@`%`            |
| GRANT INSERT ON `co_lancer`.`project_pdf` TO `colancer_client`@`%`                            |
| GRANT SELECT, INSERT ON `co_lancer`.`project_skills` TO `colancer_client`@`%`                 |
| GRANT SELECT, INSERT, UPDATE ON `co_lancer`.`project` TO `colancer_client`@`%`                |
| GRANT INSERT, UPDATE ON `co_lancer`.`reviews` TO `colancer_client`@`%`                        |
| GRANT SELECT ON `co_lancer`.`user_client` TO `colancer_client`@`%`                            |
+-----------------------------------------------------------------------------------------------+
11 rows in set (0.00 sec)
```

Table 5.3

```
Co-Lancer>show grants for 'colancer_freelancer'@'%';
+-----------------------------------------------------------------------------------------------+
| Grants for colancer_freelancer@%                                                              |
+-----------------------------------------------------------------------------------------------+
| GRANT USAGE ON *.* TO `colancer_freelancer`@`%`                                               |
| GRANT SELECT, INSERT ON `co_lancer`.`chat` TO `colancer_freelancer`@`%`                       |
| GRANT SELECT, UPDATE ON `co_lancer`.`counter` TO `colancer_freelancer`@`%`                    |
| GRANT INSERT ON `co_lancer`.`freelancer_educations` TO `colancer_freelancer`@`%`              |
| GRANT INSERT ON `co_lancer`.`freelancer_skills` TO `colancer_freelancer`@`%`                  |
| GRANT INSERT ON `co_lancer`.`freelancer_socials` TO `colancer_freelancer`@`%`                 |
| GRANT SELECT, INSERT, UPDATE ON `co_lancer`.`freelancer` TO `colancer_freelancer`@`%`         |
| GRANT SELECT ON `co_lancer`.`project_domains` TO `colancer_freelancer`@`%`                    |
| GRANT SELECT, INSERT ON `co_lancer`.`project_freelancers` TO `colancer_freelancer`@`%`        |
| GRANT SELECT ON `co_lancer`.`project_skills` TO `colancer_freelancer`@`%`                     |
| GRANT SELECT ON `co_lancer`.`project` TO `colancer_freelancer`@`%`                            |
| GRANT EXECUTE ON PROCEDURE `co_lancer`.`getprojects` TO `colancer_freelancer`@`%`             |
+-----------------------------------------------------------------------------------------------+
12 rows in set (0.00 sec)
```

Table 5.4

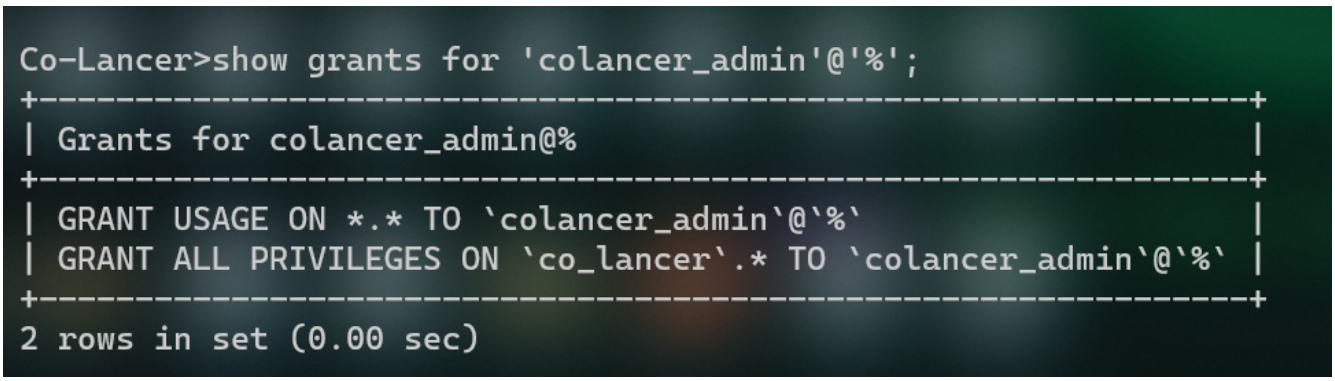
Co-Lancer>show grants for 'colancer_admin'@'%';
Grants for colancer_admin@%
GRANT USAGE ON *.* TO `colancer_admin`@`%`
GRANT ALL PRIVILEGES ON `co_lancer`.* TO `colancer_admin`@`%`
2 rows in set (0.00 sec)

Table 5.5

6. DML STATEMENTS

STATEMENTS WITH SCREEN SHOTS OF THE TABLE WITH INSERTED VALUES

1. Inserting a user profile details into USERS table

```
const insert_query = "INSERT INTO users(username, email_id, password, first_name, "+ (middle_name === 'NULL'?
'':"middle_name, ") + (last_name === 'NULL'? '':"last_name, ") + "user_type) VALUE('" + username + "', '" +
email_id + "', '" + password + "', '" + (middle_name === "NULL"? (last_name === "NULL" ? first_name: first_name
+ "', '" + last_name) : first_name + "', '" + middle_name + "', '" + last_name) + "', '"+ usertype + "');";
console.log(insert_query);
```

```
Co-Lancer>select * from users;
+----------+------------------+------------------------------------------------------------+------------+-------------+----------+-----------+
| username | email_ID         | password                                                   | first_name | middle_name | last_name | user_type |
+----------+------------------+------------------------------------------------------------+------------+-------------+----------+-----------+
| Mugund   | mugund@gmail.com | $2a$10$R1WsBulA/fwQwuel3umm5eRheQ31KYbF/hNrnbKfEMxQIu51mRfmC | Mugund     | NULL        | NULL     | Client    |
| user1    | pranjal@gmail.com | $2a$10$L45D358w.IlFiyvdm8Jv.uflAJjFJ2lHdKVS3CTnv02ZdruS5i3FK | Pranjal    | NULL        | NULL     | Freelancer |
| user10   | manish@gmail.com | $2a$10$.hIKHz.jHKp5Mt9xxil/WufRg1S7xKKB6oSQKjcLnxobBoNIrbDpO | Manish     | NULL        | Pandey   | Freelancer |
| user2    | malay@gmail.com  | $2a$10$RKkW0BHZjTP9Ob2NqQLyTOGvntxpvFc.8rSTuWAw//QJItMGpwqQm | Malay      | NULL        | Shankar  | Freelancer |
| user3    | priya@gmail.com  | $2a$10$qfOGAS/E3cIIDsCmweU5eekWwp0VA/zrPbNRwiNPaQ9fBEXRcwqWq | Priya      | NULL        | S        | Freelancer |
| user4    | disha@gmail.com  | $2a$10$gYaT0p73lkjl8V3CFPi/lOv1clrT7i7nud0qLTDBCWAzD1krR9itW | Disha      | NULL        | Raj      | Client    |
| user5    | keerthi@gmail.com | $2a$10$MIdV.bB94aDiDTuabAWUu..21clMZPePHg.2ENkYLCtR61ozCHCuu | Keerthi    | NULL        | Ramesh   | Client    |
| user6    | prakash@gmail.com | $2a$10$sPhY4m1KUA3UczOKfV19NuU7x/6Y4ORN4wvhaQkNauWqH1GbDwS3i | Prakash    | NULL        | L        | Client    |
| user9    | arman@gmail.com  | $2a$10$rHOw/PMy45.C.T8thtDsFOhIP.wJSj04aEiOMx7OZG2jsofep61gC | Arman      | NULL        | Singh    | Client    |
+----------+------------------+------------------------------------------------------------+------------+-------------+----------+-----------+
9 rows in set (0.07 sec)
```

Table 6.1

2. Inserting additional freelancer details into FREELANCER table

```
var insert_query = "INSERT INTO freelancer VALUE('FID" + freelancer_count + "', '" + dob + "', '" + country
+ "', 0, '" + username + "');";
console.log(insert_query);
```

```
insert_query = "INSERT INTO freelancer_educations VALUES " + temp.slice(0,temp.length-1) + ";";
```

```
insert_query = "INSERT INTO freelancer_skills VALUES " + temp.slice(0,temp.length-1) + ";";
```

```
insert_query = "INSERT INTO freelancer_socials VALUES " + temp.slice(0,temp.length-1) + ";";
```

```
Co-Lancer>select * from freelancer;
+---------------+------------+---------+---------+----------+
| freelancer_ID | DOB        | country | cookies | username |
+---------------+------------+---------+---------+----------+
| FID0          | 2023-11-14 | Russia  |      10 | user1    |
| FID1          | 2023-11-03 | France  |      19 | user2    |
| FID2          | 2023-11-03 | japan   |      17 | user3    |
| FID3          | 2003-06-11 | Russia  |       0 | user10   |
+---------------+------------+---------+---------+----------+
4 rows in set (0.07 sec)
```

Table 6.2

3. Inserting client details into CLIENT table

```
const insert_query = "INSERT INTO client VALUE('CID" + client_count + "', '" + country + "', '" + company +
"', '" + username + "');";
```

```
Co-Lancer>select * from client;
+-----------+---------+-------------+----------+
| client_ID | country | company     | username |
+-----------+---------+-------------+----------+
| CID0      | Germany | Cisco       | user4    |
| CID1      | Dubai   | GreenMethod | user5    |
| CID2      | Russia  | Lukoil      | user6    |
| CID3      | Japan   | Nintendo    | user9    |
+-----------+---------+-------------+----------+
4 rows in set (0.07 sec)
```

Table 6.3

4. To retrieve username and password for validation of user account

```
db_user.query("SELECT password, user_type FROM users WHERE username='"+username+"';", f
```

5. To build a client profile

```
let retrieve_query="SELECT client_id, first_name, middle_name, last_name, company FROM client JOIN user_client USING
(username) WHERE username='"+ username + "';";
db_client.query(retrieve_query, function(err, result) {
```

```
retrieve_query = "SELECT project_ID, title, description, status, url FROM client JOIN project ON client.
username=project.client_username WHERE client_id='" + client_id + "';";
db_client.query(retrieve_query, function(err, result) {
```

