

# **CS 432: Databases**

## **Assignment 3: Implementing a Web App using MySQL**

### **Topic: Placement Management**

#### **Group Members**

##### **G1:**

Pushpendra Pratap Singh (20110151)

Saatvik Rao (20110175)

Sahil Agrawal (20110178)

Sanskar Sharma (20110185)

Tanvi Dixit (20110212)

##### **G2:**

Dhyey Kumar Thummar (20110059)

Ksheer Sagar Agrawal (20110098)

Mihir Sutariya (20110208)

R Yeeshu Dhurandhar (20110152)

Shruhid Banthia (20110198)

Utkarsh Mishra (20110218)

#### **Responsibility of G1:**

The code is submitted as per the instructions.

**Responsibility of G2:**

The code is submitted as per the instructions.

## 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.

The SQL database with dummy values is submitted to GitHub..

### 3a. Steps to run the Web App

We need three different emails due to the constraint that one person can not play the role of the above three roles. We will give access to the provided mails.

**Step 1-** Download the code from GitHub.

**Step 2-** Installing packages:

```
python3 -m venv env_flask
```

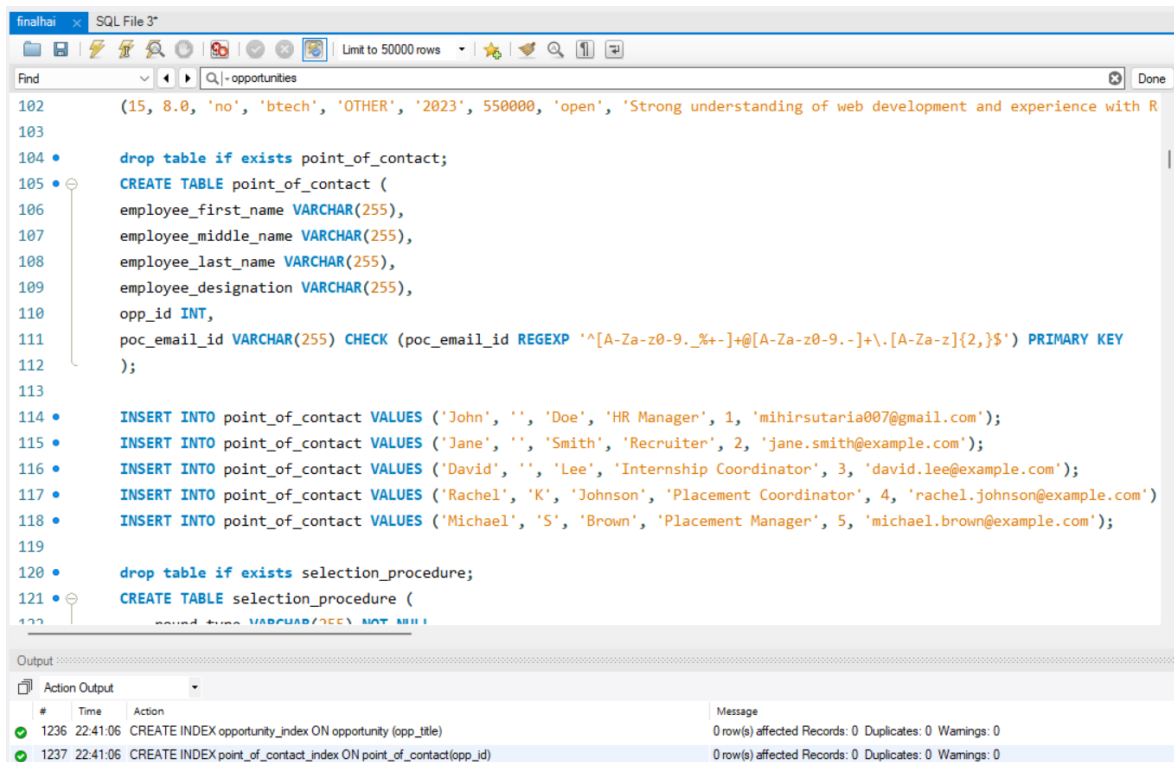
```
pip3 install flask
```

**Step 3-** To setup all the requirements for frontend as well as backend, write the command:

```
pip install -r requirements.txt
```

**Step 4-** To get access :

- For point of contact : add a point of contact details or replace the '[mihirsutaria007@gmail.com](mailto:mihirsutaria007@gmail.com)' gmail id (line 114) in the SQL file to get access to the point of contact page.



```
102 (15, 8.0, 'no', 'btech', 'OTHER', '2023', 550000, 'open', 'Strong understanding of web development and experience with R
103
104 • drop table if exists point_of_contact;
105 • CREATE TABLE point_of_contact (
106     employee_first_name VARCHAR(255),
107     employee_middle_name VARCHAR(255),
108     employee_last_name VARCHAR(255),
109     employee_designation VARCHAR(255),
110     opp_id INT,
111     poc_email_id VARCHAR(255) CHECK (poc_email_id REGEXP '^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}$') PRIMARY KEY
112 );
113
114 • INSERT INTO point_of_contact VALUES ('John', '', 'Doe', 'HR Manager', 1, 'mihirsutaria007@gmail.com');
115 • INSERT INTO point_of_contact VALUES ('Jane', '', 'Smith', 'Recruiter', 2, 'jane.smith@example.com');
116 • INSERT INTO point_of_contact VALUES ('David', '', 'Lee', 'Internship Coordinator', 3, 'david.lee@example.com');
117 • INSERT INTO point_of_contact VALUES ('Rachel', 'K', 'Johnson', 'Placement Coordinator', 4, 'rachel.johnson@example.com');
118 • INSERT INTO point_of_contact VALUES ('Michael', 'S', 'Brown', 'Placement Manager', 5, 'michael.brown@example.com');
119
120 • drop table if exists selection_procedure;
121 • CREATE TABLE selection_procedure (
122     sound_type VARCHAR(255) NOT NULL
123 )
```

| #    | Time     | Action  | Message  |
|------|----------|---|--|
| 1236 | 22:41:06 | CREATE INDEX opportunity_index ON opportunity (opp_title)       | 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0 |
| 1237 | 22:41:06 | CREATE INDEX point_of_contact_index ON point_of_contact(opp_id) | 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0 |

- For students : add your email id (and other student details) or replace the '[mihirsutaria007@gmail.com](mailto:mihirsutaria007@gmail.com)' gmail id (line 161) in the SQL file to get access to the student page.

```

141
142 • drop table if exists student;
143 • CREATE TABLE student (
144     student_id INTEGER PRIMARY KEY,
145     student_email_id VARCHAR(255),
146     student_first_name VARCHAR(255) NOT NULL,
147     student_middle_name VARCHAR(255),
148     student_last_name VARCHAR(255) NOT NULL,
149     student_image VARCHAR(255),
150     dept ENUM('CSE','EE','ME','CE','CVE','MSE') NOT NULL,
151     CPI DECIMAL(3,2) CHECK (CPI >= 0 AND CPI <= 10) NOT NULL,
152     active_backlogs ENUM('yes','no') NOT NULL,
153     gender ENUM('male','female','other') NOT NULL,
154     study_year INTEGER NOT NULL
155 );
156
157
158
159 • INSERT INTO student (student_id, student_email_id, student_first_name, student_middle_name, student_last_name, student_i
160 VALUES
161 (1, 'mihirsutaria007@gmail.com', 'Samantha', '', 'Johnson', 'https://drive.google.com/uc?export=view&id=18bYTLyqoilSHH02r
162 (2, 'shruhid.banthia@iitgn.ac.in', 'Shruhid', 'bunty', 'Doe', NULL, 'EE', 9.9, 'yes', 'male', 2),
163 (3, 'dhyeykumar.thummar@iitgn.ac.in', 'Sarah', '', 'Wilson', NULL, 'CE', 8.2, 'no', 'female', 4),
164 (4, 'rahul.mishra@iitgn.ac.in', 'Rahul', 'Raj', 'Mishra', NULL, 'CSE', 8.0, 'yes', 'male', 3)

```

- For CDS employee : Search '[mihirsutaria007@gmail.com](mailto:mihirsutaria007@gmail.com)' in the app.py file and replace all with your CDS employee email id.

The screenshot shows the Visual Studio Code interface with the 'app.py' file open. A search and replace dialog is visible, showing the search term 'mihirsutaria007@gmail.com' and the replace term 'AB'. The code in the background includes imports for Flask, SQLAlchemy, and other libraries, and defines an 'Occupation' enum with values 'STUDENT', 'CDS\_EMPLOYEE', and 'COMPANY\_POC'.

If any problems, please contact us at [<sutariya.mihirkumar@iitgn.ac.in>](mailto:sutariya.mihirkumar@iitgn.ac.in).

Run the whole SQL file in the workbench provided in the GitHub.

Run in terminal using the below code: `python app.py`

- To access the student view: <http://localhost:5000/student>
- To access the cds employee view: <http://localhost:5000/cds>
- To access the company poc (point of contact) view: <http://localhost:5000/poc>
- To logout : <http://localhost:5000/logout>

NOTE:

- To change view between student, cds employee and company poc, you have to logout from the previous view (you will have to change the email address accordingly) and then go to links mentioned above to log into the required view.
- You might face problem in running this in MacOS.

