

## Problem Statement & Requirements:

Based on the teams formed, each team is supposed to create a web application of the proposed functionalities in the previous assignments. Each team can be divided into Group G1 and Group G2; G1 is supposed to work on the front-end and G2 on the back-end sections, respectively. You can use Flask + MySQL for developing your WebApp.

The web app should support 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:

- INSERT
- UPDATE
- DELETE
- RENAME
- WHERE clause

Push it to GitHub and share the link for submission.

Please note that all the constraints mentioned in Assignment 1 and 2 should be implemented and functioning in your application.

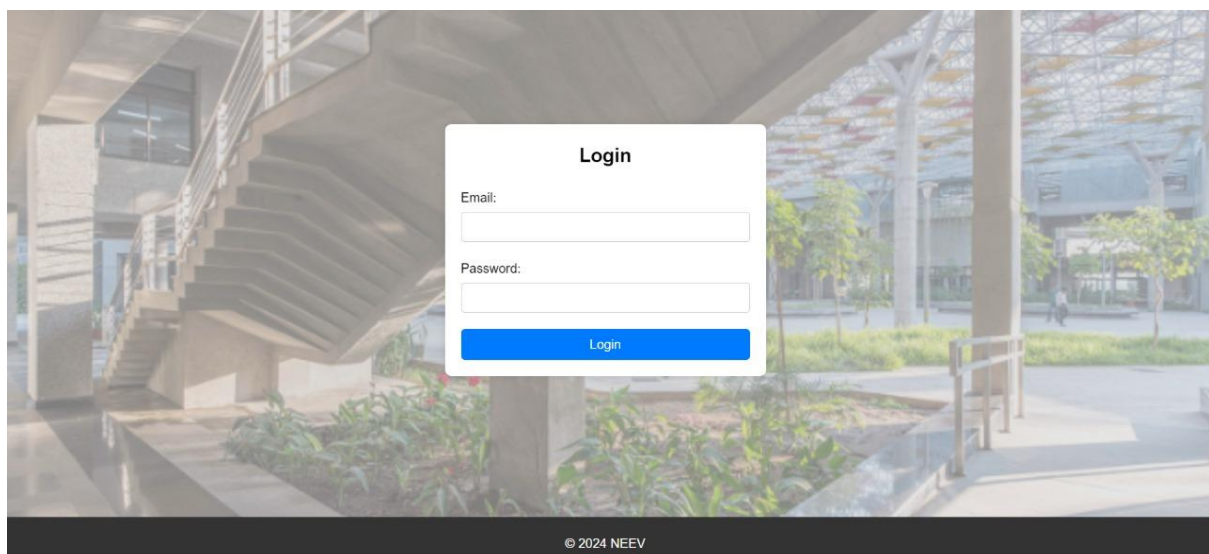
Login page with authentication of users and stakeholders.

### 3.1 Responsibility of G1:

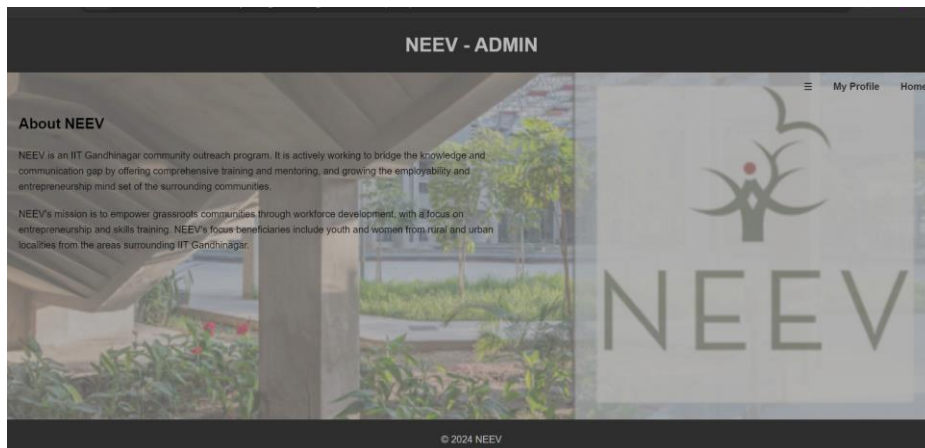
G1 is supposed to design the front end and beautify accordingly using HTML/CSS/JS, you may use any other libraries for the front end, such as Bootstrap or jQuery. But flask is mandatory.

Screenshots:

