Club Activities Management

JUNE 1, 2020

CBIT

Gandipet, Hyderabad

Project Report

Club Activities Management

by

CH.SAITEJA

V.LAHARI

E.ANUSHA

P.VISHNU VARDHAN REDDY

VINEETH SHARMA

L.MANOJ

TABLE OF CONTENTS

ACKNOWLEDGEMENT

Firstly, we are grateful to COSC for giving us this opportunity to do an internship amidst this period of lockdown. We thank Chaitanya Bharathi Institute of Technology for providing us the scope and supporting our internship. We are thankful to our faculty mentor Trupthi Mukund (Assistant Professor, Dept. of IT - CBIT) for providing us his valuable assistance for doing this project and recommending us some features. We are grateful from the bottom of our hearts to our mentor Syed Haany Ali for his valuable guidance and support all the times. Finally, we thank each of our team mates for working hard for all the days and completing this project successfully.

ABSTRACT

Club activities management is a project which is developed to provide and manage various clubs and their activities. The main purpose of this project to help the students by providing all the details of the club and the activities that are going on in the club. This application provides services online and stores all the details of the clubs present and the details of the student who are part in the club.

The present system doesn’t provide all the details of the clubs and their activities that are taking place at a place, they have to publicize by making announcements in all the classes or making poster available in the notice boards, sharing the poster through whatsapp. This type of publicity may not reach to all by our application provides an easy way of accessing all the details through a single click in their phones. Now a days many students prefer to use internet and get all the details they doesn’t prefer online. This application can be browsed from any where using internet service and all the details and activities of different clubs can be viewed.

PROBLEM STATEMENT

Students details and the club's they have enrolled and activities they have participated and courses done under different clubs

SUPER ADMIN

- Super admin login
- Receive all Club Registrations and approve them
- Dismiss club

ADMIN (one for each club)

- Club Registration
- Admin Login
- Club Profile
- Upload all the list of participants in the events
- Upload all club members with their roles

USER

- Student Login
- Home page (Sections - Clubs, Activities)
- Clubs section should contain all the clubs in which the student is a member and his role.
- Activities section should contain all the events he participated in.
- Profile page (all the academic and cocurricular details of the student)

1. **Introduction**

About Project:

Club activities management is very useful for all the students .It gives us the valuable information about the club and their activities and saves time. Our project built an app that provides all the details of the club and can search from anywhere and at any time. You can search a club for its prominent activities. The user registration can be done easily with in the app.

The project is divided in to three parts android for application building done using Java/Kotlin, Web development this is done using Django framework, HTML/CSS, Java Script, JWT(for tokens and session handling), Database is with MySQL and Rest API using Flask. The app contains the complete information of all the clubs along with their activities, members in the club and their complete details. This app can be used by any person who is having general knowledge about internet.

Overall description consists of background of the entire specific requirement. It also gives explanation about actor and function which is used. It gives explanation about architecture diagram and it also gives what we are assumed and dependencies. It also support specific requirement and also it support functional requirement, supplementary requirement other than actor which is used.

2. Project Analysis

2.1 Purpose of the Project:

The main aim of this project is to provide services to the students. The services provided are regarding all the club details, what are the events that the club is going to host, registration for the interested events. All the information regarding to the clubs can be known very easily with just a click with out searching here and there or consulting many people. The app provides the complete information about all the clubs.

It facilitates communication between head of the clubs/admins and members who are there in the club and also all the students. This will definitely help the users for the purpose of saving their valuable time which can't be got back. The user has to fill their profile for all who wants to get the services. Then they can login using their credentials for services. This can be categorized based on the type of users. IN this project we have normal users, admins who are head of the respective clubs and super admin who can manage all the things. It provides different login forms for different categories. In case of students if they want to know the details of the clubs and events they need to give their id and password for security purpose.

2.2 Existing System:

In Existing system the students has to gather information about the events from different means like notice boards, whats app or through announcements . The complete details cannot be known easily. This posses a lot of time. In order to get each piece of information we need to go for help desk.

2.2.1 Limitations of Existing System:

- The existing system is a manual system. Here the information need to be gathered or forwarded in groups, or pasted in notice boards.
- There is no sharing is possible if the data is in the form of paper.
- The manual system gives us very less security for saving data; some data may be lost due to mismanagement.

2.3 Proposed System:

The Proposed System provides an online information about the clubs and their activities. It provides all the details about the club. The development of this new system contains the following activities, which try to automate the entire process keeping in the view of database integration approach.

- User friendliness is provided in the application with various controls provided by system Rich User Interface.
- The system makes the overall project management much easier and flexible.
- It can be accessed over the Intranet.
- The club information files can be stored in centralized database which can be maintained by the system.

3. Requirement Analysis

3.1 Scope:

- It can be accessed by unlimited number of users.
- Each user will be assigned a different set of permissions for each module of the system.
- The user can have access to all the information in the site with limited services and provide extra services to admins and super admins.
- Every user must have their id and password for security purpose.
- Only registered members will be provided with communication between users and admins through mails.
- Super Administrator is created in the system already.
- The super admin can manage all the operations in the application.
- This application is best designed to be useful through internet to people of different places.

3.2 Users of the System

- Students
- Admins who are also students heads of the each club.
- Super admin can be faculty/head of all the clubs.

4. Specific Requirements

4.1 Functional and Non Functional Requirements

- **Functional Requirements:**
- Details of all the clubs and their activities.
- Secure login of all users after filling the personal profile.
- Facilitate communication between users and admins through mails.
- All the alerts and notifications are sent to mail.

- **Non-Functional Requirements:**
- 24*7 availability
- Better component design to get better performance at peak time
- Flexible service based architecture will be highly desirable for future extension.

4.2 User Interface Requirements

- Professional look and feel.
- Use of HTML/CSS for all login pages.
- Browser testing and support for Chrome, Mozilla, and Fire fox

4.3 Hardware Requirements

- Pentium 233HZ 80 GB HD, 1024 MB RAM (Server).
- Any P.C with Windows,IOS, 256 MB RAM (Client).
- Internet connection with 33.6 KBPS Modem.

