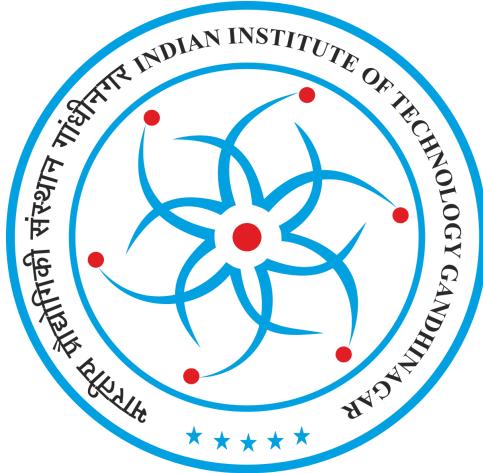


Indian Institute of Technology Gandhinagar



CS 432: Databases

Assignment 4: Deploying the DBMS

Topic: Club Management

Team: ManuDB

Amey Rangari | 21110177

Anshul Choudhary | 21110028

Manvendra Singh | 21110124

Pratham Sagar | 21110165

Shrijan Sahu | 21110203

Shubh Singhal | 21110206

Yashraj J Deshmukh | 21110245

Date: 14th April 2024

Responsibility of G1:

Feedback Collected -

We collected our feedback from Karan Khajanchi, Secretary of Sargam (Music Club of our college), who is an ideal user of our web app. His feedback included appreciation of features like issuing and returning of club equipment while checking the count of remaining quantity. The feedback included the addition of minor features like showing the events done by a club (at least to the secretary), the capability to add multiple participants of a team together, and resolving bugs that cause errors randomly.

Updates to Website -

1) More Information for Club Secretary (from 2nd feedback)

The screenshot shows a web browser window with the URL 127.0.0.1:5000/fetch_club_member/Odyssey. The page title is "Clubs Management IIT Gandhinagar". On the right, there are links for "Events" and "Logout". The main content area is titled "Odyssey:" and contains a table of club members. The table has columns for Roll Number, Email, Contact Number, Name, and Position. The data is as follows:

Roll Number	Email	Contact Number	Name	Position
2020004	2020004@iitgn.ac.in	8912345670	Aisha	SECRETARY
2021011	2021011@iitgn.ac.in	9123456789	Ayaan	GENERAL MEMBER
2021027	2021027@iitgn.ac.in	8912345678	Arnav	GENERAL MEMBER
2021001	2021001@iitgn.ac.in	8901234567	Arjun	General Member

Below the table, there are two sections: "Select an option:" with a dropdown menu set to "All member" and a "Submit" button, and "Update:" with a dropdown menu set to "Add member", an "Enter Roll Number" input field containing "Enter roll no.", a "Submit" button, and an "Add Event" button. At the bottom of the page, it says "Developed by ManuDB IIT Gandhinagar, Palaj, Gujarat".

The above picture displays the view of the club's page for the secretary of that club. In addition to viewing the members of the club like any other person registered on the website, the secretary has the privilege to update members and their role in the club as well as to create new events.

On talking with Karan, a club secretary, we realised that it would be better if we could also show the secretary a gist of all the past events that he has created, primarily if they have been approved or not. The result of this implementation is shown below:

Clubs Management IIT Gandhinagar

Logout

Odyssey:

Select an option:
All member

[Submit](#)

Update:
Add member

Enter Roll Number
Enter roll no

[Submit](#)

[Add Event](#)

Roll Number	Email	Contact Number	Name	Position
2020004	2020004@iitgn.ac.in	8912345670	Aisha	SECRETARY
2021011	2021011@iitgn.ac.in	9123456789	Ayaan	GENERAL MEMBER
2021027	2021027@iitgn.ac.in	8912345678	Arnav	GENERAL MEMBER
2021001	2021001@iitgn.ac.in	8901234567	Arjun	Coordinator

Event name	Edition	Budget	Approval Status	Overseer	Event lead
Tech Symposium	2021	15000	accepted	39	2020041
Tech Symposium	2021	15000	accepted	39	2020018
Science Fair	2023	10000	accepted	54	2021011

Developed by ManuDB
IIT Gandhinagar, Palaj, Gujarat

2) Problem of Disappearing Clubs (from 1st feedback)

Council Page - Clubs Management IIT Gandhinagar

Events Logout

Clubs Management IIT Gandhinagar

Select an option:
All member

[Submit](#)

Roll Number	Email	Contact Number	Name	Position
2020032	2020032@iitgn.ac.in	8912345670	Nandini	Secretary
2021003	2021003@iitgn.ac.in	8976543210	Reyansh	Coordinator
2021030	2021030@iitgn.ac.in	8912345670	Aadhyा	Coordinator
2021032	2021032@iitgn.ac.in	8765432101	Anvi	General Member
2022003	2022003@iitgn.ac.in	8976543210	Aisha	General Member
2022004	2022004@iitgn.ac.in	8912345670	Vikram	General Member
2022005	2022005@iitgn.ac.in	8954321098	Neha	General Member
2022006	2022006@iitgn.ac.in	8901234567	Arjun	General Member

Clubs in Technical Council:

Club Name	Description	Total Members	Actions
-----------	-------------	---------------	---------

Developed by ManuDB
IIT Gandhinagar, Palaj, Gujarat

The above picture displays an issue, where if we run a query on the council page, the clubs under the council disappear. This was an oversight on our part and we fixed it in the below version.

Select an option:

All member

[Submit](#)

Roll Number	Email	Contact Number	Name	Position
2020032	2020032@iitgn.ac.in	8912345670	Nandini	Secretary
2021003	2021003@iitgn.ac.in	8976543210	Reyansh	Coordinator
2021030	2021030@iitgn.ac.in	8912345670	Aadhyा	Coordinator
2021032	2021032@iitgn.ac.in	8765432101	Anvi	General Member
2022003	2022003@iitgn.ac.in	8976543210	Aisha	General Member
2022004	2022004@iitgn.ac.in	8912345670	Vikram	General Member
2022005	2022005@iitgn.ac.in	8954321098	Neha	General Member
2022006	2022006@iitgn.ac.in	8901234567	Arjun	General Member

Clubs in Technical Council:

Club Name	Description	Total Members	Actions
Digis	Dice is the digital sports development club of IIT Gandhinagar. It acts as a platform for creative thinkers to turn their imaginations into reality. The club intends to take human-technology interaction to a whole new level.	4	Know More about Digis
Council	Council	2	Know More about Council

3) Capability to Add Multiple Participants (from 1st feedback)

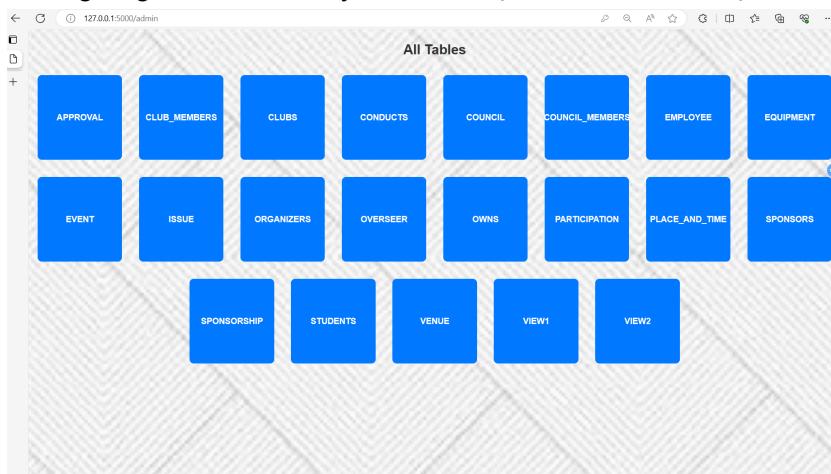
The image displays two side-by-side screenshots of a web application interface for event participation.

Left Screenshot: A simple form titled "Event Participation for ('Photography Contest', '202311')". It contains three text input fields: "Team Name:", "Roll Number:", and "Captain:". Below these is a checkbox labeled "Captain:" and a "Submit" button. At the bottom, a yellow bar displays the message "Participation added".

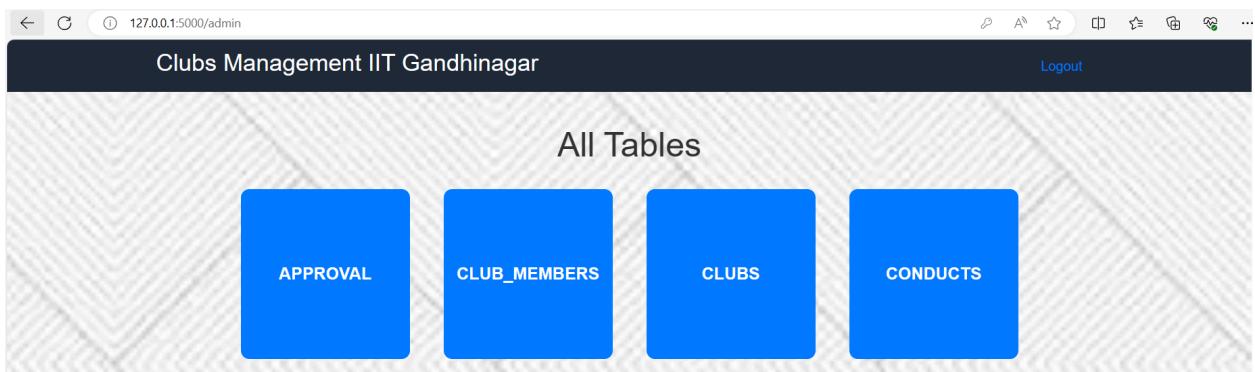
Right Screenshot: An enhanced version of the same form. It includes additional input fields for "Captain's Roll number:" and "Roll Number 1:". Below these are "Add Participant" and "Submit" buttons. The background features a light blue header with the event details.

Above is the initial website for the participation form for any given event. However, we found that an entire team should be able to register together, for which we made a newer version of the web page with small changes in the styling and added an 'add participant' button to increase the number of entries we make at once, as visible.

4) Adding Logout functionality to Admin (from 1st feedback)



This was again an oversight by us but a somewhat problematic one. With no logout for admin, the session information for admin was not deleted even when we tried to shift to student view; therefore risk of attacks. This was solved by adding a logout button to pop the admin session info and move back to the login page.



5) Registration Page (from 2nd feedback)

Since we were asked to work with authentications and registrations, we added the concept of passwords. We have already manually created roles and credentials for the dummy data personnel and stored them in the table *passwords*. We have now added a register option for our teammates and future visitors to our website to view it as students/employees by registering for this web app and setting their password.

