

CS 432: Databases

Assignment 3: Implementing a Web App using MySQL

Group Name: La Retro's

Topic: Placement Management

Group Members

G1: Aaryan Darad - 21110001

Abhay Kumar Upparwal - 21110004

Adit Kaushik - 21110010

Aditya Deshmukh - 21110014

G2: Ahaan Giriya - 21110015

Anugu Arun Reddy - 21110029

Gaurav Joshi - 21110065

Rutwik More - 21110133

3.1 Responsibility of G1:

G1 is supposed to design the front end and beautify accordingly using HTML/CSS/JS, you may use any other libraries for the front end, such as Bootstrap or jQuery. But flask is mandatory.

In this part, we will show the frontend design of the various parts of Databases.

Components of Website

There are a total of three views-

1. Student

- Profile
- Opportunities
- Status of opportunities

2. CDS

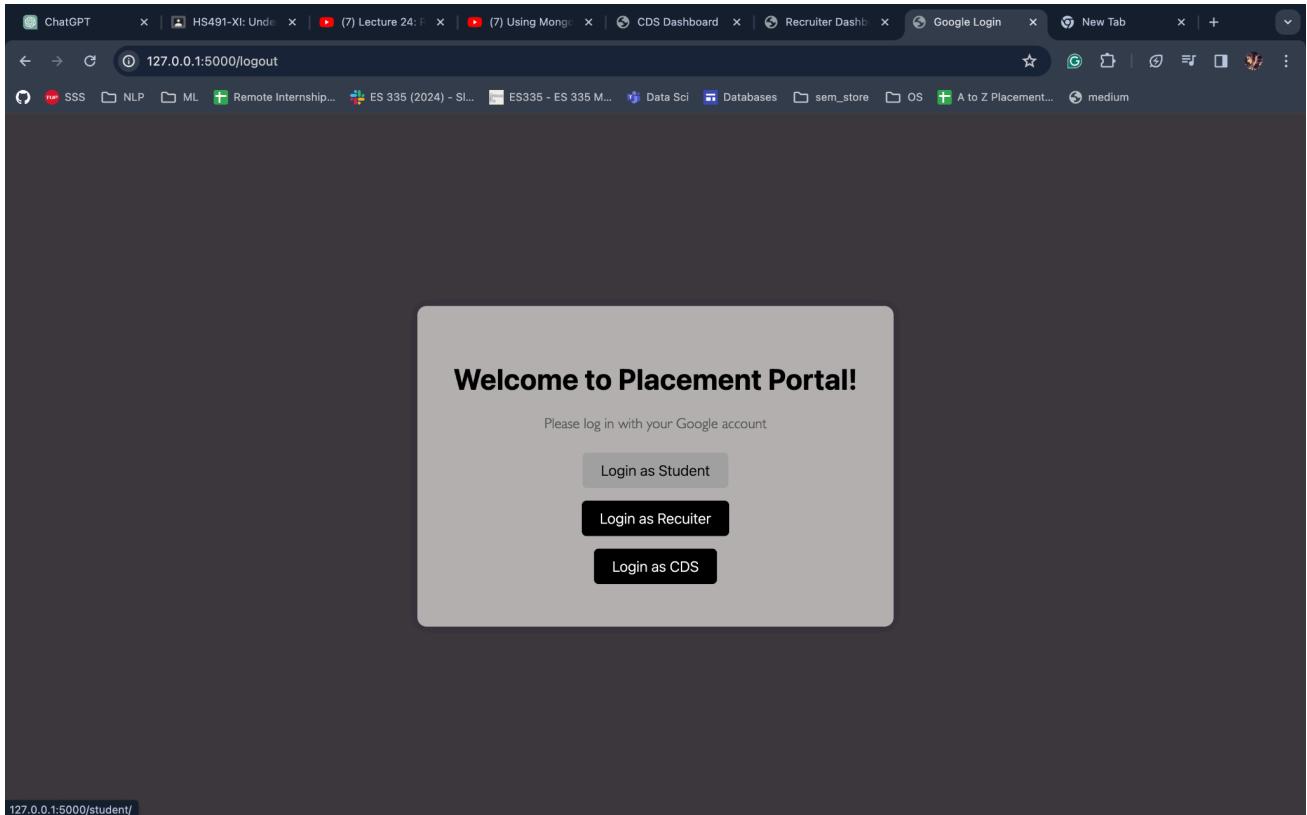
- Opportunities
- Add POC
- Student Details
- Company Details
- Add Placement Details
- See Placement Details

3. Recruiter

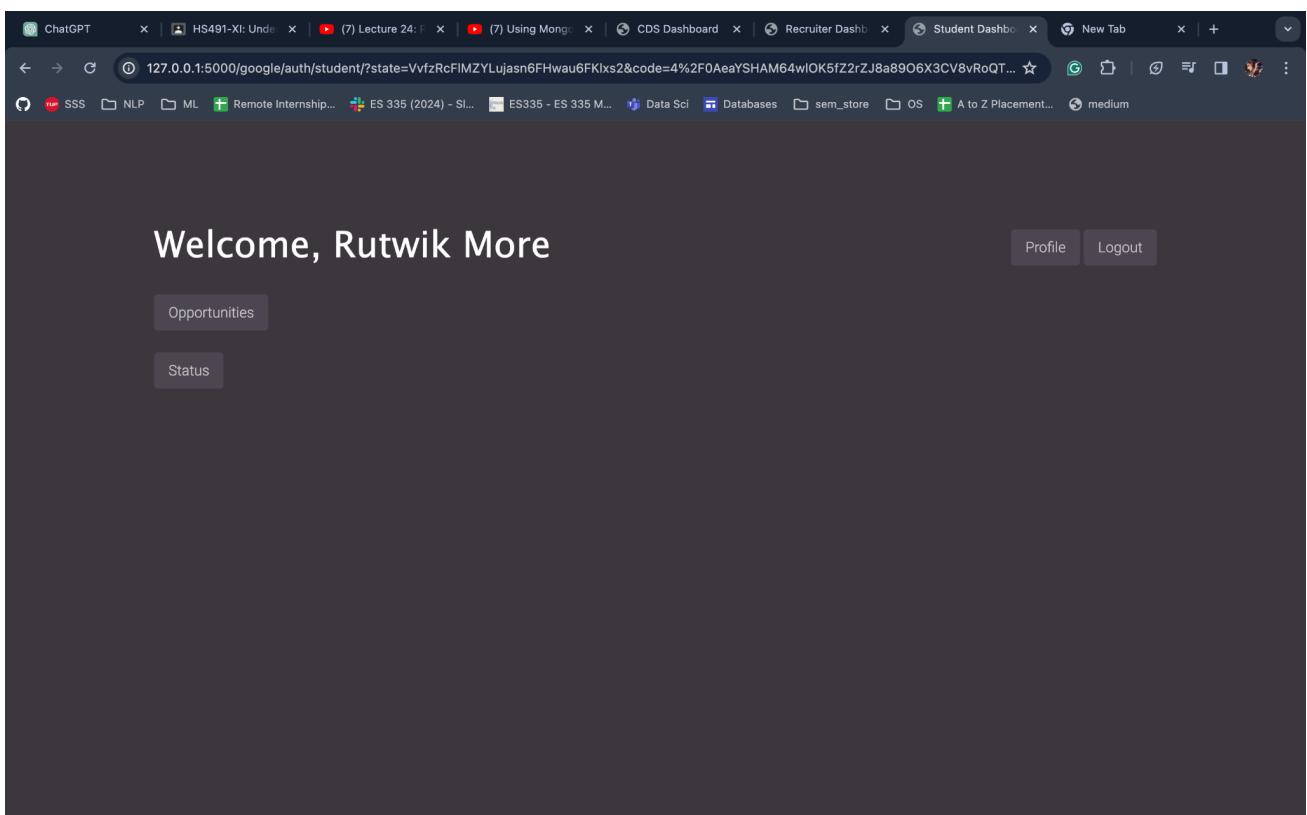
- Hosted Opportunities
- Host New Opportunities

1. Student

- Login Page



- Student Dashboard



- Student Profile

Create Profile

First Name

Middle Name

Last Name

Department

Active Backlog

Current Year

Minor

Contact Number

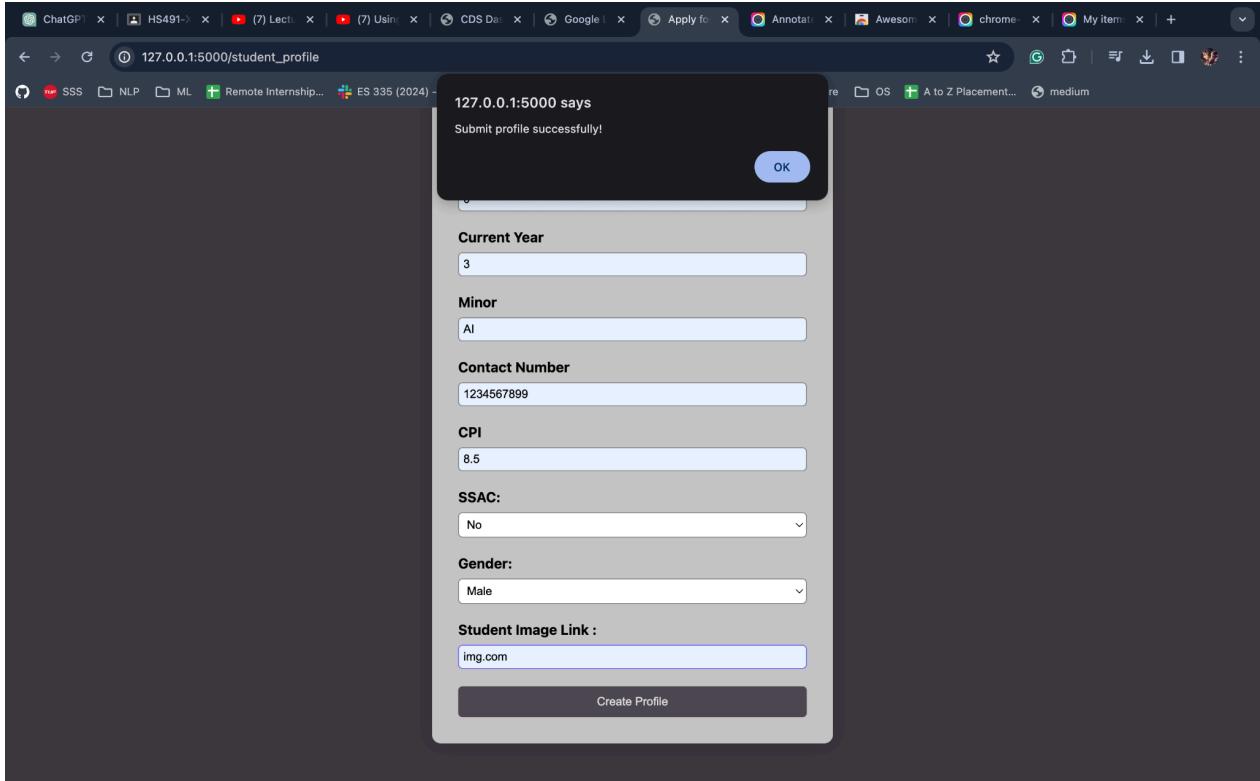
CPI

SSAC_or_not:

Gender:

Student Image Link :

- Pop Up on Creating Profile successfully



- Profile Page of Student

A screenshot of a web browser window showing the "Profile Page". The title bar indicates the URL is `127.0.0.1:5000/student_profile`. The page displays the following information:
First name : Rutwik
Middle name : Babruwan
Last name : More
Department: Computer Engineering
Active Backlog : 0
Gender : Male
Current Year : 3
Minor : AI
Email : babruwanmore@iitgn.ac.in
Contact Number : 1234567899
CPI : 8.5
SSAC_or_not : No
At the bottom left is an "Edit Profile" button, and at the top right are "Dashboard" and "Logout" buttons.