4.4 Software Requirements

- Windows, IOS
- Flask and Django
- MySQL
- Android studio

5. System Design

5.1 Behavioural Diagrams

5.1.1 Use Case Diagrams: A use case is a methodology used in system analysis in identify, clarify, and organize system requirements. The use case is made up of a set of possible sequences of interactions between systems and users in a particular environment and related to a particular goal.

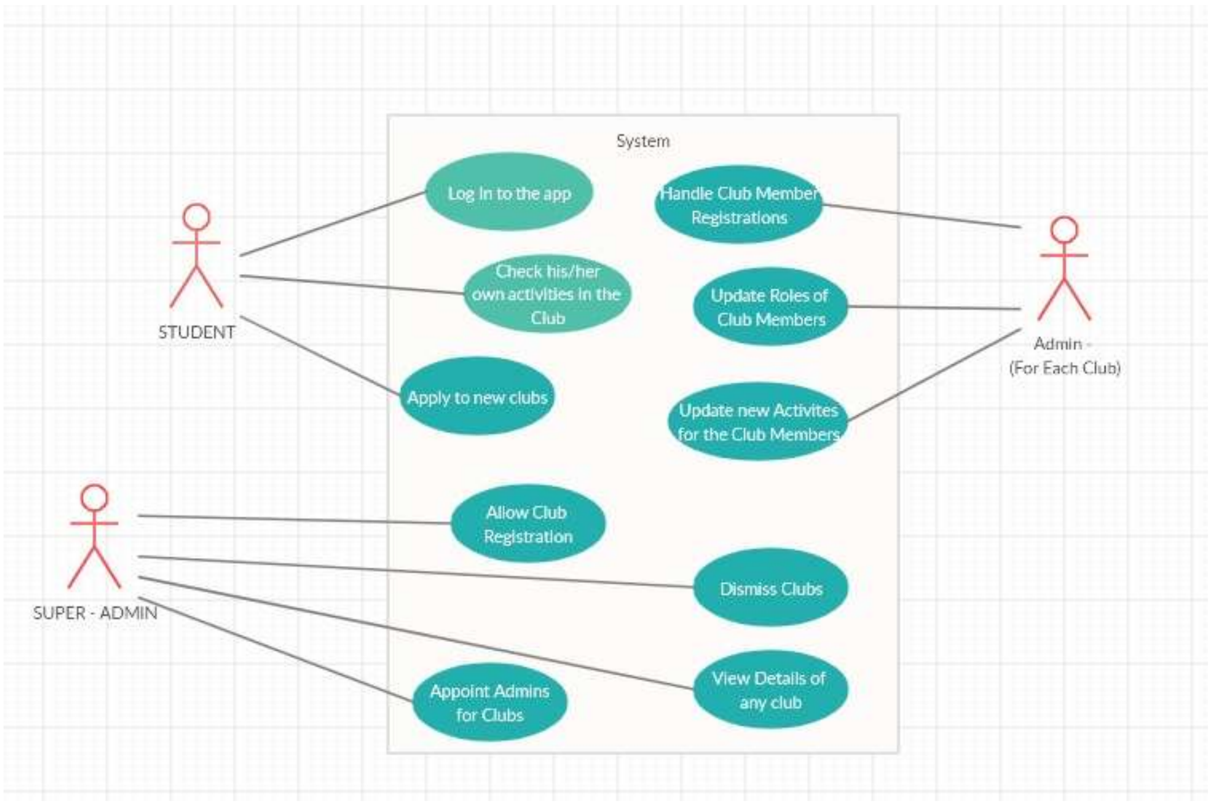


Fig.6.2.1 Use Case Diagram of the System

Name of the actor: Student

Description:

Every student has to login and fill his profile.

Pre Condition: No pre condition exists

Post condition: Home Page will be displayed

Flow of events:

- ➢ Invoke the home page.
- ➢ Views the Required information pages.
- ➢ Access required services through sign up.
- ➢ Can apply new clubs
- ➢ Can view own activities.

Name of the actor: Admin

Description:

Every club has an admin who has to manage his own club activities.

Pre Condition: Admin has to login with his credentials to view admin home page.

Post condition: Respective club details and functions will be displayed.

Flow of events:

- ➢ Invoke the home page.
- ➢ Admin can manage club registrations.
- ➢ Can modify club members roles.
- ➢ Can manage the requests made by students.
- ➢ Activities of the club can be managed.\

Name of the actor: Super Admin

Description:

Only one super admin is present. Super admin can manage all clubs.

Pre Condition: First register and Login with userid and password.

Post condition: Select the required service you will.

Flow of events:

- Invoke the home page.
- Login through UserId and password.
- Click on services in your home page.
- Click the required service you choose.

6. User roles and their description

There are three different types of users in our project .They are listed below with their description.

1. User
2. Admin (for each club)
3. Super Admin

1.User: All the students are users they can login using their id and password and use the services provided. The user has to fill the profile. Later they can edit the profile. The home page contains the clubs and activities sections. The clubs section contains all the clubs in which the student is member and his role. Activities section should contain all the events in which student is a part. Profile page contains all the academic and cocurricular activities of the student.

The student can register for any number of clubs. Student can make request to the club, so that it will appear for the admin of the respective club. Initially the accept status will be set to -1 when the admin accepts his/her request to the club it will be changed to1. The student can see what are all the events that are going to be hosted by all clubs.

2.Admin: There are many admins, each club has one admin. Admin posses all rights of the respective club. Admin has to login with his credentials provided by the super admin. Admin can change password later on by using change password option. When change password option is clicked an email with one time password (otp) and change password link will be sent to the registered email by using that admin can change password.

Admin has a right to accept the requests made by the student to join into the club. Admin can display all the members in the club. Admin can show all the events that are going to be host by the club. Admin can delete any member in the club or can edit the member in the club but admin cant delete the entire club.

Before deleting the club by super admin or deleting a member from club an email will be sent to the particular user. Whenever an admin accepts the request an email will be sent to the student stating that admin accepted your request. Admin can decide the role of the student in the club. Admin can manage all the things of his/her respective club.

3.Super Admin: There is only one super admin for all clubs. Super admin has to login with credentials. Super admin can monitor all the clubs present. Super admin can create clubs and allot admins to the clubs. When ever super admin creates a club automatically table will be created in the data base with the club name.

Super admin has a power to delete any club or any member in a club. The super admin has a higher privilege among the different user roles. Super admin can manage all clubs.