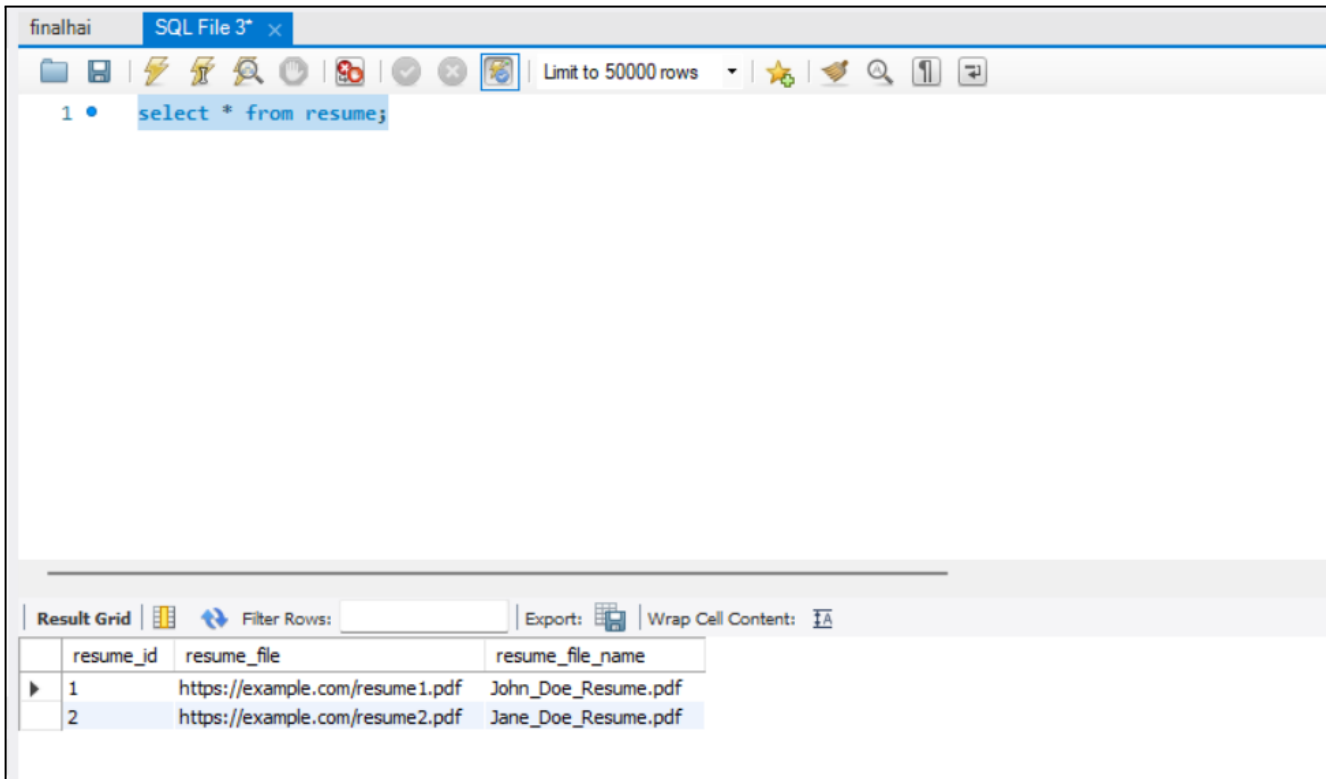
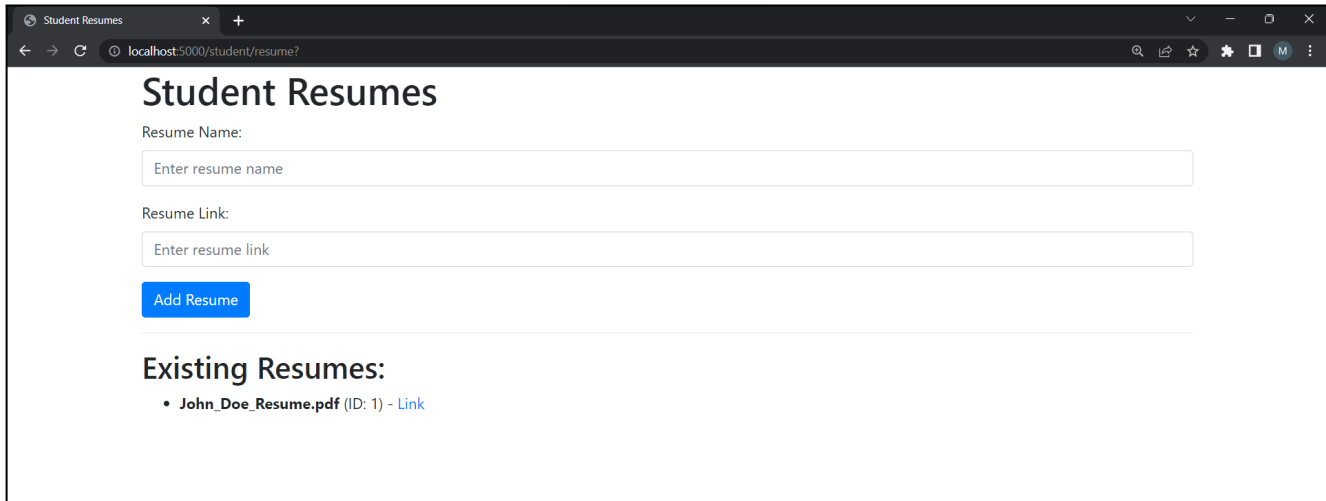
3b. Screenshots of successful execution of the dynamic operations

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:

A. INSERT

- a. Before inserting the data

This is how the page looks before inserting a resume. We can see existing resumes which were uploaded earlier.



b. **Inserting**

Here, we are inserting a resume named 'resume2' on the page. A dialogue box appears which says that the resume is submitted successfully.

Student Resumes

Resume Name:  
resume2

Resume Link:  
https://hfsadjfhkj.com

Add Resume

Existing Resumes:

- John\_Doe\_Resume.pdf (ID: 1) - Link

c. **After** insertion

Here, we have inserted 'resume2' and the changes can be seen in the SQL workbench.

SQL File 3\*

Limit to 50000 rows

```
1 • select * from resume;
```

Result Grid

|   | resume_id | resume_file                     | resume_file_name    |
|---|-----------|---------------------------------|---------------------|
| ▶ | 1         | https://example.com/resume1.pdf | John_Doe_Resume.pdf |
|   | 2         | https://example.com/resume2.pdf | Jane_Doe_Resume.pdf |
|   | 3         | https://hfsadjfhkj.com          | resume2             |

Below, we can see on the page that 'resume2' has been uploaded and comes under existing resumes tab.

-

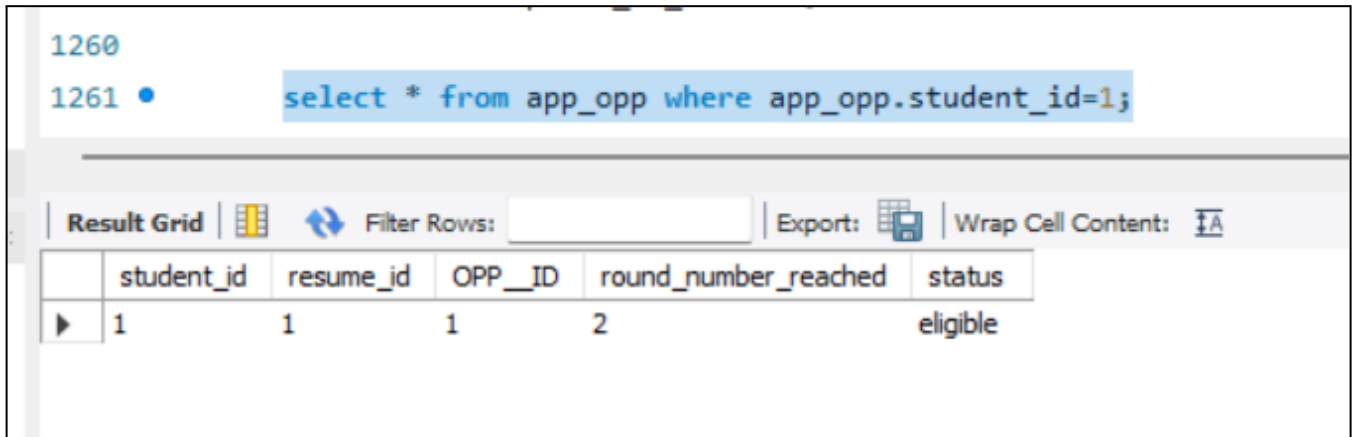
There are also other insertions we can do - add opportunity, add point of contact

## **B. UPDATE**



a. **Before state**

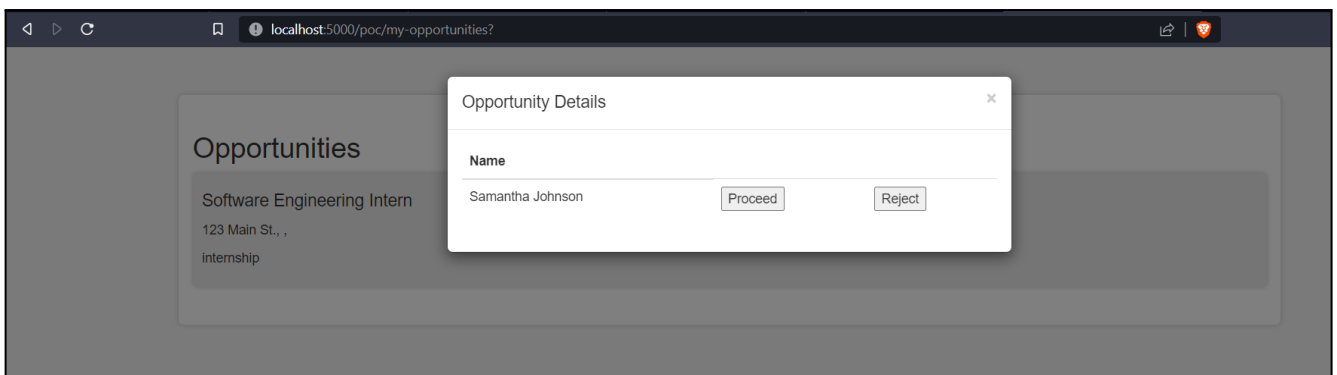
The current status is 'eligible' (in company poc view).



The screenshot shows a SQL query editor with the query `select * from app_opp where app_opp.student_id=1;` highlighted. Below the query is a 'Result Grid' table with the following data:

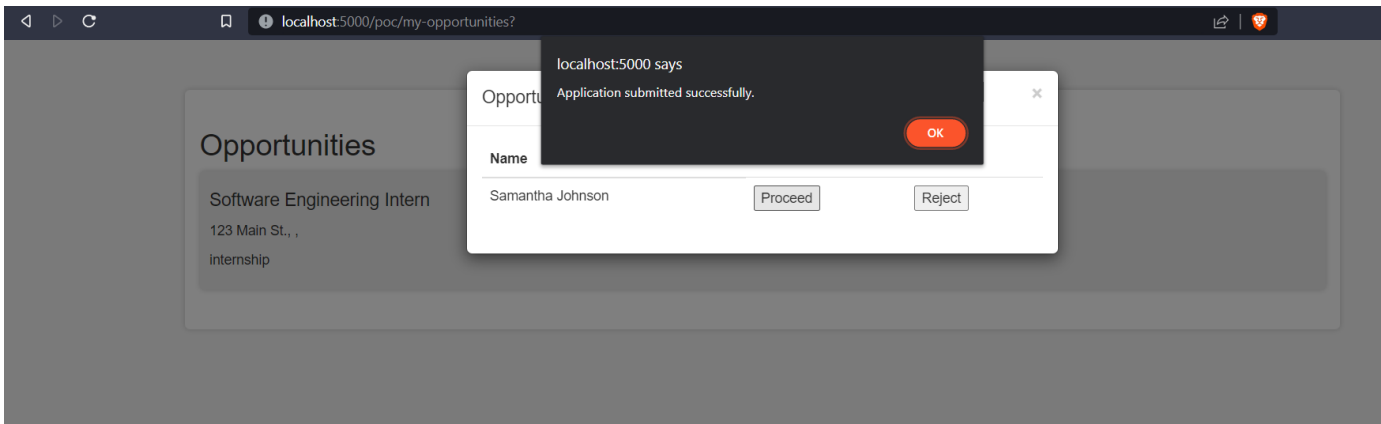
|   | student_id | resume_id | OPP_ID | round_number_reached | status   |
|---|------------|-----------|--------|----------------------|----------|
| ▶ | 1          | 1         | 1      | 2                    | eligible |

In company poc view, you can see options for the applicant.