```
Co-Lancer>select * from user_client;
+----------+------------+-------------+-----------+
| username | first_name | middle_name | last_name |
+----------+------------+-------------+-----------+
| user4    | Disha      | NULL        | Raj       |
| user5    | Keerthi    | NULL        | Ramesh    |
| user6    | Prakash    | NULL        | L         |
| user9    | Arman      | NULL        | Singh     |
+----------+------------+-------------+-----------+
4 rows in set (0.08 sec)
```

Table 6.4

6. Freelancer joining a project

```
const query = "INSERT INTO project_freelancers VALUE('" + p_id + "', '" + f_id +"');";
```

```
db_client.query("UPDATE project SET status='In Progress', start_date=(SELECT CURDATE()) WHERE project_ID='"
+p_id+"';", function(err,result) {
```

```
Co-Lancer>select * from project;
+------------+--------+------------------------------+--------+-------------+----------+-----------------+-------+-------------+------------+
| project_ID | title  | description                  | budget | status      | timeline | client_username | colab | url         | start_date |
+------------+--------+------------------------------+--------+-------------+----------+-----------------+-------+-------------+------------+
| PID0       | Title1 | Description for first project |   1000 | Completed   |       10 | user4           | NO    | repo_title1 | 2023-11-15 |
| PID1       | Title2 | Description2                 |   3000 | In Progress |       10 | user5           | YES   | NULL        | 2023-11-15 |
| PID2       | Title3 | Description3                 |   1000 | Completed   |        6 | user5           | NO    | repo_title3 | 2023-11-15 |
| PID3       | Title4 | Description4                 |   3000 | Completed   |       10 | user6           | YES   | repo_title4 | 2023-11-15 |
| PID4       | Title5 | Desctiption5                 |   2000 | Completed   |       10 | user6           | NO    | repo_title5 | 2023-11-16 |
| PID5       | Title6 | Description 6                |   2000 | Completed   |       10 | user6           | YES   | repo_title6 | 2023-11-18 |
+------------+--------+------------------------------+--------+-------------+----------+-----------------+-------+-------------+------------+
6 rows in set (0.07 sec)
```

Table 6.5

```
Co-Lancer>select * from project_freelancers;
+------------+---------------+
| project_ID | freelancer_id |
+------------+---------------+
| PID1       | FID0          |
| PID3       | FID0          |
| PID0       | FID1          |
| PID1       | FID1          |
| PID5       | FID1          |
| PID1       | FID2          |
| PID2       | FID2          |
| PID3       | FID2          |
| PID4       | FID2          |
| PID5       | FID2          |
+------------+---------------+
10 rows in set (0.28 sec)
```

Table 6.6

7. Retrieve reviews of freelancers

```
db_user.query("SELECT freelancer_ID, client_username, review, rating FROM reviews WHERE freelancer_id='" + req.body.
f_id + "';", (err, result) => {
```

```
Co-Lancer>select * from reviews;
+-----------------+---------------+--------------------+--------+
| client_username | freelancer_ID | review             | rating |
+-----------------+---------------+--------------------+--------+
| user4           | FID1          | great work         | 9      |
| user5           | FID2          | UI could be better | 7      |
| user5           | FID0          | Well done          | 10     |
| user5           | FID1          | Well done          | 10     |
| user5           | FID2          | Well done          | 10     |
+-----------------+---------------+--------------------+--------+
5 rows in set (0.06 sec)
```

Table 6.7

8. Publishing a project

```
const query = "INSERT into project VALUE('PID" + project_count + "', '" + title + "', '" + description+ "',
" + budget + ", 'Not Assigned', " + timeline + ", '" + username + "', '" + req.body.collab + "', null,
null);";
```

```
db_client.query("INSERT INTO project_domains VALUES"+list1+";", function(err, result) {
    if(err) throw err;
```

```
Co-Lancer>select * from project_domains;
+------------+------------------+
| project_ID | domain_name      |
+------------+------------------+
| PID0       | Machine Learning |
| PID0       | Data Analytics   |
| PID1       | NLP              |
| PID2       | Cyber Security   |
| PID2       | Web dev          |
| PID3       | Big Data         |
| PID3       | Computer Networks|
| PID4       | Cyber Security   |
| PID4       | Machine Learning |
| PID5       | Cyber Security   |
| PID5       | NLP              |
| PID5       | Machine Learning |
+------------+------------------+
12 rows in set (0.11 sec)
```

Table 6.8

```
db_client.query("INSERT INTO project_skills VALUES"+list2+";", function(err, result) {
    if(err) throw err
```

```
Co-Lancer>select * from project_skills;
+------------+------------+
| project_ID | skill      |
+------------+------------+
| PID0       | c          |
| PID0       | python     |
| PID0       | hadoop     |
| PID1       | python     |
| PID1       | r          |
| PID2       | html       |
| PID2       | javascript |
| PID2       | c          |
| PID3       | hadoop     |
| PID3       | spark      |
| PID3       | c          |
| PID3       | python     |
| PID4       | c          |
| PID4       | python     |
| PID4       | hadoop     |
| PID4       | r          |
| PID5       | c          |
| PID5       | python     |
| PID5       | hadoop     |
+------------+------------+
19 rows in set (0.09 sec)
```

Table 6.9

9. Uploading and displaying profile picture

```
db_user.query('INSERT INTO image SET ?', image, (err, result) => {
```

```
db_user.query('SELECT data, contentType FROM image WHERE username = ?',
```

```
Co-Lancer>select username,contentType from image;
+----------+-------------+
| username | contentType |
+----------+-------------+
| user1    | image/jpeg  |
| user1    | image/jpeg  |
| user2    | image/jpeg  |
| user3    | image/jpeg  |
| user4    | image/png   |
| user5    | image/png   |
| user6    | image/jpeg  |
| user9    | image/jpeg  |
| user10   | image/jpeg  |
+----------+-------------+
9 rows in set (0.09 sec)
```

Table 6.10

10. Uploading and displaying project requirements (PDF format)

```
db_client.query('INSERT INTO project_pdf SET ?', pdf, (err, result)
```

```
db_user.query('SELECT data, contentType FROM project_pdf WHERE title = ?', [pdfId], (err, result)
```

```
Co-Lancer>select title,contentType from project_pdf;
+--------+-----------------+
| title  | contentType     |
+--------+-----------------+
| Title1 | application/pdf |
| Title2 | application/pdf |
| Title3 | application/pdf |
| Title4 | application/pdf |
| Title5 | application/pdf |
| Title6 | application/pdf |
+--------+-----------------+
6 rows in set (0.08 sec)
```

Table 6.11

11. Project submission my freelancer

```
const query = "UPDATE project SET url='" + url + "' WHERE project_ID='" + project_id + "';";
```

12. Project accepted by client

```
let query = "UPDATE project SET status='Completed' WHERE project_ID='" + project_id + "';";
```

```
query = "DELETE FROM chat WHERE project_ID='" + project_id + "';";
```

13. Project not accepted by client and returned to freelancer

```
const query = "UPDATE project SET url="+null+" WHERE project_ID='" + project_id + "';";
```

14. Providing reviews to freelancers by clients

```
let query = "SELECT freelancer_ID FROM project_freelancers WHERE project_ID='"+project_id+"';";
```

```
query = "INSERT INTO reviews VALUES('" + username+ "','" + freelancers[i] + "', '" + review + "', '"+
rating + "');";
```

```
query = "UPDATE freelancer SET cookies=cookies+" + rating + " WHERE freelancer_id='" + freelancers[i] +
"';";
```

15. Finding the group projects that are in progress

```
db_freelancer.query("select distinct(project_id) from project_freelancers where project_ID IN (SELECT project_ID
FROM project where colab='YES' AND status<>'Completed') AND freelancer_id=(select freelancer_id from freelancer
where username='"+username+"');",(err,result) =>{
```

16. Display previous chats

```
q="SELECT username,message,timestamp FROM chat,freelancer WHERE project_ID='"+pid+"' and chat.
freelancer_id=freelancer.freelancer_id;"
```

17. Inserting a new chat into database

```
q="SELECT freelancer_id FROM freelancer WHERE username='"+final_msg['name']+"';"
```

```
q1="INSERT INTO chat VALUES('"+final_msg['project_id']+"','"+fid+"', (SELECT CURRENT_TIMESTAMP()), '"
+final_msg['text']+"');"
```

```
Co-Lancer>select * from chat;
+------------+---------------+---------------------+--------------------------+
| project_ID | freelancer_ID | timestamp           | message                  |
+------------+---------------+---------------------+--------------------------+
| PID5       | FID2          | 2023-11-18 18:59:53 | hello this is user3      |
| PID5       | FID1          | 2023-11-18 19:00:16 | hey this is user 2       |
| PID5       | FID0          | 2023-11-18 19:00:47 | hey everyone user1 here  |
+------------+---------------+---------------------+--------------------------+
3 rows in set (0.07 sec)
```

Table 6.12

18. Displaying monthly recap of freelancers

```
q1="SELECT COUNT(*) AS monthly_proj FROM project JOIN project_freelancers ON project.project_id =
project_freelancers.project_id WHERE MONTH(project.start_date) =" +month+" and project_freelancers.freelancer_id='"
+fid+"';"
```

```
q2="SELECT COUNT(*) as total FROM project JOIN project_freelancers ON project.project_id=project_freelancers.
project_id WHERE project_freelancers.freelancer_id='"+fid+"';"
```

```
q3="SELECT COUNT(*) AS in_prog FROM project JOIN project_freelancers ON project.
project_id=project_freelancers.project_id WHERE project_freelancers.freelancer_id='"+fid+"' and status='In
Progress';"
```

```
q4="SELECT COUNT(*) AS completed FROM project JOIN project_freelancers ON project.
project_id=project_freelancers.project_id WHERE project_freelancers.freelancer_id='"+fid+"' and
status='Completed';"
```

```
q5="SELECT monthly_amount('"+fid+"',"+month+") AS total_earned;"
```

19. Payment feature

```
q2="INSERT INTO payment VALUES('"+payid+"','"+pid+"','"+currentDate+"','"+currentDate+" "+time+"','Successful',"
+amount+",'"+mode+"');"
```