7. IMPLEMENTATION OF THE PROPOSED SYSTEM

7.1 Tools and technologies used

Android Studio - Android Studio is the official integrated development environment for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development.

Java - Java is a general-purpose programming language that is class-based and object-oriented, and designed to have as few implementation dependencies as possible. Android application in this project is programmed using Java in Android Studio.

Python - Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace.

Django - Django is a Python-based free and open-source web framework that follows the model-template-view architectural pattern. It is used in this project to make the web application for the admin.

Flask and Flask RESTful (for RESTful API)- Flask is a micro web framework written in Python. Flask-RESTful is an extension for Flask that adds support for quickly building REST APIs. An application programming interface (API) is a computing interface which defines interactions between multiple software intermediaries. It defines the kinds of calls or requests that can be made, how to make them, the data formats that should be used, the conventions to follow, etc. Representational state transfer (REST) is a software architectural style that defines a set of constraints to be used for creating Web services. Web service APIs that adhere to the REST architectural constraints are called RESTful APIs.

Postman - Postman is a collaboration platform for API development. Postman's features simplify each step of building an API and streamline collaboration so you can create better APIs—faster.

HTML, CSS and JavaScript - Hypertext Markup Language is the standard mark-up language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript

Bootstrap - Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development.

JSON - JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. It is based on a subset of the JavaScript Programming Language Standard. It is used in APIs to send and receive the data.

MySQL - MySQL is an open-source relational database management system (RDBMS)."SQL" is the abbreviation for Structured Query Language. A relational database organizes data into one or more data tables in which data types may be related to each other; these relations help structure the data.

7.2 REST API

The API is designed so that it can accept requests from web app and mobile app to send or receive data. The data sent can either be used to insert into the database, delete, update or retrieve information appropriately.

The API takes in some data sent by the API caller and API returns a message accordingly, whether the action to be performed by the API is successful or not – the API returns an error if the data is not found or when there is an internal server error or connection error when connecting to the database. API is useful as it maintains the common database for both admin and end users. It's a flexible way to do operations with the data to get a desired response or change in the database. This API is stored in a server or a similar location and API callers have to make calls to specific endpoints of the API.

Rest api is done using Flask End points in Rest api are:

```
1.  (Login,'/login')
```

This end point is used to login for the super admin. If the credentials are correct then an access token is generated, this access token is used for next steps. Initially we have provided the login credentials for the super admin. The super admin have many functionalities like adding a club, deleting the club, add an Admin, can view all clubs.

```
2.  (Addclub,'/addclub')
```

This is to add the new club and also creates the new admin. This also creates a club with that club name. Initially we assign default passwords for all admins.

```
3.  (Profile,'/profile')
```

This end point is to display or post the profile details of all the members. Profile include name,id,branch,hobbies,email etc.

```
4.  (Clubdelete,'/del')
```

Super admin can only delete any club. This deletes the admin which also deletes the table with that club name.

```
5.  (Clubnames,'/clubnames')
```

This is used to display all the club names which are present in the data base. This can be used by students to find what are all the clubs present .

```
6.  (Allclubdetails,'/allclub')
```

This is used to find all the club details present like how many members are present and what are all the clubs present and what are the roles of the respective members in the club.

7. `(Addclubmembers,'/addclubmembers')`

This is specific for the admins, admins can add the members in to the club by accepting the request sent by the students. Initially default accept status is -1 and when an admin accepts the request by the respective admin the accept status is changed to 1.

8. `(Changepassword,'/changepassword')`

Initially the default password is set for admins to change the default password admins can change their password. When admin clicks change password an email is sent to the respective email with otp.

9. `(Forgotpassword,'/forgotpassword')`

This feature enable to recover the password when they forget.

10. `(Clubmembers,'/clubmembers')`

This is used to find the members who are present in the respective club.

11. `(Requesttoclub,'/requesttoclub')`

This is used to send the request to the admin of the respective club. Student will send the request of joining to the club this will send the request to the admin. The admin has a functionality to view the requests if admin accepts it will change the accept status .

12. `(Adminlog,'/adminlog')`

This is admin login, there is separate login foe admins. If the credentials are correct admin will be logged in else incorrect credentials message will be displayed. After admin logs in he has some functionalities.

13. `(Displaypostevents,'/displaypostevents')`

Displaying and posting events can be done using this. All the events hosted by the respective club can be displayed.

14. `(Studentlogin,'/slogin')`

This is for student purpose all the students first has to fill the profile then they can login foe next activities. Initially the username and password are their respective ids. They can change their password by clicking change password.

Super Admin endpoints:

Endpoint	Particular	Information
/login (POST)	Data to be sent	username and password
	Response received	A message of successful login or error (internal server error or database related error.)
	Back-end working detail	It checks with the database data whether login credentials are correct or not if correct then generate an access token other wise post an error.
/addclub (POST)	Data to be sent	uid, username, password and clubname
	Response received	A message of successful creation of club
	Back-end working detail	A table in the database will be created and in admin table admin entry will be made. Accept status in the student table will be updated to 1 .
/del (POST)	Data to be sent	username
	Response received	A message of a successful deletion or error (internal server error or database related error.)
	Back-end working detail	The database gets the clubname with particular username in admin table. The club name table will be dropped and also admin entry in the admin table will be deleted.

/editadmin (POST)	Data to be sent	New-admin-id and admin-name
	Response received	The admin details will be edited
	Back-end working detail	If the id is existing it will post an error else the admin details will be edited In the profile table also details will be edited
/clubnames (GET)	Data to be sent	-
	Response received	All the entries in the admin table club names will be displayed
	Back-end working detail	Admin table club names will be fetched.
/clubmembers (GET)	Data to be sent	clubname
	Response received	All the club members in that particular club will be displayed
	Back-end working detail	All the members are retrieved from that particular club table.
/allclub (GET)	Data to be sent	-
	Response received	All the members in all the clubs are returned if successful or a message of data not present is returned or an error message is returned.
	Back-end working detail	All members details are retreived.