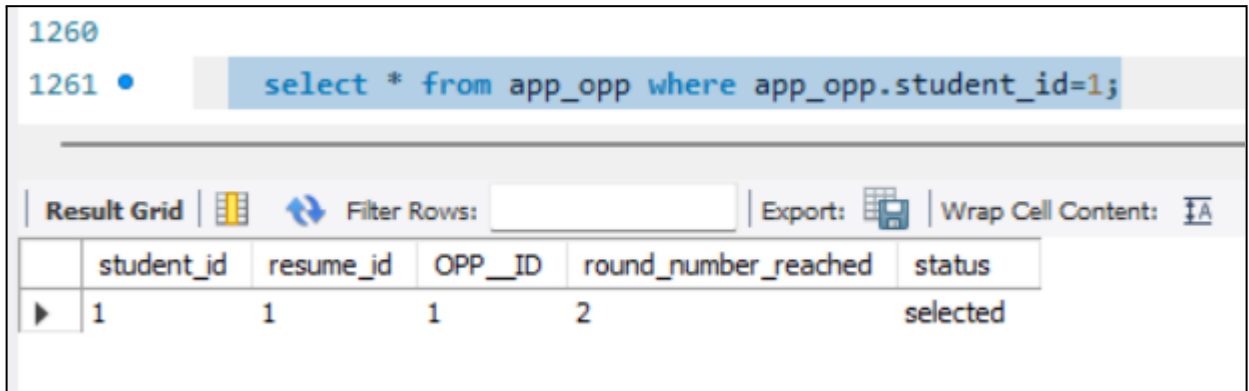
b. **Updating**

Clicking on 'proceed' (accepting student for next round).



c. **After** updating

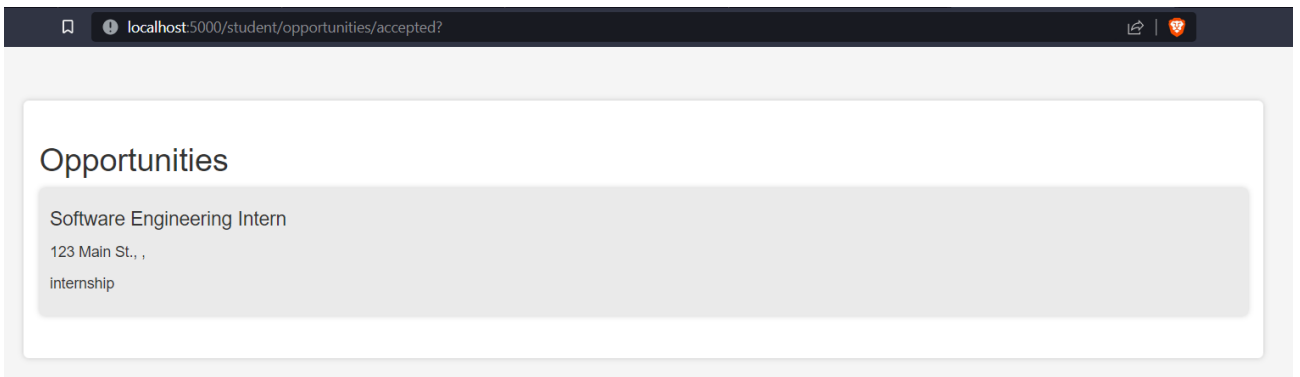
After selecting the student, the status updates to 'selected' and the round number updates to '2'.



The screenshot shows a SQL query interface with a text input field containing the query: `select * from app_opp where app_opp.student_id=1;`. Below the query, there is a toolbar with options like 'Result Grid', 'Filter Rows', 'Export', and 'Wrap Cell Content'. The main area displays a table with the following data:

|   | student_id | resume_id | OPP_ID | round_number_reached | status   |
|---|------------|-----------|--------|----------------------|----------|
| ▶ | 1          | 1         | 1      | 2                    | selected |

This is the student view, in the URL you can see that this is the opportunity in which they are accepted.



C. **DELETE**

a. **Before** deleting

We will be deleting the last column of 'Business Development Placement' opportunity from the data.

finalhai SQL File 3\*

Limit to 50000 rows

```
1 • select * from opportunity;
```

Result Grid Filter Rows: Export: Wrap Cell Content:

|   | opp_id | opp_type   | opp_title                         | address_line_1            | address_line_2 | address_line_3 | company_id |
|---|--------|------------|-----------------------------------|---------------------------|----------------|----------------|------------|
| ▶ | 1      | internship | Software Engineering Intern       | 123 Main St.              |                |                | 1          |
|   | 2      | placement  | Marketing Manager                 | 456 Oak Ave.              |                |                | 2          |
|   | 3      | internship | Data Science Intern               | 789 Elm St.               |                |                | 3          |
|   | 4      | placement  | Sales Representative              | 246 Maple Rd.             |                |                | 4          |
|   | 5      | internship | Product Management Intern         | 135 Cedar Ln.             |                |                | 5          |
|   | 6      | internship | Software Engineering Internship   | 1 Infinite Loop           | Cupertino      | California     | 6          |
|   | 7      | placement  | Hardware Engineer Placement       | One Microsoft Way         | Redmond        | Washington     | 7          |
|   | 8      | internship | Data Science Internship           | 1600 Amphitheatre Parkway | Mountain View  | California     | 8          |
|   | 9      | placement  | Marketing Placement               | 3500 Deer Creek Road      | Palo Alto      | California     | 9          |
|   | 10     | internship | Software Development Internship   | 410 Terry Avenue North    | Seattle        | Washington     | 1          |
|   | 11     | placement  | Product Manager Placement         | 1 Infinite Loop           | Cupertino      | California     | 2          |
|   | 12     | internship | Data Analyst Internship           | One Microsoft Way         | Redmond        | Washington     | 2          |
|   | 13     | placement  | Hardware Design Placement         | 1600 Amphitheatre Parkway | Mountain View  | California     | 3          |
|   | 14     | internship | Mobile App Development Internship | 3500 Deer Creek Road      | Palo Alto      | California     | 4          |
|   | 15     | placement  | Business Development Placement    | 410 Terry Avenue North    | Seattle        | Washington     | 5          |

opportunity 4 x

The page looks something like this before deleting the opportunity from the data:

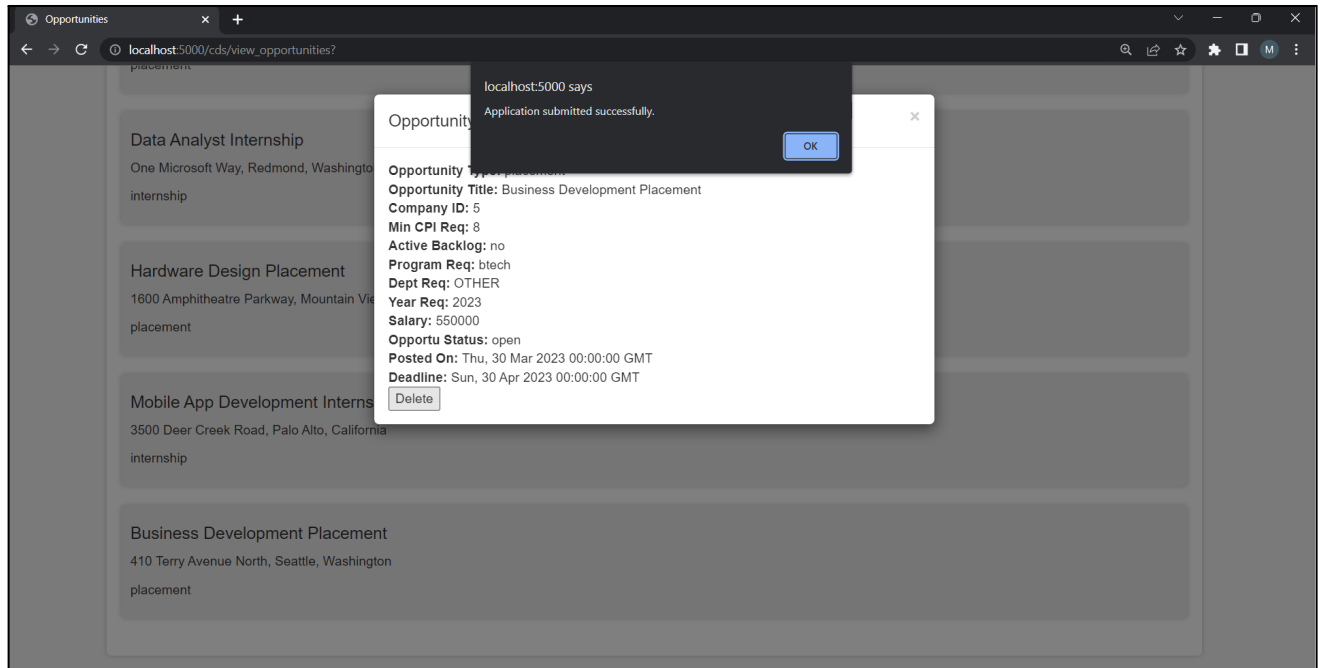
Opportunities

localhost:5000/cds/view\_opportunities?

|  |
|--|
| placement  |
| Data Analyst Internship<br>One Microsoft Way, Redmond, Washington<br>internship                |
| Hardware Design Placement<br>1600 Amphitheatre Parkway, Mountain View, California<br>placement |
| Mobile App Development Internship<br>3500 Deer Creek Road, Palo Alto, California<br>internship |
| Business Development Placement<br>410 Terry Avenue North, Seattle, Washington<br>placement     |

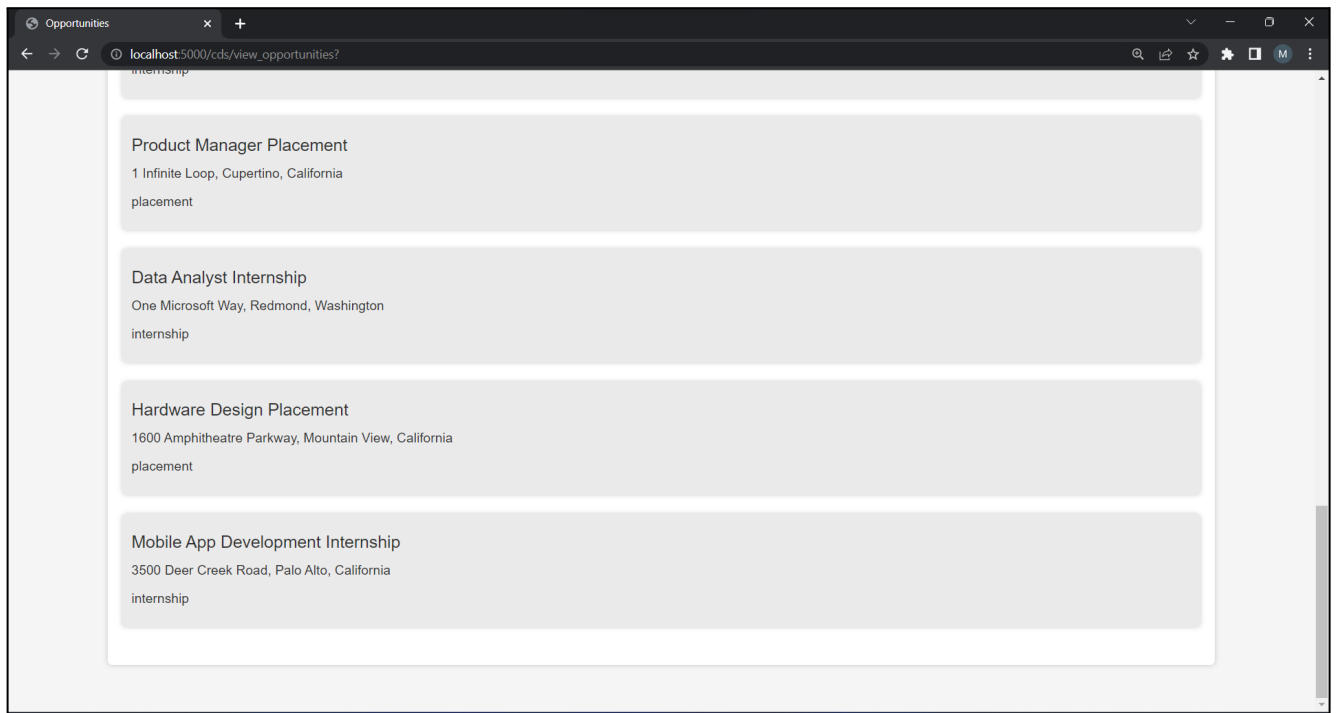
## b. Deleting

Deleting the opportunity using 'DELETE' button.