```
Co-Lancer>select * from payment;
+------------+------------+------------+---------------------+------------+--------+------------+
| payment_ID | project_ID | date       | time                | status     | amount | type       |
+------------+------------+------------+---------------------+------------+--------+------------+
| PAY0       | PID2       | 2023-11-15 | 2023-11-15 18:59:33 | Successful |   1000 | upi        |
| PAY1       | PID4       | 2023-11-16 | 2023-11-16 09:42:24 | Successful |   2000 | netbanking |
| PAY2       | PID3       | 2023-11-18 | 2023-11-18 17:37:22 | Successful |   2000 | upi        |
| PAY3       | PID5       | 2023-11-18 | 2023-11-18 23:43:59 | Successful |   2000 | upi        |
+------------+------------+------------+---------------------+------------+--------+------------+
4 rows in set (0.06 sec)
```

Table 6.13

7. QUERIES

7.1 SIMPLE QUERY WITH GROUP BY, AGGREGATE

GROUP_CONCAT and SUM aggregate queries are included in the procedures and functions

```
    (SELECT GROUP_CONCAT(domain_name) FROM project_domains WHERE
project_id=project.project_id) AS domains,
    (SELECT GROUP_CONCAT(skill) FROM project_skills WHERE project_id=project.project_id) AS
skills,
    (SELECT GROUP_CONCAT(freelancer_id) FROM project_freelancers WHERE
project_id=project.project_id) AS freelancers
```

7.2 UPDATE OPERATION

```
db_client.query("UPDATE project SET status='In Progress', start_date=(SELECT CURDATE()) WHERE project_ID='"
+p_id+"';", function(err,result) {
```

```
const query = "UPDATE project SET url='" + url + "' WHERE project_ID='" + project_id + "';";
db_client.query(query, (err, result) => {
```

```
const query = "UPDATE project SET url="+null+" WHERE project_ID='" + project_id + "';";
db_client.query(query, (err, result) => {
```

```
query = "UPDATE freelancer SET cookies=cookies+" + rating + " WHERE freelancer_id='" + freelancers[i] +
"';";
```

7.3 DELETE OPERATION

```
query = "DELETE FROM chat WHERE project_ID='" + project_id + "';";
db_admin.query(query, (err,result) => {
```

7.4 CORRELATED QUERY

```
db_client.query("UPDATE project SET status='In Progress', start_date=(SELECT CURDATE()) WHERE project_ID='"
+p_id+"';", function(err,result) {
```

7.5 NESTED QUERY

```
db_freelancer.query("select distinct(project_id) from project_freelancers where project_ID IN (SELECT project_ID
FROM project where colab='YES' AND status<>'Completed') AND freelancer_id=(select freelancer_id from freelancer
where username='"+username+"');",(err,result) =>{
```

8. STORED PROCEDURES, FUNCTIONS AND TRIGGERS

8.1 STORED PROCEDURES OR FUNCTIONS

1. **getProjects**

```
DELIMITER $$
CREATE PROCEDURE getProjects(IN domain VARCHAR(30))
BEGIN
   DECLARE p_id VARCHAR(10);
   DECLARE p_title VARCHAR(30);
   DECLARE p_desc VARCHAR(100);
        DECLARE p_budget INT;
        DECLARE p_status ENUM('Not Assigned', 'In Progress', 'Completed');
        DECLARE p_timeline INT;
        DECLARE p_req BLOB;
        DECLARE domains VARCHAR(100);
        DECLARE skills VARCHAR(100);
        DECLARE freelancers VARCHAR(100);
        DECLARE colab ENUM("YES", "NO");

SELECT
   DISTINCT
   project.project_id AS p_id,
        project.title as p_title,
        project.description as p_desc,
        project.budget as p_budget,
        project.status as p_status,
        project.timeline as p_timeline,
        project.colab as p_colab,
   (SELECT GROUP_CONCAT(domain_name) FROM project_domains WHERE
project_id=project.project_id) AS domains,
   (SELECT GROUP_CONCAT(skill) FROM project_skills WHERE
project_id=project.project_id) AS skills,
   (SELECT GROUP_CONCAT(freelancer_id) FROM project_freelancers WHERE
project_id=project.project_id) AS freelancers
FROM
   project
LEFT JOIN
        project_domains USING (project_id)
WHERE
        project_domains.domain_name = domain OR domain = "all";

END$$
DELIMITER ;
```

2. **makeFreelancerProfile**

```
DELIMITER $$
CREATE PROCEDURE makeFreelancerProfile(IN uname VARCHAR(30))
BEGIN
   DECLARE f_id VARCHAR(10);
   DECLARE cookie_value INT;

SELECT
   DISTINCT
   freelancer.freelancer_id AS f_id,
   freelancer.cookies AS cookie_value,
   (SELECT GROUP_CONCAT(project_id) FROM project_freelancers WHERE
freelancer_ID = freelancer.freelancer_ID) AS p_id,
   (SELECT GROUP_CONCAT(status) FROM project JOIN project_freelancers
USING(project_id) WHERE freelancer_ID = freelancer.freelancer_id) AS p_status,
   (SELECT GROUP_CONCAT(title) FROM project JOIN project_freelancers
USING(project_id) WHERE freelancer_ID = freelancer.freelancer_id) AS p_title,
   (SELECT GROUP_CONCAT(description) FROM project JOIN project_freelancers
USING(project_id) WHERE freelancer_ID = freelancer.freelancer_id) AS p_desc,
   (SELECT GROUP_CONCAT(skill) FROM freelancer_skills WHERE freelancer_ID
= freelancer.freelancer_ID) AS skill,
   (SELECT GROUP_CONCAT(experience) FROM freelancer_skills WHERE
freelancer_ID = freelancer.freelancer_ID) AS experience,
   (SELECT GROUP_CONCAT(media) FROM freelancer_socials WHERE
freelancer_ID = freelancer.freelancer_ID) AS social_media,
   (SELECT GROUP_CONCAT(userhandle) FROM freelancer_socials WHERE
freelancer_ID = freelancer.freelancer_ID) AS userhandle
FROM
   freelancer
WHERE
   freelancer.username = uname;

END$$
DELIMITER ;
```

3. monthly_amount

```
DELIMITER $$
CREATE FUNCTION monthly_amount(fid VARCHAR(10), month INT)
RETURNS INT
DETERMINISTIC
BEGIN
   DECLARE total INT;

   SELECT SUM(amount) AS total_amt
   INTO total
   FROM payment
   WHERE payment.project_id IN (SELECT project_id FROM project_freelancers
WHERE freelancer_id = fid)
   AND MONTH(payment.date) = month;
   RETURN total;
END $$
DELIMITER ;
```

8.2 TRIGGERS

1. update_freelancer_counter

```
DELIMITER $$
CREATE TRIGGER update_freelancer_counter
        AFTER INSERT ON freelancer
        FOR EACH ROW
        BEGIN
                UPDATE counter SET freelancer_count = freelancer_count + 1;
        END;
$$
DELIMITER ;
```

2. update_client_counter

```
DELIMITER $$
CREATE TRIGGER update_client_counter
        AFTER INSERT ON client
        FOR EACH ROW
        BEGIN
                UPDATE counter SET client_count = client_count + 1;
        END;
$$
DELIMITER ;
```

3. **update_project_counter**

```
DELIMITER $$
CREATE TRIGGER update_project_counter
        AFTER INSERT ON project
        FOR EACH ROW
        BEGIN
                UPDATE counter SET project_count = project_count + 1;
        END;
$$
DELIMITER ;
```

4. **update_payment_counter**

```
DELIMITER $$
CREATE TRIGGER update_payment_counter
        AFTER INSERT ON payment
        FOR EACH ROW
        BEGIN
                UPDATE counter SET payment_count = payment_count + 1;
        END;
$$
DELIMITER ;
```