Admin endpoints:

/adminlog (POST)	Data to be sent	Userid, password and clubname
	Response received	A message of successful login is returned if successful or else a message of error is returned. (database or internal server error)
	Back-end working detail	The login credentials will be cross checked with the database details if correct then an access token will be generated else error as invalid credentials is returned
/addclubmembers (POST)	Data to be sent	Clubid, stuid, eventname and eventdate
	Response received	A message of successful insertion is returned if successful or else a message of error is returned. (database or internal server error)
	Back-end working detail	The details are inserted in the respective club table created when the club is created.
/delmembers (POST)	Data to be sent	stuid, clubname
	Response received	A message of successful deletion is returned or else a message of error is returned. (database or internal server error)
	Back-end working detail	The entry in the student having the stuid and clubname are deleted.

User endpoints:

/profile (GET)	Data to be sent	-
	Response received	A list of JSONs of distinct student profile are returned if successful or an error message is returned for connection error.
	Back-end working detail	All the profile table details will be fetched
/profile (POST)	Data to be sent	Stuid,name,branch,year,grade,cactivities, hobbies,phoneno,emailid
	Response received	A message of success is returned if successful or else error message is returned
	Back-end working detail	All the given details will be inserted in to the profile table.
/requesttoclub (GET)	Data to be sent	clubname
	Response received	Name,stuid,branch and year if worked successfully or else an error message is returned.
	Back-end working detail	The row corresponding to the clubname is searched for and returned if found. If the accept status =-1.
/requesttoclub (POST)	Data to be sent	cid,stuid,clubname,crole,acceptstatus
	Response received	Success or else an error message is returned.

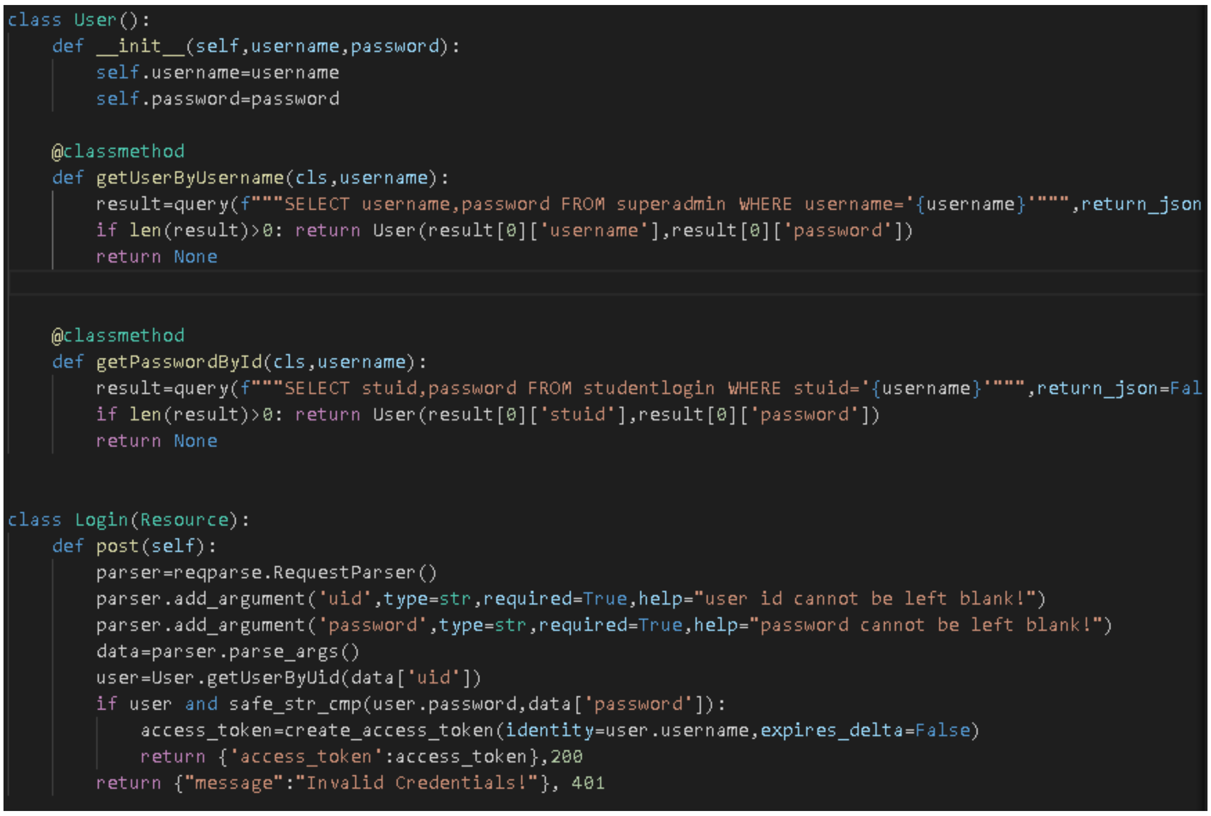
		If the accept status is not equal to -1 then updates Takes place else error message will be displayed
	Back-end working detail	Updation will be done
/eventmembers (GET)	Data to be sent	Clubname,eventname,eventdate
	Response received	All the members of that particular clubname, eventname and event date are returned
	Back-end working detail	Details will be fetched based on the inputs given.
/changepassword (POST)	Data to be sent	otp, password
	Response received	Change password is successful or not or an error message is returned
	Back-end working detail	The otp will be verified and if correct the password will be updated else an error message will be displayed.

Note that all of these endpoints – Super admin ,Admin endpoints and User endpoints require the respective JWT (JSON Web Token) authentication for the corresponding endpoints (excluding the /admin-login, /user-login, /user-register endpoints).

Providing an invalid JSON Web token makes an API issue an invalid token response stating signature verification failed.