## c. After deletion

After deleting, the opportunity disappears from the 'veiw opportunity' page.



The opportunity also disappears from the data.

finalhai SQL File 3\*

Limit to 50000 rows

1 • `select * from opportunity;`

Result Grid Filter Rows: Export: Wrap Cell Content:

|   | opp_id | opp_type   | opp_title                         | address_line_1            | address_line_2 | address_line_3 | company_id |
|---|--------|------------|-----------------------------------|---------------------------|----------------|----------------|------------|
| ▶ | 1      | internship | Software Engineering Intern       | 123 Main St.              |                |                | 1          |
|   | 2      | placement  | Marketing Manager                 | 456 Oak Ave.              |                |                | 2          |
|   | 3      | internship | Data Science Intern               | 789 Elm St.               |                |                | 3          |
|   | 4      | placement  | Sales Representative              | 246 Maple Rd.             |                |                | 4          |
|   | 5      | internship | Product Management Intern         | 135 Cedar Ln.             |                |                | 5          |
|   | 6      | internship | Software Engineering Internship   | 1 Infinite Loop           | Cupertino      | California     | 6          |
|   | 7      | placement  | Hardware Engineer Placement       | One Microsoft Way         | Redmond        | Washington     | 7          |
|   | 8      | internship | Data Science Internship           | 1600 Amphitheatre Parkway | Mountain View  | California     | 8          |
|   | 9      | placement  | Marketing Placement               | 3500 Deer Creek Road      | Palo Alto      | California     | 9          |
|   | 10     | internship | Software Development Internship   | 410 Terry Avenue North    | Seattle        | Washington     | 1          |
|   | 11     | placement  | Product Manager Placement         | 1 Infinite Loop           | Cupertino      | California     | 2          |
|   | 12     | internship | Data Analyst Internship           | One Microsoft Way         | Redmond        | Washington     | 2          |
|   | 13     | placement  | Hardware Design Placement         | 1600 Amphitheatre Parkway | Mountain View  | California     | 3          |
|   | 14     | internship | Mobile App Development Internship | 3500 Deer Creek Road      | Palo Alto      | California     | 4          |

## D. RENAME

### a. Before renaming

CDS employee view, here we can rename student names and other details.

Student Details

localhost:5000/cds/student\_profile?

### Student Details

|  |   |   |
|--|---|---|
| <p>changed oxezh</p> <p>Department: ME</p> <p>CPI: 8</p> <p>Edit</p>   | <p>Shruhriddoe</p> <p>Department: EE</p> <p>CPI: 9.9</p> <p>Edit</p>  | <p>Sarah Wilson</p> <p>Department: CE</p> <p>CPI: 8.2</p> <p>Edit</p> |
| <p>Mihir Taylor</p> <p>Department: CSE</p> <p>CPI: 9.9</p> <p>Edit</p> | <p>Emily Clark</p> <p>Department: MSE</p> <p>CPI: 9.1</p> <p>Edit</p> | <p>nywxd gpsxd</p> <p>Department: ME</p> <p>CPI: 9</p> <p>Edit</p>    |
| <p>rpwac jmdeo</p> <p>Department: ME</p> <p>CPI: 7</p>                 | <p>othek dntrh</p> <p>Department: ME</p> <p>CPI: 8</p>                | <p>byahc wzpeh</p> <p>Department: ME</p> <p>CPI: 8</p>                |

Data before renaming this student.

| Result Grid  |            |                               |                    |                     |                   |               |      |      |                 |        |
|--------------|------------|-------------------------------|--------------------|---------------------|-------------------|---------------|------|------|-----------------|--------|
| Filter Rows: |            |                               |                    |                     |                   |               |      |      |                 |        |
|              | student_id | student_email_id              | student_first_name | student_middle_name | student_last_name | student_image | dept | CPI  | active_backlogs | gender |
| ▶            | 2          | shruhridd.banthia@iitgn.ac.in | Shruhridd          | bunty               | Doe               | NULL          | EE   | 9.90 | yes             | male   |
| ✱            | NULL       | NULL                          | NULL               | NULL                | NULL              | NULL          | NULL | NULL | NULL            | NULL   |

b. **Renaming**

Process

Student Details

changed oxezh  
Department: ME  
CPI: 8  
Edit

Mihir Taylor  
Department: CSE  
CPI: 9.9  
Edit

rpwac jmdeo  
Department: ME  
CPI: 7

Edit Student Details

Student Id:  
2

First Name:  
sanskar

Middle Name:  
ji

Last Name:  
sharma

Image:  
trial

Department:  
ME

CPI:  
8

Sarah Wilson  
Department: CE  
CPI: 8.2  
Edit

nywxd gpsxd  
Department: ME  
CPI: 9  
Edit

byahc wzpeh  
Department: ME  
CPI: 8

c. **After state**

Changed the named of the student .

Student Details

changed oxezh

Department: ME

CPI: 8

Edit

sanskar sharma

Department: ME

CPI: 8

Edit

Sarah Wilson

Department: CE

CPI: 8.2

Edit

Mihir Taylor

Department: CSE

CPI: 9.9

Edit

Emily Clark

Department: MSE

CPI: 9.1

Edit

nywxd gpsxd

Department: ME

CPI: 9

Edit

rpwac jmdeo

Department: ME

CPI: 7

othek dntrh

Department: ME

CPI: 8

byahc wzpeh

Department: ME

CPI: 8

1254 • `select * from student where student_id=2;`

1255

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

| student_id | student_email_id            | student_first_name | student_middle_name | student_last_name | student_image | dept | CPI  | active_backlogs | gender | study_year |
|------------|-----------------------------|--------------------|---------------------|-------------------|---------------|------|------|-----------------|--------|------------|
| 2          | shruhid.banthia@iitgn.ac.in | sanskar            | ji                  | sharma            | trial         | ME   | 8.00 | no              | male   | 2          |

## E. WHERE clause

Student Dashboard

Profile

My Resumes

Opportunities ▾

Applied

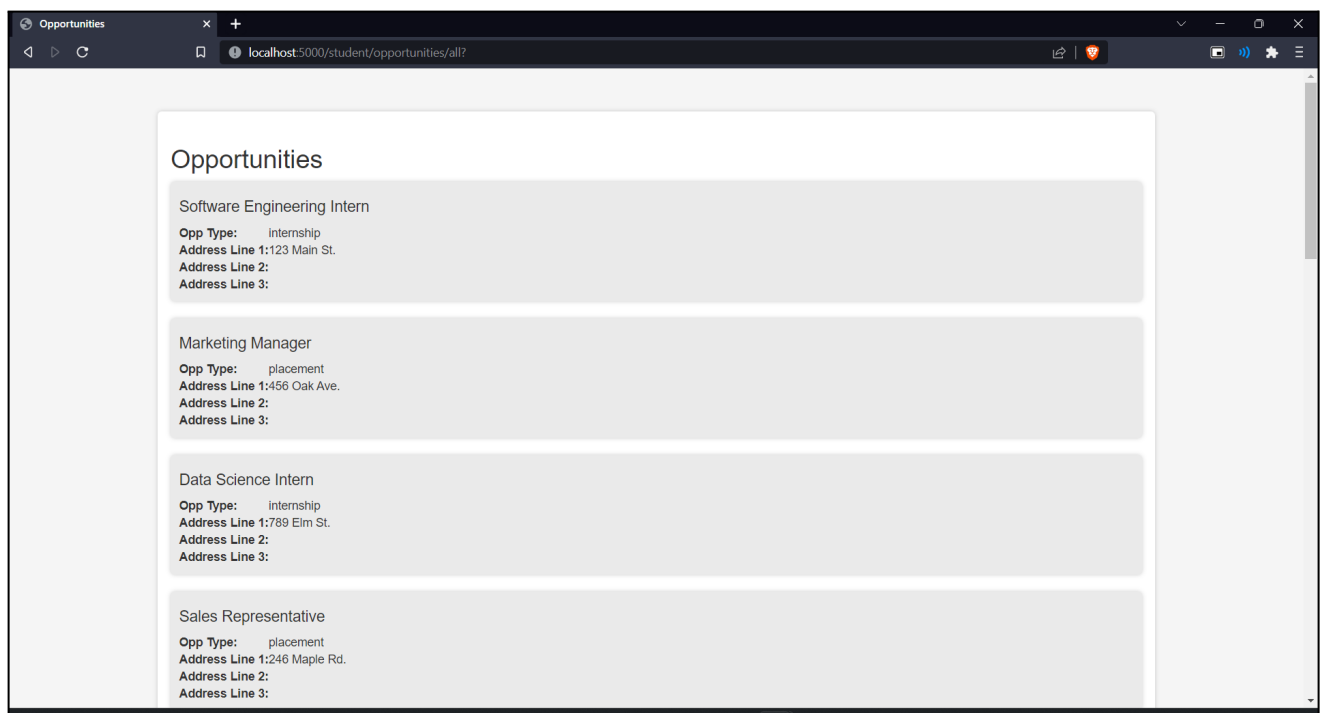
All Opportunities

Accepted

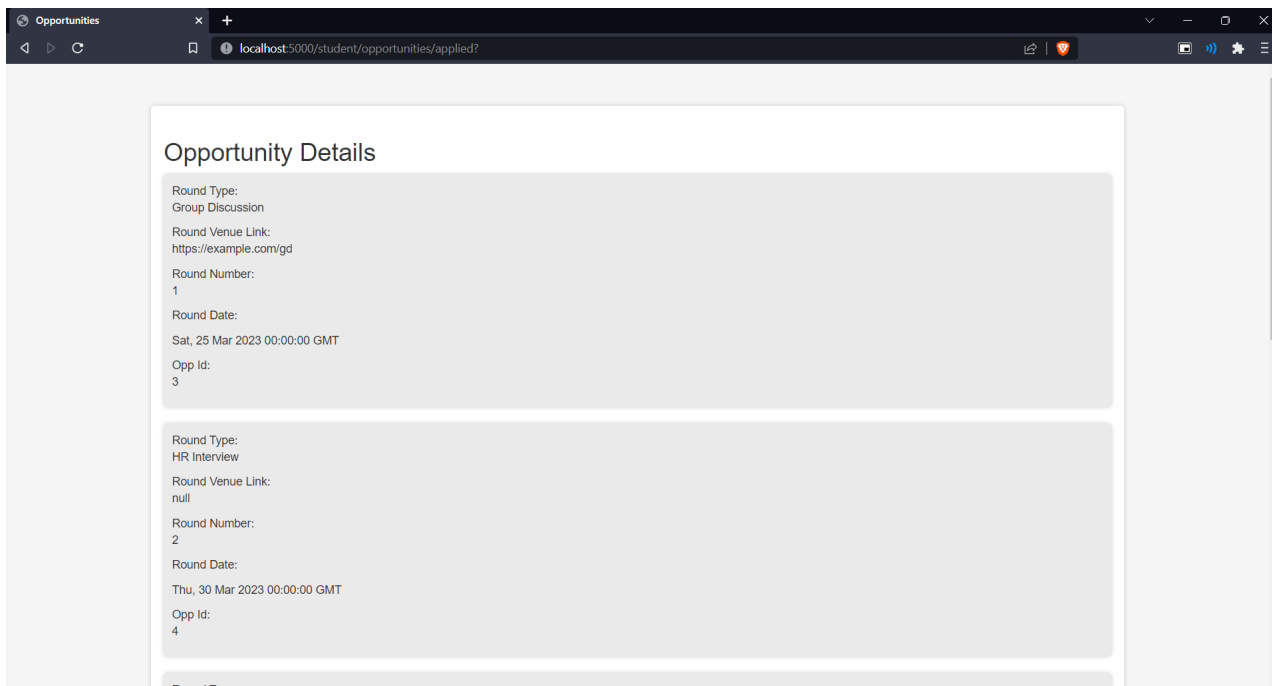
Rejected

Copyright © 2023

On clicking 'All Opportunities' it shows all available opportunities.