```
Co-Lancer>select *  from counter;
+--------------+------------------+---------------+---------------+
| client_count | freelancer_count | payment_count | project_count |
+--------------+------------------+---------------+---------------+
|            4 |                4 |             6 |             6 |
+--------------+------------------+---------------+---------------+
1 row in set (0.06 sec)
```

Table 8.1

9. FRONT END DEVELOPMENT

Home Page:

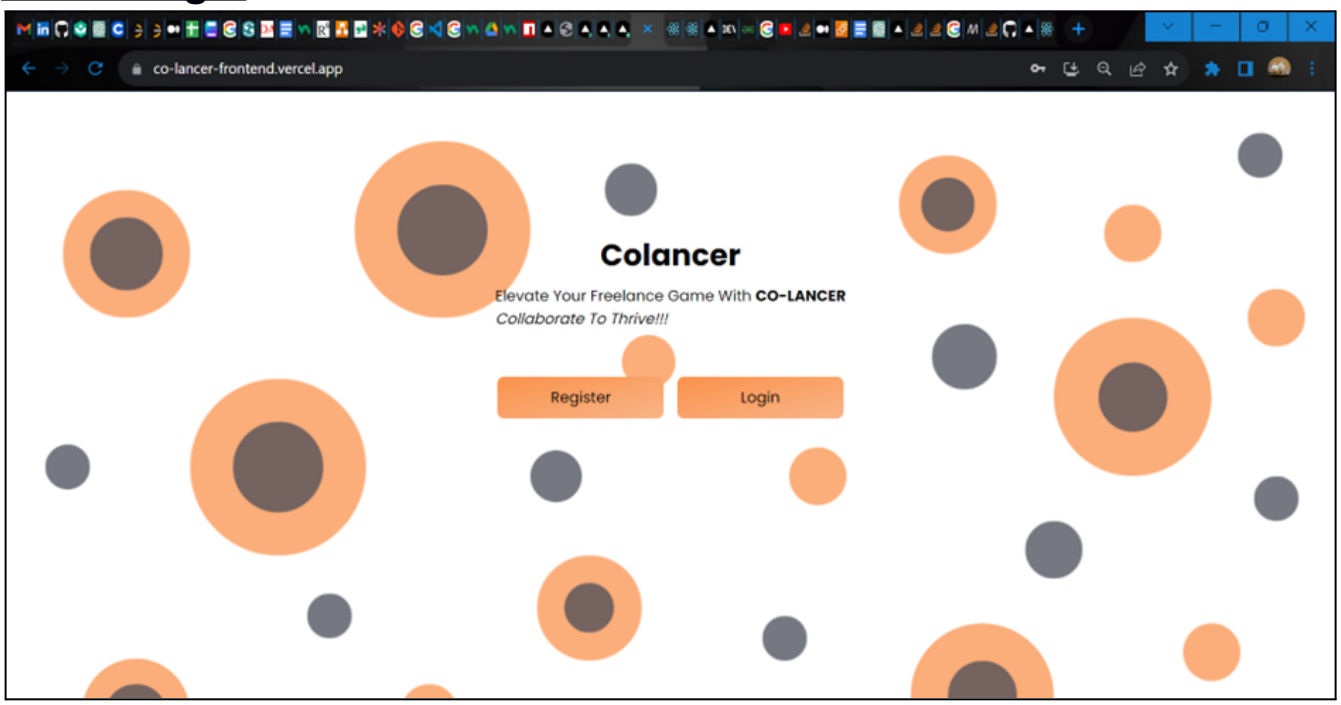

Fig. 9.1

Registration:

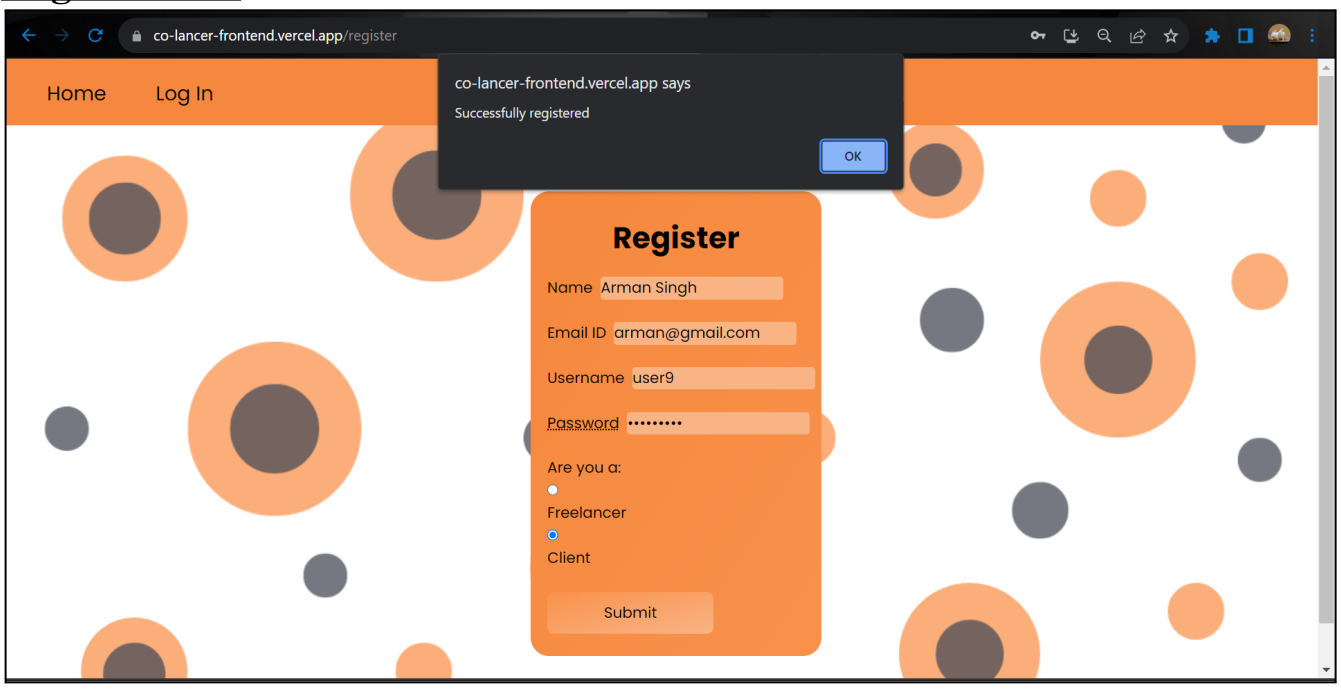

Fig. 9.2

Client Registration

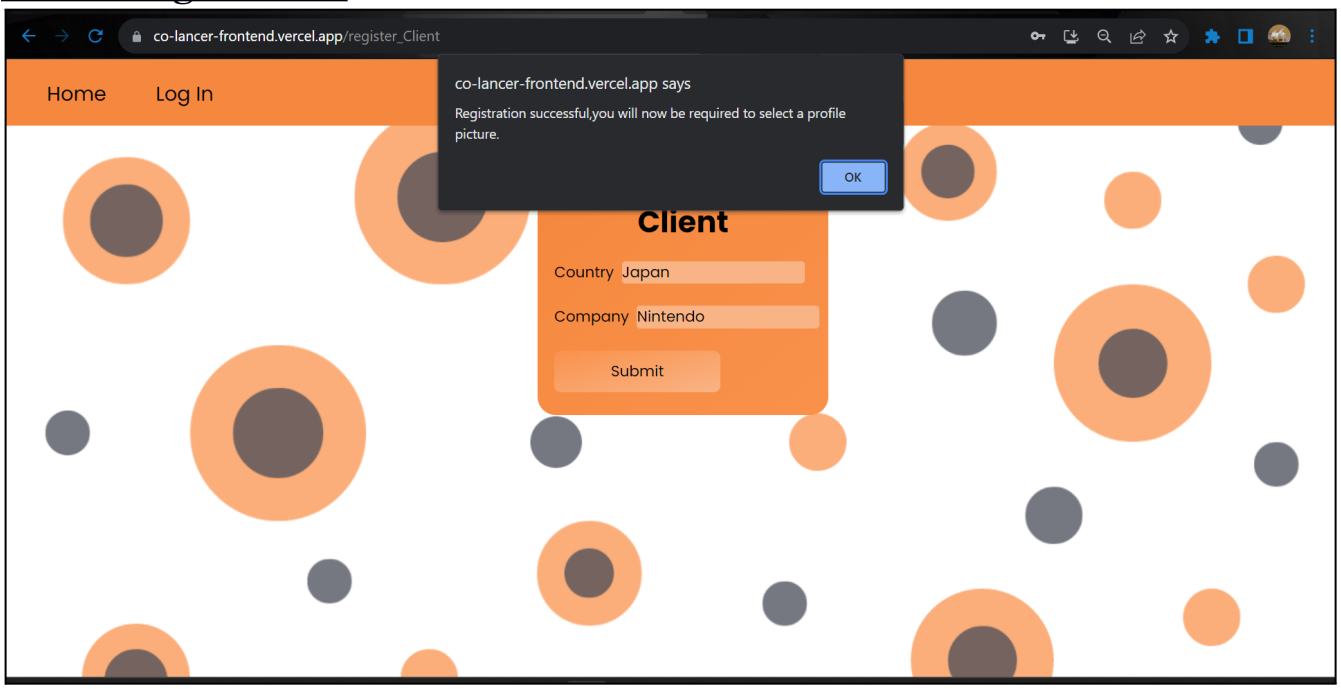

Fig. 9.3

Upload Profile

Max profile image size is 1MB for security reasons

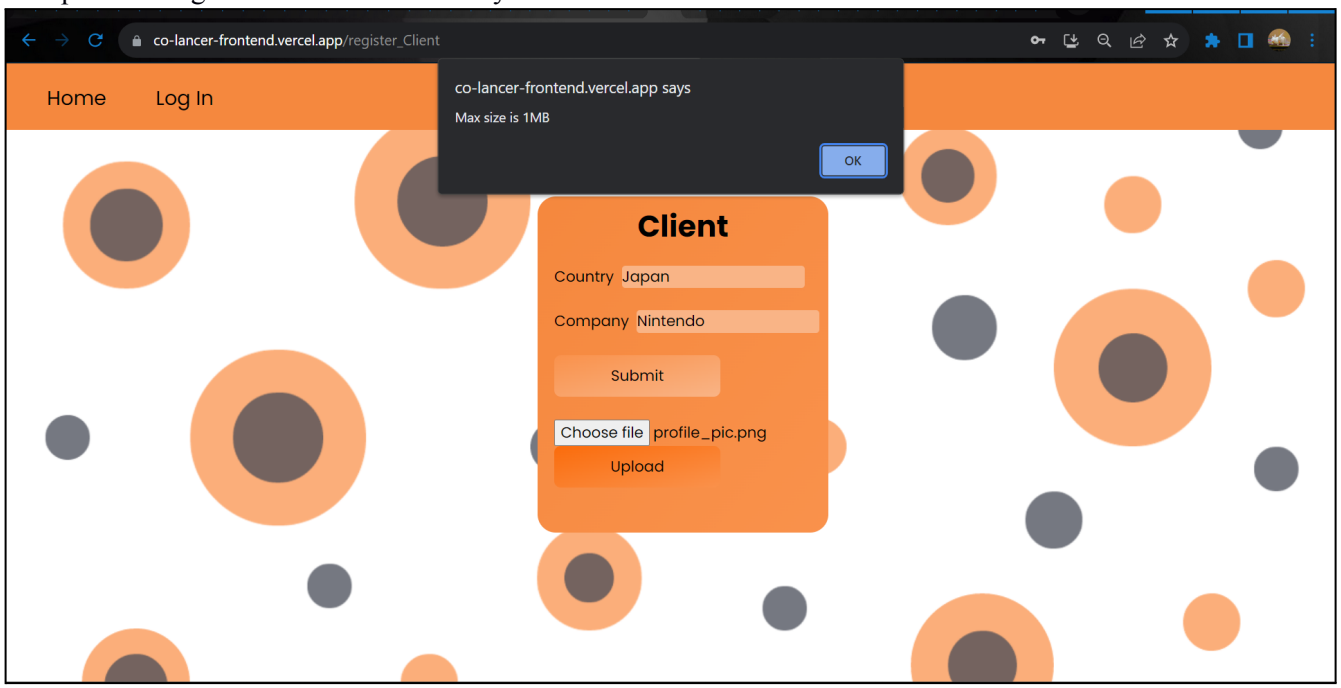

Fig. 9.4