Clubs Management IIT Gandhinagar

Student Login **Employee Login** **Admin Login**

Email:
Enter email

Password:
Enter password

Login

Email:
Enter email

Password:
Enter password

Login

Email:
Enter email

Password:
Enter password

Login

New User? Register Here:

Register

Developed by ManuDB
IIT Gandhinagar, Palaj, Gujarat

Clubs Management IIT Gandhinagar

Student Registration

UserID: Enter userID

Email: Enter email

Password: Enter password

Register

Employee Registration

UserID: Enter userID

Email: Enter email

Password: Enter password

Register

Developed by ManuDB
IIT Gandhinagar, Palaj, Gujarat

Privileges of Different Users -

(A) Student :

We have two different types of students-

i) General Students

These are all the students who are either not affiliated with any club or council or are just general members or coordinators there. They can only see the below-shown details of all members/secretaries/coordinators.

The screenshot shows the 'Club Members' section of the website. At the top left, there is a dropdown menu labeled 'Select an option:' with 'All member' selected. To the right of the dropdown is the title 'Club Members'. Below the title, there are five columns: 'Roll Number', 'Email', 'Contact Number', 'Name', and 'Position'. At the bottom left of the form area is a blue 'Submit' button. At the very bottom of the page, in a dark footer bar, it says 'Developed by ManuDB IIT Gandhinagar, Palaj, Gujarat'.

ii)Secretary

Secretaries have all the privileges of general members. Additionally, Secretaries can add/remove any member, add/remove any coordinator and add any upcoming event. The image shows the student view of the secretary of the Odyssey club.

This screenshot shows both the 'Club Members' and 'Events' sections. The 'Club Members' section is identical to the one above. The 'Events' section is located below it. It has a 'Select an option:' dropdown with 'All member' selected and a blue 'Submit' button. Below this is an 'Update:' section with a dropdown set to 'Add member' and a text input field labeled 'Enter Roll Number' containing 'Enter roll no'. There is also a blue 'Submit' button for this section. At the bottom of the events section is a blue 'Add Event' button. The events table has columns: 'Event name', 'Edition', 'Budget', 'Approval Status', 'Overseer', and 'Event lead'. The data in the table is as follows:

Event name	Edition	Budget	Approval Status	Overseer	Event lead
Tech Symposium	2021	15000	accepted	39	2020041
Tech Symposium	2021	15000	accepted	39	2020018
Science Fair	2023	10000	accepted	54	2021011

At the bottom of the page, in a dark footer bar, it says 'Developed by ManuDB IIT Gandhinagar, Palaj, Gujarat'.

(B) Employee :

We have three different types of Employees-

i) Student Affairs Staff

If the employee has a designation of Student Affairs Staff, he/she has the power to approve any event proposed by any club/council. He/She can update the status of any event to accept/reject/pending.

The screenshot shows the 'Clubs Management IIT Gandhinagar' dashboard. At the top right, there are links for 'Clubs and Councils' and 'Logout'. In the center, under 'Events for Approval:', there is a table with four columns: Event Name, Edition, Budget, and Approval Status. The table contains three rows: Art Exhibition (Edition 202408, Budget 0, Status pending), Comedy Night (Edition 202301, Budget 8000, Status pending), and Tech Symposium (Edition 2021, Budget 15000, Status accepted). Below the table is a form titled 'Update Status' with dropdown menus for 'Event Name' (Art Exhibition, 202408) and 'Status' (Accepted), and a 'Submit' button. At the bottom, it says 'Developed by ManuDB IIT Gandhinagar, Palaj, Gujarat'.

Event Name	Edition	Budget	Approval Status
Art Exhibition	202408	0	pending
Comedy Night	202301	8000	pending
Tech Symposium	2021	15000	accepted

ii) WARDEN

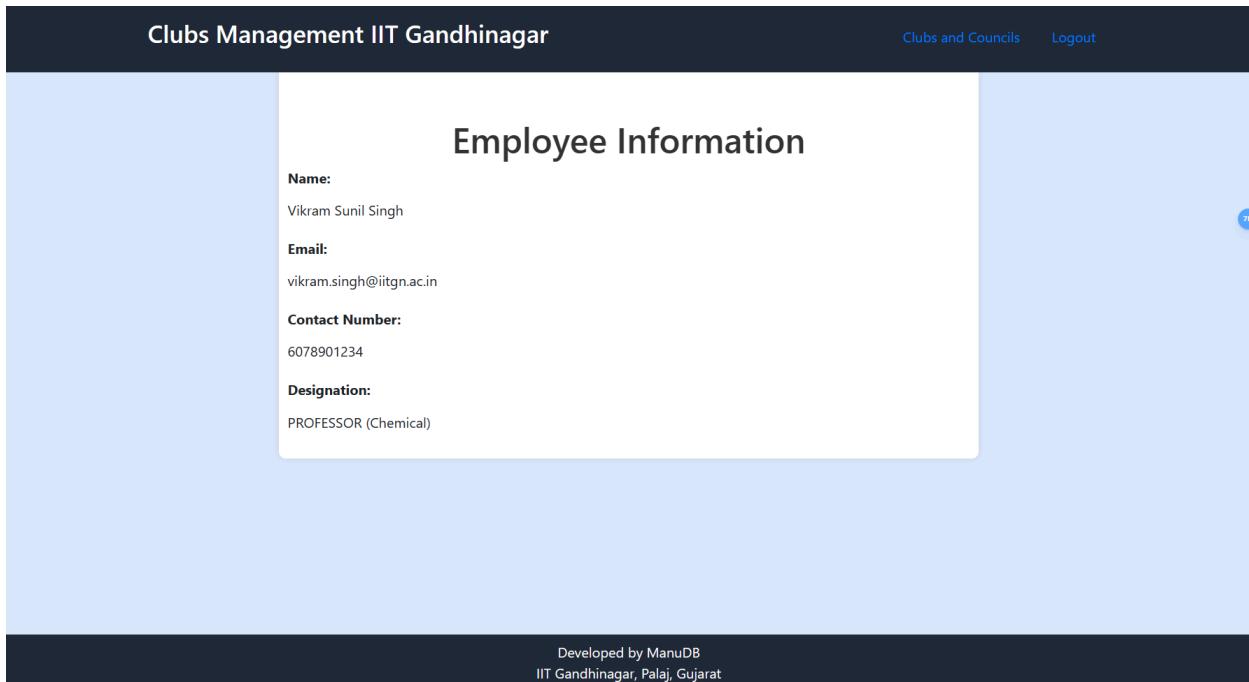
Warden has privileges to manage the venue of the events. The below screenshot is the employee view for WARDEN (Mechanical) of the shown events.

The screenshot shows the 'Clubs Management IIT Gandhinagar' dashboard. At the top right, there are links for 'Clubs and Councils' and 'Logout'. In the center, under 'Venues Managed:', there is a table with six columns: Name, EVENT_NAME, EDITION, DATE, START_TIME, and END_TIME. The table contains ten rows of venue information. At the bottom, it says 'Developed by ManuDB IIT Gandhinagar, Palaj, Gujarat'.

Name	EVENT_NAME	EDITION	DATE	START_TIME	END_TIME
Central Arcade	Artisan Market	202108	2021-08-14	9:30:00	18:30:00
Hockey Ground	Reverb	2020	2020-02-03	18:00:00	20:00:00
Hockey Ground	Fashion Fiesta	2023	2023-03-05	18:00:00	20:00:00
Hockey Ground	Tech Symposium	2021	2021-11-11	9:00:00	17:00:00
Hockey Ground	Photography Contest	202311	2023-11-10	5:00:00	22:00:00
Hockey Ground	Science Fair	2023	2025-01-15	11:00:00	17:00:00
Hostel Area	Rangoli	2023	2024-10-30	6:00:00	17:00:00
Hostel Area	Rangoli	2022	2022-10-28	6:00:00	20:00:00
Hostel Area	Photography Contest	202311	2024-11-10	5:00:00	22:00:00
Rangmanch	Dance Showdown	2021	2021-10-21	20:30:00	22:30:00

iii) PROFESSOR

The employees with the “PROFESSOR” designation have no special privileges. The image shows the employee view for a PROFESSOR (Chemical).



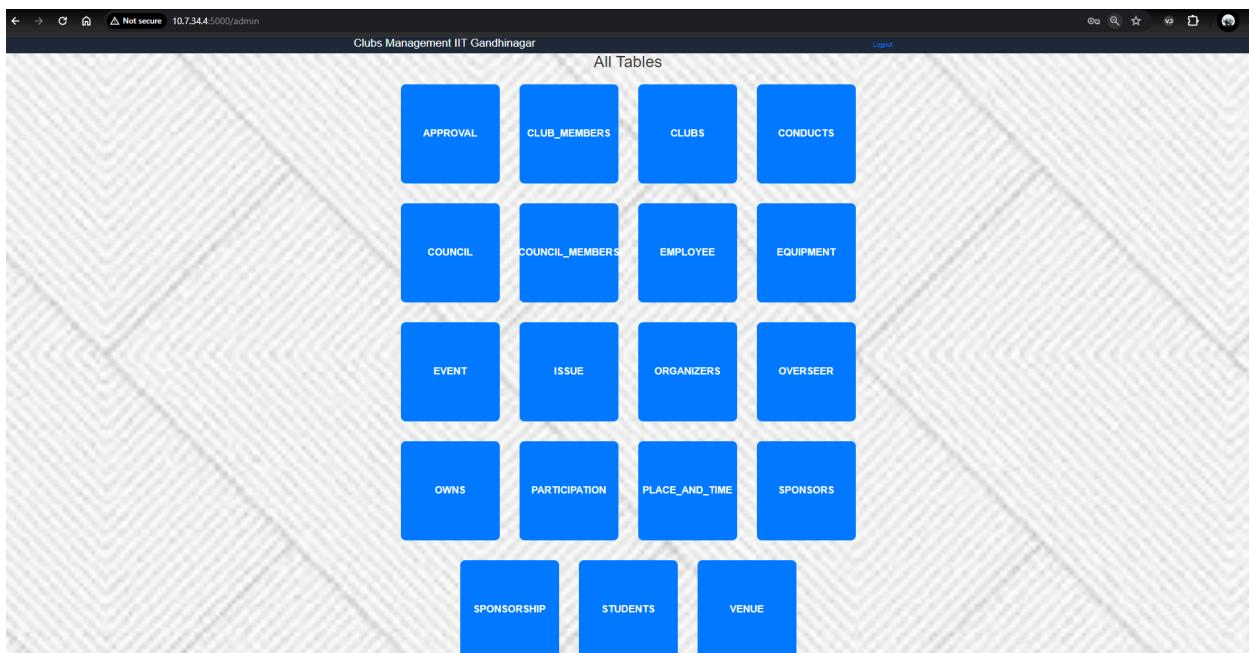
A screenshot of a web application titled "Clubs Management IIT Gandhinagar". The main content area is titled "Employee Information" and displays the following details for a user:

- Name:** Vikram Sunil Singh
- Email:** vikram.singh@iitgn.ac.in
- Contact Number:** 6078901234
- Designation:** PROFESSOR (Chemical)

The footer of the page indicates it was "Developed by ManuDB IIT Gandhinagar, Palaj, Gujarat".