The following are the applied opportunities.



On clicking 'Applied', it shows the 'All Opportunities' which we have applied to. Here we do not explicitly implement the where clause but the where clause is used at multiple instances in the backend for example:

```
query = "SELECT * FROM student WHERE student.student_id in (SELECT app_opp.student_id FROM app_opp WHERE app_opp.opp_id = %s);"
```

Here we can see the where clause being used in the code where we select the students with only the required id.



## Components and views of whole Website

There is total of three views:

### 1. Student

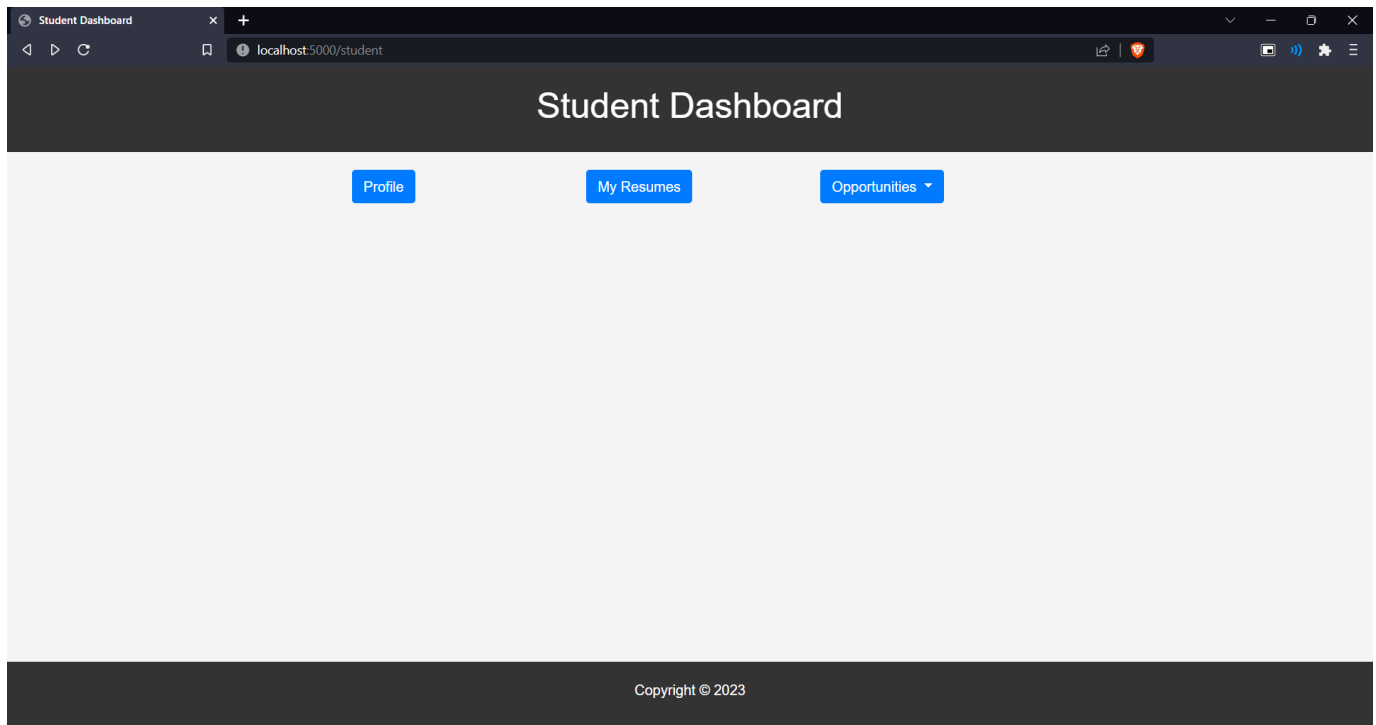
- Profilemo
- My Resume
- Opportunities
  - Applied
  - All Opportunities
  - Accepted
  - Rejected

### 2. CDS Employee (Admin)

### 3. Company Point of Contact

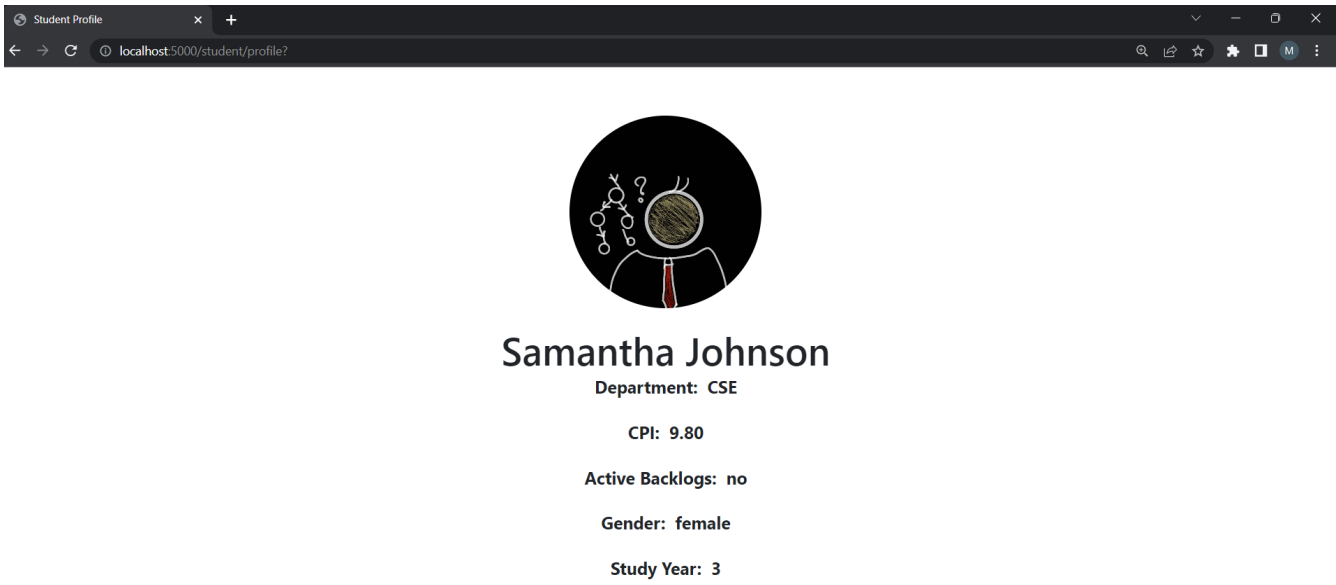
## Student's View

- Dashboard for student



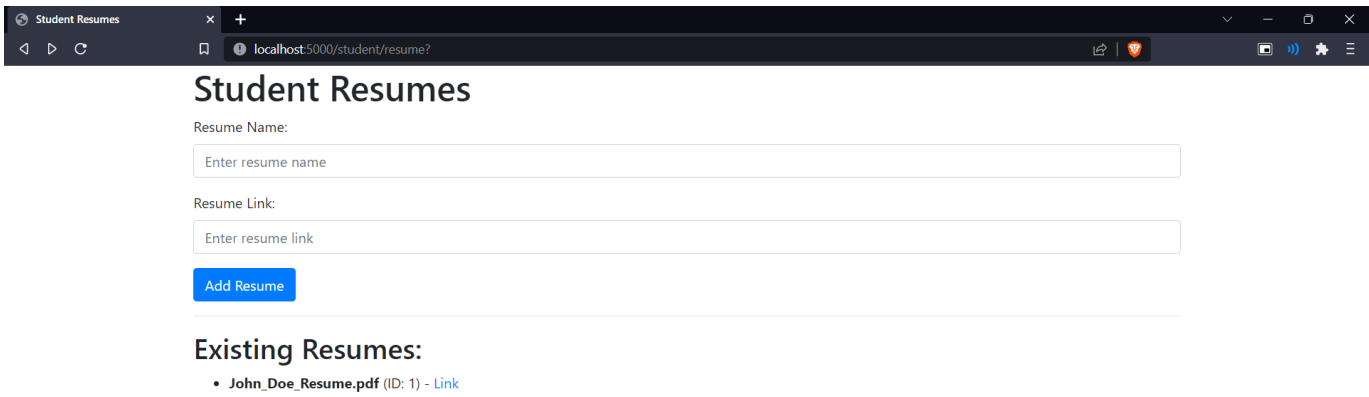
student/profile

- View of the student profile page

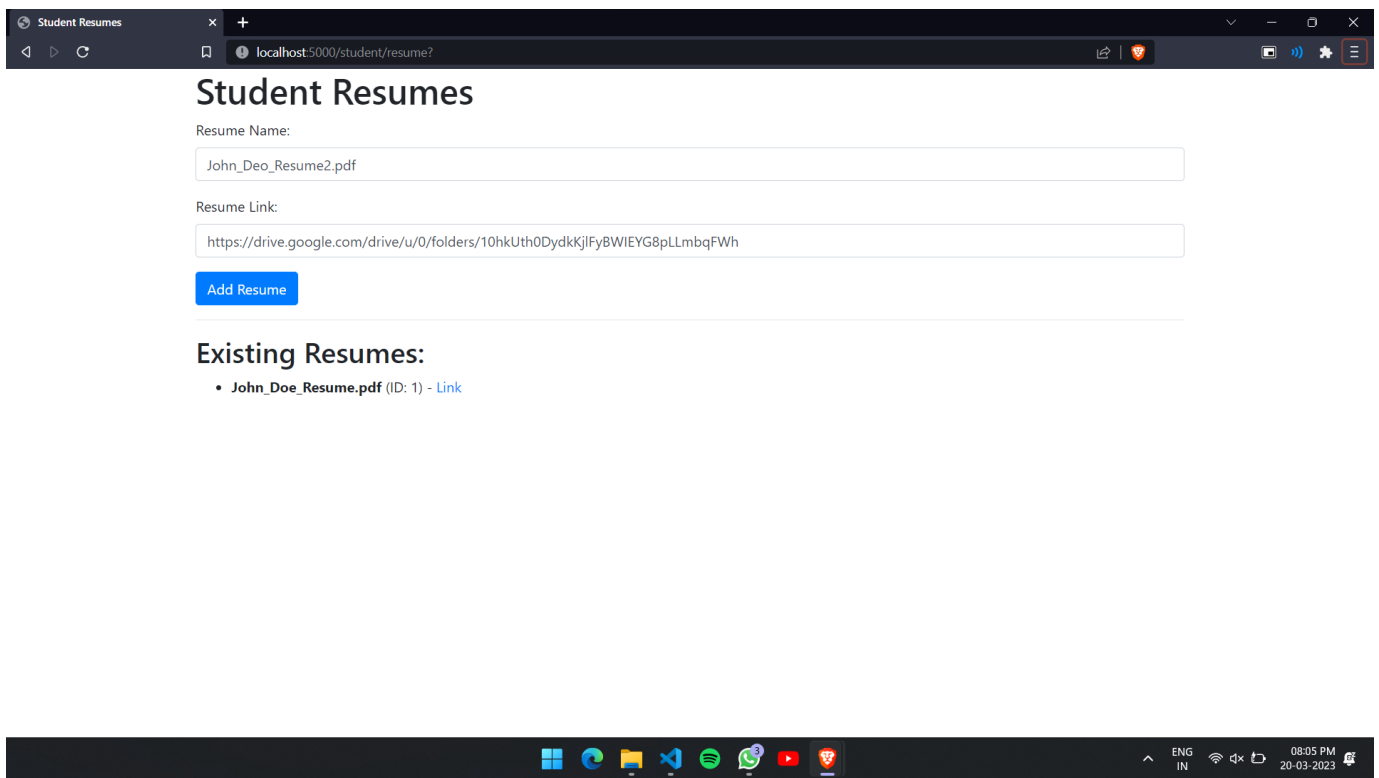


/student/resume?