- Student can also edit their profile

Edit Profile

First Name
Rutwik

Middle Name
Babruwan

Last Name
More

Department
Computer Engineering

Active Backlog
0

Current Year
3

Minor
AI

Contact Number
1234567899

CPI
8.5

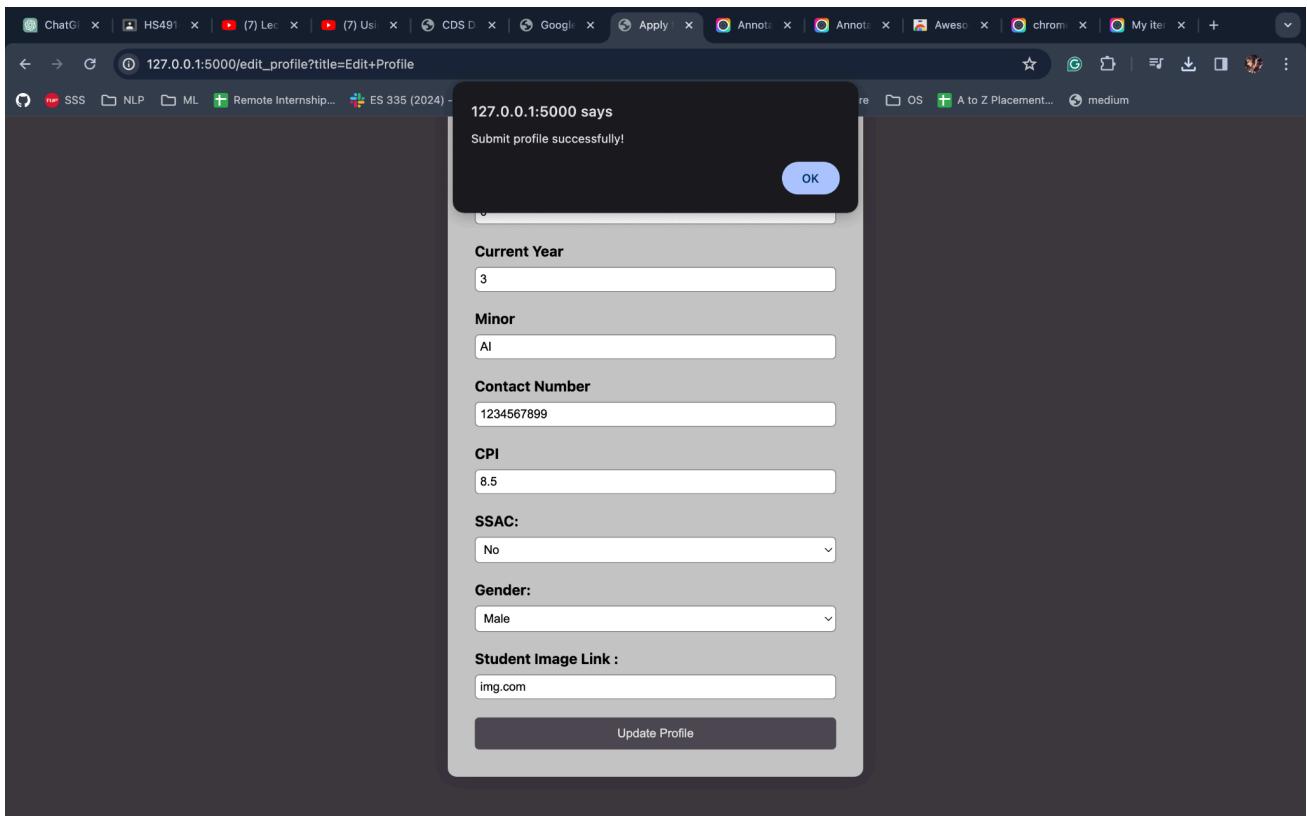
SSAC:
No

Gender:
Male

Student Image Link :
img.com

Update Profile

- Pop Up after editing the profile successfully



- Students can see the listed opportunities.

Opportunities								Dashboard	Logout
COMPANY	TITLE	DEPARTMENT	DESCRIPTION	SALARY	CPI REQUIREMENT	NUMBER OF POSITION	MORE DETAILS	APPLY	
Tesla	SDE	BTech	file.com	120000	6.0	2	<button>Details</button>	<button>Apply</button>	
Devin	ML	BTech MTech	bhoot.com	80000	7.0	4	<button>Details</button>	<button>Apply</button>	

- Student can see each opportunity in detail after clicking Details button

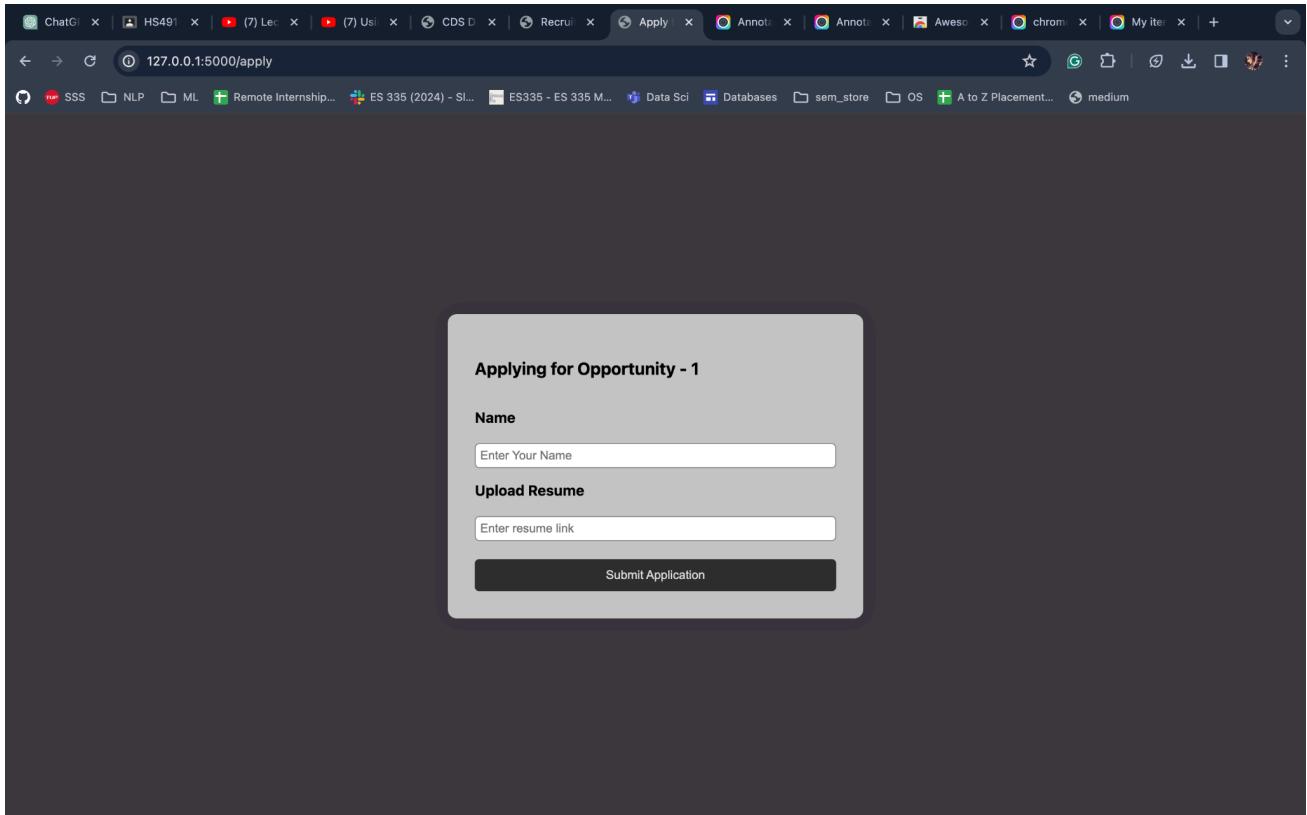
The screenshot shows a web browser window with the URL 127.0.0.1:5000/opportunity_details_student/1. The page title is "Opportunity Details". On the right, there are "Dashboard" and "Logout" buttons. The main content area displays various job details:

- Title : SDE
- Positions : 2
- Salary : 120000
- Requirements : file.com
- CPI requirement : 6.0
- Active Backlog : 0
- Student Year Requirement : 4
- Program requirement : BTech
- Job Description : job.com

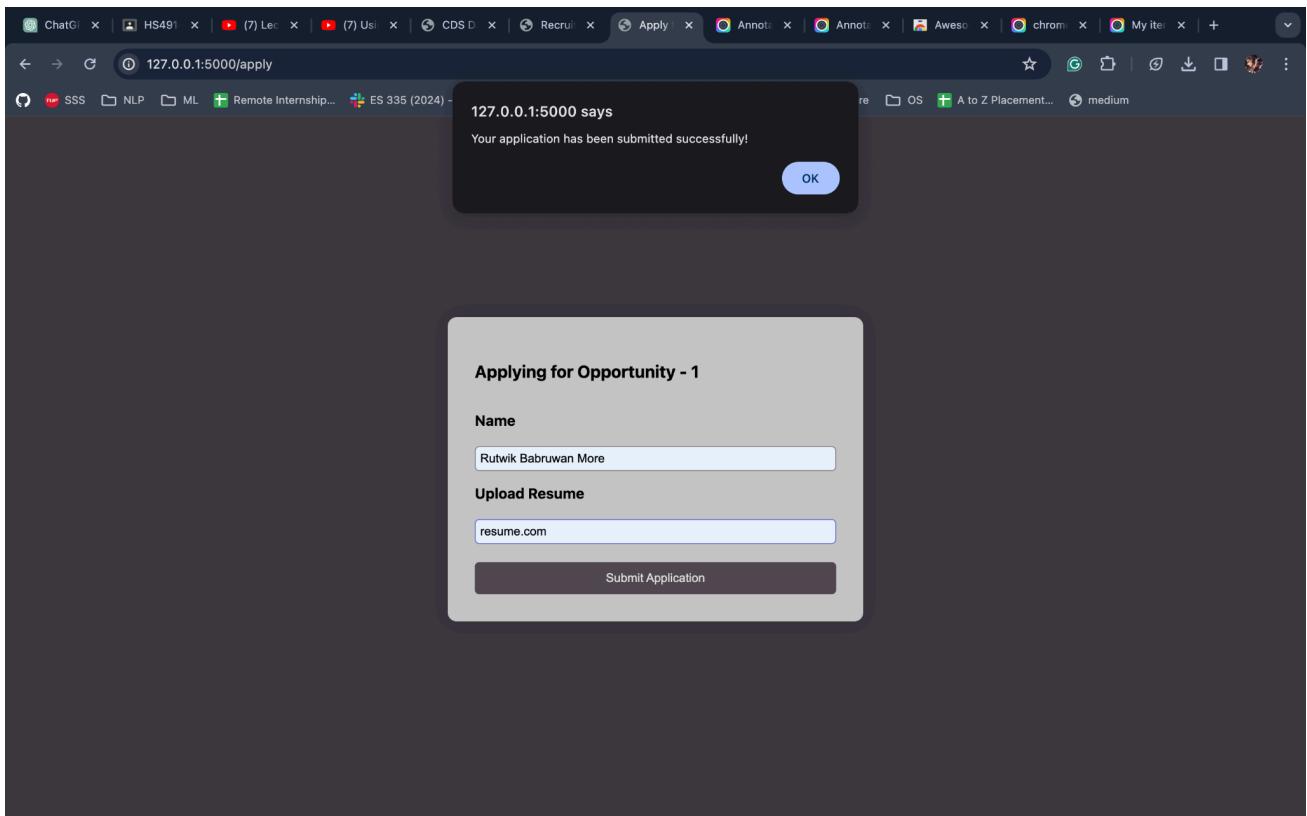
Below this, under "Round Details:", is a table:

ROUND NUMBER	TYPE	VENUE	DATE	START TIME	END TIME
1	Online	Gmeet	2024-04-26	13:20:00	13:30:00
2	Offline	AB 9/101	2024-04-27	14:20:00	15:40:00

- Student can also apply for the opportunity by clicking on Apply button



- Student get pop up after successful application



- Student can see the status of applied opportunities.