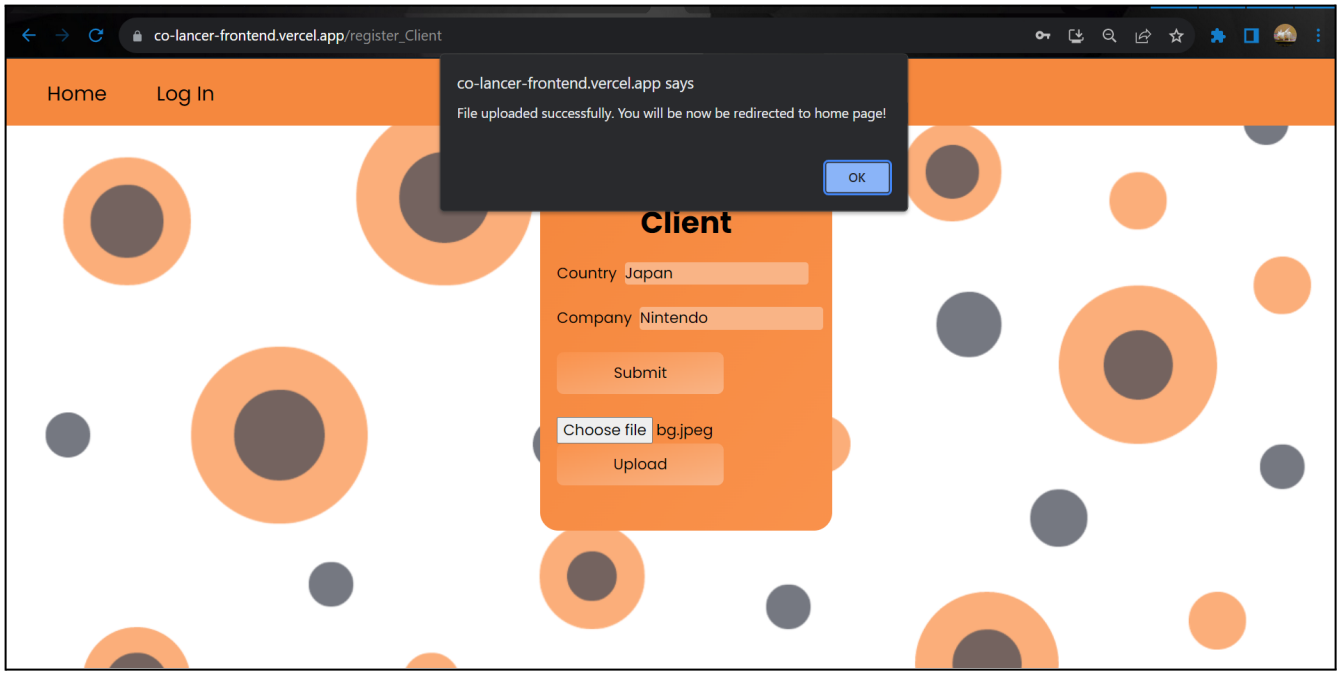

Fig. 9.5

Freelancer Registration

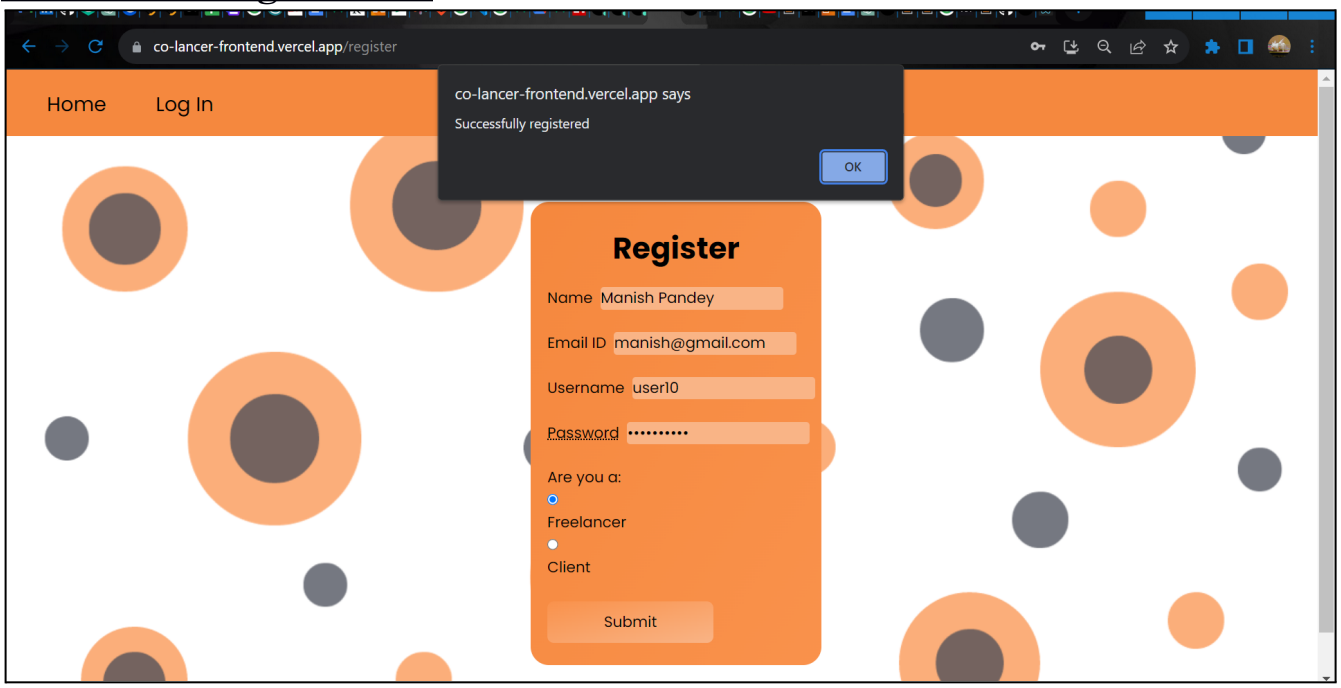

Fig. 9.6

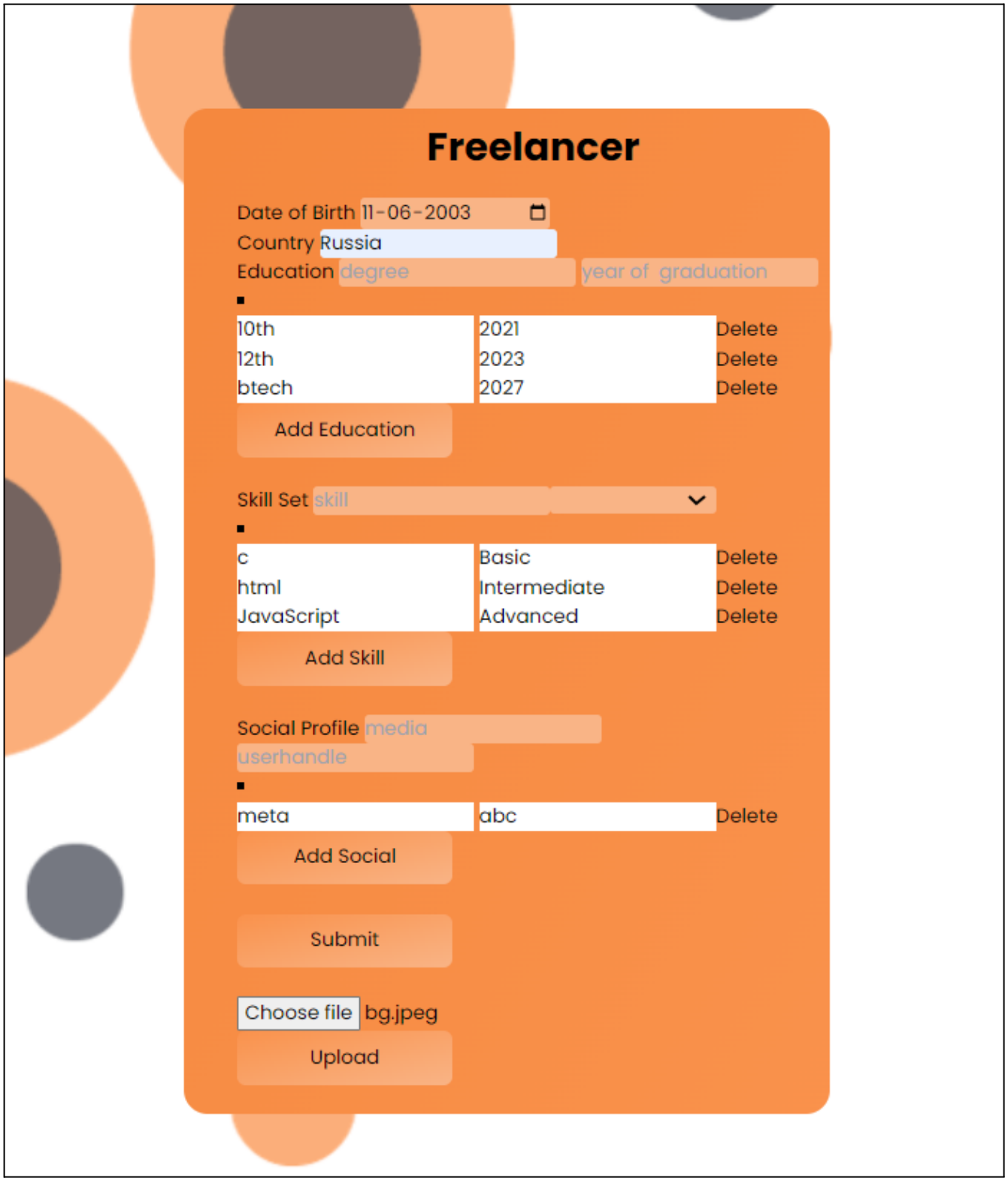
Freelancer
Date of Birth 11-06-2003
Country Russia
Education degree year of graduation
10th 2021 Delete
12th 2023 Delete
btech 2027 Delete
Add Education
Skill Set skill
c Basic Delete
html Intermediate Delete
JavaScript Advanced Delete
Add Skill
Social Profile media
userhandle
meta abc Delete
Add Social
Submit
Choose file bg.jpeg
Upload

Fig. 9.7

Login:

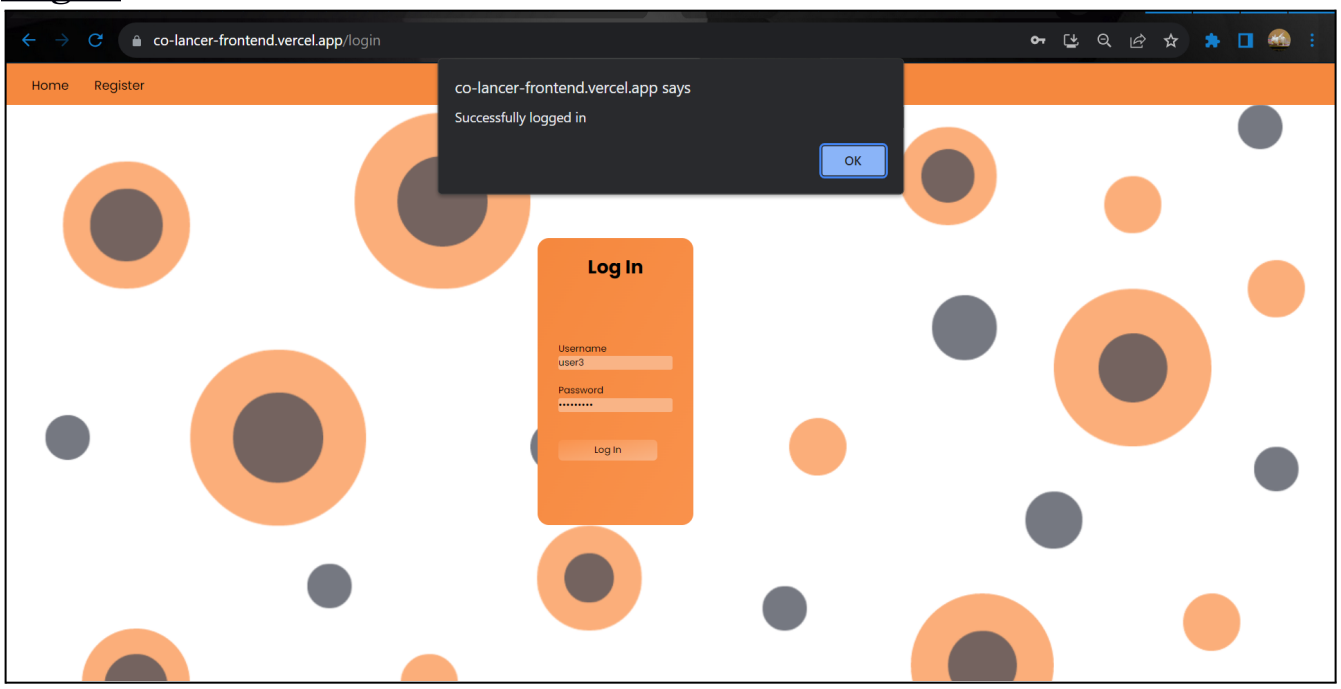

Fig. 9.8

Freelancer Profile:

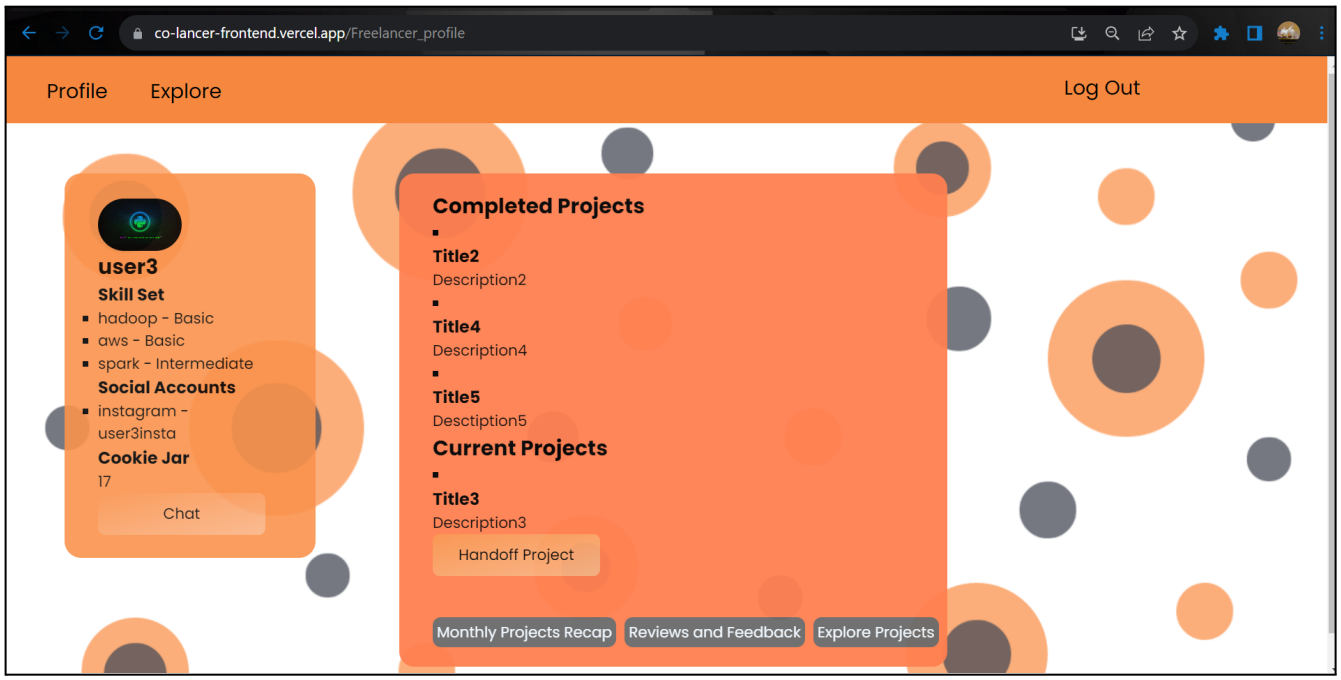

Fig. 9.9

Monthly Recap:

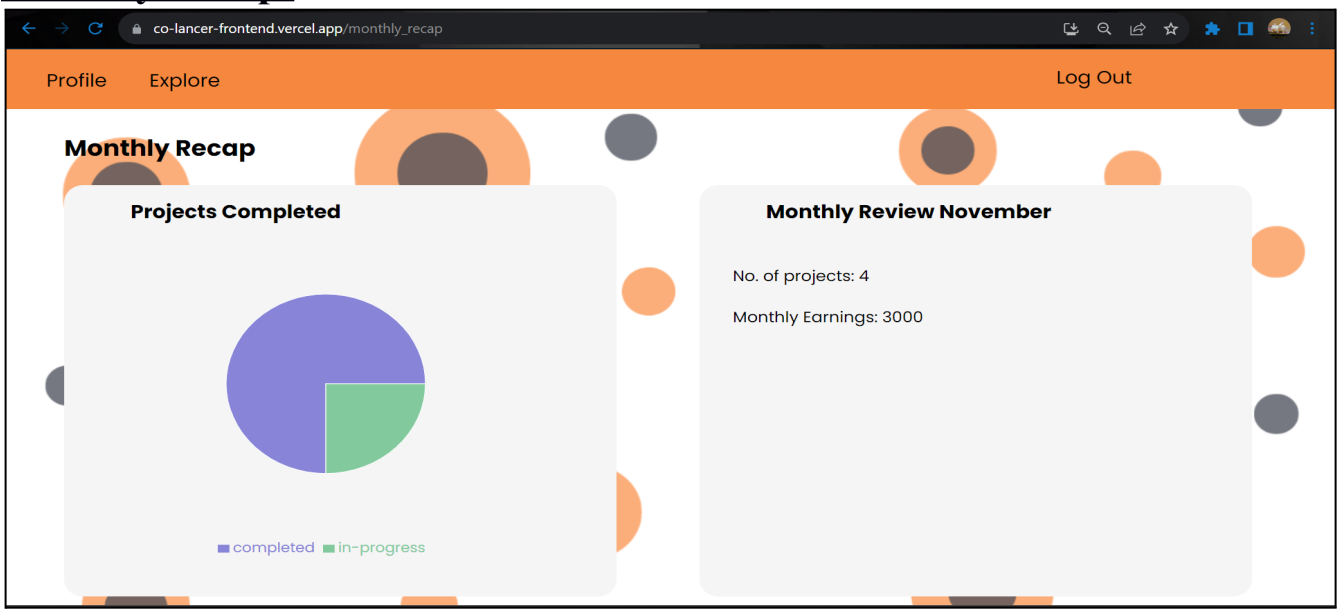

Fig. 9.10

Chat

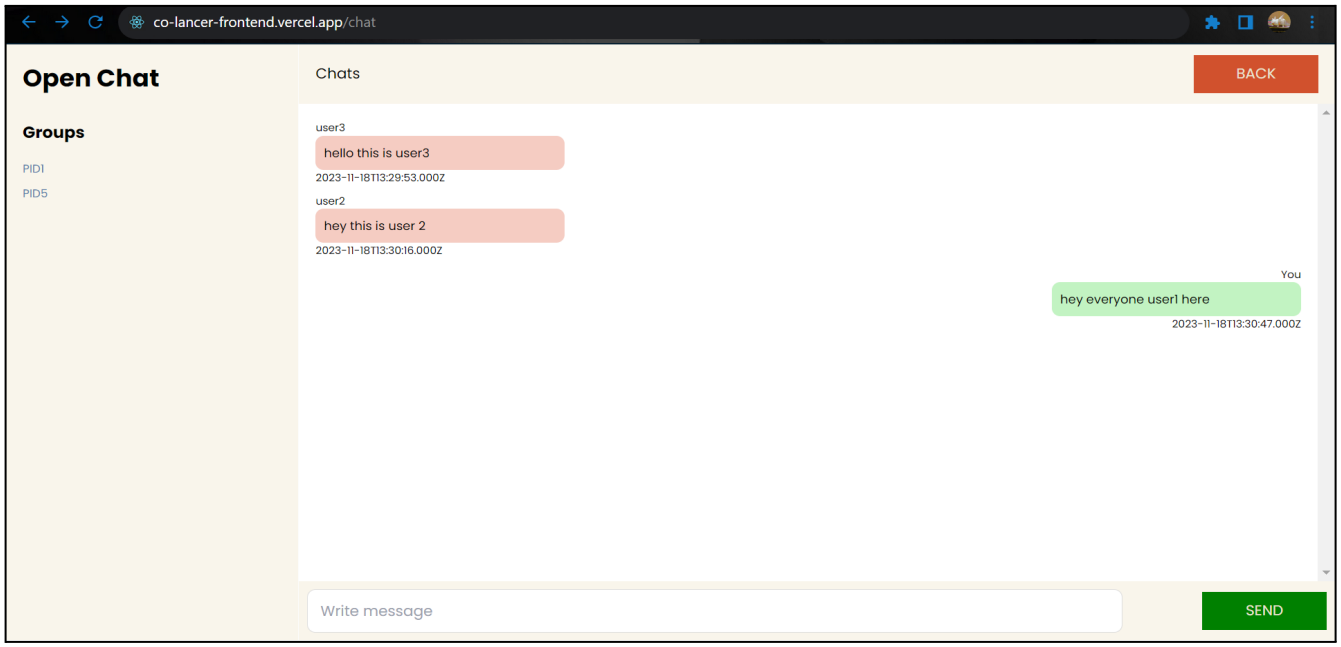

Fig. 9.11

Reviews:

Fig. 9.12