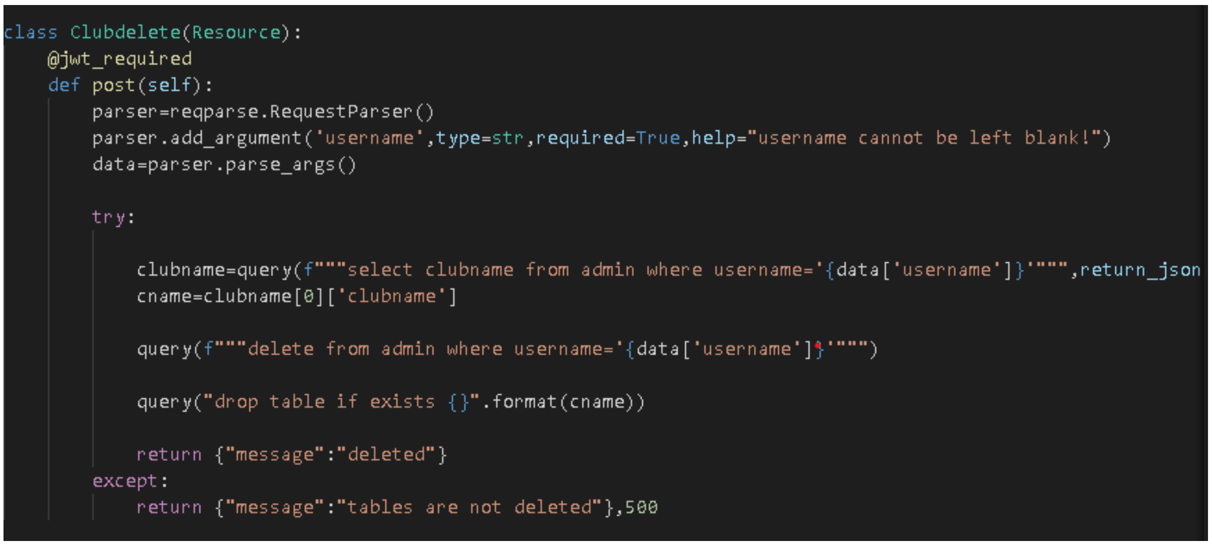
Not providing JWT for the JWT restricted endpoints will return an error stating that authorization is required and that the token is missing.

Creation of access token for super admin:



```python
class User():
    def __init__(self,username,password):
        self.username=username
        self.password=password

    @classmethod
    def getUserByUsername(cls,username):
        result=query(f"""SELECT username,password FROM superadmin WHERE username='{username}'""",return_json
        if len(result)>0: return User(result[0]['username'],result[0]['password'])
        return None


    @classmethod
    def getPasswordById(cls,username):
        result=query(f"""SELECT stuid,password FROM studentlogin WHERE stuid='{username}'""",return_json=Fal
        if len(result)>0: return User(result[0]['stuid'],result[0]['password'])
        return None


class Login(Resource):
    def post(self):
        parser=reqparse.RequestParser()
        parser.add_argument('uid',type=str,required=True,help="user id cannot be left blank!")
        parser.add_argument('password',type=str,required=True,help="password cannot be left blank!")
        data=parser.parse_args()
        user=User.getUserByUid(data['uid'])
        if user and safe_str_cmp(user.password,data['password']):
            access_token=create_access_token(identity=user.username,expires_delta=False)
            return {'access_token':access_token},200
        return {"message":"Invalid Credentials!"}, 401
```

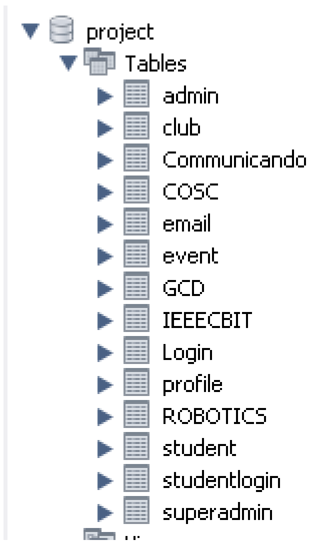
For club deletion by super admin:



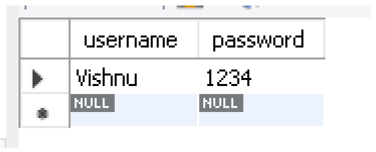
```python
class Clubdelete(Resource):
    @jwt_required
    def post(self):
        parser=reqparse.RequestParser()
        parser.add_argument('username',type=str,required=True,help="username cannot be left blank!")
        data=parser.parse_args()

        try:

            clubname=query(f"""select clubname from admin where username='{data['username']}'""",return_json
            cname=clubname[0]['clubname']

            query(f"""delete from admin where username='{data['username']}'""")

            query("drop table if exists {}".format(cname))

            return {"message":"deleted"}
        except:
            return {"message":"tables are not deleted"},500
```

7.3 Database schema

These are all the tables in our project schema

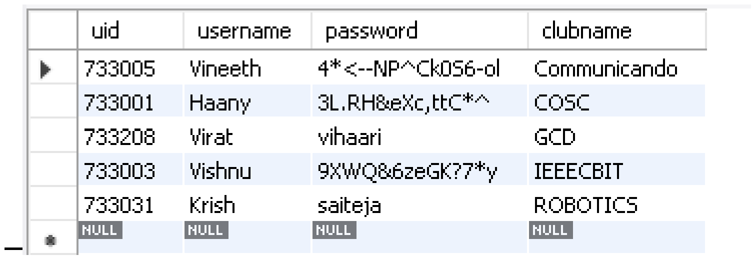

project
Tables
admin
club
Communicando
COSC
email
event
GCD
IEEECBIT
Login
profile
ROBOTICS
student
studentlogin
superadmin

Super Admin table stores the login credentials of super admin



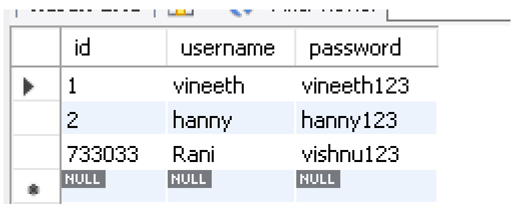
username	password
Vishnu	1234
NULL	NULL

Admin table stores the details of admins for respective clubs.



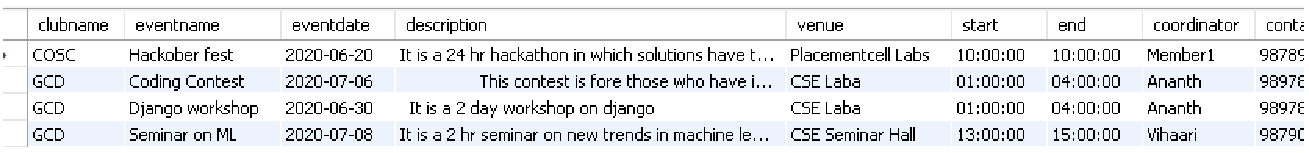
uid	username	password	clubname
733005	Vineeth	4*<--NP^Ck0S6-ol	Communicando
733001	Haany	3L.RH&eXc,ttC*^	COSC
733208	Virat	vihaari	GCD
733003	Vishnu	9XWQ&6zeGK?7*y	IEEECBIT
733031	Krish	saiteja	ROBOTICS
NULL	NULL	NULL	NULL

Login table stores all the login details like username and password.