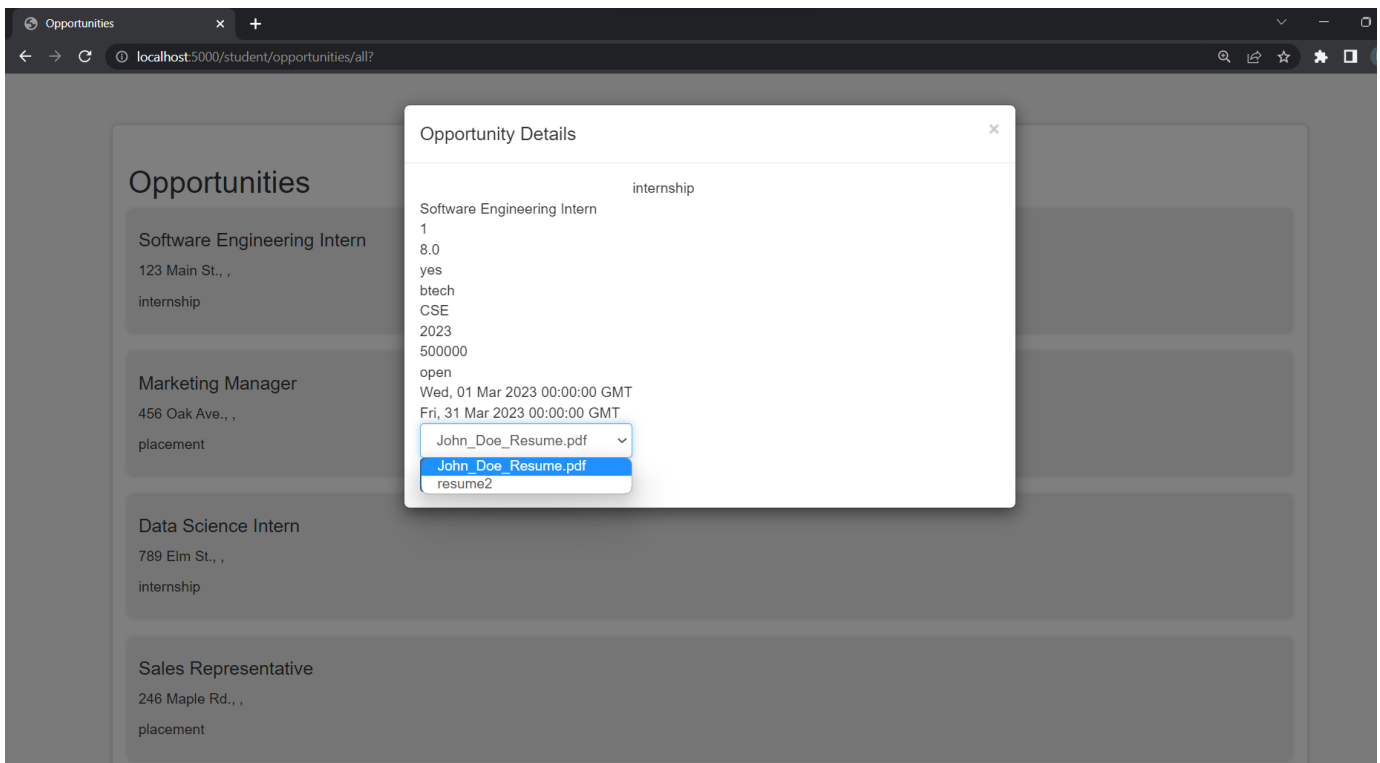
Here student can enter their resume and it will be saved in database with the name given by the student.



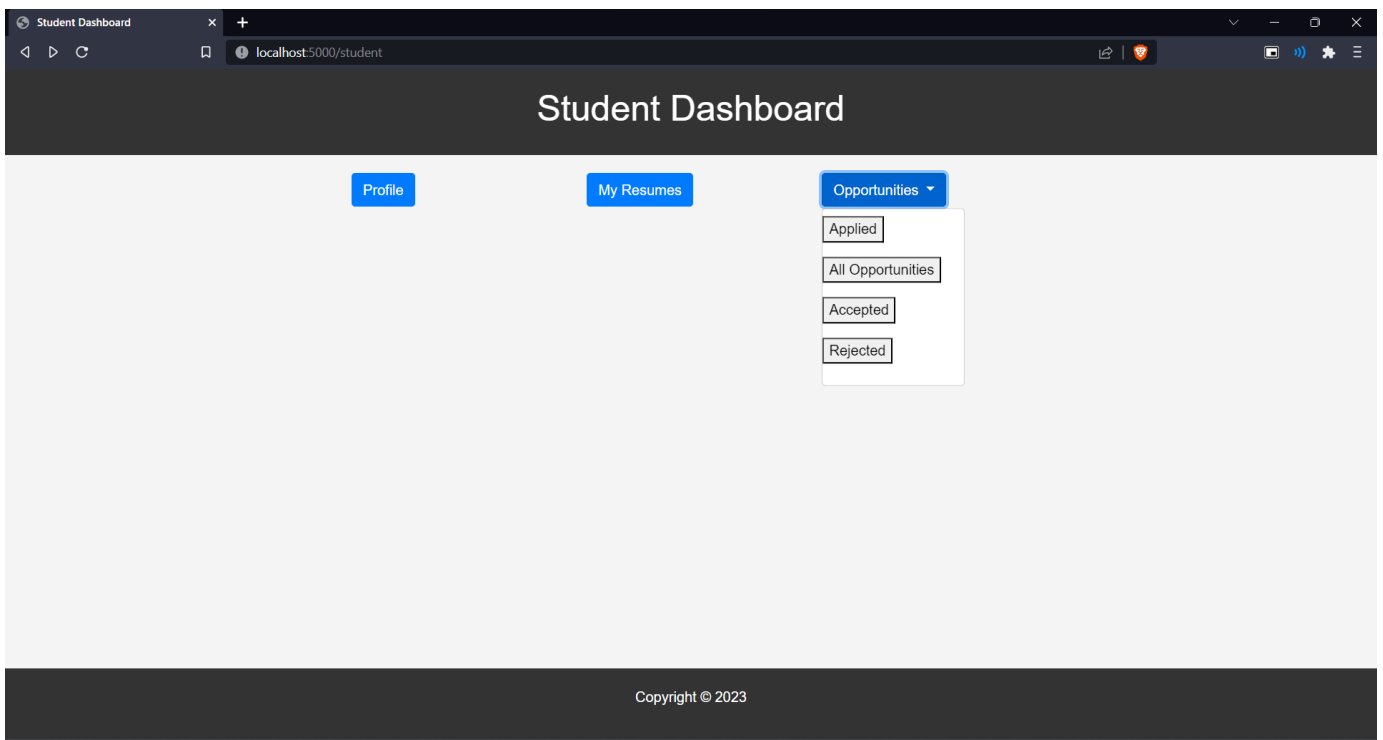
Adding new resume:



After inserting one resume:

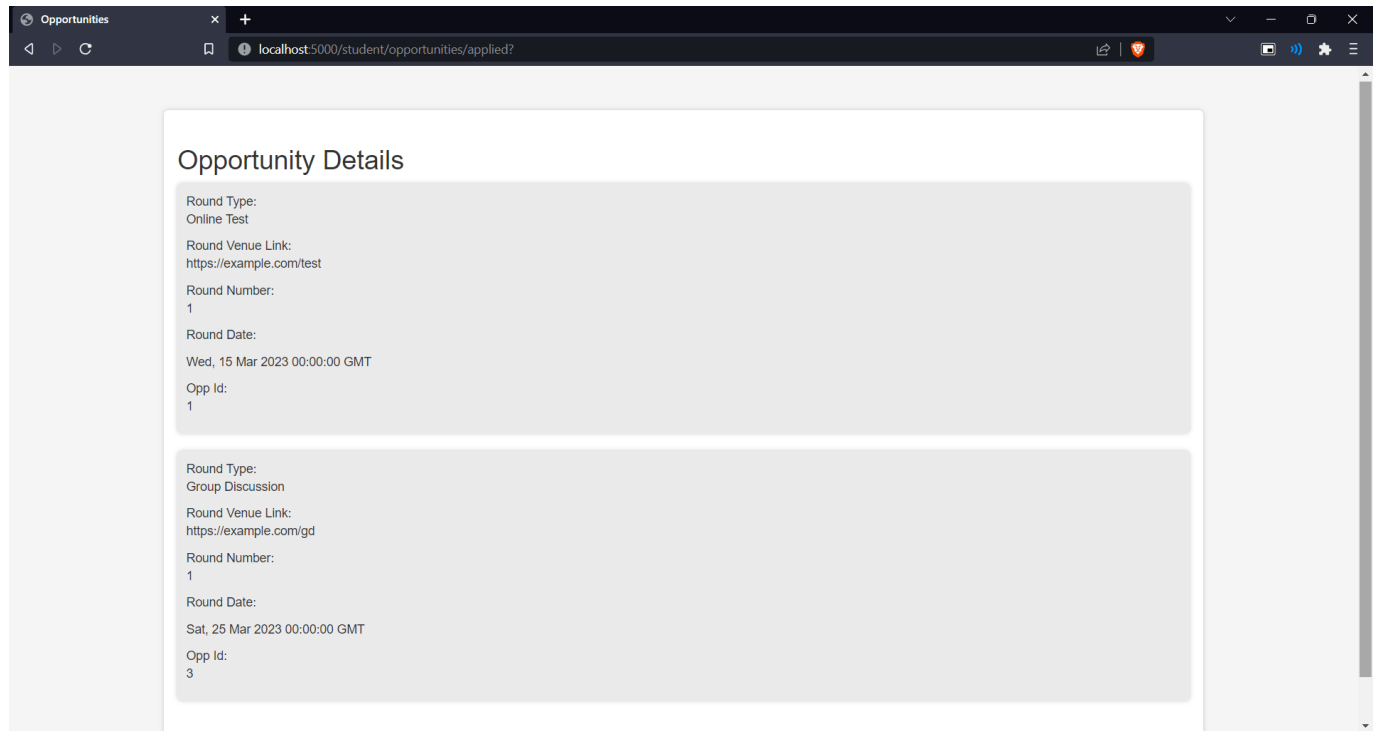


There are four options in opportunities: Applied, All Opportunities, Accepted, Rejected