(C) Admin :

Admin can add/remove any record in any of the tables shown in the image. He/She can rename any of these tables. The image shows the admin view.



A screenshot of a web application titled "Clubs Management IIT Gandhinagar" showing the "All Tables" section. The page lists 17 tables arranged in a grid:

All Tables			
APPROVAL	CLUB_MEMBERS	CLUBS	CONDUCTS
COUNCIL	COUNCIL_MEMBERS	EMPLOYEE	EQUIPMENT
EVENT	ISSUE	ORGANIZERS	OVERSEER
OWNS	PARTICIPATION	PLACE_AND_TIME	SPONSORS
SPONSORSHIP	STUDENTS	VENUE	

Viewing Table: clubs

Enter New Data

Column Name	Data Entry
CLUB_NAME	<input type="text"/>
DESCRIPTION	<input type="text"/>
COUNCIL_NAME	<input type="text"/>

Buttons: INSERT, DELETE, RENAME

Existing Data

CLUB_NAME	DESCRIPTION	COUNCIL_NAME
16Pixels	Photography club	Cultural Council
Abhinaya	Abhinaya, the drama club at IITGN, brings together all the acting and theatre enthusiasts to experience the theatre environment from script writing to acting. The drama club is actively engaged in making short films.	Cultural Council
Annuity	Financial Club	PDC
Anveshanam	Research society of college which promotes field of research	None
Art@IITGN	This is an initiative that promotes and facilitates artistic expressions and engagements on campus. Art as a medium and practice nurtures liberal, inclusive,	None

Admin gets the following view for each of the tables in our database. They can insert new entries, delete old ones, or rename the table in our database.

Responsibility of G2:

Concurrent multi-user access -

Accessing (GET) page with two different IPs.

```
Windows PowerShell
PS C:\Users\prath\Desktop\Database-project> python app.py
Connected to MySQL database successfully!
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://10.7.34.4:5000
Press CTRL+C to quit
* Restarting with watchdog (windowsapi)
Connected to MySQL database successfully!
* Debugger is active!
* Debugger PIN: 271-250-678
127.0.0.1 - - [13/Apr/2024 18:45:08] "GET / HTTP/1.1" 200 -
10.7.34.4 - - [13/Apr/2024 18:45:11] "GET / HTTP/1.1" 200 -
10.7.34.4 - - [13/Apr/2024 18:45:38] "GET / HTTP/1.1" 200 -
10.7.34.4 - - [13/Apr/2024 18:45:38] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [13/Apr/2024 18:45:41] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [13/Apr/2024 18:45:42] "GET /favicon.ico HTTP/1.1" 404 -
```

Now, we will login as follows: 127.0.0.1 - Student login , 10.7.34.4 - Admin login

```
Received email: admin@iitgn.ac.in
Received password: manu9b
10.7.34.4 - - [13/Apr/2024 20:48:53] "POST / HTTP/1.1" 302 -
Admin
['approval', 'club_members', 'clubs', 'conducts', 'council', 'council_members', 'employee', 'equipment', 'event', 'issue', 'organizers', 'overseer', 'owns', 'participation', 'place_and_time', 'sponsors', 'sponsorship', 'students', 'venue']
10.7.34.4 - - [13/Apr/2024 20:48:53] "GET /admin HTTP/1.1" 200 -
Received email: 20200904@iitgn.ac.in
Received password: 1234567890
User: {'ROLL_NO': '20200904', 'EMAIL': '20200904@iitgn.ac.in', 'CONTACT_NO': '89123456789', 'FIRST_NAME': 'Aisha', 'MIDDLE_NAME': 'Meera', 'LAST_NAME': 'Khan', 'STREET': 'Indiranagar 100 Feet Road', 'CITY': 'Bangalore', 'STATE': 'Karnataka', 'PINCODE': '456789', 'DATE_OF_BIRTH': datetime.date(1999, 1, 30), 'AGE': 25, 'PROGRAMME': 'MSc', 'DISCIPLINE': 'ChE', 'YEAR': 2020}
127.0.0.1 - - [13/Apr/2024 20:49:02] "POST / HTTP/1.1" 302 -
Student
127.0.0.1 - - [13/Apr/2024 20:49:02] "GET /studentInfo HTTP/1.1" 200 -
127.0.0.1 - - [13/Apr/2024 20:49:02] "GET /20231283171925_IMG_2141.JPG HTTP/1.1" 404 -
127.0.0.1 - - [13/Apr/2024 20:49:02] "GET /20231283175437_IMG_2169.JPG HTTP/1.1" 404 -
127.0.0.1 - - [13/Apr/2024 20:49:02] "GET /20231283171925_IMG_2141.JPG HTTP/1.1" 404 -
```

Student's view:

The screenshot shows a web browser window for the 'Clubs Management IIT Gandhinagar' system. The URL is 127.0.0.1:5000/studentinfo. The page title is 'Student Information'. The content displays personal details of a student named Aisha Meera Khan, including Name, Email, Roll Number, Contact Number, Address, Date of Birth, Age, and Programme. The details are as follows:

- Name:** Aisha Meera Khan
- Email:** 2020004@iitgn.ac.in
- Roll Number:** 2020004
- Contact Number:** 8912345670
- Address:** Indiranagar 100 Feet Road, Bangalore, Karnataka
- Date of Birth:** Sat, 30 Jan 1999 00:00:00 GMT
- Age:** 25
- Programme:** MSc ChE '2020

Admin's view:

The screenshot shows the 'All Tables' section of the 'Clubs Management IIT Gandhinagar' system. The URL is 10.73.4.5000/admin. The page title is 'All Tables'. It displays a grid of 16 blue buttons, each representing a database table: APPROVAL, CLUB_MEMBERS, CLUBS, CONDUCTS, COUNCIL, COUNCIL_MEMBERS, EMPLOYEE, EQUIPMENT, EVENT, ISSUE, ORGANIZERS, OVERSEER, OWNS, PARTICIPATION, PLACE_AND_TIME, SPONSORS, SPONSORSHIP, STUDENTS, and VENUE.

Viewing Table: clubs		
Existing Data		
CLUB_NAME	DESCRIPTION	COUNCIL_NAME
16Pots	Photography club	Cultural Council
Abhinaya	Abhinaya, the drama club at IITGN, brings together all the acting and theatre enthusiasts to experience the theatre environment from script writing to acting. The drama club is actively engaged in making short films.	Cultural Council
Annuity	Financial Club	PDC
Anveethanam	Research society of college which promotes field of research	None
Art@IITGN	This is an initiative that promotes and facilitates artistic expressions and engagements on campus. Art as a medium and practice nurtures liberal, inclusive, interdisciplinary, and futuristic values.	None
Chess Club	Promotes chess culture in college	Sports Council
Cinemateque	Film-screening club	Cultural Council
Coding Initiative	Coding initiative aims to inculcate among students the skills of problem solving through programming and coding.	None
Digs	Dice is the digital sports development club of IIT Gandhinagar. It acts as a platform for creative thinkers to turn their imaginations into reality. The club intends to take human-technology interaction to a whole new level.	Technical Council
Graap	Competitive coding club of the college	Technical Council
LDI	LDI is to promote leadership skills among the IITGN student community through experiential learning and engaging activities. It believes in going beyond monotonous talks and instead focus on active involvement.	None
LitFoc	Club to promote skills related to literary arts	Cultural Council
Mean Mechanics	Robotics club	Technical Council
Mells	Web and app development club	Technical Council
Odyssey	The Odyssey club of IITGN attempts to foster the knowledge and participation of students of the institute in the field of astronomy. The club provides students with technical learning and hands on experiences on astronomical equipments.	Technical Council
Palatte	The club aim is to promote all forms of art in college and to provide students with equal opportunity to enhance as well as showcase their talent in art by the means of the workshop and annual exhibitions.	Cultural Council
Quizzing Society	Club which promotes quizzing culture and skills among community	Cultural Council
Sargam	The music club at IITGN gives the community an opportunity to be a part of different bands within the institute and to perform at different cultural events. The club appreciates and promotes various genres of music.	Cultural Council
Step-Up	The club aim is to encourage students to take up different dance forms and provide a platform to perform these styles collectively. The Dance Club organizes expert and beginner level workshops for students, faculty and staff of the institute.	Cultural Council

Enter New Data		
Column Name	Data Entry	
CLUB_NAME	<input type="text"/>	
DESCRIPTION	<input type="text"/>	
COUNCIL_NAME	<input type="text"/>	
<input type="button" value="INSERT"/>	<input type="button" value="DELETE"/>	<input type="button" value="RENAME"/>

Here the Admin and student(General Secretary) tried adding same entry to events table simultaneously.

10.0.5.40 – Admin

127.0.0.1 – student

```
10.0.5.40 - - [13/Apr/2024 22:48:43] "POST /view_table/event HTTP/1.1" 200 -
127.0.0.1 - - [13/Apr/2024 22:48:43] "POST /addEvent HTTP/1.1" 200 -
```

Both of the requests got posted without any error and only one entry gets reflected due to foreign key constraint. But if there is a delay in posting , i.e. If admin tries adding the same entry later we get an error message suggesting duplicate entry.

```
10.0.5.40 - - [13/Apr/2024 22:49:32] "POST /view_table/event HTTP/1.1" 500 -
Traceback (most recent call last):
  File "C:\Users\singh\AppData\Local\Programs\Python\Python311\Lib\site-packages\mysql\connector\connection_cext.py", line 661, in cmd_query
    self._mysql.query(
_mysql.connector.MySQLInterfaceError: Duplicate entry 'hey-23' for key 'event.PRIMARY'
```