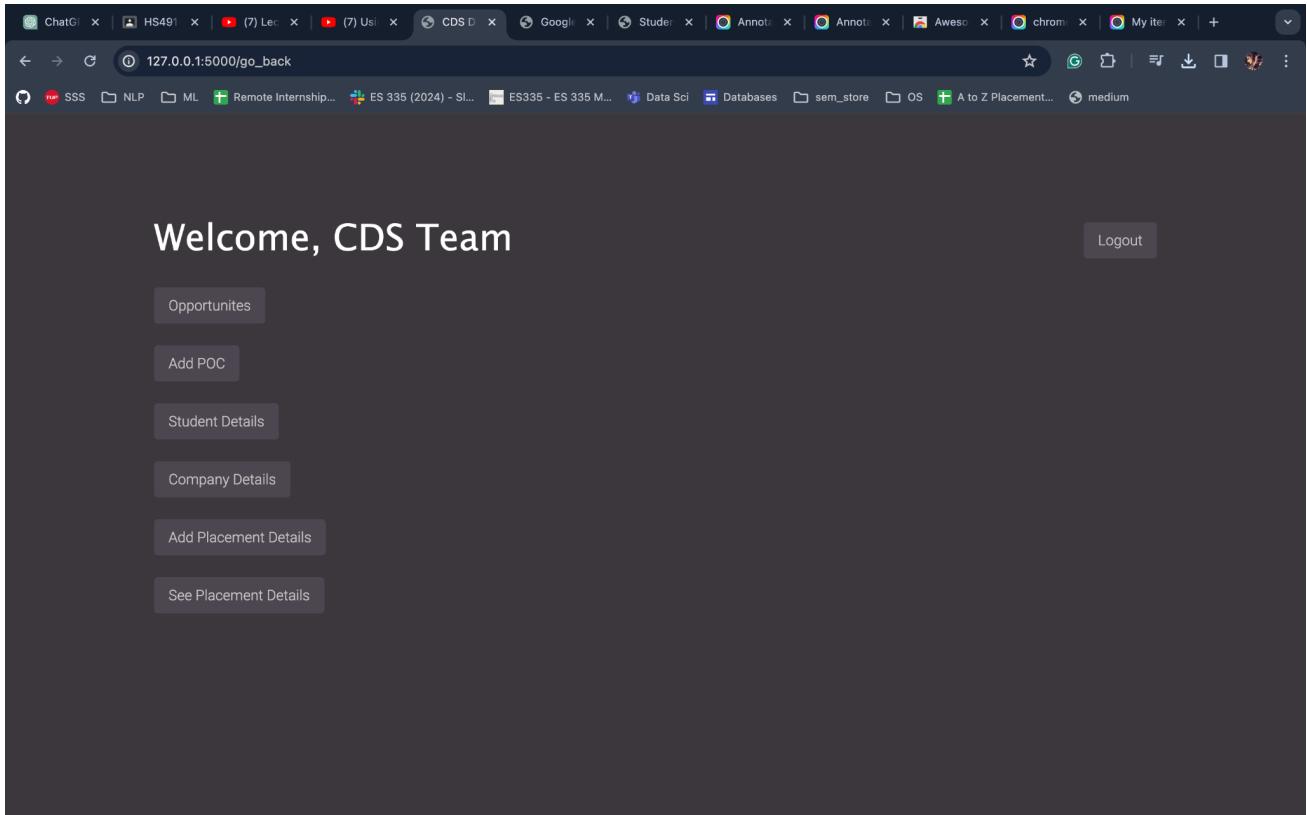
The screenshot shows a web browser window with the URL `127.0.0.1:5000/status_opp_student`. The page title is "Opportunities Status". The content is a table with three columns: COMPANY, TITLE, and STATUS. A single row is present, showing Tesla in the COMPANY column, SDE in the TITLE column, and Accepted in the STATUS column. At the top right of the page, there are "Dashboard" and "Logout" buttons. The browser's address bar and various tabs are visible at the top.

COMPANY	TITLE	STATUS
Tesla	SDE	Accepted

Dashboard Logout

1. CDS Employee:

- Dashboard of CDS



- CDS can see all the opportunities listed by recruiter by clicking on opportunity.

COMPANY	TITLE	DEPARTMENT	DESCRIPTION	SALARY	REMOTE	CPI REQUIREMENT	NUMBER OF POSITION	POC EMAIL
Tesla	SDE	BTech	file.com	120000	Yes	6.0	2	rutwikmore1440@gmail.com
Devin	ML	BTech MTech	bhoot.com	80000	Yes	7.0	4	rutwikmore1440@gmail.com

- CDS can add Person of Contact details by clicking on Add POC button.

Create Profile for POC

First Name:
Alan

Middle Name:
M

Last Name:
Walker

Email:
rutwikmore1440@gmail.com

Designation:
CEO

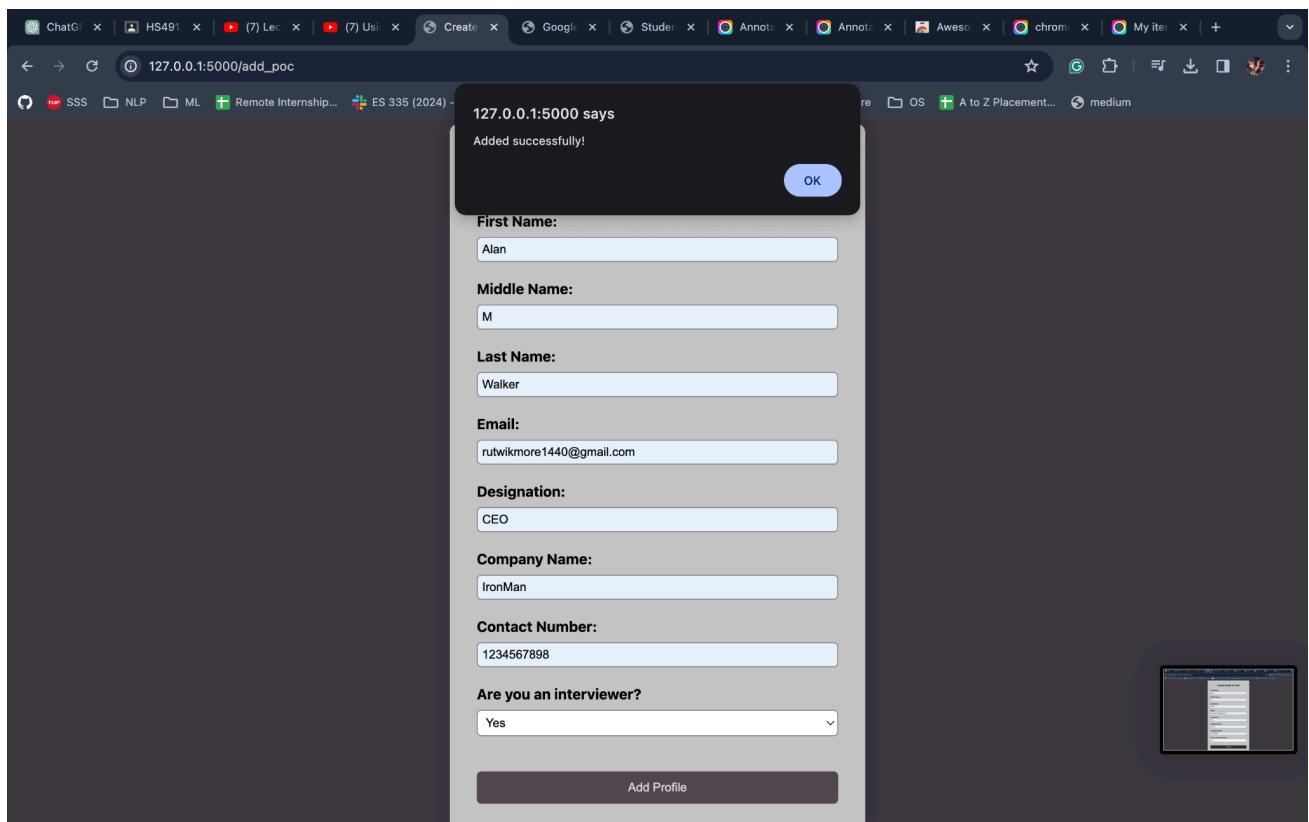
Company Name:
IronMan

Contact Number:
1234567898

Are you an interviewer?
Yes

Add Profile

- CDS will get pop up after successfully adding the POC



- CDS can see the Details of each student.

Student Details

Name	Degree	Email	Action
Alice Johnson	Electrical Engineering	alice.johnson@iitgn.ac.in	View Details
Bob Williams	Mechanical Engineering	bob.williams@iitgn.ac.in	View Details
Emily Brown	Chemical Engineering	emily.brown@iitgn.ac.in	View Details
Michael Taylor	Civil Engineering	michael.taylor@iitgn.ac.in	View Details
Sophia Anderson	Biomedical Engineering	sophia.anderson@iitgn.ac.in	View Details
Rutwik More	Computer Engineering	babruwanmore@iitgn.ac.in	View Details

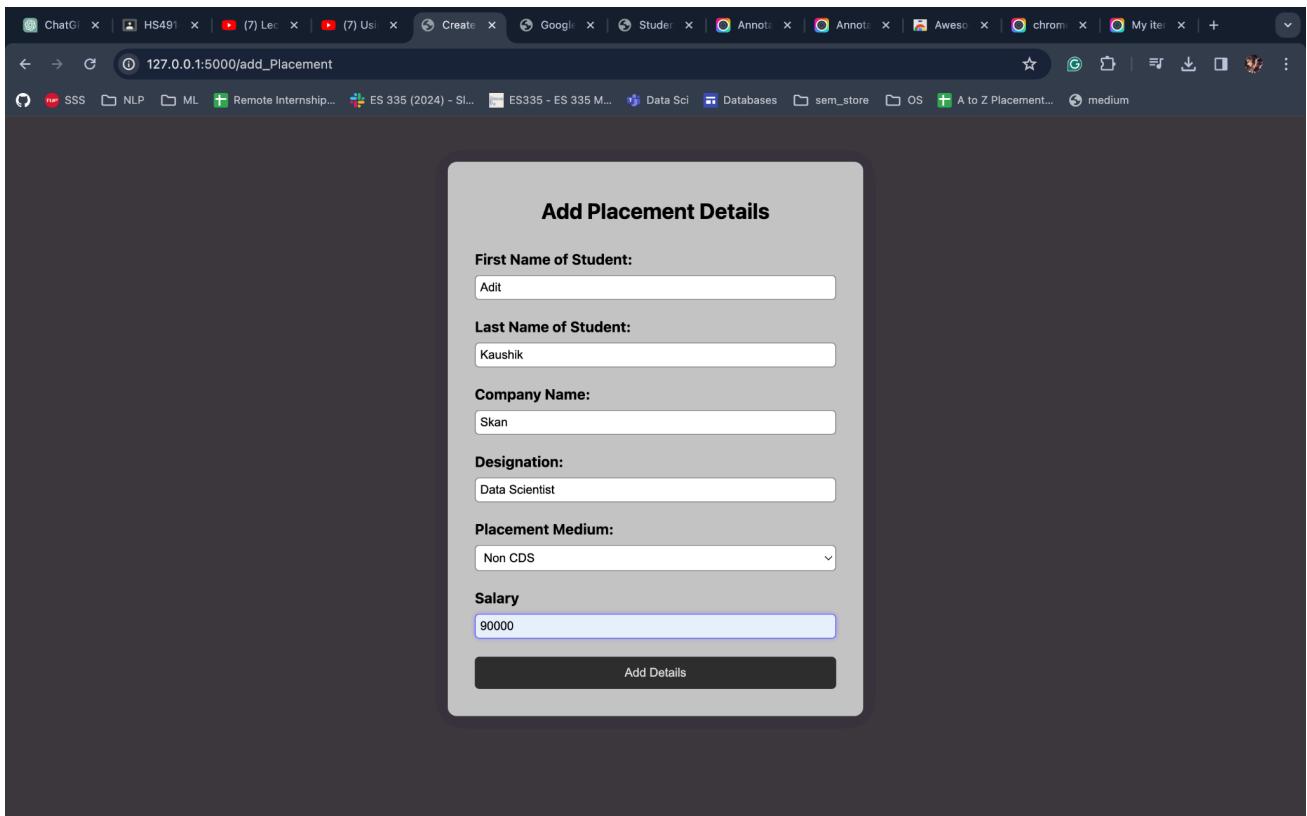
- CDS can See student details in detail by clicking on the button 'View Details'.