Explore Projects:

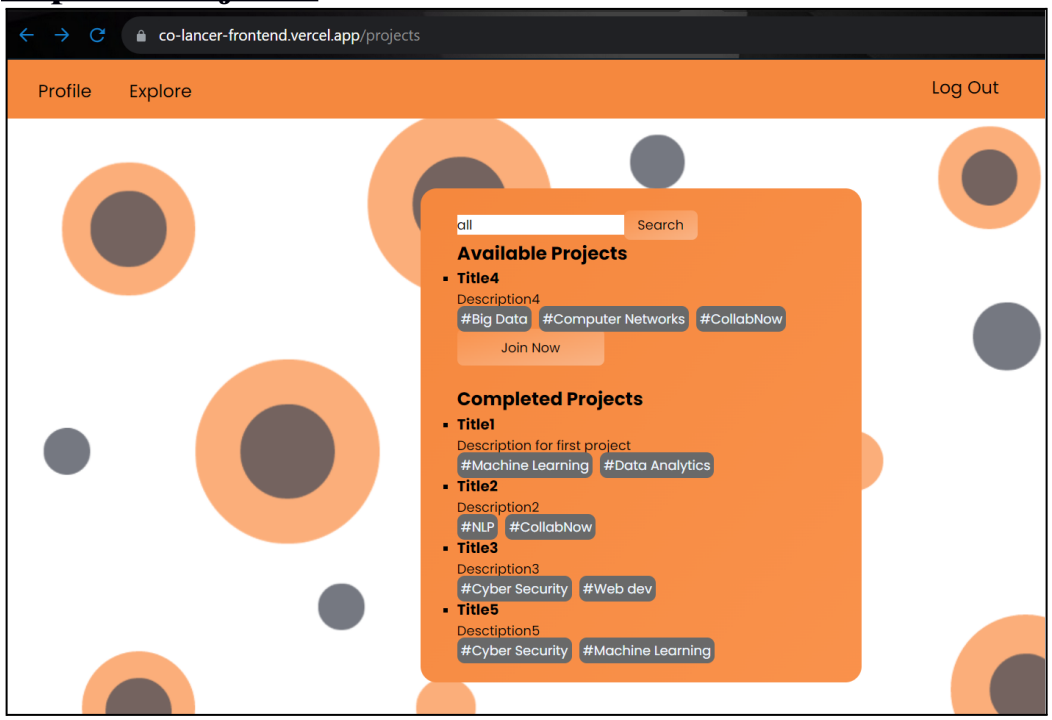

Fig. 9.13

Project Details:

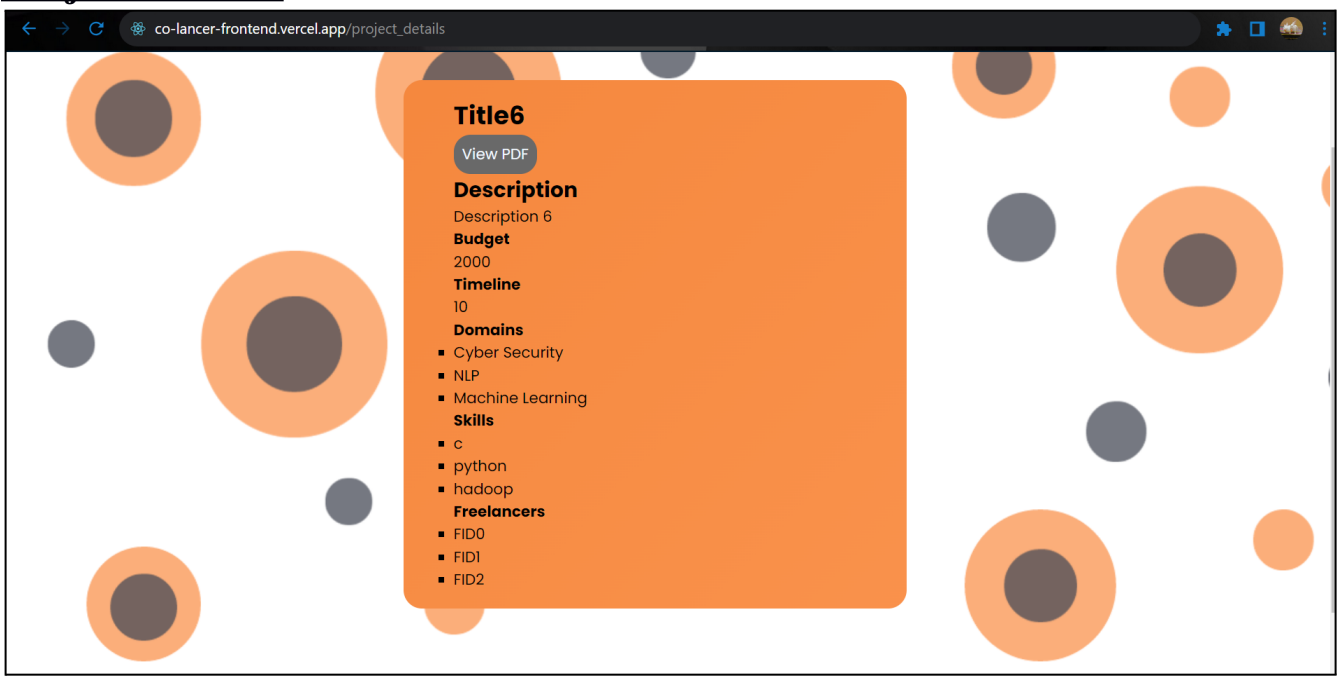

Fig. 9.14

PDF Display:

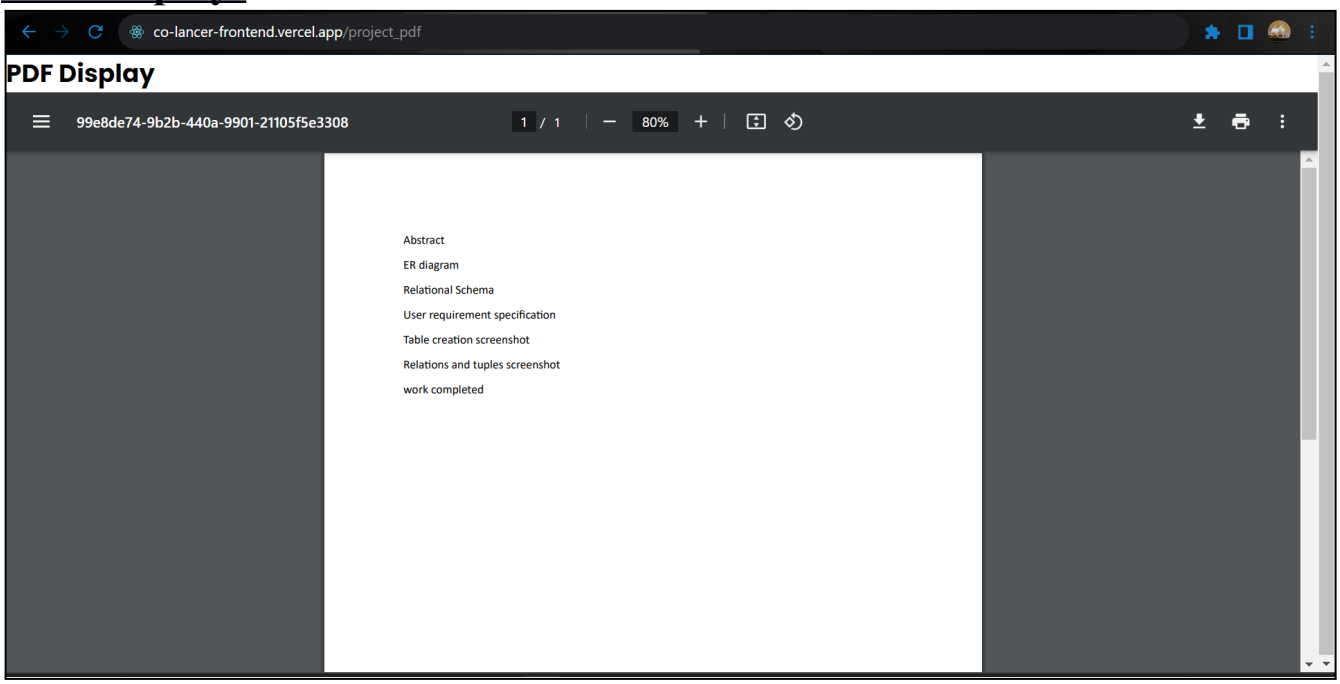

Fig. 9.15

Client Profile:

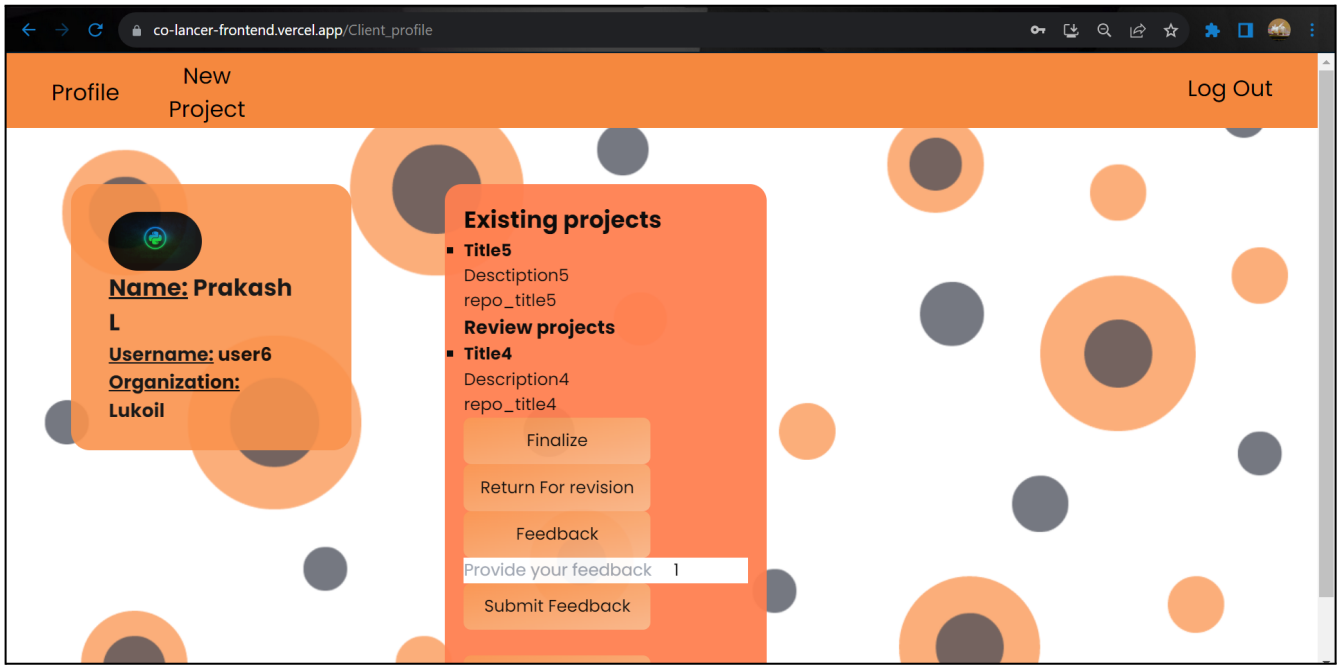

Fig. 9.16

Creating new project:

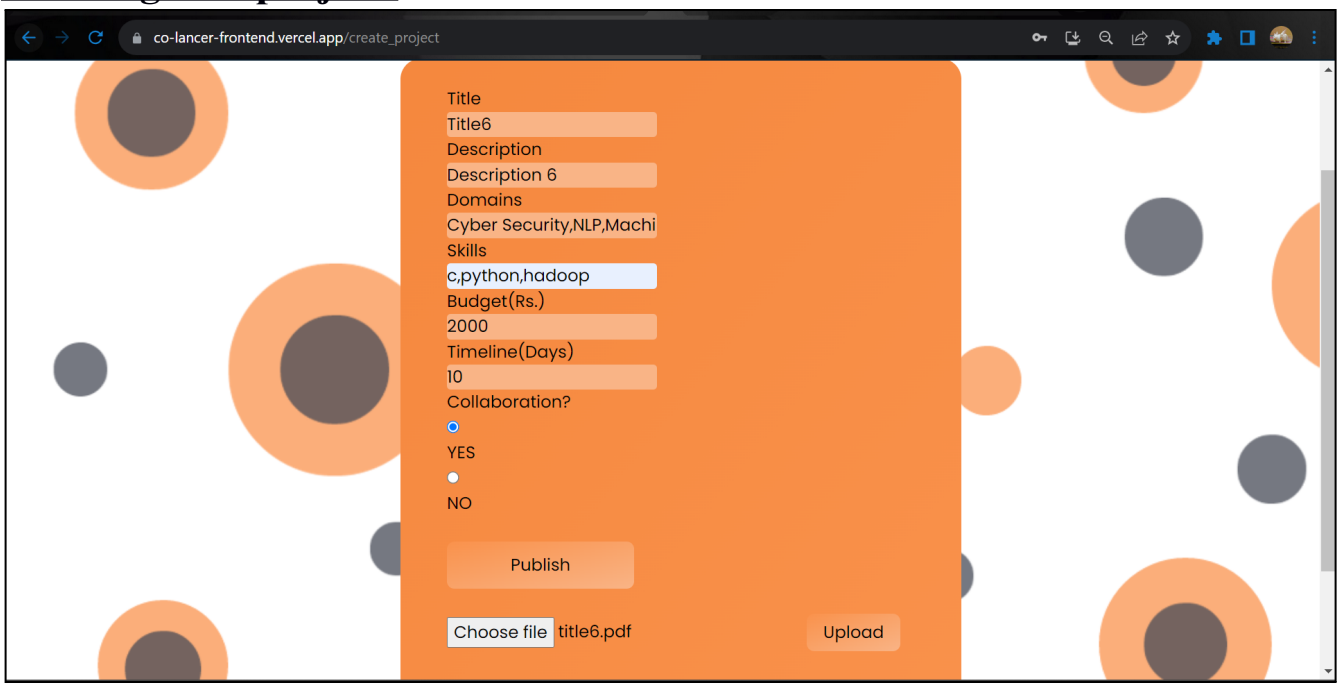

Fig. 9.17

Payment:

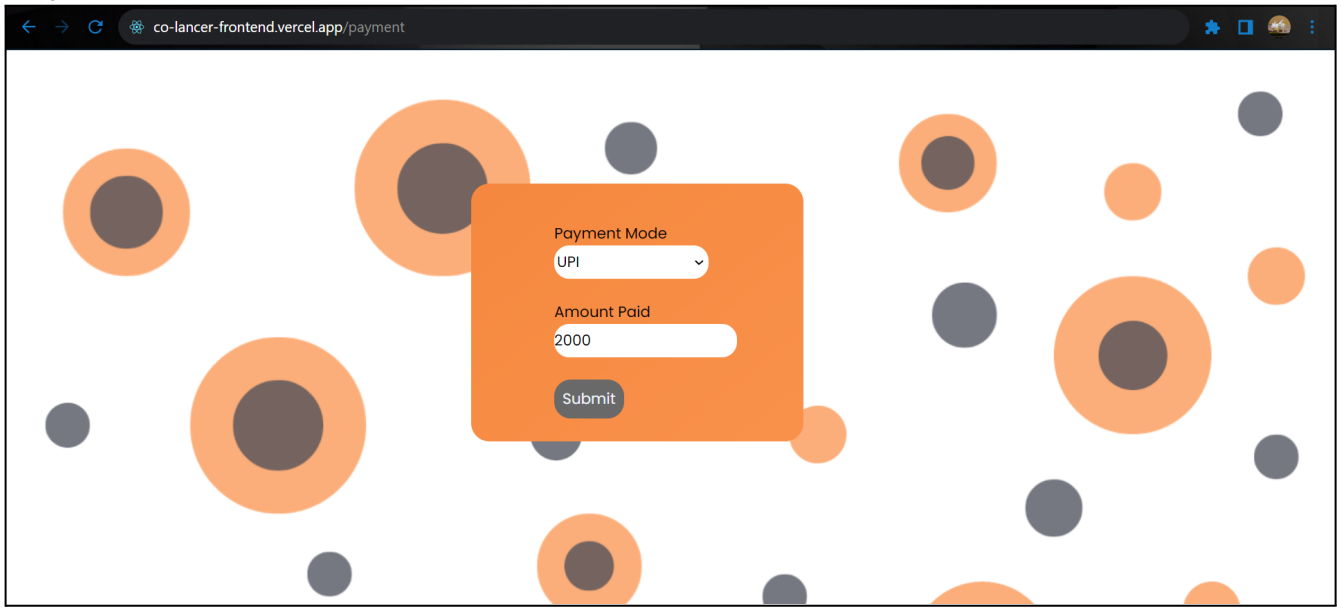

Fig. 9.18

SOURCE CODE:

https://github.com/QubitMatrix/Co-Lancer

REFERENCES

[1] https://www.w3schools.com/sql

[2] https://vercel.com/

[3] https://dev.to/tradecoder/how-to-run-front-end-and-backend-together-in-react-js-and-express-js-with-concurrently-package-5fga

[4] https://www.geeksforgeeks.org/how-to-deploy-mern-application-on-vercel/

[5] https://stackoverflow.com/