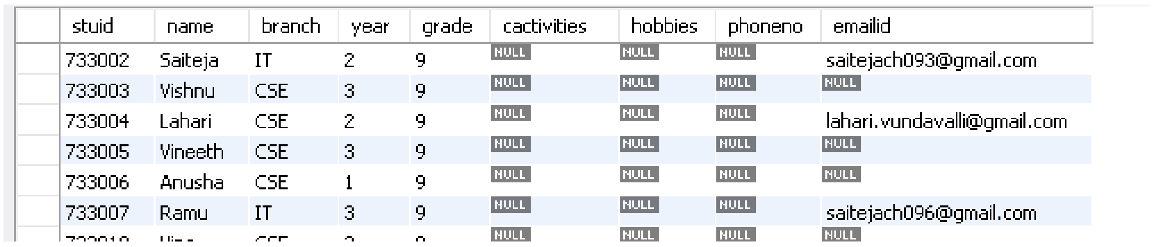
id	username	password
1	vineeth	vineeth123
2	hanny	hanny123
733033	Rani	vishnu123
NULL	NULL	NULL

Event table stores all the events of all the clubs along with description of the event, when the event is going to be , time of the event, venue of the event along with contact number of the head.



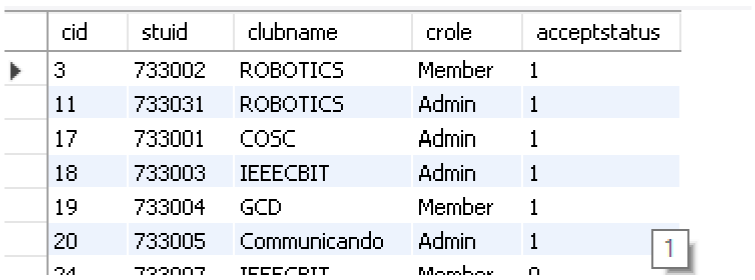
clubname	eventname	eventdate	description	venue	start	end	coordinator	conta
COSC	Hackober fest	2020-06-20	It is a 24 hr hackathon in which solutions have t...	Placementcell Labs	10:00:00	10:00:00	Member1	98789
GCD	Coding Contest	2020-07-06	This contest is fore those who have i...	CSE Laba	01:00:00	04:00:00	Ananth	98978
GCD	Django workshop	2020-06-30	It is a 2 day workshop on django	CSE Laba	01:00:00	04:00:00	Ananth	98978
GCD	Seminar on ML	2020-07-08	It is a 2 hr seminar on new trends in machine le...	CSE Seminar Hall	13:00:00	15:00:00	Vihaari	98790

Profile table contains all the details of the students like student id, name of the student, branch, year , grade, what are the cocurricular activities, hobbies of the student, phone number and email id is must.



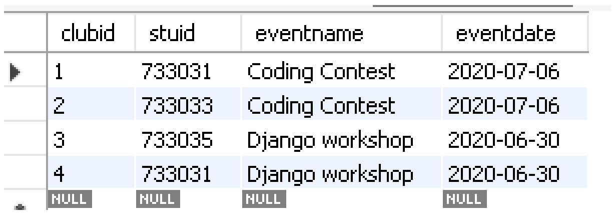
stuid	name	branch	year	grade	cactivities	hobbies	phoneno	emailid
733002	Saiteja	IT	2	9	NULL	NULL	NULL	saitejach093@gmail.com
733003	Vishnu	CSE	3	9	NULL	NULL	NULL	NULL
733004	Lahari	CSE	2	9	NULL	NULL	NULL	lahari.vundavalli@gmail.com
733005	Vineeth	CSE	3	9	NULL	NULL	NULL	NULL
733006	Anusha	CSE	1	9	NULL	NULL	NULL	NULL
733007	Ramu	IT	3	9	NULL	NULL	NULL	saitejach096@gmail.com

Student table stores the student id, what club name he/she is requested to join and role in the club as well as the accept status whether it is -1 or 1.-1 states request is not accepted and 1 states it is accepted by the admin of respective club.



cid	stuid	clubname	crole	acceptstatus
3	733002	ROBOTICS	Member	1
11	733031	ROBOTICS	Admin	1
17	733001	COSC	Admin	1
18	733003	IEEECBIT	Admin	1
19	733004	GCD	Member	1
20	733005	Communicando	Admin	1

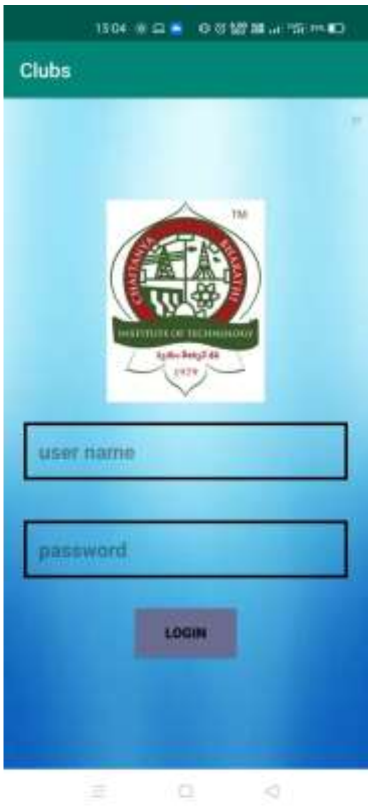
This is the club table for respective clubs. Where the clubid which is auto incremented , id of the student, event name , event date i.e., when the event has hosted.