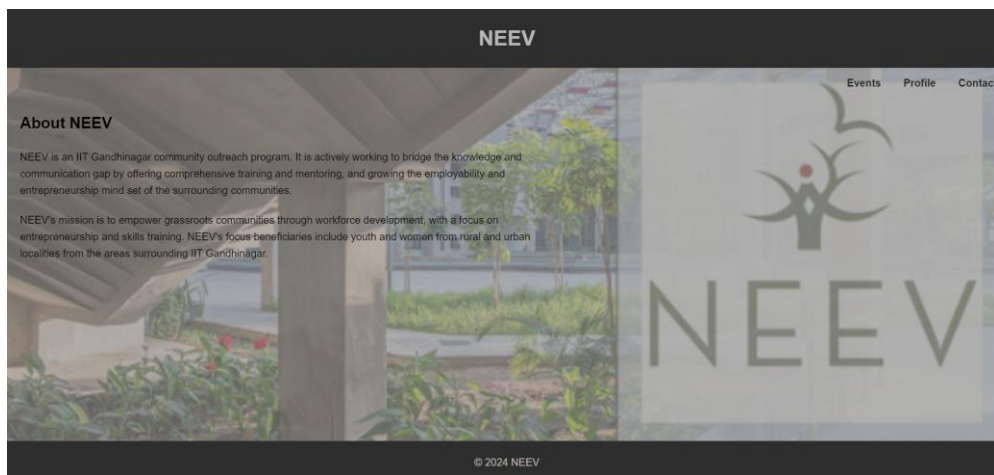
- **Login:** Login interface



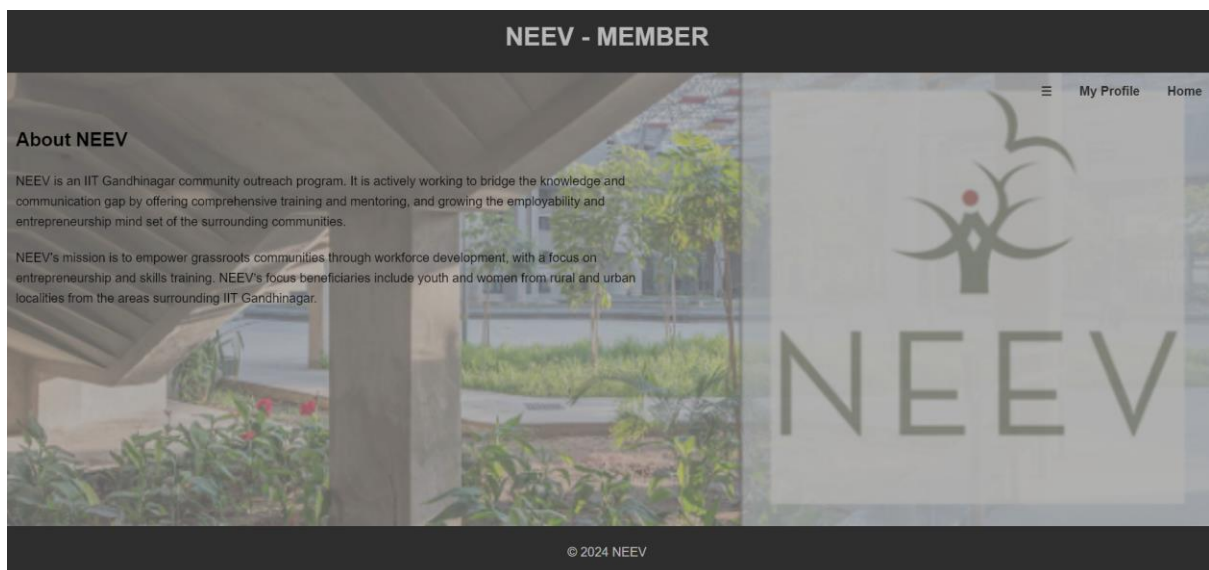
- **Admin Dashboard:** The front page for the Admin.



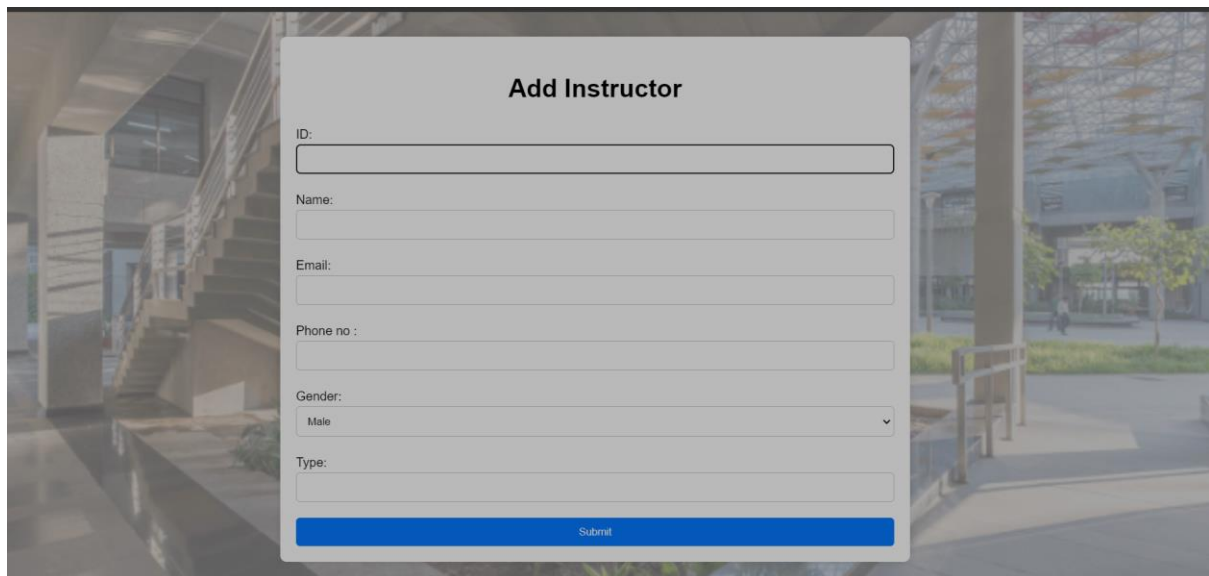
**Main Dashboard: The main page for NEEV members.**



- **Neev\_members Dashboard: This page is displayed for the neev members**



- **Add Instructor page:** Admin can add Instructor using the information below.