[6] https://medium.com/nerd-for-tech/how-to-setup-mysql-database-on-aws-rds-relational-database-service-f5a186ccbadb

[7] https://react.dev/

[8] https://nodejs.org/en/

[9] https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_GettingStarted.CreatingConnecting.MySQL.html

[10] https://techtavern.wordpress.com/2013/06/17/mysql-triggers-and-amazon-rds/

[11] https://docs.aws.amazon.com/

[12] https://www.npmjs.com/

[13] https://developer.mozilla.org/en-US/docs/Web/API/Response/json

[14] https://www.codementor.io/@parthibakumarmurugesan/what-is-env-how-to-set-up-and-run-a-env-file-in-node-1pnyxw9yxj

[15] https://dev.mysql.com/doc/

[16] https://www.ibm.com/docs/en/qmf/12.2.0?topic=privileges-sql-grant-statement

[17] https://reactrouter.com/en/main/hooks/use-navigate

[18] https://www.geeksforgeeks.org/mysql-on-delete-cascade-constraint/

[19] https://www.postman.com/

[20] https://tailwindcss.com/

[21] https://github.com/NasreenKhalid/Blog-React-CRUD-MYSQL

[22] https://www.tabnine.com/code/javascript/functions/mysql/createConnection

APPENDIX A DEFINITIONS, ACRONYMS AND ABBREVIATIONS

ACRONYMS

Sl. No.	Acronyms	Definition
1.	SQL	Structured Query Language
2.	ER model	Entity Relationship Model
3.	DDL statements	Data Definition language
4.	DML statements	Data Manipulation language

DEFINITIONS

Sl.No	Terms	Definitions
1.	DDL statements	Statements that affect structure of table
2.	DML statements	Statements that add, change or delete data
3.	Freelancers	Individuals who are looking for projects to work on
4.	Clients	Companies/ individuals who offer projects