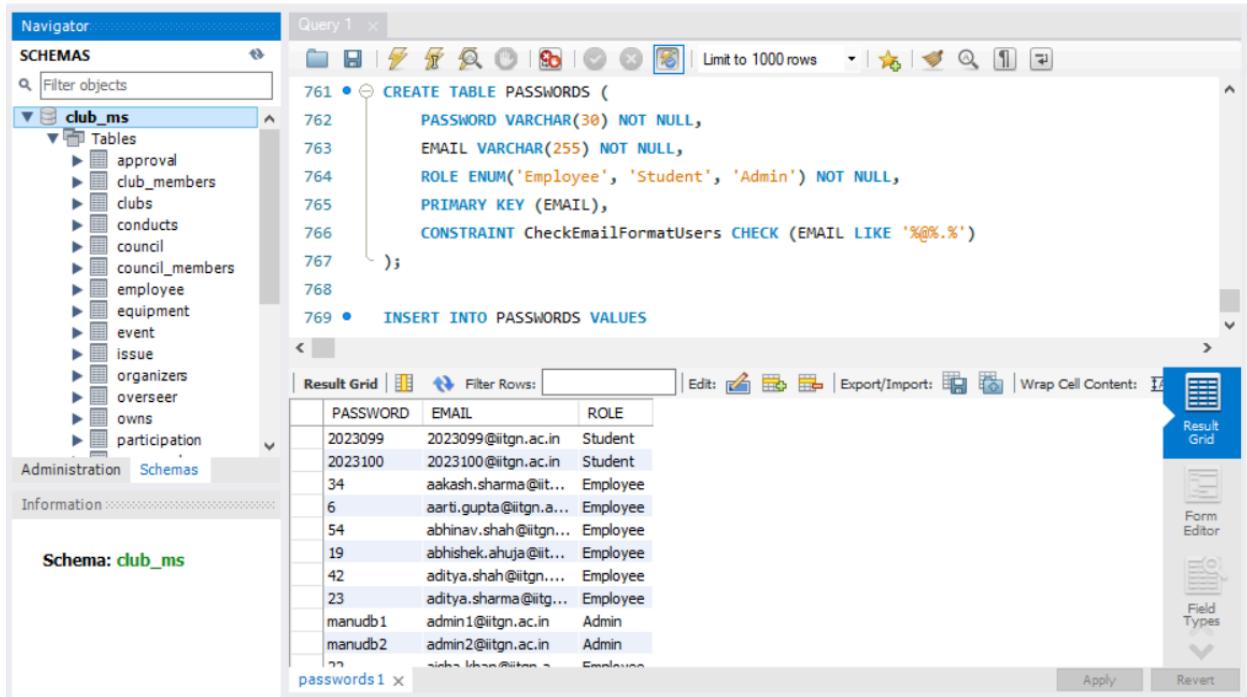
Changes in Database -

We discussed with a few stakeholders regarding the database schema of our website and we did not receive any request that required any concrete changes to it. However, we made the following changes to our data as was required:

1) A new table - Passwords

Earlier we had kept the login pretty simple. For students and employees, we just looked up our database and gave access. For admin, we created another email entry. Since all the email ids existed in our database, we did not create a register page for our website. But in conversation with TA Ritesh, he advised us to have a registration page as they would require to create their ids and log-in to check, plus we would also require actual emails for Google authentication for login when any of the team members or the TAs visit our websites. Hence, we created the data

for our table *Passwords*, ran the Python script that we used in assignment II to generate SQL commands for insertion, and finally created the table in our database. Alongside this, we created a webpage for new users to register themselves and added Google verification to our website.



The screenshot shows the MySQL Workbench interface. On the left, the Navigator pane displays the schema structure under 'club_ms'. The 'Tables' section lists various tables like approval, club_members, clubs, conducts, council, council_members, employee, equipment, event, issue, organizers, overseer, owns, and participation. The central area is a 'Query 1' window containing the following SQL code:

```

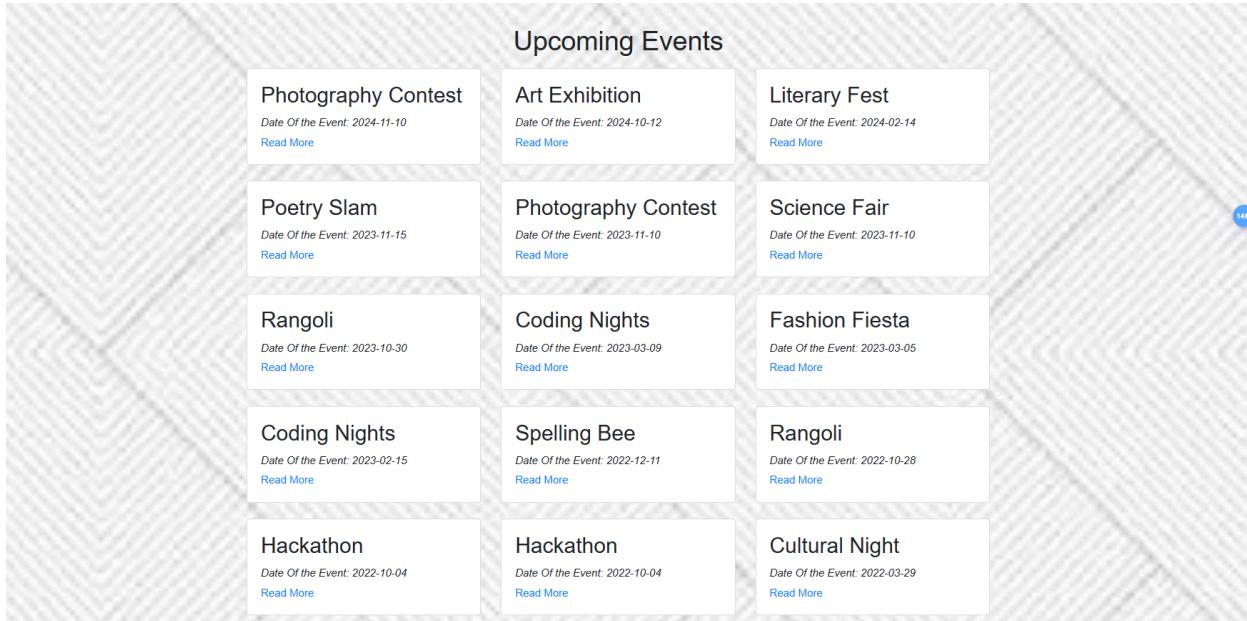
CREATE TABLE PASSWORDS (
    PASSWORD VARCHAR(30) NOT NULL,
    EMAIL VARCHAR(255) NOT NULL,
    ROLE ENUM('Employee', 'Student', 'Admin') NOT NULL,
    PRIMARY KEY (EMAIL),
    CONSTRAINT CheckEmailFormatUsers CHECK (EMAIL LIKE '%@%.%')
);

INSERT INTO PASSWORDS VALUES
    (2023099, '2023099@litgn.ac.in', 'Student'),
    (2023100, '2023100@litgn.ac.in', 'Student'),
    (34, 'akash.sharma@it...', 'Employee'),
    (6, 'aarti.gupta@itgn.a...', 'Employee'),
    (54, 'abhinav.shah@itgn...', 'Employee'),
    (19, 'abhishek.ahuja@it...', 'Employee'),
    (42, 'aditya.shah@itgn....', 'Employee'),
    (23, 'aditya.sharma@itg...', 'Employee'),
    (manudb1, 'admin1@itgn.ac.in', 'Admin'),
    (manudb2, 'admin2@itgn.ac.in', 'Admin'),
    (22, 'nisha.khan@itgn.a...', 'Employee')
;
  
```

The 'Result Grid' tab shows the inserted data:

PASSWORD	EMAIL	ROLE
2023099	2023099@litgn.ac.in	Student
2023100	2023100@litgn.ac.in	Student
34	akash.sharma@it...	Employee
6	aarti.gupta@itgn.a...	Employee
54	abhinav.shah@itgn...	Employee
19	abhishek.ahuja@it...	Employee
42	aditya.shah@itgn....	Employee
23	aditya.sharma@itg...	Employee
manudb1	admin1@itgn.ac.in	Admin
manudb2	admin2@itgn.ac.in	Admin
22	nisha.khan@itgn.a...	Employee

2) Date of Events updated



The screenshot shows a website page titled 'Upcoming Events' featuring a grid of twelve event cards:

- Photography Contest** - Date Of the Event: 2024-11-10 | [Read More](#)
- Art Exhibition** - Date Of the Event: 2024-10-12 | [Read More](#)
- Literary Fest** - Date Of the Event: 2024-02-14 | [Read More](#)
- Poetry Slam** - Date Of the Event: 2023-11-15 | [Read More](#)
- Photography Contest** - Date Of the Event: 2023-11-10 | [Read More](#)
- Science Fair** - Date Of the Event: 2023-11-10 | [Read More](#)
- Rangoli** - Date Of the Event: 2023-10-30 | [Read More](#)
- Coding Nights** - Date Of the Event: 2023-03-09 | [Read More](#)
- Fashion Fiesta** - Date Of the Event: 2023-03-05 | [Read More](#)
- Coding Nights** - Date Of the Event: 2023-02-15 | [Read More](#)
- Spelling Bee** - Date Of the Event: 2022-12-11 | [Read More](#)
- Rangoli** - Date Of the Event: 2022-10-28 | [Read More](#)
- Hackathon** - Date Of the Event: 2022-10-04 | [Read More](#)
- Hackathon** - Date Of the Event: 2022-10-04 | [Read More](#)
- Cultural Night** - Date Of the Event: 2022-03-29 | [Read More](#)

On the events page we were showing all the events from our database. But since it would make sense to only show events in which the students can still participate, we added a date filter to only show events after current date. But in the data we created in assignment II, only two events passed this filter.

The screenshot shows a web browser window titled "Event Page" with the URL "127.0.0.1:5000/events". The main content is titled "Upcoming Events" and lists two events:

- Photography Contest**
Date Of the Event: 2024-11-10
[Read More](#)
- Art Exhibition**
Date Of the Event: 2024-10-12
[Read More](#)

So, we are updating our database, changing the event dates, so that more events pass the filter and get displayed.

The screenshot shows a web browser window titled "Event Page" with the URL "127.0.0.1:5000/events". The main content is titled "Upcoming Events" and lists nine events:

- Science Fair**
Date Of the Event: 2025-01-15
[Read More](#)
- Photography Contest**
Date Of the Event: 2024-11-10
[Read More](#)
- Rangoli**
Date Of the Event: 2024-10-30
[Read More](#)
- Art Exhibition**
Date Of the Event: 2024-10-12
[Read More](#)
- Hackathon**
Date Of the Event: 2024-10-04
[Read More](#)
- Coding Nights**
Date Of the Event: 2024-08-15
[Read More](#)
- Art Battle**
Date Of the Event: 2024-07-18
[Read More](#)
- Coding Nights**
Date Of the Event: 2024-06-09
[Read More](#)
- Film Festival**
Date Of the Event: 2024-05-07
[Read More](#)
- Cooking Competition**
Date Of the Event: 2024-04-28
[Read More](#)
- Art Exhibition**
Date Of the Event: 2024-10-12
[Read More](#)

3) Availability of Equipments updated

Due to a minor error in the Python script to convert Excel data to SQL command conversion in assignment II, most of our equipments were listed as unavailable, due to which no user could issue them. We have fixed the script and updated our database to resolve this issue.