### Add Instructor

ID:

Name:

Email:

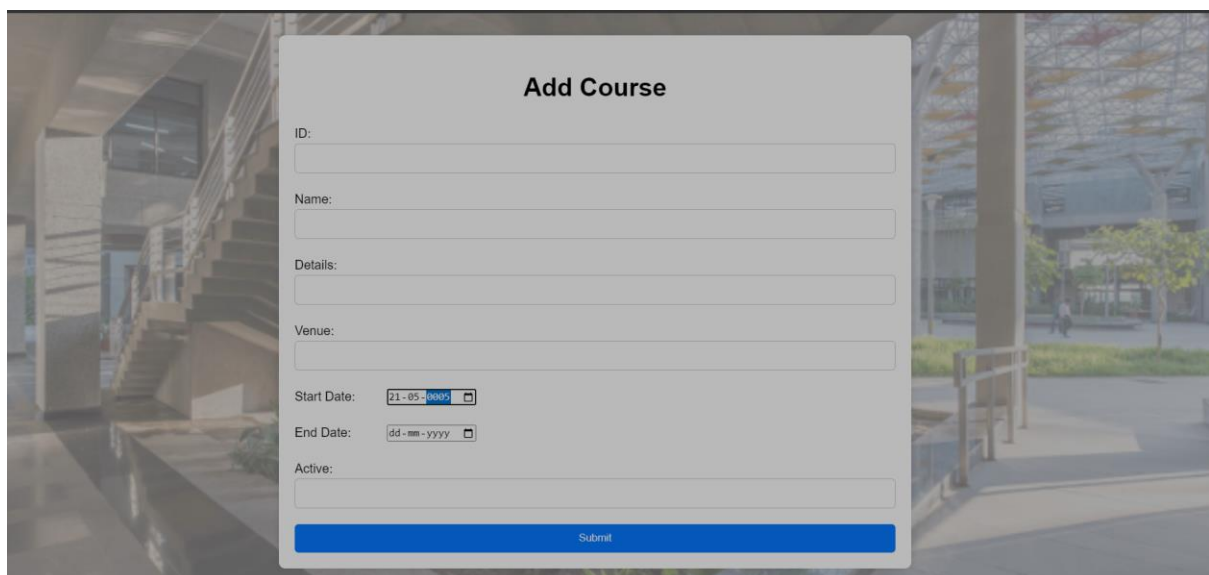
Phone no :

Gender:

Type:

[Submit](#)

- **Add Course page:** Admin can add the course using following information below.



### Add Course

ID:

Name:

Details:

Venue:

Start Date:

End Date:

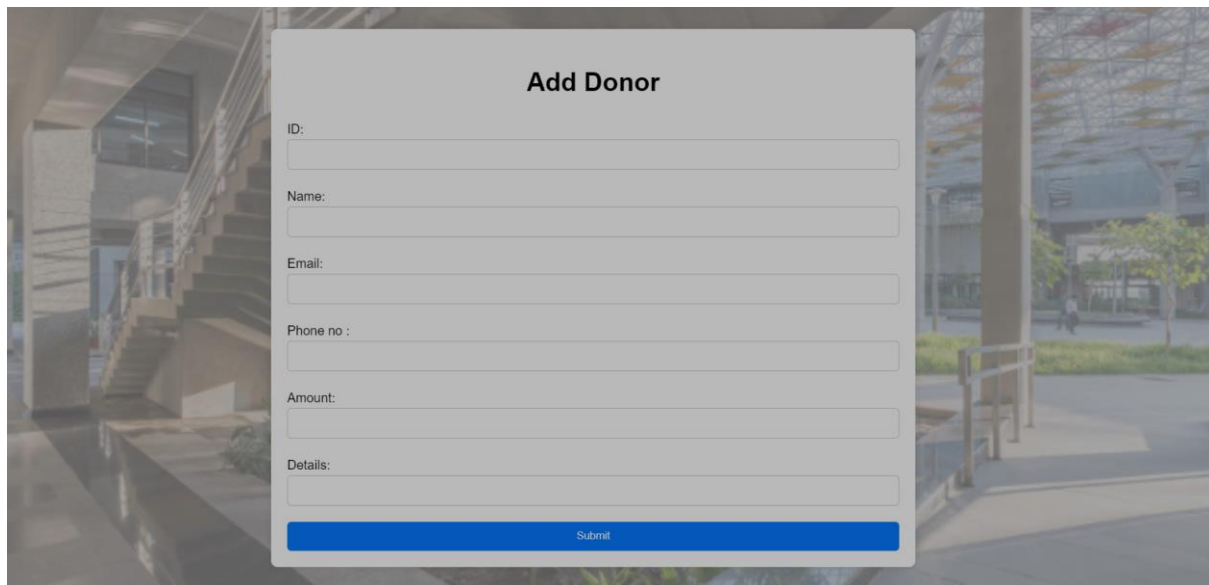
Active:

[Submit](#)

- **Add Volunteer:** Admin can add volunteer using below information.

NEEV								
Name	Email	Phone	Identification_id	Photo	Gender	Address		
Steve Smith	steve@example.com	8934567089	123456789	<a href="#">Link</a>	male	123 Main St, City	<a href="#">Update</a>	<a href="#">Delete</a>
Sarah Davis	sarah@example.com	9089786547	456789012	<a href="#">Link</a>	female	321 Pine St, County	<a href="#">Update</a>	<a href="#">Delete</a>
Himani	himanitrivedi1874@gmail.com	9327035695	2354646	<a href="#">Link</a>	female	Ahme	<a href="#">Update</a>	<a href="#">Delete</a>

- **Add Volunteer:** Admin can add donor using below information.