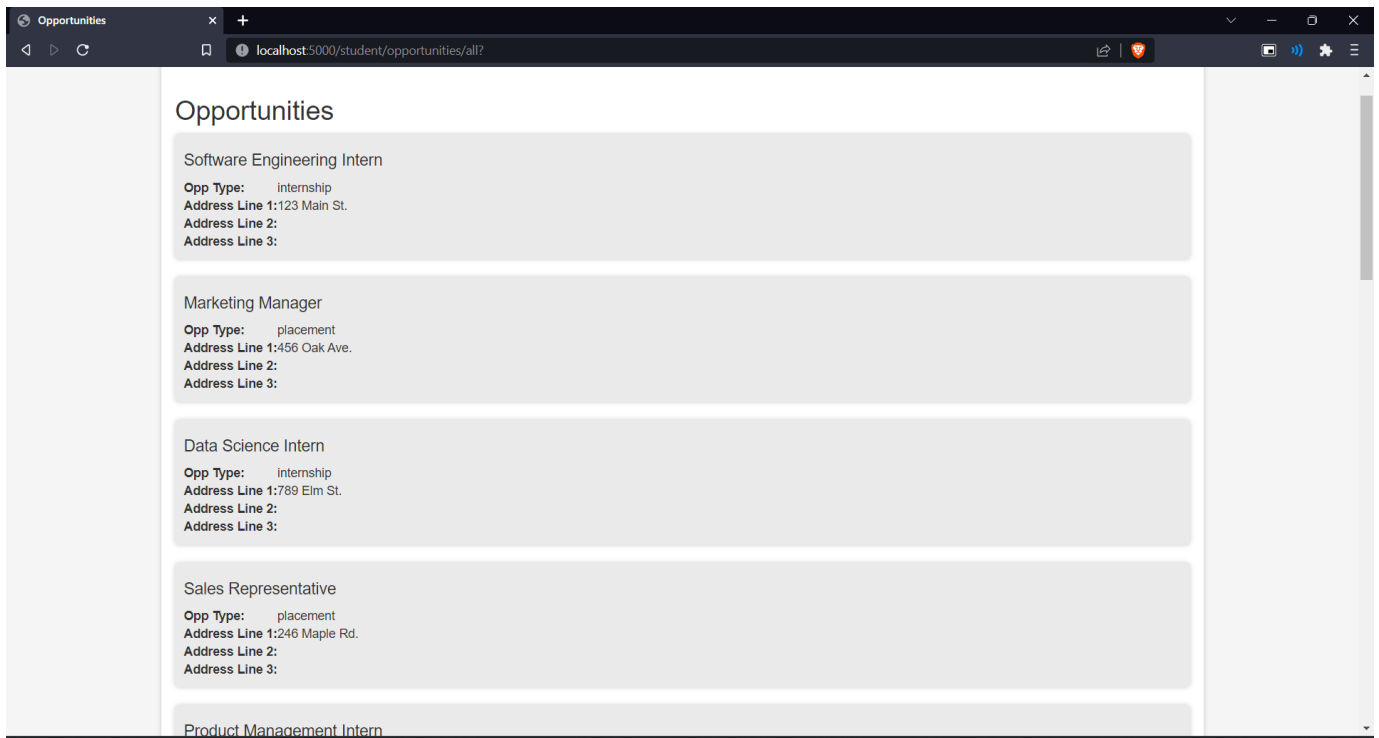


student/opportunities/applied

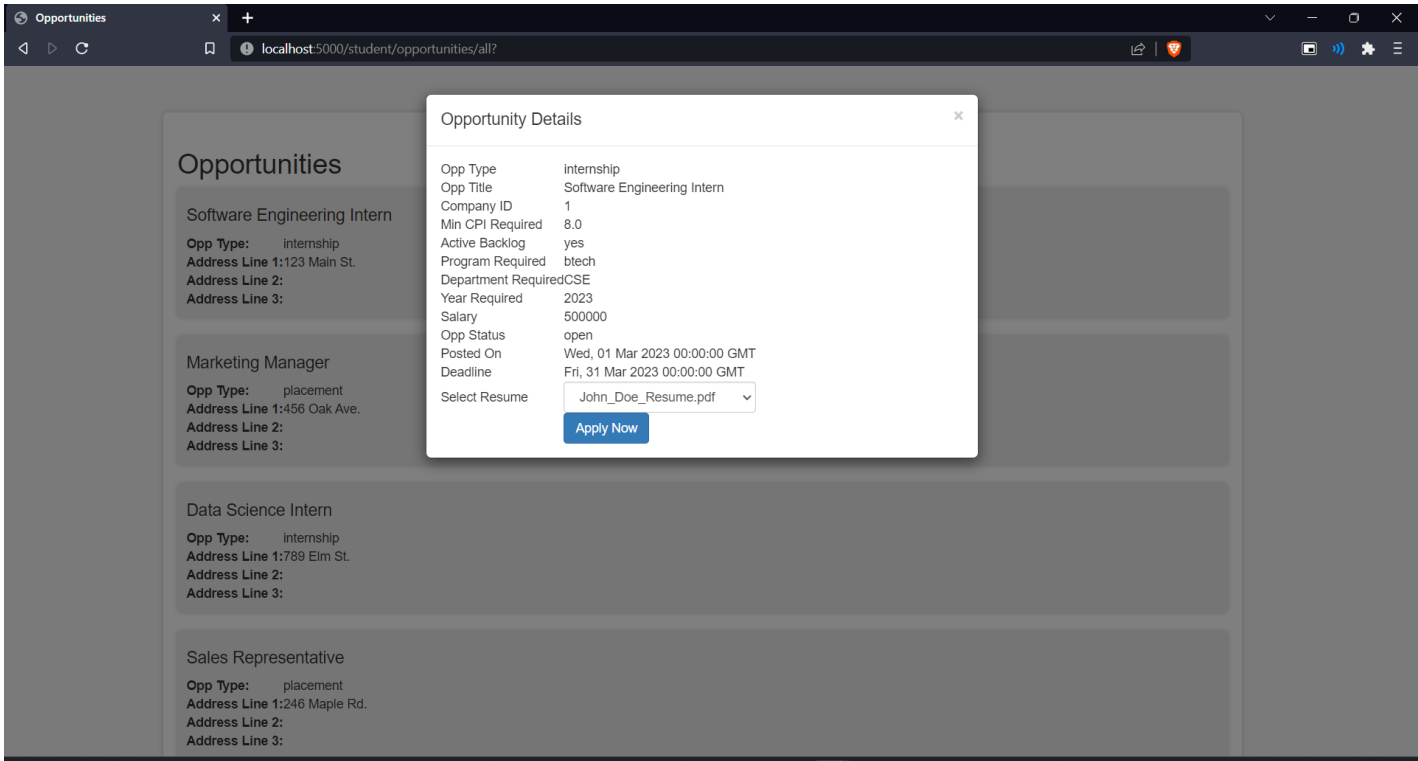
Here student can see all the opportunities on which student has applied.



Here students can see all the opportunities on which they can apply.

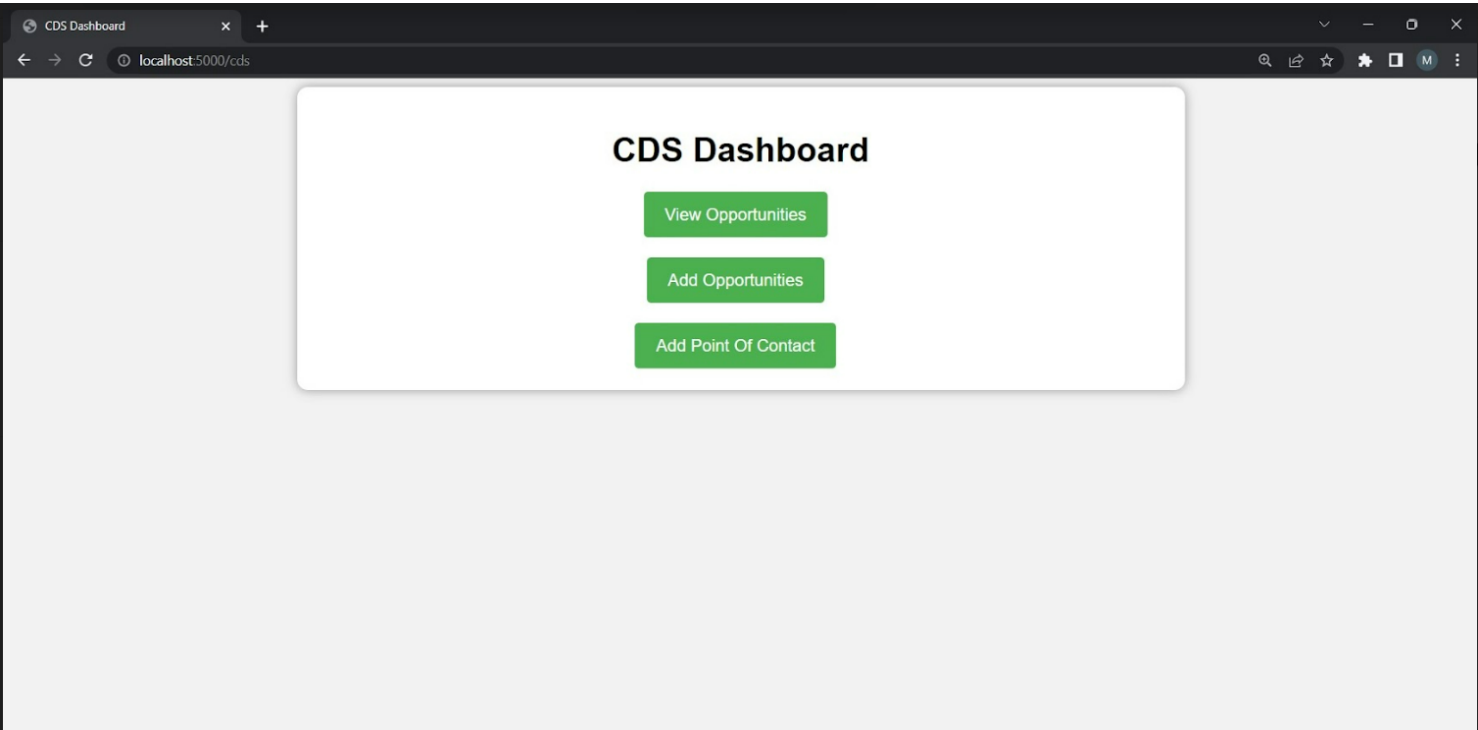


Student can see the details of the opportunities available by clicking on it. It has the option of selecting the one of the uploaded resume and applying.