Viewing Table: equipment

Existing Data

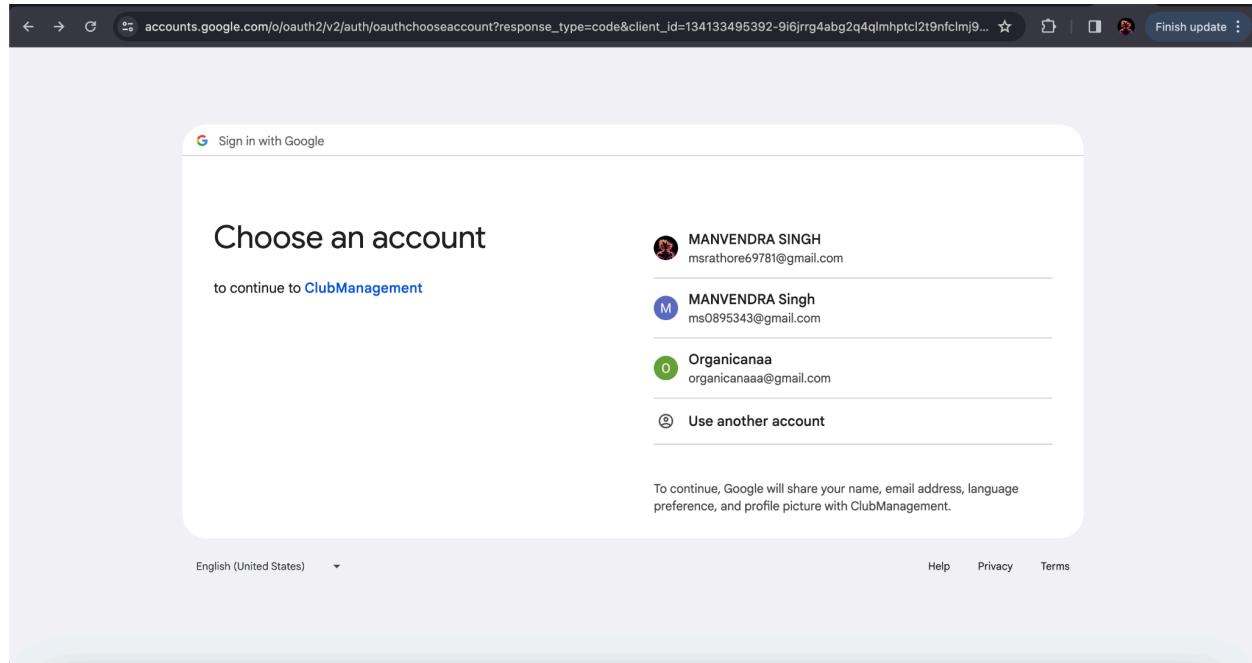
EQUIPMENT_ID	NAME	QUANTITY	STATE	ACQUISITION_DATE	AVAILABILITY	IMG
1	GUITAR	1	GOOD	2022-01-01	1	b'108fffe00104aa4f
12	Wig	6	BAD	2017-09-14	1	b'Abhinaya'
123	Nintendo Switch Lite	4	Good	2020-03-04	0	b'Digis'
129	Pablo Picasso Brushes	1	Good	2022-07-14	0	b'Palette'
321	Giorgione Paintbrushes	12	Good	2020-06-11	1	b'Palette'
1001	Hubble Space Telescope Replica	1	Good	2021-10-15	0	b'Odyssey'
1234	Zumba Fitness DVD Set	1	Good	2023-04-08	0	b'Step-Up'
2062	Star Charts Set	1	Good	2020-04-26	0	b'Odyssey'
2183	Spotlight	2	Good	2021-06-19	0	b'Abhinaya'
2345	Claude Monet Watercolors	1	Good	2023-11-18	0	b'Palette'
2543	Oculus VR headset	1	Good	2019-04-30	0	b'Digis'
3210	Fender Stratocaster Guitar	1	Good	2022-03-25	0	b'Sargam'
3452	Korg Synthesizer	1	Good	2019-08-06	0	b'Sargam'
3456	Costume Designer Software	1	Good	2020-07-20	0	b'Abhinaya'
3457	Salvador Dali Palette Knife	1	BAD	2019-03-05	0	b'Palette'
4321	Leonardo Da Vinci Paints	1	Good	2021-08-10	1	b'Digis'
4567	Newtonian Reflector Telescope	1	Good	2022-07-02	1	b'Odyssey'

Viewing Table: equipment

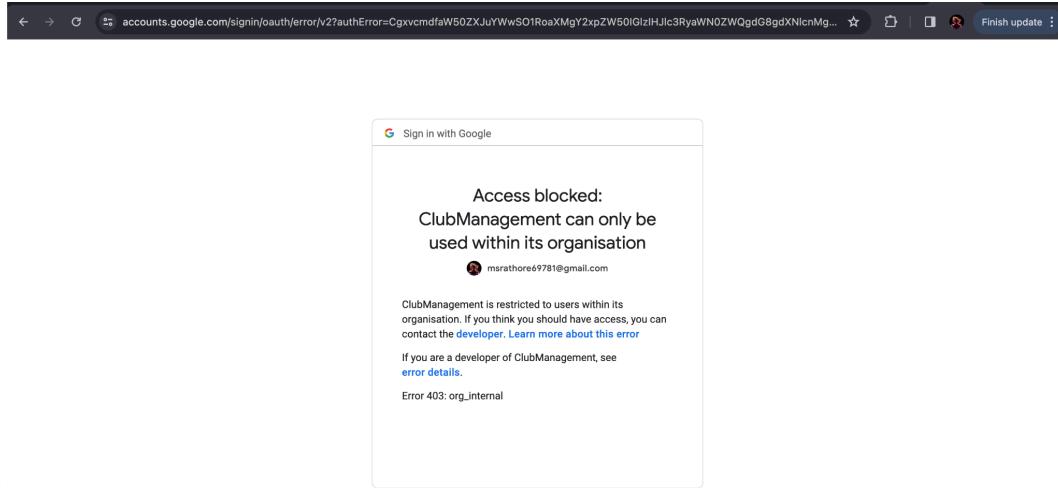
Existing Data

EQUIPMENT_ID	NAME	QUANTITY	STATE	ACQUISITION_DATE	AVAILABILITY	club_name
1	GUITAR	1	Good	2022-01-01	1	Palette
12	Wig	6	Bad	2017-09-14	1	Abhinaya
123	Nintendo Switch Lite	4	Good	2020-03-04	1	Sargam
129	Pablo Picasso Brushes	1	Good	2022-07-14	1	Palette
321	Giorgione Paintbrushes	12	Good	2020-06-11	1	Palette
1001	Hubble Space Telescope Replica	1	Good	2021-10-15	1	Odyssey
1234	Zumba Fitness DVD Set	1	Good	2023-04-08	1	Step-Up
2062	Star Charts Set	1	Good	2020-04-26	1	Odyssey
2183	Spotlight	2	Good	2021-06-19	1	Abhinaya
2345	Claude Monet Watercolors	1	Good	2023-11-18	1	Sargam
2543	Oculus VR headset	1	Good	2019-04-30	0	Digis
3210	Fender Stratocaster Guitar	1	Good	2022-03-25	1	Sargam
3452	Korg Synthesizer	1	Good	2019-08-06	1	Sargam
3456	Costume Designer Software	1	Good	2020-07-20	1	Abhinaya
3457	Salvador Dali Palette Knife	1	Bad	2019-03-05	1	Digis
3459	Leonardo Da Vinci Paints	1	Good	2022-08-22	1	Palette
4321	Newtonian Reflector Telescope	1	Good	2021-08-10	1	Odyssey
4567	Celestron Telescope	1	Good	2022-07-02	1	Odyssey

Google Authentication -



We tried to run the application on a browser where iit gn gmail id was not available. When we used the email id other than with domain (**iitgn.ac.in**) we get an error block message from google given below.



API APIs & Services

Edit app registration

To protect you and your users, Google only allows apps using OAuth to use Authorized Domains. The following information will be shown to your users on the consent screen.

OAuth consent screen

Authorized domains

When a domain is used on the consent screen or in an OAuth client's configuration, it must be pre-registered here. If your app needs to go through verification, please go to the [Google Search Console](#) to check if your domains are authorized. [Learn more](#) about the authorized domain limit.

Developer contact information

Email addresses * msrathore69781@gmail.com

These email addresses are for Google to notify you about any changes to your project.

SAVE AND CONTINUE **CANCEL**

Learn

How is this info presented to users?

This is the consent screen that users see

API APIs & Services

Client ID for Web application

Enabled APIs & services

Credentials

Name * Web client 2

The name of your OAuth 2.0 client. This name is only used to identify the client in the console and will not be shown to end users.

The domains of the URIs you add below will be automatically added to your OAuth consent screen as authorized domains.

Authorized JavaScript origins

For use with requests from a browser

Authorized redirect URIs

For use with requests from a web server

URIs 1 * http://127.0.0.1:5000/google/auth/

Additional information

Client ID 134133495392-9j6jrg4abg2q4lmhptcl2t9nfclmj9.apps.googleusercontent.com

Creation date April 14, 2024 at 10:50:43 PM GMT+5

Client secrets

If you are in the process of changing client secrets, you can manually rotate them without downtime. [Learn more](#)

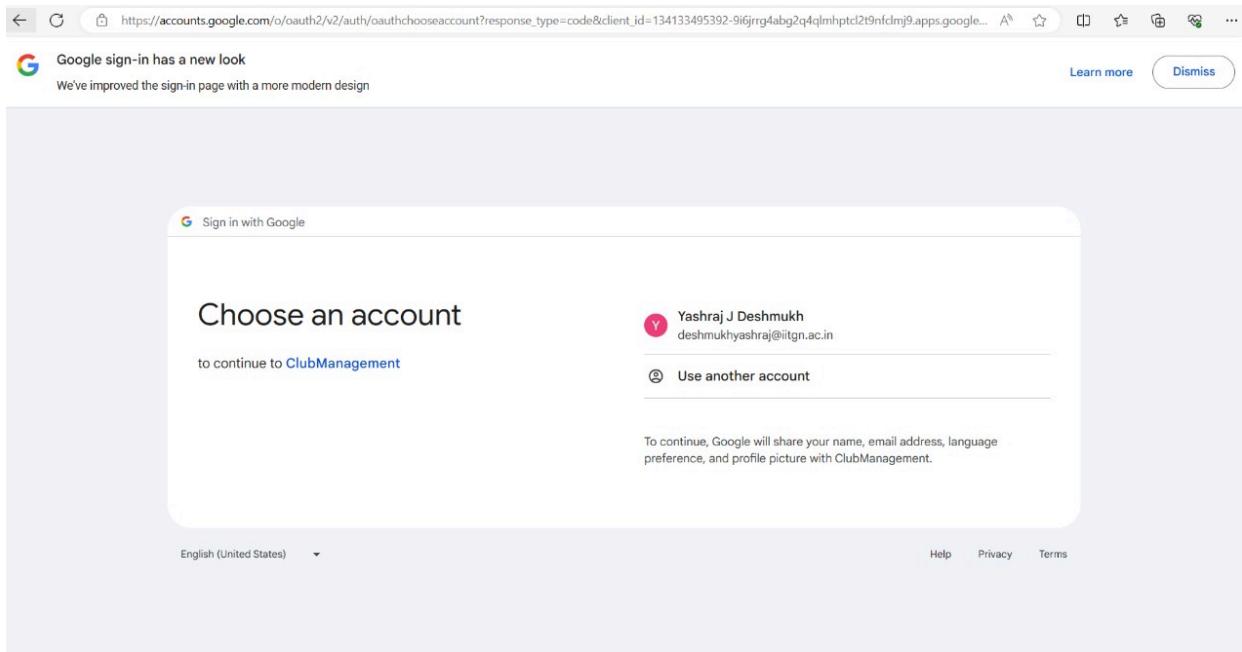
Client secret GOCPSPX-1bzBDtUqdXibzawiOCdExxaefXO