**Add Donor**

ID:

Name:

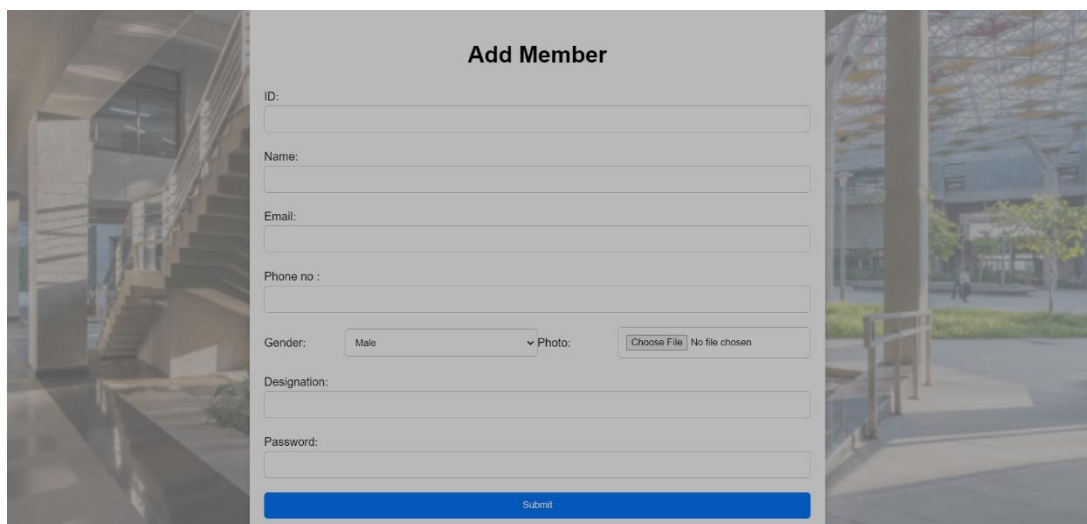
Email:

Phone no :

Amount:

Details:

- **Add Neev member:** Admin can add the course using following information below.



**Add Member**

ID:

Name:

Email:

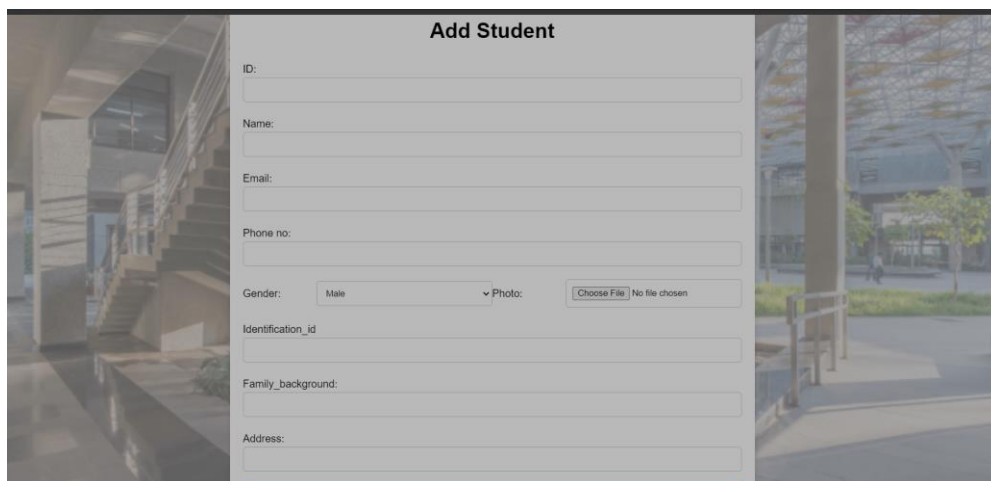
Phone no :

Gender:  Photo:  No file chosen

Designation:

Password:

- **Add Student:** members of Neev can add the student using following information below.



**Add Student**

ID:

Name:

Email:

Phone no:

Gender:  Photo:  No file chosen

Identification\_id

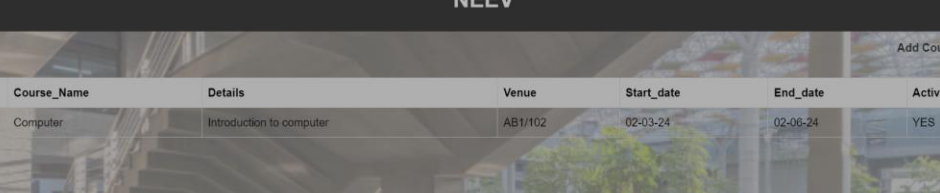
Family\_background:

Address:



- Courses: Below interfaces shows the list of courses in NEEV. The Column 'Active' shows the active course is the course available in specific time period

# NEEV



[Add Course](#)    [Back](#)

Id	Course_Name	Details	Venue	Start_date	End_date	Active
1	Computer	Introduction to computer	AB1/102	02-03-24	02-06-24	YES

© 2024 NEEV

- The page too see the past events.


## Neev - Past Events

Home


### Event

During January to March 2024, NEEV IITGN, in partnership with Desai Foundation Trust, successfully organized nine projects and activities, including skill development courses, workshops, and awareness sessions for 286 youth and women hailing from rural areas in Gandhinagar. A total of 37 participants completed the Stitching Skill Training Course, 24 participants completed the Beautician Skill Training Course, 15 participants completed the Tally Skill Training Course, 12 participants completed the Spoken English Training Course, 20 participants completed the Computer Hardware Training Course, and 25 participants completed the Training Of Trainers in Sewing, all of whom received certificates for their achievements. These courses were offered entirely free of cost to the participants. Additionally, NEEV conducted awareness sessions in Banking and Financial Literacy for 56 participants, Menstrual Health Management for 55 participants, and Career Development Skills for 43 participants.