Student Profile

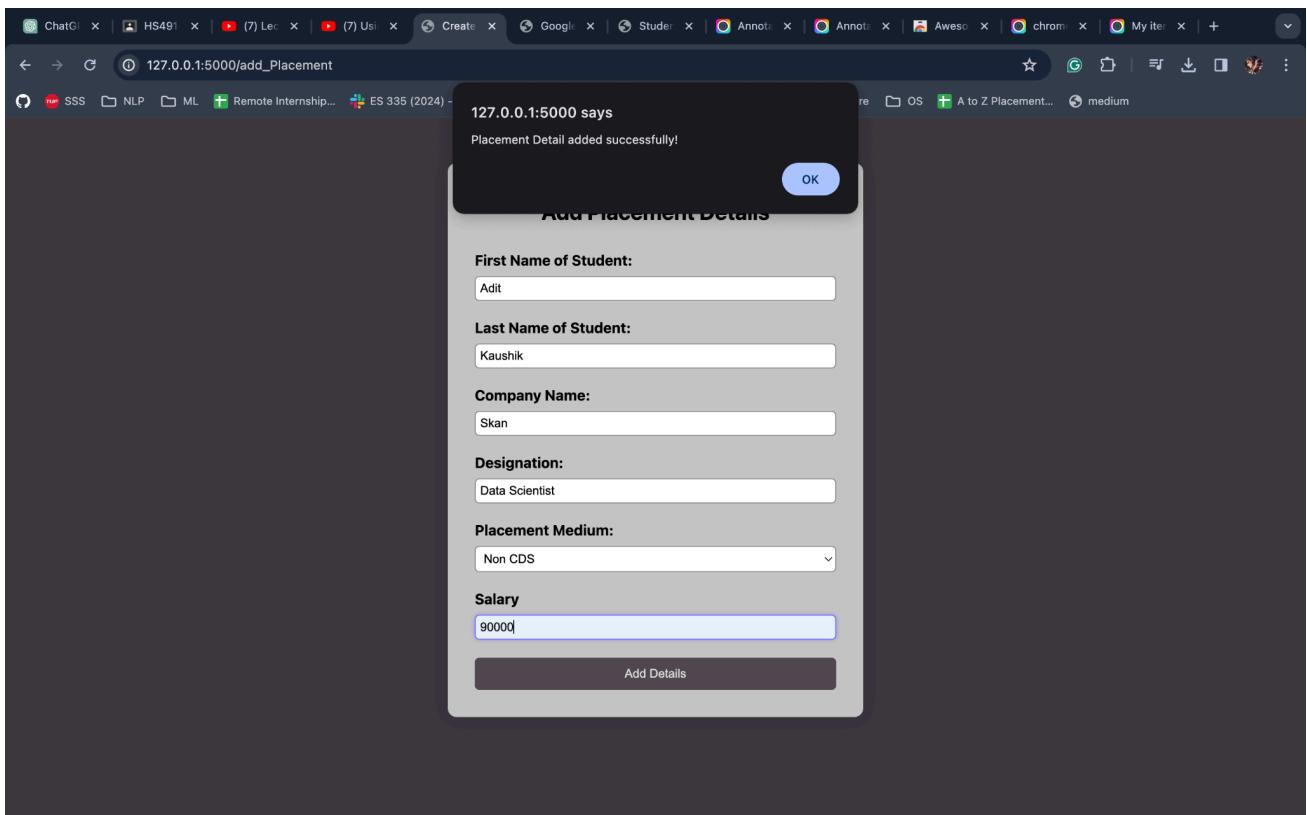
Fist name : Rutwik
Middle name : Babruwan
Last name : More
Department: Computer Engineering
Active Backlog : 0
Gender : Male
Current Year : 3
Minor : AI
Email : babruwanmore@iitgn.ac.in
Contact Number : 1234567899
CPI : 8.5
SSAC_or_not : No

[Go Back](#)

- CDS can add placement details of each student.



- CDS will get pop up after adding the details successfully.



- CDS can see placement details of each student.

The screenshot shows a web application titled "Placement Details". The page has a dark background with white text. At the top right are "Dashboard" and "Logout" buttons. Below is a table with columns: STUDENT NAME, COMPANY, SALARY, ROLE, and MEDIUM. Two rows of data are shown:

STUDENT NAME	COMPANY	SALARY	ROLE	MEDIUM
Adit Kaushik	Skan	90000	Data Scientist	Non_CDS
Gaurav Joshi	Purdue	100000	Researcher	CDS

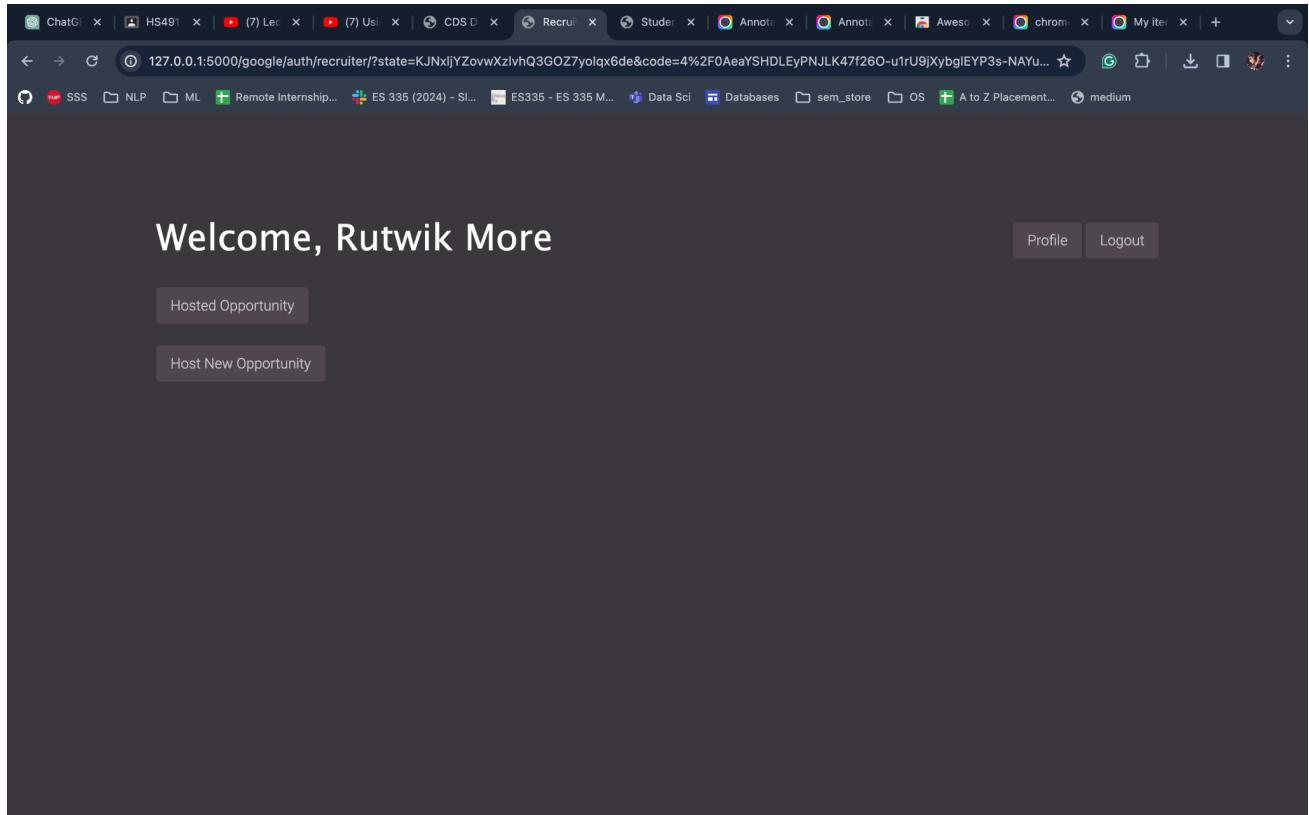
- CDS can see details of all companies.

The screenshot shows a web application titled "Company Details". The page has a dark background with white text. At the top right are "Dashboard" and "Logout" buttons. Below is a table with columns: COMPANY, PERSON OF CONTACT, EMAIL, and CONTACT NO. Two rows of data are shown:

COMPANY	PERSON OF CONTACT	EMAIL	CONTACT NO
Avengers	Captain America	rutwik1440@gmail.com	987654321
IronMan	Alan Walker	rutwikmore1440@gmail.com	1234567898

2. Person of Contact (POC):

- Dashboard of Recruiter



Recruiter can host new Opportunities.