Creation date April 14, 2024 at 10:50:43 PM GMT+5

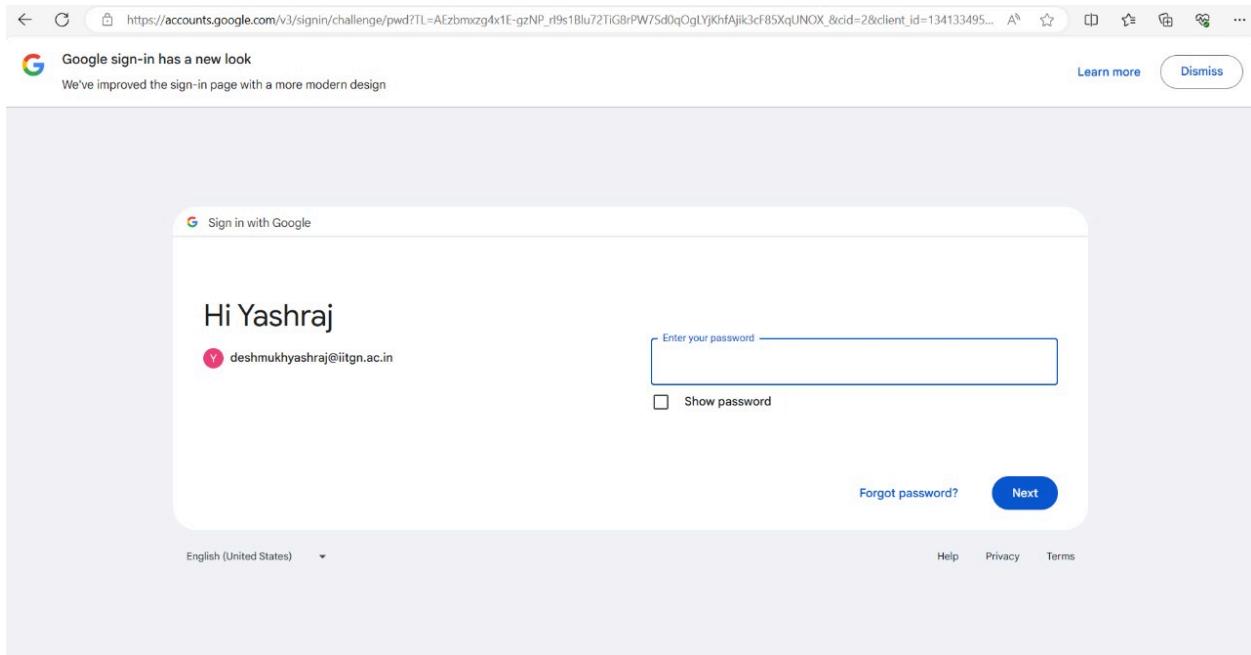
Status Enabled

+ ADD SECRET

The above screenshot shows what client id and client secret we have used in our application for the functionality of the google authentication. When the user clicks on the register button that is present in the login page it will direct the user to url_for(google) and then to the </google/auth/>



When the user uses an email id with domain(iitgn.ac.in) we will be directed to register page.



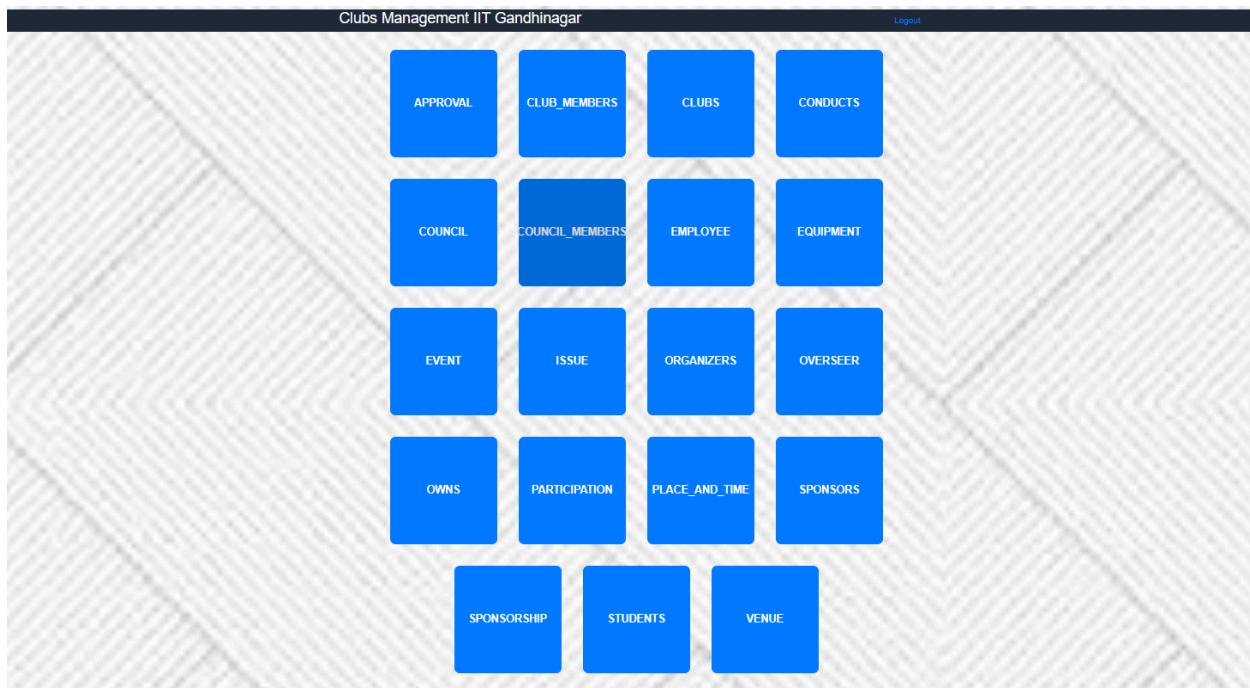
Responsibility of G1 & G2:

Attacks and Defence -

We have carried out the following - SQL Injection, URL Tampering, Session Hijacking - and stratified our website against them accordingly.

1) SQL Injection -

To execute an SQL injection, we exploit unfiltered input. By manipulating input fields, we aim to compel the backend server to run our injected SQL code, even if the application doesn't provide direct means to execute such commands. Success in SQL injection hinges on bypassing application defenses to trigger the execution of injected SQL code.



Following page shows all the tables that can be viewed by admin. Note that we don't intend to give the rights to the *passwords* table even to admin. For utmost security and privacy, we have kept viewing these credentials exclusive to MySQL Workbench.

But even though we have not provided a way on the webpage to access the *passwords*' table, a user logged-in as an admin can still view it through an sql injection. We have not written separate sql queries and created individual webpages for each table but have used functionality of flask to fetch variables from URL to populate the webpage with details related to the requested table. This is done by the following command:

```

@app.route('/view_table/<table_name>', methods=['GET', 'POST'])
def view_table(table_name):
    # Establish a connection to MySQL
    conn = mysql.connector.connect(**mysql_config)
    cursor = conn.cursor()

    # Fetch column names for the specified table
    cursor.execute(f"SHOW COLUMNS FROM {table_name}")
    columns = [column[0] for column in cursor.fetchall()]

    cursor.execute(f"select * from {table_name}")
    data = cursor.fetchall()

```

But the same query has a demerit as it can fetch any table that exists in the database even if the user has no authorisation to access it. This security flaw resulted in this:

The screenshot shows a web browser window with the URL `127.0.0.1:5000/view_table/passwords`. The page title is "Viewing Table: passwords". Below the title, there is a section titled "Enter New Data" with three input fields for "PASSWORD", "EMAIL", and "ROLE". At the bottom of this section are three buttons: "INSERT", "DELETE", and "RENAME". Below this section is another titled "Existing Data" which displays a table with the following data:

PASSWORD	EMAIL	ROLE
2020001	2020001@iitgn.ac.in	Student
2020002	2020002@iitgn.ac.in	Student
2020003	2020003@iitgn.ac.in	Student
2020004	2020004@iitgn.ac.in	Student
2020005	2020005@iitgn.ac.in	Student

As you can see, the admin can not only view the entire credential database, but can even alter its content. It's a substantial breach of every user's integrity as if some malicious personnel were to get access to an admin's login, they can basically lock everyone out of the system.