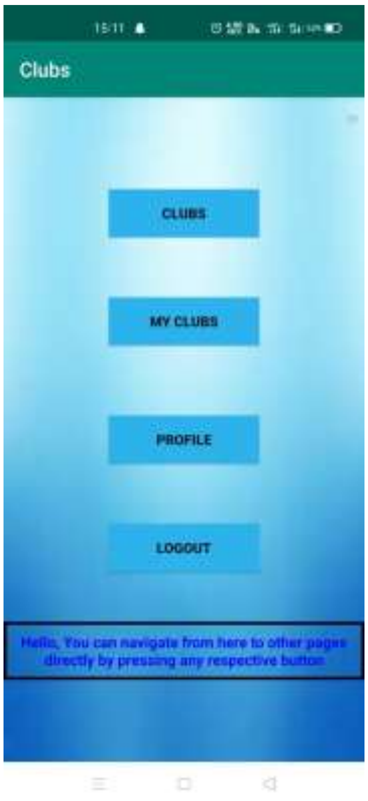
clubid	stuid	eventname	eventdate
1	733031	Coding Contest	2020-07-06
2	733033	Coding Contest	2020-07-06
3	733035	Django workshop	2020-06-30
4	733031	Django workshop	2020-06-30
NULL	NULL	NULL	NULL

7.4 Mobile Application

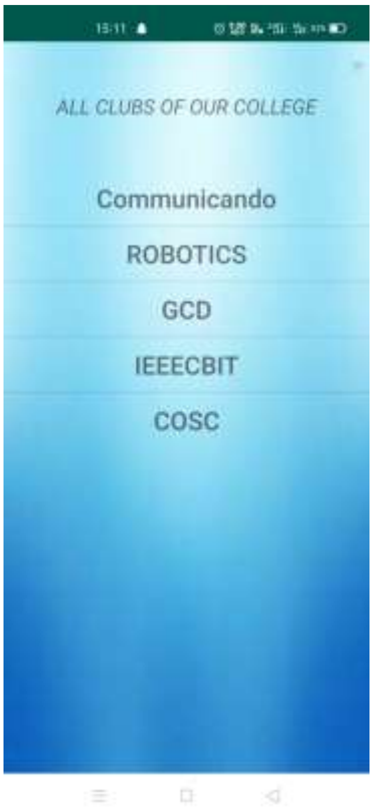
The mobile application, opening, takes user to a login page. The user has to enter the username and password. The user can log in by providing the correct username and password



Once that is done, User will be directed to the dashboard page where the clubs, my clubs, profile, logout buttons are displayed. The user can navigate to the other page using by clicking on buttons.



Hello, You can navigate from here to other pages directly by pressing any respective button



ALL CLUBS OF OUR COLLEGE

Communicando

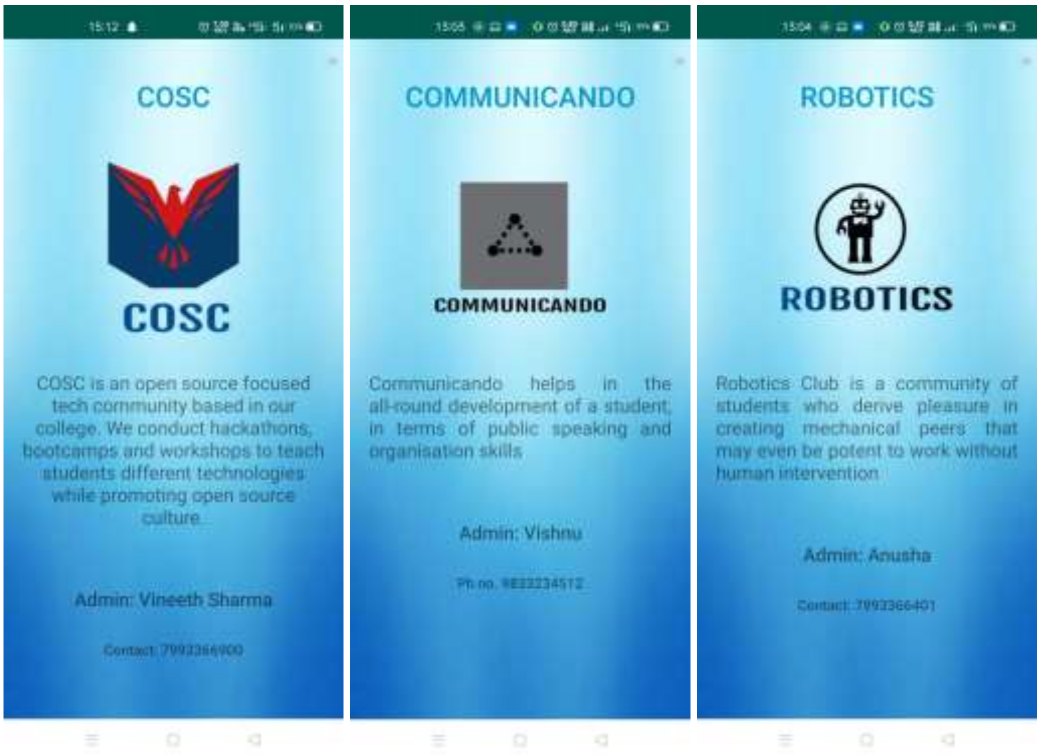
ROBOTICS

GCD

IEEECBIT

COSC

In the dashboard, after selecting clubs the user can view all clubs present in the college. If the user wants to know any information about the club then the user can click on any club so that the details about that club will be displayed.



COSC is an open source focused tech community based in our college. We conduct hackathons, bootcamps and workshops to teach students different technologies while promoting open source culture.

Admin: Vineeth Sharma

Contact: 7993366900

Communicando helps in the all-round development of a student, in terms of public speaking and organisation skills

Admin: Vishnu

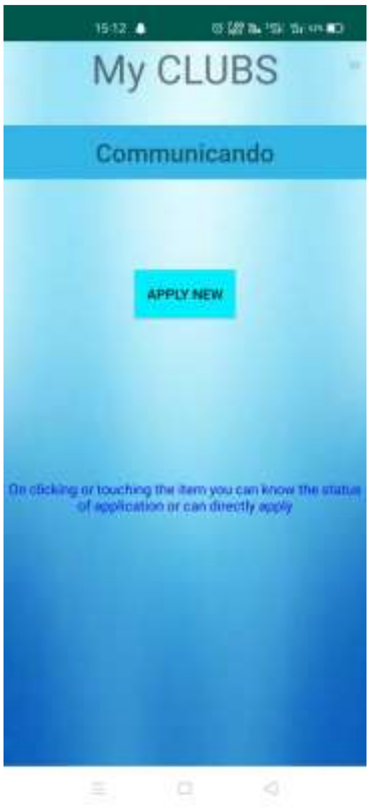
Ph no. 9833234512

Robotics Club is a community of students who derive pleasure in creating mechanical peers that may even be potent to work without human intervention