- To know the contact details



# Neev


[Events](#)

[Logout](#)

[Contact](#)

[My Profile](#)

Phone no-99898989

Email-neev@iitgn.ac.in

Instagram-neev@iitgn

## About NEEV

NEEV is an IIT Gandhinagar community outreach program. It is actively working to bridge the knowledge and communication gap by offering comprehensive training and mentoring, and growing the employability and entrepreneurship mind set of the surrounding communities.

NEEV's mission is to empower grassroots communities through workforce development, with a focus on entrepreneurship and skills training. NEEV's focus beneficiaries include youth and women from rural and urban localities from the areas surrounding IIT Gandhinagar.

Similarly, the page for Neev members is present. But the neev members can add Volunteer, Students, Courses, Instructor. Also, neev members cannot add Donor and Any other neev member.

## Responsibility of G2:

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

- Query for Login Authentication, for both admin side and member side.

```
71
72 @app.route('/login_auth',methods=['POST'])
73 def login_auth():
74     if request.method == 'POST':
75         con=mysql_app.connect()
76         cur=con.cursor()
77         cur.execute("select * from admin where Email=%s and Password=%s",(request.form['email'],request.form['password']))
78         data_admin=cur.fetchall()
79
80         cur.execute("select * from neev_members where Email=%s and Password=%s",(request.form['email'],request.form['password']))
81         data_member=cur.fetchall()
82
83         cur.close()
84         con.close()
85
86         if len(data_member) == 0 and len(data_admin) == 0:
87             return render_template('login.html',error=1)
88         elif len(data_admin) == 1:
89             session['member_type']='admin'
90         else:
91             session['member_type']='member'
92
93         global session_Set
94         session_Set=1
95         session['set']=1
96         session['email']=request.form['email']
97         session['password']=request.form['password']
98         return redirect(url_for('dashboard',session=session)) #1=> member and 2=>admin
99     else:
100         return render_template('home.html')
```

- Query to display profile

```
9
10 @app.route('/profile')
11 def display_profile():
12     if isSessionSet():
13         con=mysql_app.connect()
14         cur=con.cursor()
15         if session['member_type'] == 'admin':
16             cur.execute("select * from admin where Email=%s and Password=%s",(session['email'],session['password']))
17         else:
18             cur.execute("select * from neev_members where Email=%s and Password=%s",(session['email'],session['password']))
19         data=cur.fetchone()
20         cur.close()
21         con.close()
22     else :
23         return redirect(url_for('home'))
24
25     return render_template('profile.html',data=data,type=session['member_type'])
```

- Query to activate course:

```

755 def activate_course(id):
756     def activate_course(id):
757         if isSessionSet():
758             con=mysql_app.connect()
759             cur=con.cursor()
760             cur.execute("select * from course where C_ID=%s",id)
761             data=cur.fetchall()
762
763             cur.execute("select * from instructors")
764             instructors=cur.fetchall()
765
766             cur.close()
767             con.close()
768             current_date = date.today().isoformat()
769
770             return render_template('activate_course.html',data=data[0],ins=instructors,date=current_date)
771         else:
772             return redirect(url_for('home'))
773
774 @app.route('/activate_course/done_activate',methods=['POST'])
775 def done_activate():
776     if request.method == 'POST':
777         con=mysql_app.connect()
778         cur=con.cursor()
779
780         cur.execute("INSERT INTO active_courses (I_ID, C_ID, start_date, end_date) VALUES (%s,%s,%s,%s)",(request.form['instructor'],request.form['courseId'],request.form['start_date'],request.form['end_date']))
781         con.commit()
782
783         cur.close()
784         con.close()
785         return redirect(url_for('courses'))
786
787     con.close()
788     return redirect(url_for('courses'))
789
790 else:
791     return redirect(url_for('home'))
792
793 @app.route('/update/<id>')
794 def update_link(id):
795     if isSessionSet():
796         con=mysql_app.connect()
797         cur=con.cursor()
798         cur.execute("select * from course where C_ID=%s",id)
799         course_data=cur.fetchall()
800         cur.close()
801         con.close()
802         return render_template('update_course.html',data=course_data)
803     else:
804         return redirect(url_for('home'))
805
806 @app.route('/update_course',methods=['POST'])

```

- Query to update the link:

```

788 @app.route('/update/<id>')
789 def update_link(id):
790     if isSessionSet():
791         con=mysql_app.connect()
792         cur=con.cursor()
793         cur.execute("select * from course where C_ID=%s",id)
794         course_data=cur.fetchall()
795         cur.close()
796         con.close()
797         return render_template('update_course.html',data=course_data)
798     else:
799         return redirect(url_for('home'))
800

```

- Query to add course details:

```

818 @app.route('/course_details/<id>')
819 def course_details(id):
820     if isSessionSet():
821         con=mysql_app.connect()
822         cur=con.cursor()
823         cur.execute("select * from course where C_ID=%s",(id))
824         course_data=cur.fetchall()
825
826         cur.execute("select name from active_courses natural join instructors where C_ID=%s",(id))
827         inst=cur.fetchall()
828         inst=tuple(item for subtuple in inst for item in subtuple)
829
830         cur.execute("select * from active_courses where C_ID=%s",(id))
831         active_part=cur.fetchall()
832
833         stu=[]
834         stu_len=[]
835         vol=[]
836         vol_len=[]
837         all_stu=[]
838         all_vol=[]
839         for i in active_part:
840             cur.execute("select s_id,name,email from students where s_id in (select s_id from active_courses NATURAL join student_course where active_id=%s)",(i[4]))
841             rel=cur.fetchall()
842             stu.append(rel)
843             stu_len.append(len(rel))
844
845             cur.execute("select v_id,name from volunteer where v_id in ( select v_id from active_courses NATURAL join volunteer_course where active_id=%s)",(i[4]))
846             rel_vol=cur.fetchall()
847             vol.append(rel_vol)
848
849             vol.append(rel_vol)
850             vol_len.append(len(rel_vol))
851
852             cur.execute("select S_ID,name from students where S_ID NOT IN (select S_ID from active_courses NATURAL join student_course where active_id=%s)",(i[4]))
853             all_stu.append(cur.fetchall())
854
855             cur.execute("select v_id,name,email from volunteer where v_id not in ( select v_id from active_courses NATURAL join volunteer_course where active_id=%s)",(i[4]))
856             all_vol.append(cur.fetchall())
857
858         cur.close()
859         con.close()
860
861         return render_template('course_details.html',course=course_data, instructors=inst, stud_not_in_course=all_stu,
862                                vol_not_in_course=all_vol, active_course=active_part,students=stu,stu_count=stu_len,volunteers=vol,vol_count=vol_len)
863     else:
864         return redirect(url_for('home'))