To solve this problem, we implemented a simple check before executing our SQL query.

```

@app.route('/view_table<table_name>', methods=['GET', 'POST'])
def view_table(table_name):
    # Establish a connection to MySQL
    conn = mysql.connector.connect(**mysql_config)
    cursor = conn.cursor()

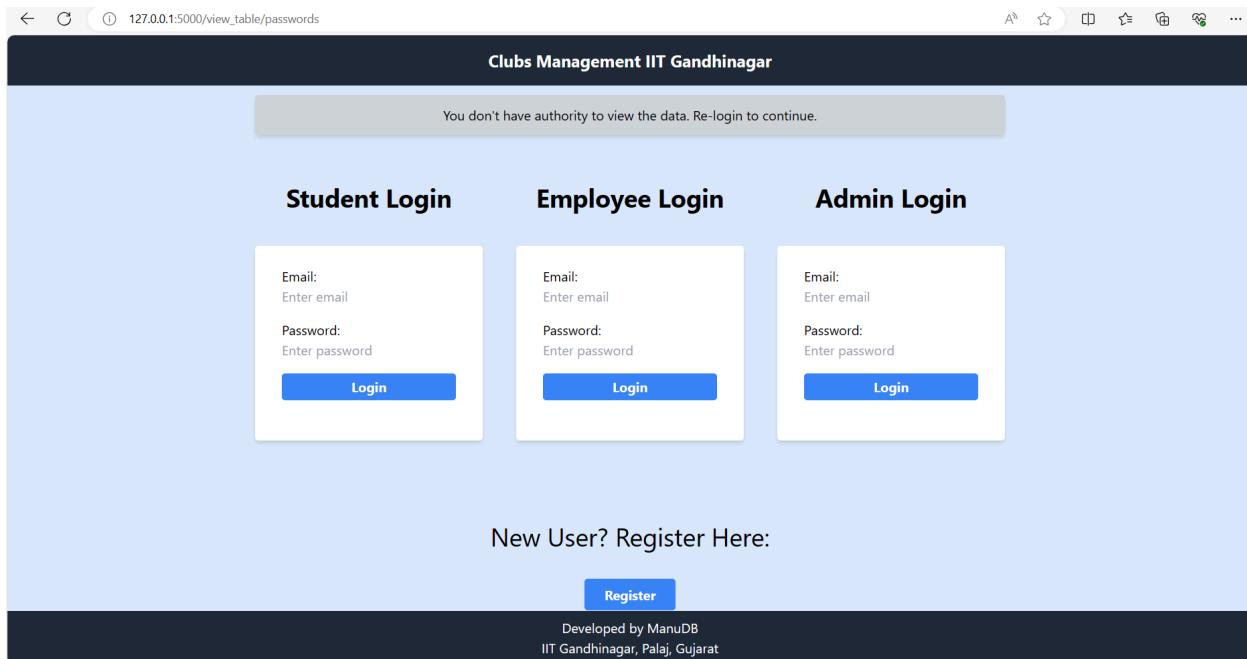
    if table_name == "passwords":
        return render_template('login.html', message = "You don't have authority to view the data.\nRe-login to continue.")

    try:
        # Fetch column names for the specified table
        cursor.execute(f"SHOW COLUMNS FROM {table_name}")
        columns = [column[0] for column in cursor.fetchall()]

        cursor.execute(f"select * from {table_name}")
        data = cursor.fetchall()
    except:
        return render_template('admin.html', message = "Requested table does not exist.")

```

And we obtain the following results against this attack:



2) URL Tampering -

URL attacks, also known as URL parameter manipulation or tampering, involve modifying parameters within a URL to manipulate the behavior of a web application. Attackers may alter parameters such as query strings, form data, or cookies to bypass security controls, access unauthorized resources, or perform other malicious actions.

Our database has mainly three types of roles (not counting sub-roles in students) - Student, Employee, and a few admins. For our assignment III itself, we had implemented few checks to ensure that a user with one role cannot venture out to the webpage of another.

```

@app.route('/studentInfo')
def student_info():
    # Check if the user is logged in, if not redirect to login page
    if 'loggedin' not in session:
        session['message'] = "Please log in first."
        return redirect(url_for('login'))

    print(session['loggedin'])

    if session['loggedin'] == "Admin" or session['loggedin'] == "Employee":
        session['message'] = "Log-in as student to access student-Info page."
        return redirect(url_for('login'))

    return render_template('studentInfo.html', studentInfo = session['user_info'])

```

But there were a few webpages for which we did not implement the above user check. For example, an employee can't view the equipment or event page. But by just typing the URL into these pages in the URL they are able to view them.

Here we are logged-in as an employee but are still able to visit the equipments page.

Another issue is that we have applied login-checks on pages that require user data or some session info to render its contents. But on other pages, which are static, one can view them without log-in.

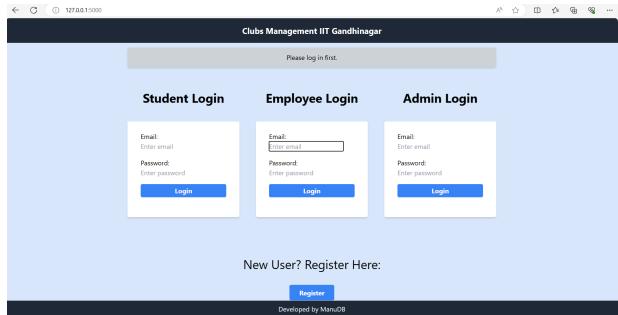
Council Name	Description	Total Members	Actions
Academic Council	Manages academic affairs of students	6	Know More about Academic Council
Cultural Council	Manages cultural events of the college	10	Know More about Cultural Council
IRP Council	Brings industrial opportunities and exposure to students through projects	6	Know More about IRP Council
PCC	Responsible for skill development and training for building careers	6	Know More about PCC
Sports Council	Promoting sports	5	Know More about Sports Council
Technical Council	Responsible for promoting technical culture of the college	9	Know More about Technical Council
Welfare Council	Looking out for hostel and daily amenities of students	6	Know More about Welfare Council

Club Name	Description	Total Members	Actions
Anveenathan	Research society of college which promotes field of research	3	Know More about Anveenathan

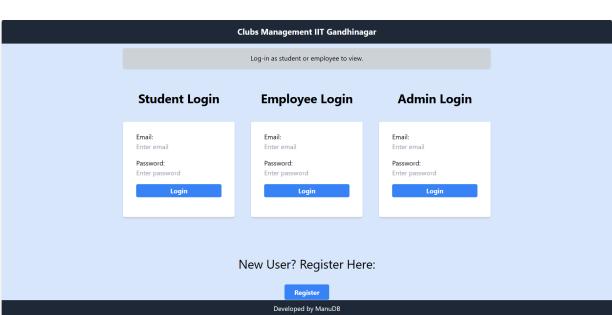
Here we are yet to login but can still visit the councils and its subsequent pages

The simple fix to these issues is to apply checks, like we had for studentInfo, employeeInfo, and admin pages, before loading any webpage as shown in the past code snippet.

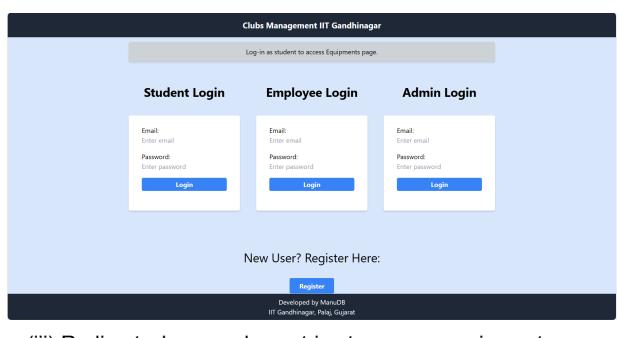
(i) Redirect when accessing website without login



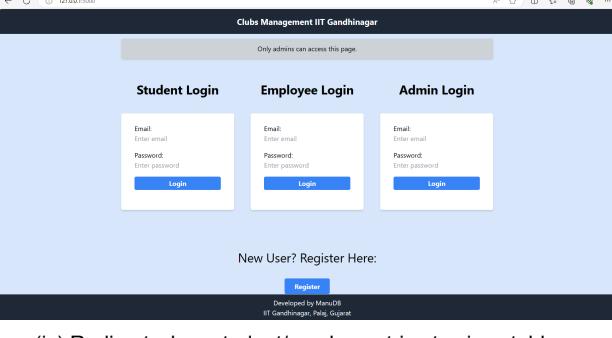
(ii) Redirect when admin tries to access events page



(iii) Redirect when employee tries to access equipment page



(iv) Redirect when student/employee tries to view_table



3) Session Hijacking -

Session hijacking involves an attacker gaining unauthorized access to a user's session ID or cookie. By obtaining this information, the attacker can impersonate the user and perform actions as if they were the legitimate user.

On our website, when a student logs in we run queries to see whether he is a secretary of any club or council so that accordingly we can display the info on the webpage. We store this result in the session map. When he or she would logout, their session would get clear, and the following user can start afresh. However instead of clicking on logout when the user returns to the login page via backtracking, the session keys are not cleared. This is not a problem when another student logs in as the results of the above-mentioned queries would overwrite the values of the past session. However, employees don't have `session["council_secretary"]` or `session["club_secretary"]` in their login query, meaning that the session details for a secretary can be reused by an employee even though he should not have that authorization.

```
127.0.0.1 - - [14/Apr/2024 16:33:03] "GET / HTTP/1.1" 200 -
Received email: 2020004@iitgn.ac.in
Received roll number: 2020004
User: {'ROLL_NO': 2020004, 'EMAIL': '2020004@iitgn.ac.in', 'CONTACT_NO': 8912345670, 'FIRST_NAME': 'Aisha', 'MIDDLE_NAME': 'Meera', 'LAST_NAME': 'Khan', 'STREET': 'Indiranagar 100 Feet Road', 'CITY': 'Bangalore', 'STATE': 'Karnataka', 'PINCODE': 456789, 'DATE_OF_BIRTH': datetime.date(1999, 1, 30), 'AGE': 25, 'PROGRAMME': 'MSc', 'DISCIPLINE': 'ChE', 'YEAR': 2020}
```

```

Student
True
[{"EVENT_NAME": "Tech Symposium", "EDITION": 2021, "BUDGET": 15000, "APPROVAL_STATUS": "accepted", "Overseer": 39, "Event_Lead": 2020041}, {"EVENT_NAME": "Tech Symposium", "EDITION": 2021, "BUDGET": 15000, "APPROVAL_STATUS": "accepted", "Overseer": 39, "Event_Lead": 2020018}, {"EVENT_NAME": "Science Fair", "EDITION": 2023, "BUDGET": 10000, "APPROVAL_STATUS": "accepted", "Overseer": 54, "Event_Lead": 2021011}]

127.0.0.1 - - [14/Apr/2024 16:33:26] "GET /clubs/Odyssey HTTP/1.1" 200 -
Received email: vikram.singh@iitgn.ac.in
Received employee id: 7
User: {'EMPLOYEE_ID': 7, 'EMAIL': 'vikram.singh@iitgn.ac.in', 'FIRST_NAME': 'Vikram', 'MIDDLE_NAME': 'Sunil', 'LAST_NAME': 'Singh', 'DEPARTMENT': 'Chemical', 'DESIGNATION': 'PROFESSOR', 'CONTACT_NO': 6078901234}

127.0.0.1 - - [14/Apr/2024 16:33:52] "GET /council_members/Technical%20Council HTTP/1.1" 200 -
Employee
True
[{"EVENT_NAME": "Tech Symposium", "EDITION": 2021, "BUDGET": 15000, "APPROVAL_STATUS": "accepted", "Overseer": 39, "Event_Lead": 2020041}, {"EVENT_NAME": "Tech Symposium", "EDITION": 2021, "BUDGET": 15000, "APPROVAL_STATUS": "accepted", "Overseer": 39, "Event_Lead": 2020018}, {"EVENT_NAME": "Science Fair", "EDITION": 2023, "BUDGET": 10000, "APPROVAL_STATUS": "accepted", "Overseer": 54, "Event_Lead": 2021011}]
127.0.0.1 - - [14/Apr/2024 16:33:54] "GET /clubs/Odyssey HTTP/1.1" 200 -

```

As you can see above, the employee with id-7 was able to reuse the session info of the student with roll_number-2020004 (who happens to be the Odyssey club secretary).