Create Opportunity

Opportunity Title:
SDE

Company Name:
Tesla

Number of Positions:
2

Specific Requirements File Link:
file.com

Minimum CPI Requirement:
6

Number of Active Backlogs:
0

Student Year Requirement:
4

Program Requirement:
BTech

Job Description File Link:
job.com

Salary:
120000

Number of Rounds:
2

Round 1

Round Type:
Online

Round Date:
26 / 04 / 2024

Venue:
Gmeet

Start Time:
01 : 20 PM

End Time:
01 : 30 PM

Round 2

Round Type:
Offline

Round Date:
27 / 04 / 2024

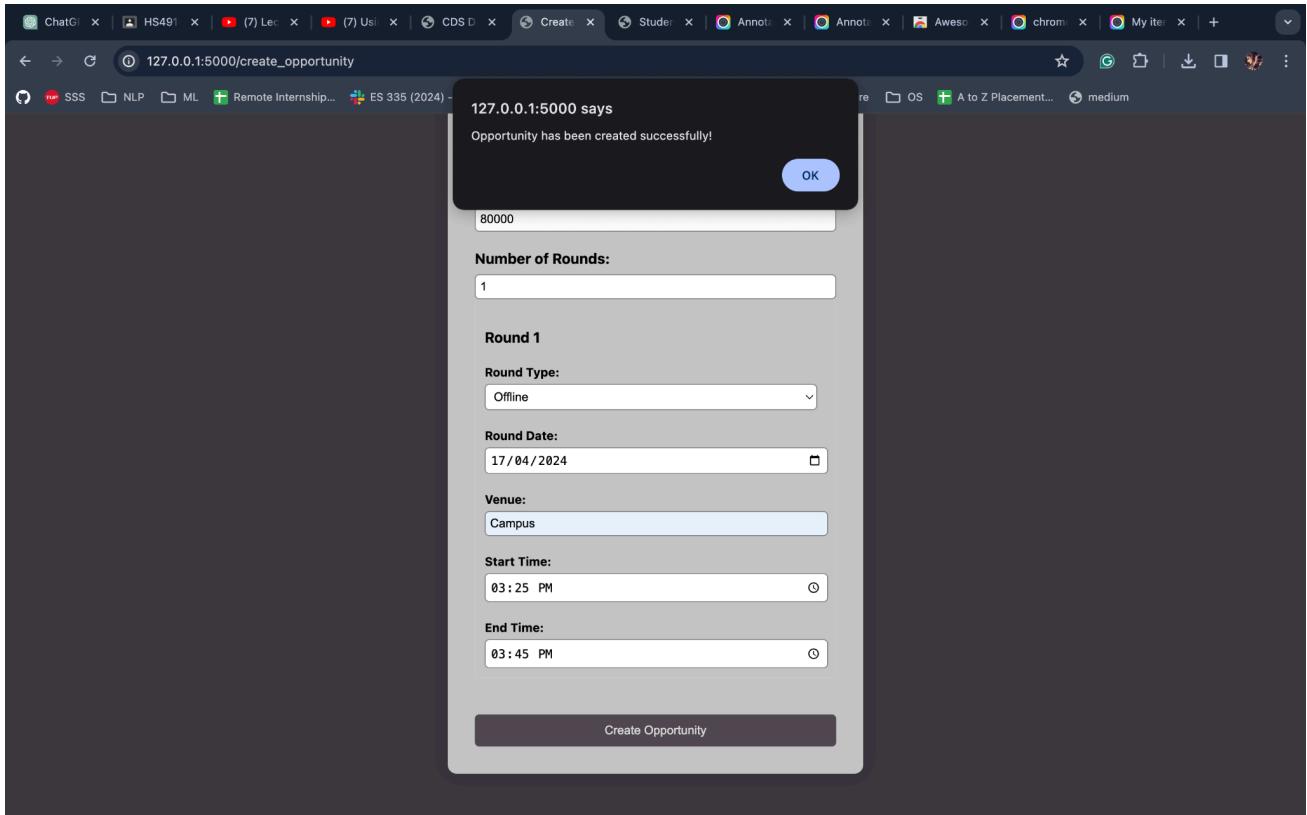
Venue:
AB 9/101

Start Time:
02 : 20 PM

End Time:
03 : 40 PM

Create Opportunity

- Recruiter will get pop after successfully adding the opportunity.



- Recruiter can see opportunities details of hosted opportunities.

The screenshot shows a web page titled "Opportunity Details". The page displays various job specifications: Title: SDE, Positions: 2, Salary: 120000, Requirements: file.com, CPI requirement: 6.0, Active Backlog: 0, Student Year Requirement: 4, Program requirement: BTech, and Job Description: job.com. Below this, a section titled "Round Details:" lists two rounds. The first round is "Online" at "Gmeet" on "2024-04-26" from "13:20:00" to "13:30:00". The second round is "Offline" at "AB 9/101" on "2024-04-27" from "14:20:00" to "15:40:00". Navigation buttons for "Dashboard" and "Logout" are visible at the top right.

ROUND NUMBER	TYPE	VENUE	DATE	START TIME	END TIME
1	Online	Gmeet	2024-04-26	13:20:00	13:30:00
2	Offline	AB 9/101	2024-04-27	14:20:00	15:40:00

- Recruiter can view applications submitted by student.

Hosted Opportunities

TITLE	POSITIONS	SALARY	APPLICATIONS	OPPORTUNITY DETAILS	DELETE OPPORTUNITY
SDE	2	120000	View Applications	Opportunity Details	Delete Opportunity
ML	4	80000	View Applications	Opportunity Details	Delete Opportunity

- Recruiter can change the status of opportunity of student.

Opportunities Status

STUDENT NAME	RESUME	
Rutwik More	resume.com	<div style="border: 1px solid #ccc; padding: 5px; display: inline-block;"> Pending Round : 1 Round : 2 ✓ Accepted Rejected </div> Update Status

3.2 Responsibility of G2:

G2 has to work on integrating MySQL in WebApp. This group works on the proper working of the backend.

The code is submitted as per the instructions.

3.3 Responsibility of G1 & G2:

1. Create the database of the designed logical schema in your previous assignment number 2, enter dummy/imaginary values into it, and show these dummy entries on the web page. (10M)

The SQL database with dummy values is submitted to GitHub.

A screenshot of a GitHub repository page. At the top, there are navigation links for 'main' (with a dropdown arrow), '5 Branches' (with a dropdown arrow), '0 Tags', a search bar with placeholder 'Go to file', a 't' icon, an 'Add file' button, and a green 'Code' button. Below this is a commit history table with the following data:

rutwik1440 Merge pull request #4 from adityadeshmukh369/Iter5		
files added		be24248 · yesterday
data_files	Small changes	3 days ago
static	Error Correction	yesterday
templates	Update README.md	3 days ago
README.md	Merge branch 'main' into Iter3	3 days ago
app.py	Update database.sql	3 days ago
database.sql	Error Correction	yesterday
db.yaml	Create requirements.txt	3 days ago
requirements.txt	files changed	3 days ago
tempCodeRunnerFile.py		

A screenshot of the GitHub code editor showing the 'Placement-System / database.sql' file. The file contains the following SQL code:

```
3 CREATE TABLE Student (
4     Student_ID INT PRIMARY KEY,
5     Student_First_Name VARCHAR(50) NOT NULL,
6     Student_Middle_Name VARCHAR(50),
7     Student_Last_Name VARCHAR(50) NOT NULL,
8     Department VARCHAR(100) NOT NULL,
9     Active_Backlog INT DEFAULT 0,
10    Gender ENUM('Male', 'Female', 'Other') NOT NULL,
11    Year INT NOT NULL,
12    Student_Image VARCHAR(255) DEFAULT NULL,
13    Minors VARCHAR(255),
14    Student_Email_Id VARCHAR(100) NOT NULL,
15    Contact_number VARCHAR(100) NOT NULL,
16    CPI DECIMAL(3,1) CHECK (CPI >= 0 AND CPI <= 10),
17    SSAC_or_not ENUM('Yes', 'No'),
18    CONSTRAINT chk_gender CHECK (Gender IN ('Male', 'Female', 'Other')),
19    CONSTRAINT chk_year CHECK (Year IN (1,2,3,4)),
20    CONSTRAINT chk_email_format CHECK (Student_Email_Id LIKE '%@iitgn.ac.in')
21 );
22
23 SELECT * FROM Student;
24 INSERT INTO Student (Student_ID, Student_First_Name, Student_Middle_Name, Student_Last_Name, Department, Active_Backlog, Gender, Year,
25 VALUES
26 (2, 'Alice', 'M', 'Johnson', 'Electrical Engineering', 1, 'Female', 3, 'alice.johnson@iitgn.ac.in', '+91 9876543210', 8.5, 'No'),
27 (3, 'Bob', NULL, 'Williams', 'Mechanical Engineering', 0, 'Male', 2, 'bob.williams@iitgn.ac.in', '+91 7777777777', 9, 'No'),
28 (4, 'Emily', 'J', 'Brown', 'Chemical Engineering', 2, 'Female', 1, 'emily.brown@iitgn.ac.in', '+91 5555555555', 7, 'Yes'),
29 (5, 'Michael', 'A', 'Taylor', 'Civil Engineering', 1, 'Male', 3, 'michael.taylor@iitgn.ac.in', '+91 3333333333', 8.5, 'No'),
30 (6, 'Sophia', 'K', 'Anderson', 'Biomedical Engineering', 0, 'Female', 2, 'sophia.anderson@iitgn.ac.in', '+91 9999999999', 6.5, 'Yes');
```

2. Implementing the tasks as per the section Section 2: Problem Statement and Requirements. (15M)

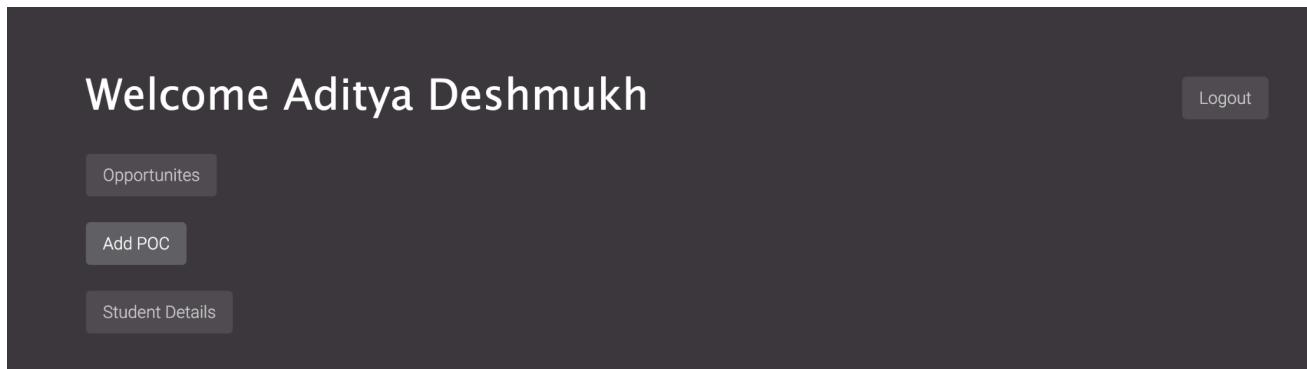
The web app supports the dynamic execution (the changes done by the user through the web app shall be reflected

in the main database and web page also) of the following functions on your database:

Task 1: INSERT

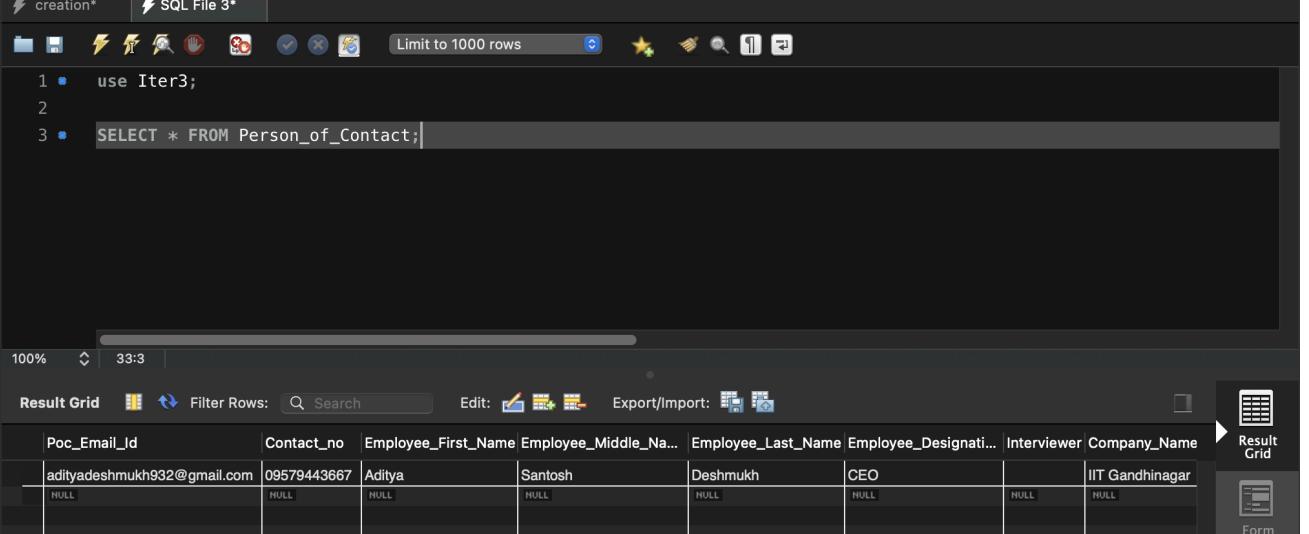
We are inserting a new Person of Contact through a CDS user into the database.