```

- Query to add volunteer:

```

156 @app.route('/update_volunteer',methods=['POST','GET'])
157 def update_volunteer():
158     if request.method == 'POST':
159         con=mysql_app.connect()
160         cur=con.cursor()
161
162         sql="UPDATE volunteer SET Name=%s, Email=%s,Identification_ID=%s,Gender=%s,Address=%s,Photo=%s,Phone_No=%s WHERE v_id=%s"
163
164
165         if request.files['photo'].filename == '':
166             sql_2="UPDATE volunteer SET Name=%s, Email=%s,Identification_ID=%s,Gender=%s, Address=%s,Phone_No=%s WHERE M_id=%s"
167
168             cur.execute(sql_2,(request.form['name'],request.form['email'],
169                                request.form['adhar_id'],request.form['gender'],request.form['designation'],
170                                request.form['phone'],request.form['v_id']))
171             con.commit()
172             app.logger.info("nothing")
173         else:
174             file = request.files['photo']
175             filename= file.filename.rsplit('.', 1)[0].lower() + str(int(random.random() * 4500)) + '.' + \
176             file.filename.rsplit('.', 1)[1].lower()
177             filepath=os.path.join(app.config['UPLOAD_FOLDER'],filename)
178
179             if file and allowed_file(file.filename):
180                 file.save(filepath)
181             else:

```



```

180         file.save(filepath)
181     else:
182         return render_template('update_member.html', error=2, path=app.config['UPLOAD_FOLDER'])
183
184     cur.execute(sql, (request.form['name'], request.form['email'], request.form['adhar_id'], request.form['gender'], request.form['designation'], filename, request.form['password']))
185     con.commit()
186     app.logger.info("else")
187
188     cur.close()
189     con.close()
190     return redirect(url_for('volunteers'))
191 else:
192     return redirect(url_for('home'))
193

```

- Query to remove volunteer Delete volunteer

```

195  def delete_volunteer(s_id):
196      if isSessionSet():
197          sql="DELETE FROM volunteer WHERE v_id=%s"
198
199          con=mysql_app.connect()
200          cur=con.cursor()
201          cur.execute(sql,s_id)
202          con.commit()
203
204          cur.close()
205          con.close()
206
207          return redirect(url_for('volunteers'))
208      else:
209          return redirect(url_for('home'))
210
211  @app.route('/add_volunteer_link')

```

**For all students, members, volunteer, Instructor. All the queries and same to add and delete.**

## 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.

NEEV									
								Add Student	Back
Name	Email	Phone	Identification_id	Photo	Gender	Family background	Address		
Michael Lee	michael@example.com		4567890123	<a href="#">Link</a>	Male	Lower class	789 Oak St, Village	<a href="#">Update</a>	<a href="#">Delete</a>
David Wilson	david@example.com		1357902468	<a href="#">Link</a>	Male	Lower middle class	654 Cedar St, Town	<a href="#">Update</a>	<a href="#">Delete</a>
Emily Davis	emily@example.com		2468013579	<a href="#">Link</a>	Female	Upper class	987 Maple St, City	<a href="#">Update</a>	<a href="#">Delete</a>
Olivia Taylor	olivia@example.com		7024579135	<a href="#">Link</a>	Female	Middle class	753 Elm St, Town	<a href="#">Update</a>	<a href="#">Delete</a>
James Brown	james@example.com	2465778065	147852369	<a href="#">Link</a>	Male	Lower Middle class	159 Cedar St, City	<a href="#">Update</a>	<a href="#">Delete</a>
Sophia Garcia	sophia@example.com	9327035695	3691478520	<a href="#">Link</a>	Female	Upper middle class	852 Pine St, Town	<a href="#">Update</a>	<a href="#">Delete</a>
himani_1874	himanitrivedi1874@gmail.com	9327035695	2346	<a href="#">Link</a>	Male	Upper class	sfd gsg	<a href="#">Update</a>	<a href="#">Delete</a>

The web app should support 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:

### 1. INSERT

#### ○ Before Adding:

Add Instructor

Name:

XYZ

Email:

himanitrivedi1874@gmail.com

Phone no :

9327035695

Identification\_id

24325436

Gender:

Female

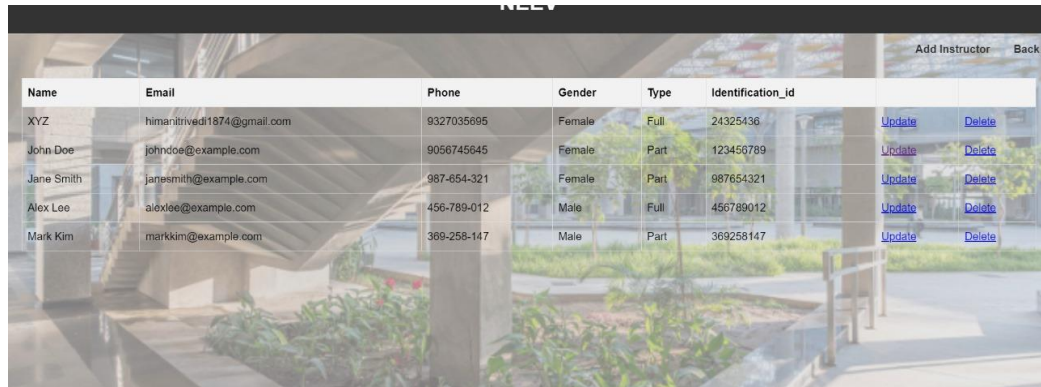
Type:

Full

Submit

- **After Adding:**

We are adding the instructor:

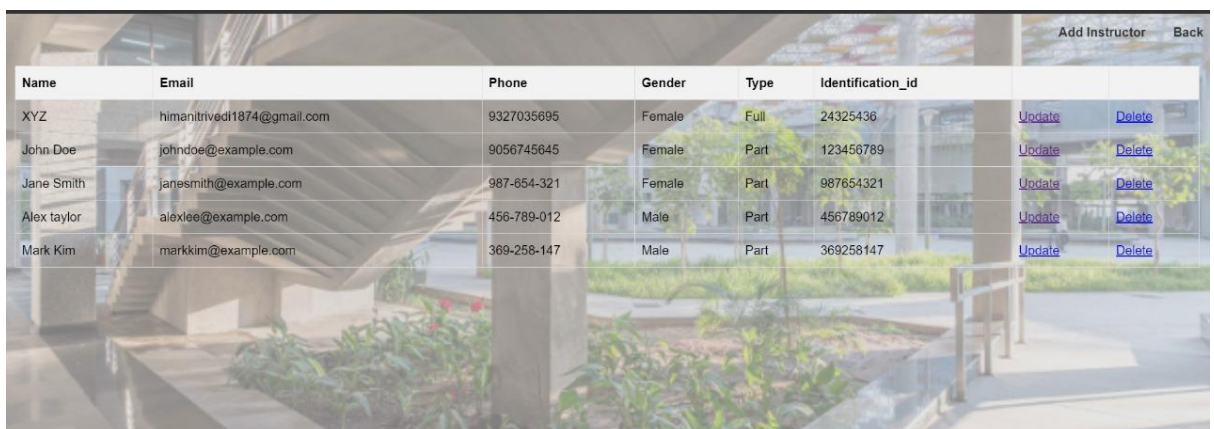


Name	Email	Phone	Gender	Type	Identification_id		
XYZ	himanitivedi1874@gmail.com	9327035695	Female	Full	24325436	<a href="#">Update</a>	<a href="#">Delete</a>
John Doe	johndoe@example.com	9056745645	Female	Part	123456789	<a href="#">Update</a>	<a href="#">Delete</a>
Jane Smith	janesmith@example.com	987-654-321	Female	Part	987654321	<a href="#">Update</a>	<a href="#">Delete</a>
Alex Lee	alexlee@example.com	456-789-012	Male	Full	456789012	<a href="#">Update</a>	<a href="#">Delete</a>
Mark Kim	markkim@example.com	369-258-147	Male	Part	369258147	<a href="#">Update</a>	<a href="#">Delete</a>

## 2. Update:

- **Before:**

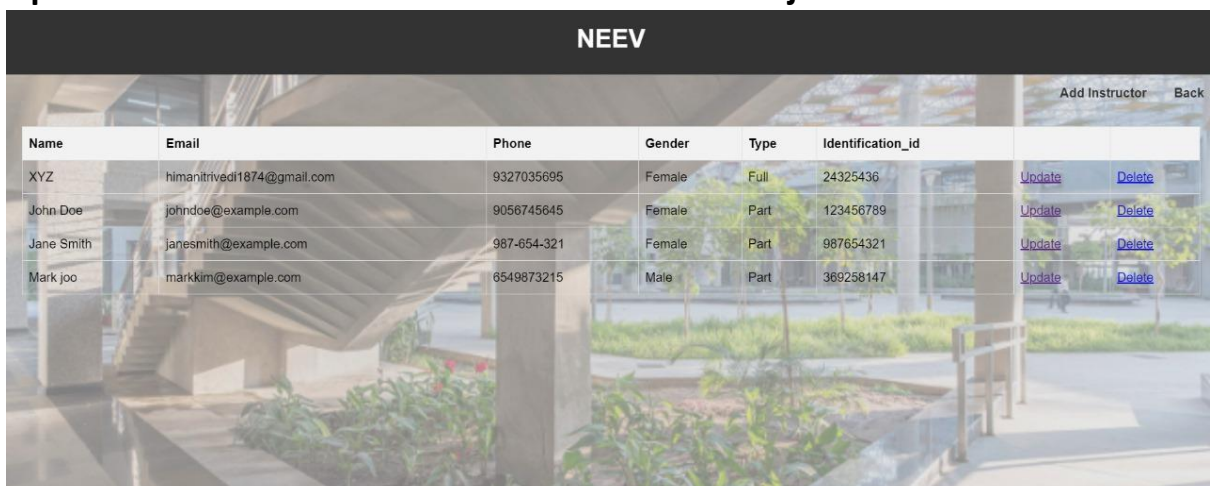
Here the surname of Mark is Kim



Name	Email	Phone	Gender	Type	Identification_id		
XYZ	himanitivedi1874@gmail.com	9327035695	Female	Full	24325436	<a href="#">Update</a>	<a href="#">Delete</a>
John Doe	johndoe@example.com	9056745645	Female	Part	123456789	<a href="#">Update</a>	<a href="#">Delete</a>
Jane Smith	janesmith@example.com	987-654-321	Female	Part	987654321	<a href="#">Update</a>	<a href="#">Delete</a>
Alex Taylor	alexlee@example.com	456-789-012	Male	Part	456789012	<a href="#">Update</a>	<a href="#">Delete</a>
Mark Kim	markkim@example.com	369-258-147	Male	Part	369258147	<a href="#">Update</a>	<a href="#">Delete</a>