Roll Number	Email	Contact Number	Name	Position

Event name	Edition	Budget	Approval Status	Overseer	Event lead
Tech Symposium	2021	15000	accepted	39	2020041
Tech Symposium	2021	15000	accepted	39	2020018
Science Fair	2023	10000	accepted	54	2021011

Both SRN-202004 and EID-7 could view this page.

The simple fix to the problem is to clear the session whenever we revisit the login page. This might cause some inconvenience as users might have to re-login frequently, but this ensures that no malicious user can hijack another user's session.

```

@app.route('/', methods=['GET', 'POST'])
def login():
    # message = None
    if "message" in session:
        message = session["message"]
    else:
        message = None
    # session["message"] = None

    session.clear()

```

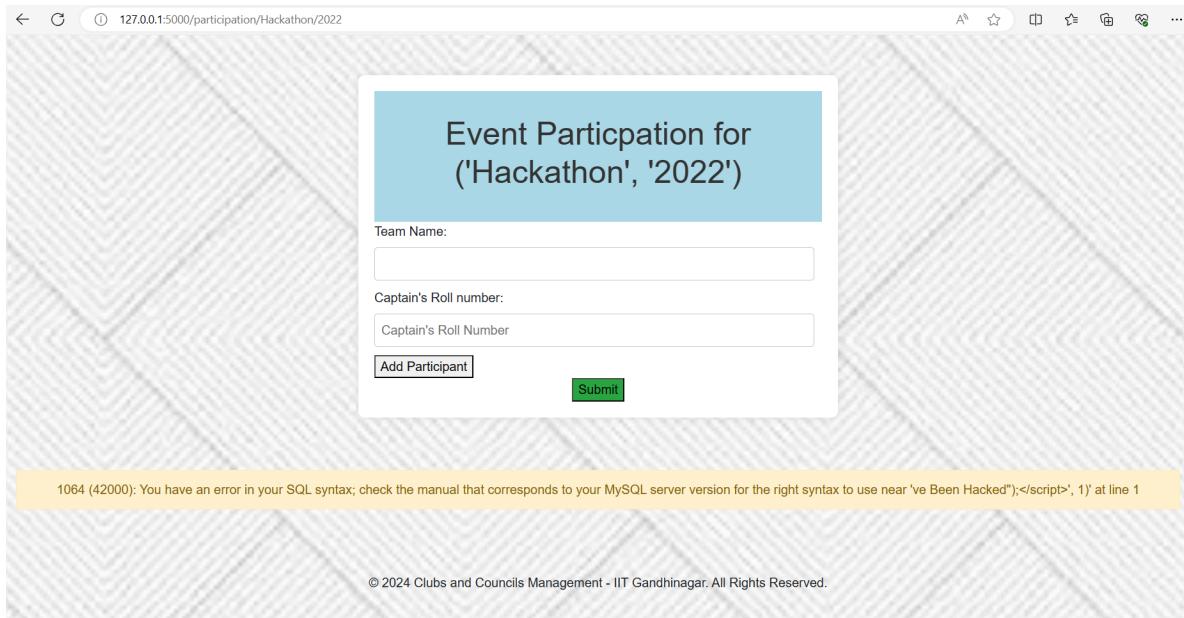
4) XSS Attack -

Cross-Site Scripting (XSS) is a type of security vulnerability commonly found in web applications. It occurs when an attacker injects malicious scripts into web pages viewed by other users. These scripts are typically executed in the context of a victim's browser, allowing the attacker to steal sensitive information, manipulate web page content, hijack user sessions, and perform other malicious actions.

We have tried performing XSS attacks on various sections of our web pages as shown below.

```
student
[{"EVENT_NAME": "Tech Symposium", "EDITION": 2021, "BUDGET": 15000, "APPROVAL_STATUS": "accepted", "Overseer": 39, "Event_Lead": 2020041}, {"EVENT_NAME": "Tech Symposium", "EDITION": 2021, "BUDGET": 15000, "APPROVAL_STATUS": "accepted", "Overseer": 39, "Event_Lead": 2020018}, {"EVENT_NAME": "Science Fair", "EDITION": 2023, "BUDGET": 10000, "APPROVAL_STATUS": "accepted", "Overseer": 54, "Event_Lead": 2021011}]
Error : 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '<script>alert("You've Been Hacked");</script>, 'General Member')' at line 1
127.0.0.1 - [14/Apr/2024 18:55:35] "POST /update_club_members/Odyssey HTTP/1.1" 200 -
```

Output of attack on updating club members



Output of attack on participation form



127.0.0.1:5000/view_table/issue

ISSUE_TIME	<input type="text"/>
RETURN_TIME	<input type="text"/>
PURPOSE	<input type="text"/>

Buttons: INSERT, DELETE, RENAME

Existing Data

EQUIPMENT_ID	ROLL_NO	ISSUE_TIME	RETURN_TIME	PURPOSE
321	2021015	2023-10-28 15:00:00	2023-10-28 21:00:00	Art Competition
2543	2020007	2023-11-14 19:00:00	2023-11-14 23:00:00	Club demonstration
9876	2022011	2023-11-17 16:00:00	2023-11-17 19:00:00	Dance Event
12	2021025	2023-11-19 17:30:00	2023-11-19 21:30:00	Drama Performance
4567	2020004	2023-11-30 21:30:00	2023-12-01 01:00:00	Club demonstration
9850	2023017	2023-12-25 19:30:00	2023-12-25 22:30:00	For drama audience
8910	2022014	2023-12-30 10:00:00	2023-12-30 15:00:00	Art Competition
3456	2020008	2024-02-14 22:00:00	2024-02-14 22:00:00	Gaming Night
7894	2021034	2024-02-14 22:30:00	2024-02-14 23:45:00	Performance
12	2020002	2024-04-14 04:50:59	2024-04-14 04:51:09	trial
321	2020002	2024-04-14 19:23:28	None	<script>alert("Security Compromised");</script>

Output of attack equipment issue (No effect on the website)

Clearly, our website is robust against these XSS attacks. This is because all the inputs taken from the user on the webpage and all the variables shared to the front end have SQL queries running in the intermediate phase. This part ensures that all constraints are matched and handles any undesired outcome or irrelevant input. This means there is no harm to our website or our database. The worst we can get is a warning pop-up for wrong data entry.

Relations and Constraints -

i) All the database-defined constraints are directly handled, for eg below is an instance of adding a new entry for roll no 2020004, which violates the primary key constraint and a message for it pops up at the top.

Error: 1062 (23000): Duplicate entry '2020004' for key 'students.PRIMARY'

Viewing Table: students

Enter New Data

Column Name	Data Entry
ROLL_NO	2020004
EMAIL	blabla@itgn.ac.in
CONTACT_NO	8912345671
FIRST_NAME	Geeta
MIDDLE_NAME	B
LAST_NAME	C

ii) Below is the error message when a foreign key constraint is being violated by adding a participation for an event which does not exist yet.

Error: 1452 (23000): Cannot add or update a child row: a foreign key constraint fails ('club_ms`.`participation', CONSTRAINT 'participation_ibfk_2' FOREIGN KEY ('EVENTS_NAME') REFERENCES 'event' ('EVENT_NAME', 'EDITION') ON DELETE CASCADE ON UPDATE CASCADE)

Viewing Table: participation

Enter New Data

Column Name	Data Entry
EVENTS_NAME	noevent
EDITIONS	69
ROLL_NO	2020004
TEAM_NAME	myteam
TEAM_CAPTAIN	1

INSERT **DELETE** **RENAME**

Existing Data

EVENTS_NAME	EDITIONS	ROLL_NO	TEAM_NAME	TEAM_CAPTAIN
Reverb	2022	2020009	Sush	1
Reverb	2022	2020018	Sush	0

iii) Below is an example of implicit handling of a self-defined constraint, where one team must have one and exactly one captain. There were not many ways to implement this within the database itself, however, while registering participation from a team, we are making sure of it ourselves.

Event Participation for ('Coding Nights', '202303')

Team Name:

Captain's Roll number:

Roll Number 1:

Roll Number 2:

Add Participant **Submit**

© 2024 Clubs and Councils Management - IIT Gandhinagar. All Rights Reserved.

iv) Example of an entry being made of a non-existent roll number, popping up the corresponding error.

Event Participation for ('Photography Contest', '202311')

Team Name:

Captain's Roll number:

Add Participant Submit

1264 (22003): Out of range value for column 'ROLL_NO' at row 1

© 2024 Clubs and Councils Management - IIT Gandhinagar. All Rights Reserved.

v) Implementation focussing not allowing issue count of a particular equipment to exceed the specified total quantity. Below is an example where an equipment spotlight which already has 2 entries and a total count of 2 cannot be issued again and a message is visible, which indicates failure of request.

Clubs Management IIT Gandhinagar Clubs and Councils Events Logout

Equipment Handling

Equipment Action:

Issue

Not available at the moment

202003	2103	2024-04-14 23:41:52	None	practice
202003	2103	2024-04-14 23:42:03	None	demo

EQUIPMENT_ID	NAME	QUANTITY	STATE	ACQUISITION_DATE	AVAILABILITY	club_name
2103	Spotlight	2	Good	2021-06-19	1	Abhinaya
2245	Claude Monet Watercolors	1	Good	2022-11-19	1	Soram

Contribution:

Team G1 -

Shubh Singhal:

- Took feedback from the stakeholders and made strategies to combat the suggestions.
- Implemented various changes in the web app as per the feedback.
- Worked on documenting the part of constraint implementation.

Amey Rangari:

- Handled frontend modifications based on the feedback received from stakeholders
- Helped modify the login and registration webpages
- Implemented the frontend part of allowing club secretaries to view respective club events and their approval status.

Anshul Choudhary:

- Wrote all the privileges of different class of users and included their screenshots
- Helped in setting up defences and performing attacks on the website
- Worked in the final documentation of the report

Shrijan Sahu:

- Helped in designing the frontend for the register page.
- Also implemented the changes due to feedback.
- Worked in the final documentation of the report.

Team G2 -

Yashraj Deshmukh:

- Implemented the changes in the database as per the feedback received
- Created the passwords table and worked on the back end of the registration page
- Performed various attacks on our website and set up defenses against them

Pratham Sagar:

- Implemented suggestions received in feedback on the Participation page where now multiple participants can be added to the team and improved its front-end styling.
- Hosted the site such that anyone can now access the website under the IITGN-SSO network.
- Handled Concurrency, i.e., multiple users can access the website and view and update under such that concurrency is respected.

Manvendra Singh:

- Worked with google authentication part to ensure that only IIT Gandhinagar students should be allowed to register.
- Helped in the backend part after the first feedback.
- Helped in implementing the changes in the database as per the feedback received.