Here, we first log in a CDS user, and click on ADD POC, which will open a form



Then we add details of the Person of Contact to add the form.

Initially: The database had only 1 person as the Person of Contact.



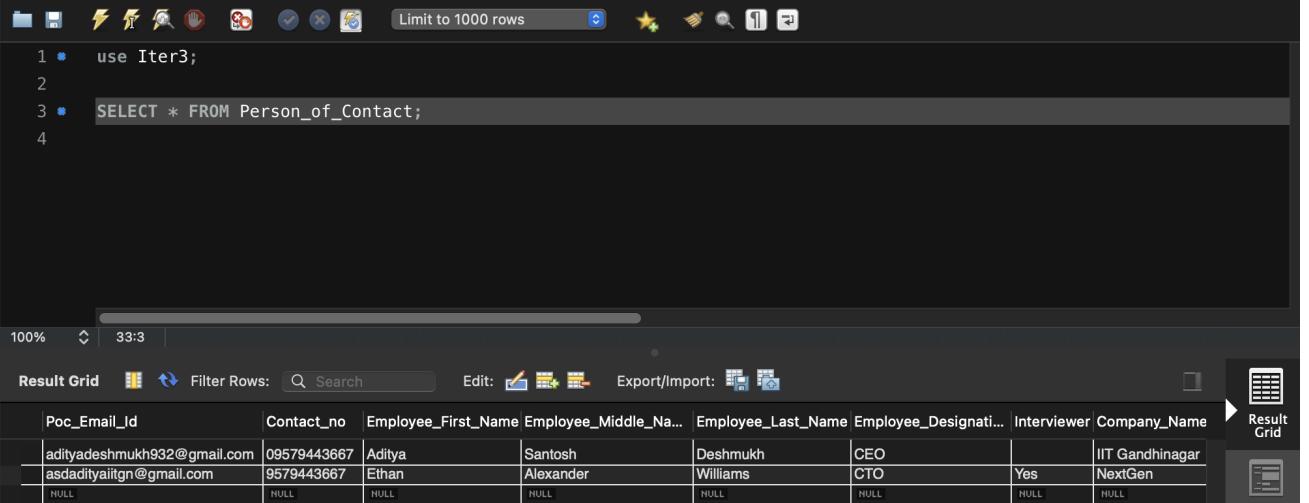
A screenshot of the MySQL Workbench interface. The SQL tab shows the following query:

```
1 • use Iter3;
2
3 • SELECT * FROM Person_of_Contact;
```

The results grid displays one row of data:

Poc_Email_Id	Contact_no	Employee_First_Name	Employee_Middle_Na...	Employee_Last_Name	Employee_Designati...	Interviewer	Company_Name
adityadeshmukh932@gmail.com	09579443667	Aditya	Santosh	Deshmukh	CEO		IIT Gandhinagar

Now: The database also have a Person of Contact Ethan Alexander Williams



A screenshot of the MySQL Workbench interface. The SQL tab shows the following query:

```
1 • use Iter3;
2
3 • SELECT * FROM Person_of_Contact;
4
```

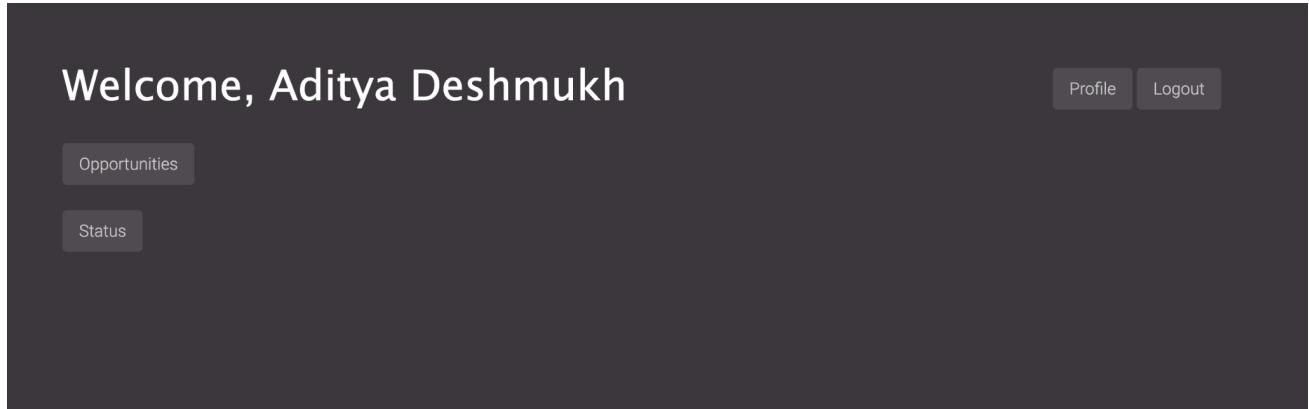
The results grid displays two rows of data:

Poc_Email_Id	Contact_no	Employee_First_Name	Employee_Middle_Na...	Employee_Last_Name	Employee_Designati...	Interviewer	Company_Name
adityadeshmukh932@gmail.com	09579443667	Aditya	Santosh	Deshmukh	CEO		IIT Gandhinagar
asdadityaiitgn@gmail.com	9579443667	Ethan	Alexander	Williams	CTO	Yes	NextGen

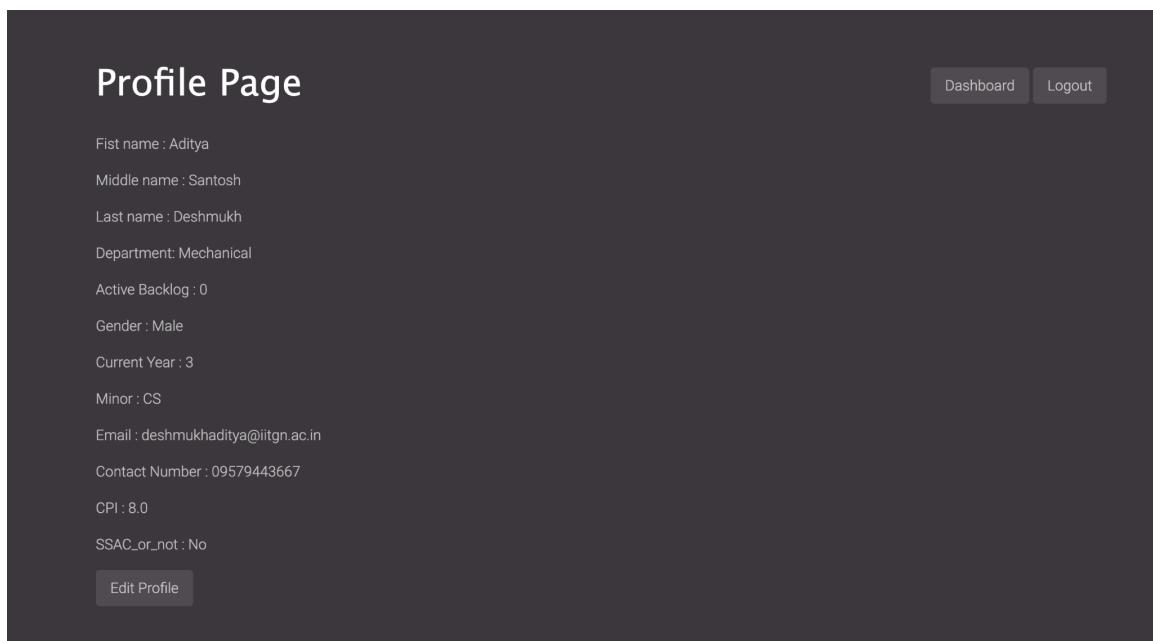
Task 2: UPDATE

Here, we will be updating the values of the students profile.

Here, open in the Student view, and choose profile.



On the profile page, we click edit profile to make updates.



We will get a prefilled form with the information of the student, and we can make changes.

Edit Profile

First Name
Aditya

Middle Name
Santosh

Last Name
Deshmukh

Department
Mechanical

Active Backlog
0

Current Year
3

Minor
CS

Contact Number
09579443667

CPI
8.0

SSAC_or_not:
No

Gender:
Male

Student Image Link :
HAWHGARE

Update Profile

The form after updating. Now has Active Backlog as 1 in place of 0, and Current Year is updated from 3 to 4.

Edit Profile

First Name
Aditya

Middle Name
Santosh

Last Name
Deshmukh

Department
Mechanical

Active Backlog
1

Current Year
4

Minor
CS

Contact Number
09579443667

CPI
8.0

SSAC_or_not:
No

Gender:
Male

Student Image Link :
HAWHGARE

Update Profile

As the Pop up says, the updated changes have been saved.

127.0.0.1:5000 says

Submit profile successfully!

OK

Current Year

4

Minor

CS

Contact Number

09579443667

CPI

8.0

SSAC_or_not:

No



Gender:

Male



Student Image Link :

HAWHGARE

Update Profile

Initially: Before making the updations:-

```
1 • use Iter3;
2
3 • SELECT * FROM Student;
4 • drop table student;
5
```

Result Grid Filter Rows: Search Edit: Export/Import:

Student_ID	Student_First_Na...	Student_Middle_Na...	Student_Last_Na...	Department	Active_Backlog	Gender	Year	Student_Image	Minors	Student_En
2	Alice	M	Johnson	Electrical Engineering	1	Female	3	HULL	CS	alice.johnso
3	Bob	NULL	Williams	Mechanical Engineering	0	Male	2	NULL	NULL	bob.williams
4	Emily	J	Brown	Chemical Engineering	2	Female	1	HULL	NULL	emily.brown
5	Michael	A	Taylor	Civil Engineering	1	Male	3	NULL	NULL	michael.tayl
6	Sophia	K	Anderson	Biomedical Engineering	0	Female	2	NULL	NULL	sophia.ande
7	Aditya	Santosh	Deshmukh	Mechanical	0	Male	3	HAWHGARE	CS	deshmukha
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Result Grid Form Editor Field Types

After: Making the changes, see, the Active backlog, and year of the student have been changed.

```
1 • use Iter3;
2
3 • SELECT * FROM Student;
4 • drop table student;
5
```

Task 3: DELETE

In this part, we will delete a Hosted opportunity from the list of opportunities.

The screenshot shows a dark-themed web application interface. At the top, there's a header with the title "Hosted Opportunities" and two buttons: "Dashboard" and "Logout". Below the header, there's a table-like structure with columns: "TITLE", "POSITIONS", "SALARY", "APPLICATIONS", "OPPORTUNITY DETAILS", and "DELETE OPPORTUNITY". Under "TITLE", it says "SDE". Under "POSITIONS", it says "4". Under "SALARY", it says "80000". Under "APPLICATIONS", there are three buttons: "View Applications", "Opportunity Details", and "Delete Opportunity".

After you press delete, the opportunity will be deleted.

The screenshot shows the same dark-themed web application. The header "Hosted Opportunities" and buttons "Dashboard" and "Logout" are visible. Below the header, a message "No opportunities hosted." is displayed. The rest of the page is empty, indicating that all opportunities have been deleted.