- **After:**

Updated the surname and mobile.no. of the Mark joo



Name	Email	Phone	Gender	Type	Identification_id		
XYZ	himanitivedi1874@gmail.com	9327035695	Female	Full	24325436	<a href="#">Update</a>	<a href="#">Delete</a>
John Doe	johndoe@example.com	9056745645	Female	Part	123456789	<a href="#">Update</a>	<a href="#">Delete</a>
Jane Smith	janesmith@example.com	987-654-321	Female	Part	987654321	<a href="#">Update</a>	<a href="#">Delete</a>
Mark joo	markkim@example.com	6549873215	Male	Part	369258147	<a href="#">Update</a>	<a href="#">Delete</a>

- It also shows the image not available if the image is not uploaded



3.Delete:

- Before

The screenshot shows the 'Before' state of the application. It features a table with 6 instructors. The table has columns for Name, Email, Phone, Gender, Type, and Identification\_id. Each row has 'Update' and 'Delete' links. The background is a blurred image of a modern building interior.

Name	Email	Phone	Gender	Type	Identification_id		
XYZ	himantrivedi1874@gmail.com	9327035695	Female	Full	24325436	<a href="#">Update</a>	<a href="#">Delete</a>
John Doe	john.doe@example.com	9056745645	Female	Part	123456789	<a href="#">Update</a>	<a href="#">Delete</a>
Jane Smith	janesmith@example.com	987-654-321	Female	Part	987654321	<a href="#">Update</a>	<a href="#">Delete</a>
Alex Taylor	alexlee@example.com	456-789-012	Male	Part	456789012	<a href="#">Update</a>	<a href="#">Delete</a>
Mark Kim	markkim@example.com	369-258-147	Male	Part	369258147	<a href="#">Update</a>	<a href="#">Delete</a>

- After:  
Deleted the Instructor named Alex Taylor

The screenshot shows the 'After' state of the application. The instructor named Alex Taylor has been removed from the table. The table now contains 5 instructors. The background is the same blurred image of a modern building interior.

Name	Email	Phone	Gender	Type	Identification_id		
XYZ	himantrivedi1874@gmail.com	9327035695	Female	Full	24325436	<a href="#">Update</a>	<a href="#">Delete</a>
John Doe	john.doe@example.com	9056745645	Female	Part	123456789	<a href="#">Update</a>	<a href="#">Delete</a>
Jane Smith	janesmith@example.com	987-654-321	Female	Part	987654321	<a href="#">Update</a>	<a href="#">Delete</a>
Mark joo	markkim@example.com	6549873215	Male	Part	369258147	<a href="#">Update</a>	<a href="#">Delete</a>



4. Rename:

Our system doesn't require any rename operation to do so we haven't implemented/shown it and there is no point to rename table, database or column.

5. WHERE clause:

The Course list:

NEEV

Add StudentBack

Name	Email	Phone	Identification_id	Photo	Gender	Family background	Address		
Michael Lee	michael@example.com		4567890123	<a href="#">Link</a>	Male	Lower class	789 Oak St, Village	<a href="#">Update</a>	<a href="#">Delete</a>
David Wilson	david@example.com		1357902468	<a href="#">Link</a>	Male	Lower middle class	654 Cedar St, Town	<a href="#">Update</a>	<a href="#">Delete</a>
Emily Davis	emily@example.com		2468013579	<a href="#">Link</a>	Female	Upper class	987 Maple St, City	<a href="#">Update</a>	<a href="#">Delete</a>
Olivia Taylor	olivia@example.com		7024579135	<a href="#">Link</a>	Female	Middle class	753 Elm St, Town	<a href="#">Update</a>	<a href="#">Delete</a>
James Brown	james@example.com	2465778065	147852369	<a href="#">Link</a>	Male	Lower Middle class	159 Cedar St, City	<a href="#">Update</a>	<a href="#">Delete</a>
Sophia Garcia	sophia@example.com	9327035695	3691478520	<a href="#">Link</a>	Female	Upper middle class	852 Pine St, Town	<a href="#">Update</a>	<a href="#">Delete</a>
himani_1874	himanitrivedi1874@gmail.com	9327035695	2346	<a href="#">Link</a>	Male	Upper class	sfd gsg	<a href="#">Update</a>	<a href="#">Delete</a>

The Course Detail:

Mathematics

1

From: 2024-03-31

To: 2024-04-30

Instructor: John Doe

Number of Students: 1

Student Id	Name	
3	Michael Lee	

Number of Volunteers: 1

Volunteer Id	Name	
1	Steve Smith	

Students List:

David Wilson, Add

© 2024 NEEV



Contribution:

G1 Priya Darji, Om Ambekar, Diya Parashar

G2: Himani Trivedi, Dhru Jain

G1 & G2: Priya Darji, Om Ambekar, Diya Parashar Himani Trivedi, Dhru Jain

Documentation: Dhru Jain