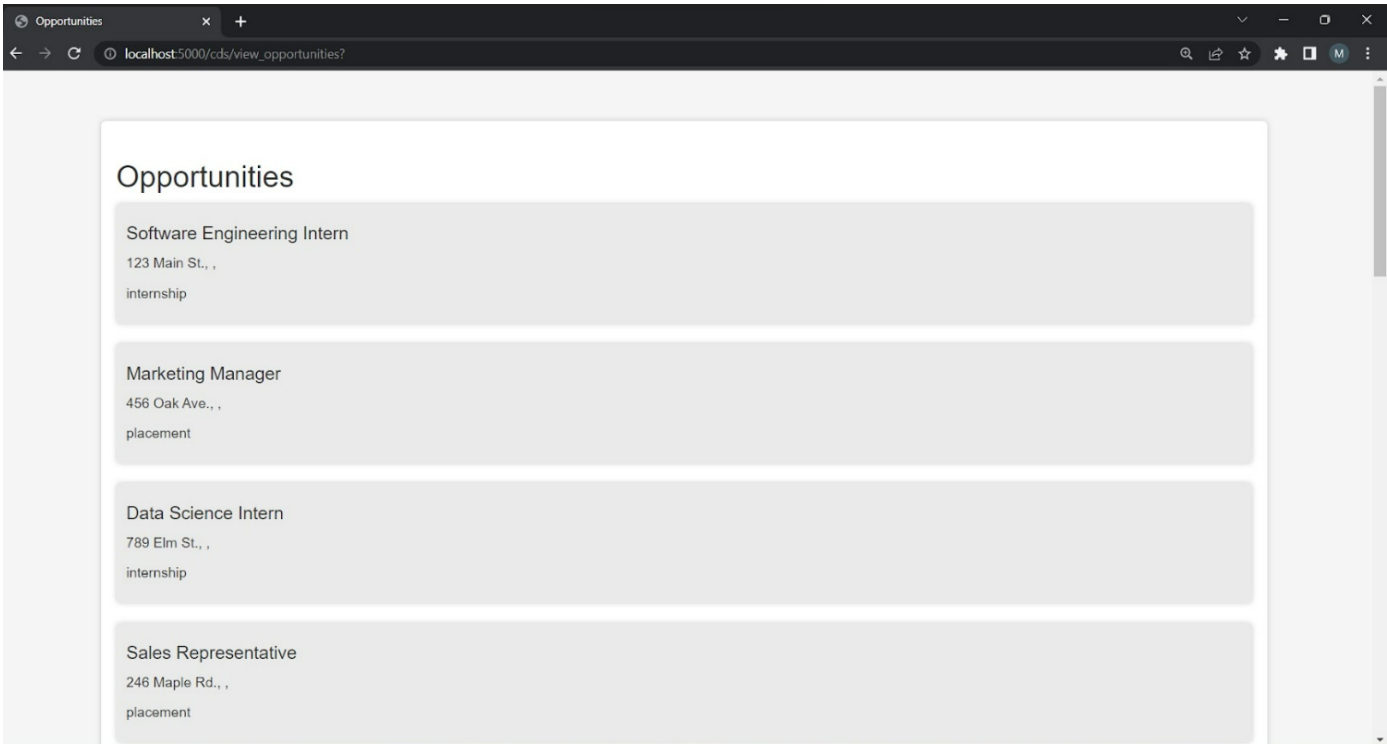


CDS employee view:

As soon as the employee of the CDS sign's in,, it opens up the below dashboard.



Upon clicking on view opportunities, the following page opens up, where we list down all opportunities available.



Upon clicking on add opportunities, the following form opens up where the cds employee can add an opportunity by filling the required details.

Opportunity Form

## Opportunity Form

### Opportunity Details

Opportunity Type:

Internship

Internship

Placement

Address Line 1:

Address Line 2:

Address Line 3:

Opportunity Form

## Requirements

Opportunity ID:

Minimum CPI Requirement:

Active Backlog:

Select an option

Program Requirement:

Select a program

Department Requirement:

Select a department

Year Requirement:

Salary:

Upon clicking on add point of contact, the following form opens up such that the CDS employee can add POC by filling the below requirements.

Point of Contact Form

First Name:

Middle Name:

Last Name:

Designation:

Opp ID:

Email:

## Google Authentication









Any person can login only if their email id is available in the database.

Sign in - Google Accounts

accounts.google.com/o/oauth2/v2/auth/oauthchooseaccount?response\_type=code&client\_id=515566156695-um1sos28i4a2ftr37eaot6l4clvovjs.app... |

Sign in with Google

Choose an account  
to continue to [database project](#)

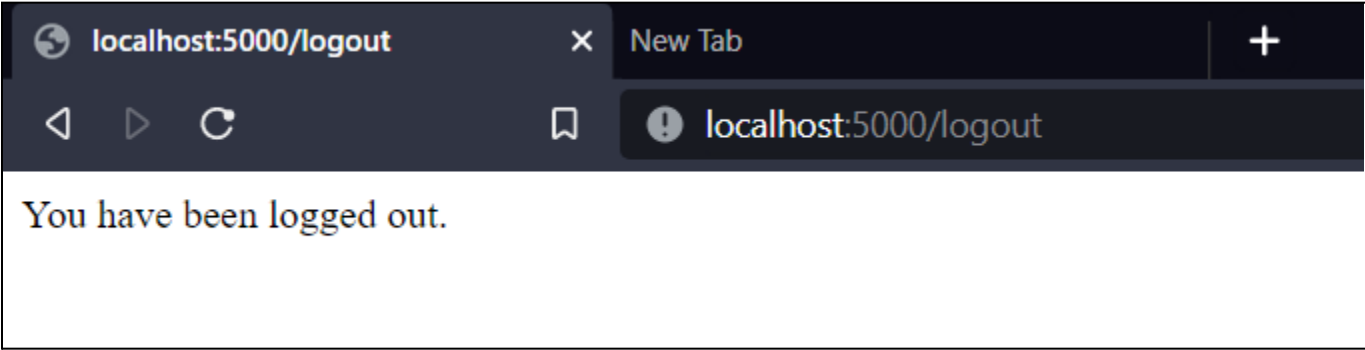
-  Dhyey Thummar  
dhyeythummar8@gmail.com
-  Dhyey Thummar  
dhyeykumar.thummar@iitgn.ac.in
-  Michael Skywalker  
sky752walker@gmail.com
-  John Doom  
legend752star@gmail.com
-  Dhyey Thummar  
dhyeythummar7@gmail.com
-  Dhyey Max  
starhacker752@gmail.com
-  DigiS IITGN  
digiis@iitgn.ac.in
-  Use another account

To continue, Google will share your name, email address,

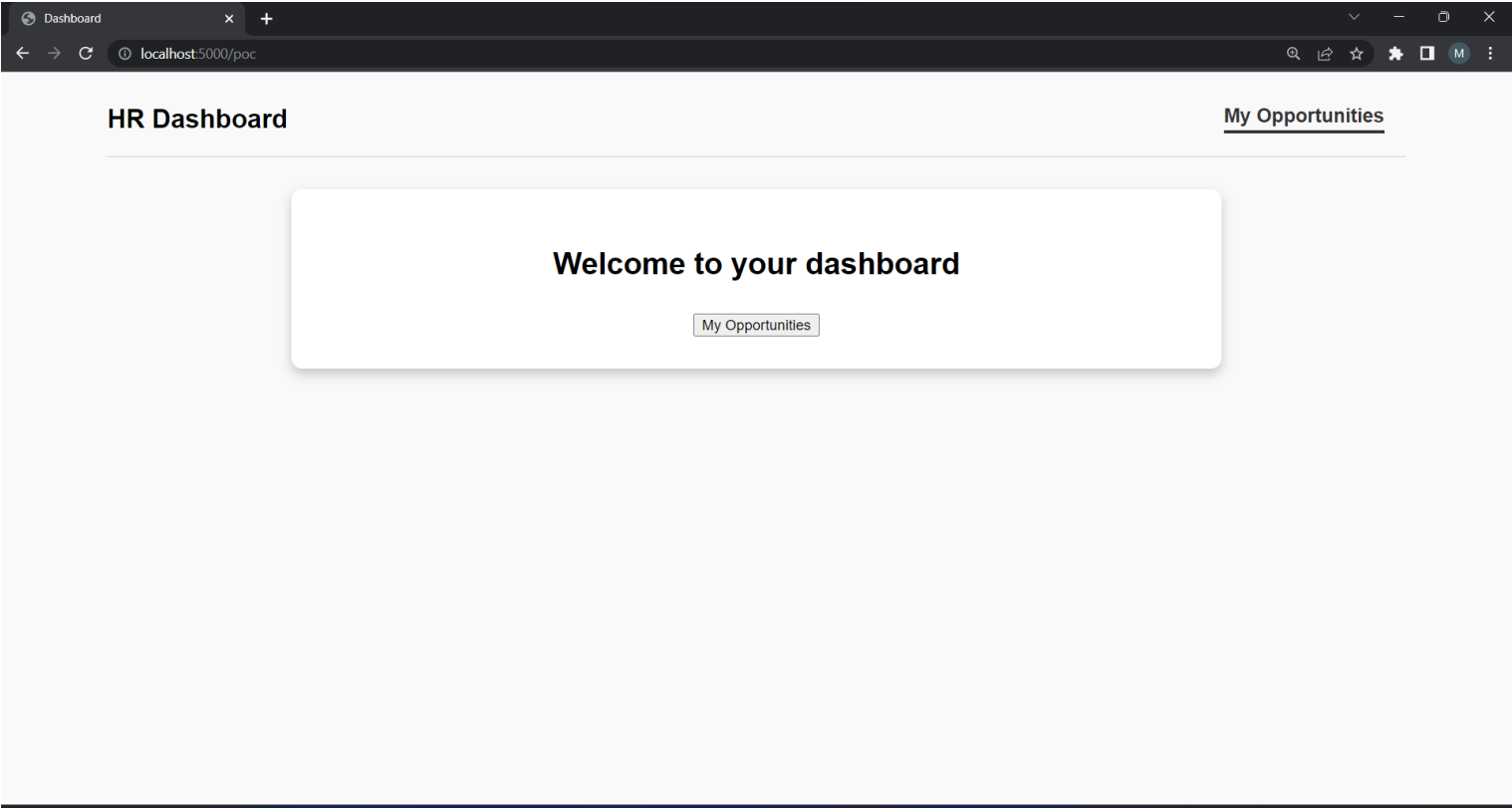


- Profile
- 
- My Resumes
- Opportunities

Logging out:



HR view:



## **Contributions:**

### **G1:**

#### **Pushpendra Pratap Singh (20110151)**

- Worked on the front end of CDS pages and POC pages.
- Worked on the integration of front end and back end. Worked on implementing successful requests in the frontend.
- Fixed the bugs in frontend..

#### **Saatvik Rao (20110175)**

- Wrote the code for front end of student pages
- Wrote the steps to run the Web App
- Added and explained screenshots of successful execution

#### **Sahil Agrawal (20110178)**

- Wrote the code for the front end of student pages.
- Wrote the code for the front end of cds pages.

#### **Sanskar Sharma (20110185)**

- Designed the frontend views
- Designed the HTML Pages and wrote the Javascript, CSS and JQuery code for most of the pages
- Integrated Frontend and Backend with Team
- Described all the GET and POST requests that are needed for the application to operate

#### **Tanvi Dixit (20110212)**

- Worked on the frontend of the web app
- Created and assisted HTML codes of web pages for all views of the web application
- Checked for any inconsistencies in the cardinalities while running of the application
- Writing and formatting the report

### **G2:**

#### **Dhyeykumar Thummar (20110059)**

- Handled POST requests (for student, CDS employee and poc) in app.py.
- Wrote all the required SQL queries for the post requests.
- Helped in debugging the code.

#### **Ksheer Sagar Agrawal (20110098)**

- Started the repository & templates for the front & backend.
- Wrote code for backend, wrote different sql queries.
- Checked flask, mysql connection & all necessary pip installs.

#### **Mihir Sutariya (20110208)**

- Wrote code for backend, wrote different sql queries. Wrote code for google authentication.
- Gave all the user's unique occupation.
- Fixed bugs and review code pushed on github.

**R Yeeshu Dhurandhar (20110152)**

- 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

**Shruhid Banthia (20110198)**

- Wrote code for backend, wrote different sql queries.
- Integrated the frontend and backend components. Fixed bugs and review code pushed on github
- Contributed for get requests and SQL commands for student required for different requests

**Utkarsh Mishra (20110218)**

- Populated the SQL database with required data following the constraints mentioned in the previous assignment
- Helped in documentation