Initially: Before making the updatons-

The screenshot shows a database query tool with a dark theme. At the top, there's a toolbar with various icons and a "Limit to 1000 rows" dropdown. Below the toolbar, a SQL query is written:

```
1 • use Iter3;
2
3 • SELECT * FROM Opportunity;
```

Below the query, the results are displayed in a grid. The grid has the following columns: Opp_ID, Opp_Title, No_of_Positions, Specific_Requirements_file, Min_CPI_req, No_Active_Backlo..., Student_year_r..., Program_req, and Job_Description_file. There are two rows of data:

Opp_ID	Opp_Title	No_of_Positions	Specific_Requirements_file	Min_CPI_req	No_Active_Backlo...	Student_year_r...	Program_req	Job_Description_file
1	Machine Learning	3	https://req/home https://requirement/drive/home	6.0	0	3	Btech	https://drive/home/req https://program_equipment/dri
2	SDE	4		8.0	0	3	Btech	

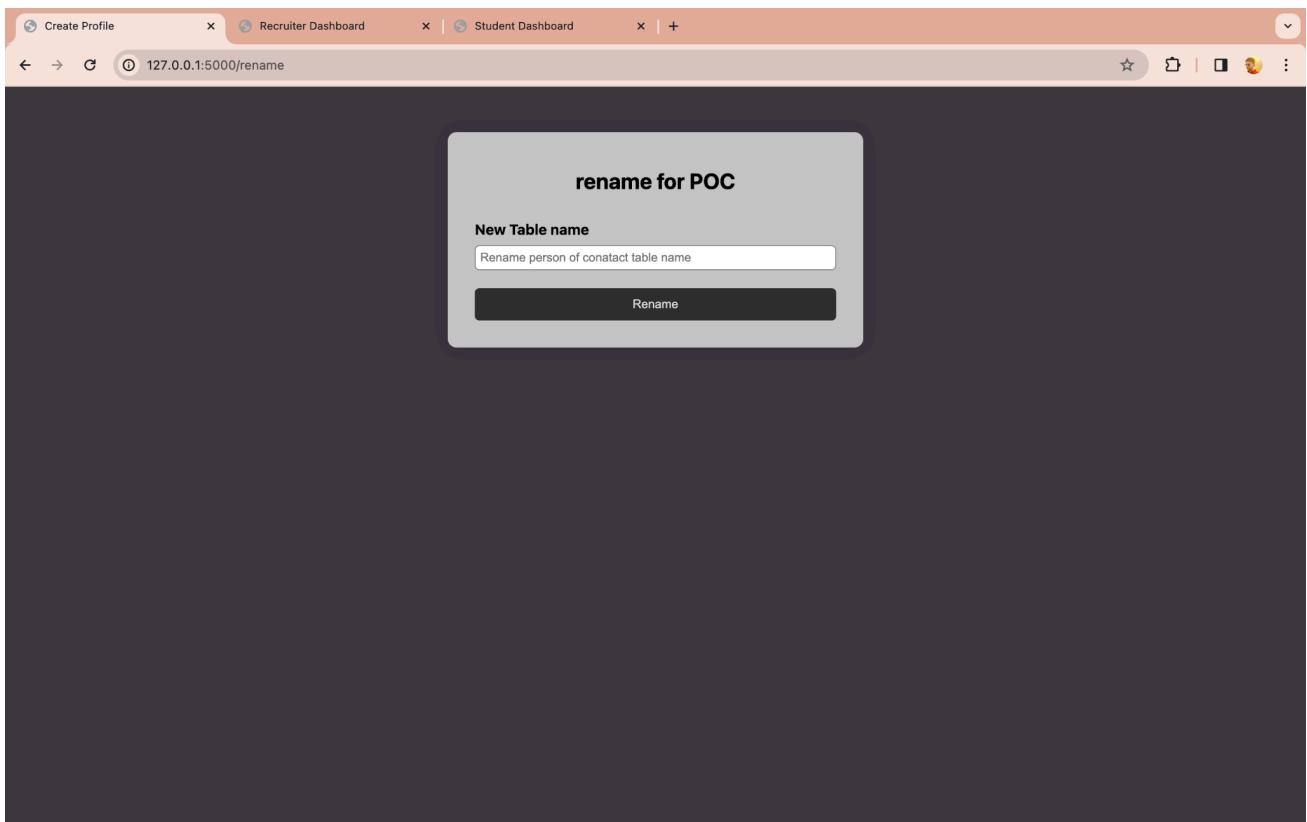
After: After deleting the SDE opportunity

```
1 • use Iter3;
2
3 • SELECT * FROM Opportunity;
4
```

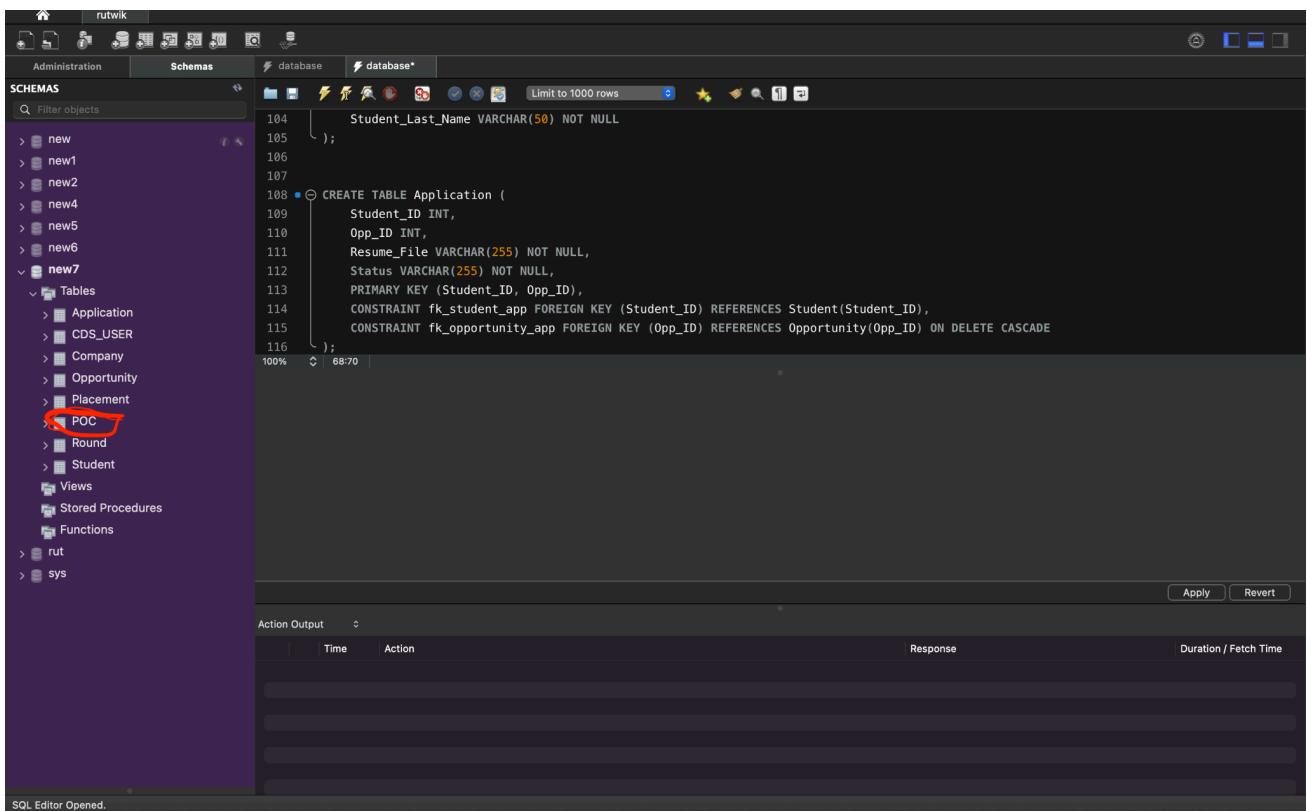
Task 4: RENAME

In this task, we will change the table name of the person of Contact.

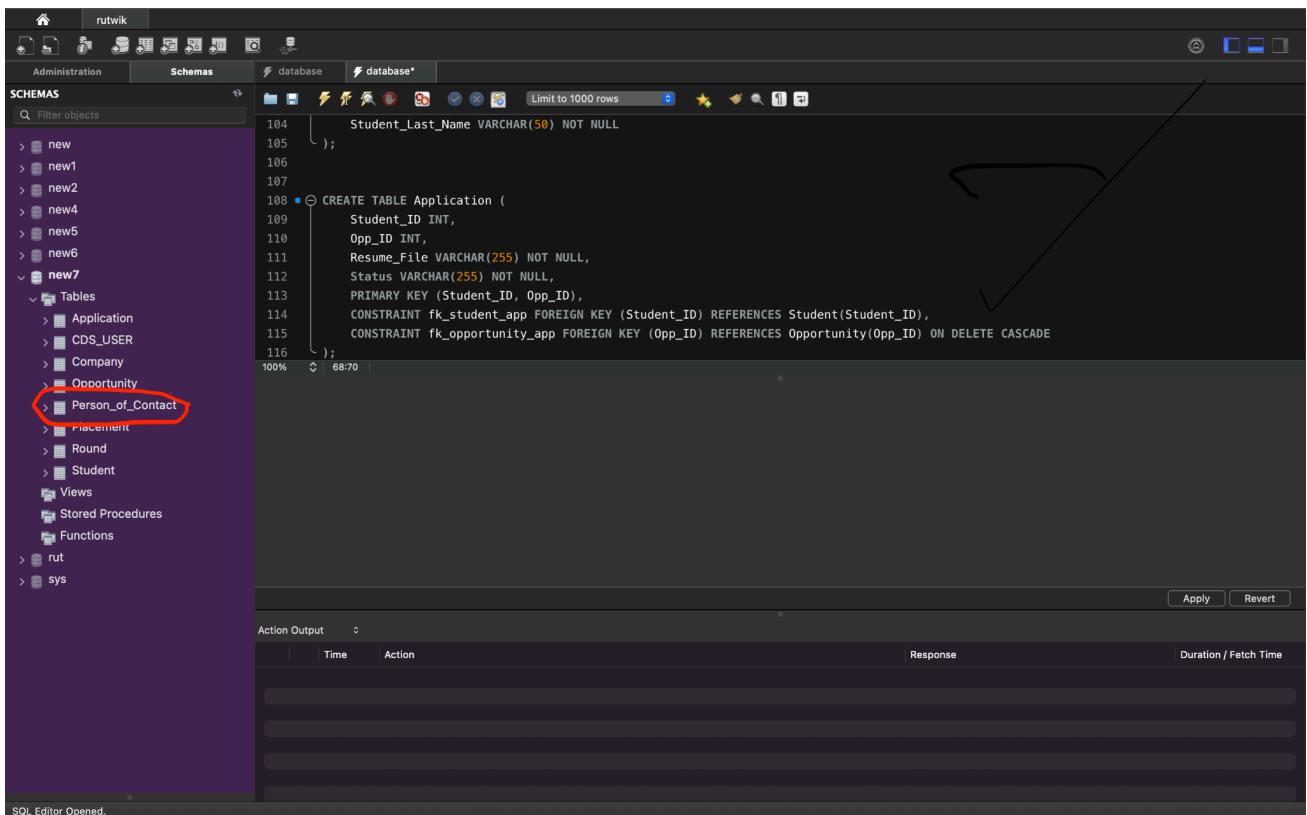
Here we have opened the rename option in CDS. Where we can rename table name of POC.



We can see before renaming table name is POC.



After naming the table name of POC changes to Person_of_Contact



The screenshot shows the Oracle SQL Developer interface. On the left, the 'Schemas' tree view is open, showing several schemas like 'new', 'new1', 'new2', 'new4', 'new5', 'new6', and 'new7'. Under 'new7', there is a 'Tables' node which contains tables such as 'Application', 'CDS_USER', 'Company', 'Opportunity', 'Person_of_Contact', 'Placement', 'Round', 'Student', 'Views', 'Stored Procedures', and 'Functions'. The 'Person_of_Contact' table is circled in red. The main panel displays a SQL editor with the following code:

```

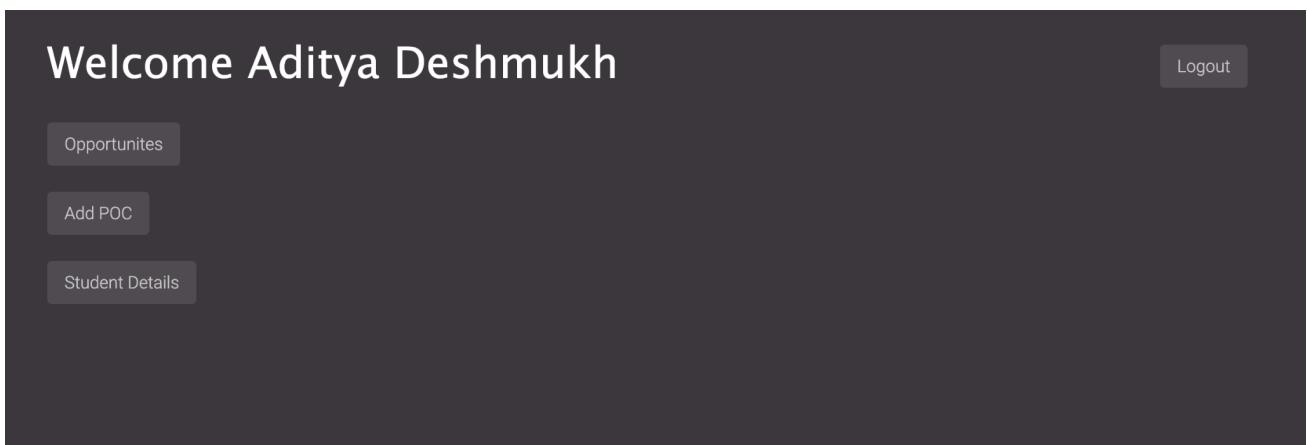
104     Student_Last_Name VARCHAR(50) NOT NULL
105   );
106
107 CREATE TABLE Application (
108   Student_ID INT,
109   Opp_ID INT,
110   Resume_File VARCHAR(255) NOT NULL,
111   Status VARCHAR(255) NOT NULL,
112   PRIMARY KEY (Student_ID, Opp_ID),
113   CONSTRAINT fk_student_app FOREIGN KEY (Student_ID) REFERENCES Student(Student_ID),
114   CONSTRAINT fk_opportunity_app FOREIGN KEY (Opp_ID) REFERENCES Opportunity(Opportunity_ID) ON DELETE CASCADE
115 );
116

```

Task 5: WHERE clause

For the where clause, we will show how CDS can fetch the details of all the students.

In CDS view, select Student Details.



The screenshot shows a CDS (Cloud Data Store) view. At the top, it says 'Welcome Aditya Deshmukh'. Below that are three buttons: 'Opportunities', 'Add POC', and 'Student Details'. In the top right corner, there is a 'Logout' button. The rest of the screen is mostly blank.

Here are the details of all the students of the database that we have currently.

Student Details

[Dashboard](#) [Logout](#)

Alice Johnson

Electrical Engineering
alice.johnson@iitgn.ac.in

[View Details](#)

Bob Williams

Mechanical Engineering
bob.williams@iitgn.ac.in

[View Details](#)

Emily Brown

Chemical Engineering
emily.brown@iitgn.ac.in

[View Details](#)

Michael Taylor

Civil Engineering
michael.taylor@iitgn.ac.in

[View Details](#)

Sophia Anderson

Biomedical Engineering
sophia.anderson@iitgn.ac.in

[View Details](#)

Aditya Deshmukh

Mechanical
deshmukhaditya@iitgn.ac.in

[View Details](#)

Here, you can see WHERE in the 4th line

```
def get_Details(student_email):
    cur = mysql.connection.cursor()
    # Fetch the user profile data from the database
    cur.execute("SELECT * FROM Student WHERE Student_Email_Id = %s", (student_email,))
    user_profile = cur.fetchone()
    cur.close()
    return user_profile
```

3. Write-up in the README file, which should include the following:
 - a. Steps to run your web app [initial setup, executing the required tasks, etc.].

Here is a detailed Step by Step Instruction on how to setup and run the database-

Step 1: Clone the code from Github:

<https://github.com/adityadeshmukh369/DBMS Assignment 2>

Step 2: Making and activating Virtual Environment for the work:

python -m venv env

source /path_to_venv/bin/activate (usually source env/bin/activate)

Step 3: Setting up the requirements for the virtual environment

pip install -r requirements.txt

Step 4: Setting up the Database in your local machine:-

- a. Open MYSQL and copy paste the contents of databases.sql and run it

```

1 DROP database IF EXISTS project;
2 CREATE database Iter1;
3 USE Iter1;
4 SET foreign_key_checks = 0;
5
6 CREATE TABLE Student (
7   Student_ID INT PRIMARY KEY,
8   Student_First_Name VARCHAR(50) NOT NULL,
9   Student_Middle_Name VARCHAR(50),
10  Student_Last_Name VARCHAR(50) NOT NULL,
11  Department VARCHAR(100) NOT NULL,
12  Active_Backlog INT DEFAULT 0,
13  Gender ENUM('Male', 'Female', 'Other') NOT NULL,
14  Year INT NOT NULL,
15  Student_Image VARCHAR(255) DEFAULT NULL,
16  Minors VARCHAR(255),
17  Student_Email_Id VARCHAR(100) NOT NULL,
18  Contact_number VARCHAR(100) NOT NULL,
19  CONSTRAINT chk_gender CHECK (Gender IN ('Male', 'Female', 'Other')),
20  CONSTRAINT chk_year CHECK (Year IN (1,2,3,4)),
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59

```

Student_ID	Opp_ID	Resume_File	Status
HULL	HULL	HULL	HULL
Student 5	Opportunity 6	Person_of_Contact 7	Application 8

Action	Time	Response	Duration / Fetch Time
CREATE TABLE Academic_Info (Academic_Info_ID INT PRIMARY KEY, Transcript_File VARCHAR(255) NOT NULL)	14:44:29	0 row(s) affected	0.0032 sec
CREATE TABLE Student_Academic_Relationship (Student_ID INT, Academic_Info_ID INT, PRIMARY KEY (Student_ID, Academic_Info_ID))	14:44:29	0 row(s) affected	0.0079 sec
CREATE TABLE Placement_Student (Placement_ID INT, Student_ID INT, CONSTRAINT fk_placement FOREIGN KEY (Placement_ID) REFERENCES Placement (Placement_ID), CONSTRAINT fk_student FOREIGN KEY (Student_ID) REFERENCES Student (Student_ID))	14:44:29	0 row(s) affected	0.0055 sec
CREATE TABLE Application (Student_ID INT, Opp_ID INT, Resume_File VARCHAR(255) NOT NULL, Status VARCHAR(100))	14:44:29	0 row(s) affected	0.0078 sec
SELECT * FROM Application LIMIT 0, 1000	14:44:29	0 row(s) returned	0.00050 sec / 0.0000...

Step 5: Make changes to the db.yml, edit and put your password.

```
mysql_host: "127.0.0.1"
mysql_user: "root"
mysql_password: "Put_your_Password"
mysql_db: "Iter1"
images_folder: "/Users/rutwikmore/Documents/DBMS_Assignment_2/data_files/images"
resume_folder: "/Users/rutwikmore/Documents/DBMS_Assignment_2/data_files/resumes"
```

Step 6: Running the code:-

```
python app.py
```

Step 7: Giving control over Student, CDS and Poc:

Student: The student has to login using their IITGN id.

CDS: Add your email id to become a CDS User and then you can use its features.

The screenshot shows the MySQL Workbench interface. On the left, the 'Schemas' tree view is open, showing the 'Iter1' schema with various tables like Academic_Info, Application, CDS_USER, Opportunity, Person_of_Contact, Placement, Placement_Student, Student, and Student_Academic_Relati...'. In the center, the SQL editor contains the following code:

```
1 USE Iter1;
2 INSERT INTO CDS_USER (Email) VALUES ('your_email@iitgn.ac.in');
3 select * from cds_user;
```

The second line, which inserts the email, is highlighted with a yellow oval. Below the SQL editor, the 'Result Grid' shows the results of the 'select * from cds_user' query. It displays two rows of data under the 'Email' column:

Email
deshmukhaditya@iitgn.ac.in
your_email@iitgn.ac.in
HULL

At the bottom, the 'Action Output' pane shows the execution log with several entries, including the insertion of the new user record.

POC: Enter the CDS portal as a CDS member and add an email as a Person of Contact. Then you can login as a POC and start making, accepting opportunities.

- b. Screenshots of successful execution of the dynamic operations defined in Section 2: Problem Statement and Requirements. You can design the web app in such a way that a dialogue box pops up that the query has been successfully executed. Also, you will have to post these snapshots of the dialogue box (in the readme file) for every respective query. Or else, you can directly show the before and after changes reflected in your concerned table in the web app in the README file.*

We have attached the screenshot above.

Important Links

- 1) ER Diagram: Draw.io file:

<https://app.diagrams.net/?client=1#G1yCpFEy0jyzqNLsKFg6Lim8za2PPASyPd%7B%22pageId%22%3A%22n4RG28VcgyLvL5a97sLb%22%7D>

- 2) Code for the Data Generation process: Google Colab notebook:

<https://colab.research.google.com/drive/1thD8FtA0pBjFwbtnWxz85L4WhKXOa4s->

- 3) Generated Dataset for the Database: Google Drive:

https://drive.google.com/drive/folders/1lr40jKZI_w58t7_DY7AVztuXKtXFc2A2

- 4) SQL Files, and data dump:

https://drive.google.com/drive/folders/1_MqzNJfXfNdrbH-PaVhR7fRd7r1Yt9vL

- 5) Github Repository of project:

<https://github.com/adityadeshmukh369/Placement-System>

Contributions:

Name	Group	Roll Number	Contribution
Anugu Arun Reddy	G2	21110029	<ul style="list-style-type: none"> • Integrated database with the web app. • Debugged parts of the frontend code. • Contributed to documentation.
Aditya Deshmukh	G1	21110014	<ul style="list-style-type: none"> • Worked on google authentication of the users • Wrote code for frontend for student and recruiter pages • Integrated dynamic form filling for number of round for recruiter • Worked on documentation
Abhay Kumar Upparwal	G1	21110004	<ul style="list-style-type: none"> • Debugged errors in the code at frontend. • Contributed to the documentation.
Aaryan Darad	G1	21110001	<ul style="list-style-type: none"> • Worked on making the report of the project • Debugged some parts of the code where the webapp from the database had glitches. • Contributed to the documentation part of the project.
Gaurav Joshi	G2	21110065	<ul style="list-style-type: none"> • Worked on making the report of the project. • Made contributions to brain storming the design of the webapp. • Contributed to requirements of the project.
Adit Kaushik	G1	21110010	<ul style="list-style-type: none"> • Contributed for get requests and SQL commands for company POC and CDS Employee required for different requests • Wrote the setup requirements and steps to run the Web App. • Contributed to the documentation

More Rutwik	G2	21110133	<ul style="list-style-type: none"> • Wrote code for the backend, and wrote different SQL queries. • Developed all HTML pages of CDS. • Wrote backend functionalities for routing of some of the pages.
Ahaan Giriya	G2	21110015	<ul style="list-style-type: none"> • Helped in making the frontend • Formatted the document