Admin: Anusha

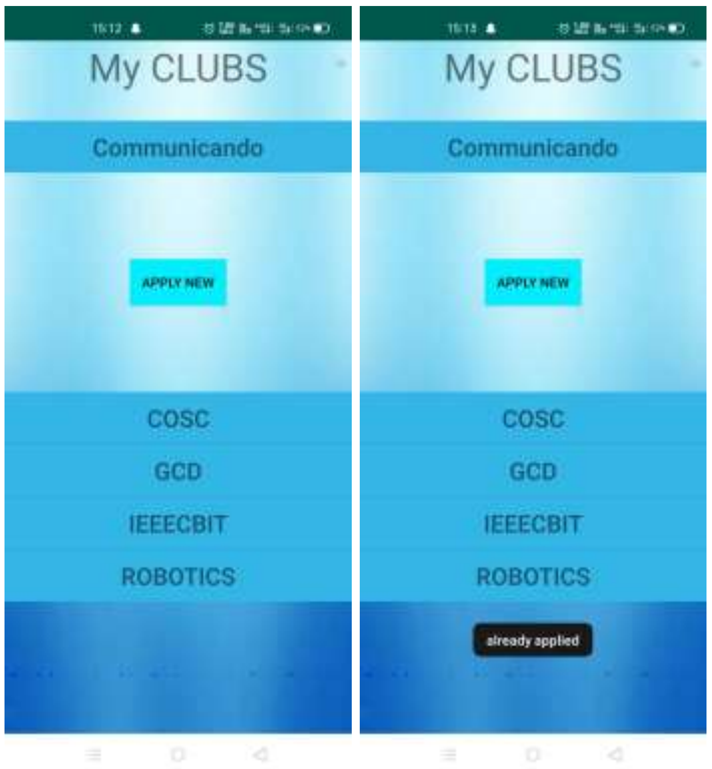
Contact: 7993366401

When user clicks on any club the club description and who is the admin of the club and the contact info the admin will be displayed.



On clicking or touching the item you can know the status of application or can directly apply

When user clicks my clubs it will be directed to this page, where all the club names where the user is part of will be displayed.


My CLUBS

Communicando

APPLY NEW

COSC

GCD

IEEECBIT

ROBOTICS

My CLUBS

Communicando

APPLY NEW

COSC

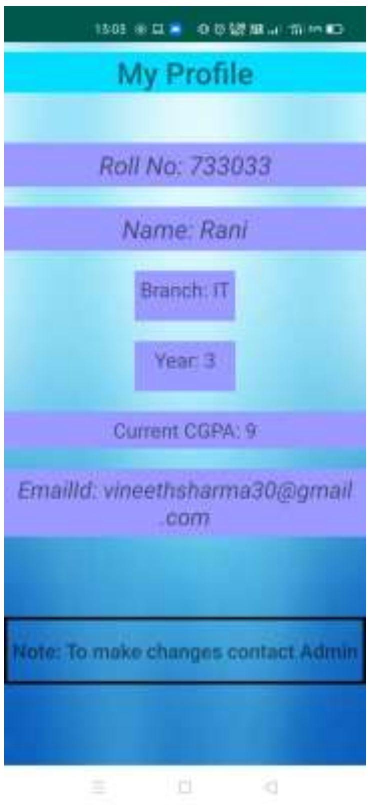
GCD

IEEECBIT

ROBOTICS

already applied

When user clicks apply new button , then the remaining clubs where user is not part in will be displayed and when user clicks that club then it will be applied directly else it shows message as "already applied"


My Profile

Roll No: 733033

Name: Rani

Branch: IT

Year: 3

Current CGPA: 9

EmailId: vineethsharma30@gmail.com

Note: To make changes contact Admin

When user clicks profile button the profile of the user will be displayed as shown

additions in Manifest file:

```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission
android:name="android.permission.ACCESS_NETWORK_STATE"/>
```

Login connection:

```
private static final String url = "jdbc:mysql://skillup-team-
06.cxgok3weok8n.ap-south-
1.rds.amazonaws.com:3306/project?characterEncoding=utf8&useSSL=false&useUni
code=true";
private static final String user = "admin";
private static final String pass = "coscskillup";

Context context;

BackgroundWorker(Context ctx) {
    context = ctx;
}

@Override
protected String doInBackground(String... params) {

    try {
        String user_name = params[0];
        String password = params[1];
        Class.forName("com.mysql.jdbc.Driver");
        Connection con = DriverManager.getConnection(url, user, pass);
        System.out.println("Databasecction success");

        String result = "Database Connection Successful\n";
        PreparedStatement st = con.prepareStatement("select id from
Login where username = ? and password = ?");
        st.setString(1,user_name);
        st.setString(2,password);
        ResultSet rs = st.executeQuery();
       while(rs.next())
       {
           result = rs.getString(1).toString();
       }

        return result;
    } catch (Exception e) {
        e.printStackTrace();
    }

    return null;
}
```

Applying for clubs:

```
 try {
        String id = strings[0];
        String cname = strings[1];
        int stuid = Integer.parseInt(id);
        Class.forName("com.mysql.jdbc.Driver");
        Connection con = DriverManager.getConnection(url, user, pass);
        System.out.println("Databasecction success");
        boolean check = false;
        String result = "Database Connection Successful\n";
        PreparedStatement stmt1 = con.prepareStatement(" select
acceptstatus from student where stuid = ? and clubname=?");
        stmt1.setInt(1, stuid);
        stmt1.setString(2, cname);
        ResultSet rs1 = stmt1.executeQuery();
        if (!rs1.next()) {
            PreparedStatement stmt = con.prepareStatement("insert into
student(stuid,clubname,crole,acceptstatus) values(?,?,'member','0')");
            stmt.setInt(1, stuid);
            stmt.setString(2, cname);
            int out = stmt.executeUpdate();
            if (out == 0)
                return "0";
            else
                return "1";
        } else
            return "2";

    } catch (Exception e) {
        e.printStackTrace();
        return String.valueOf(e);
    }
}
```

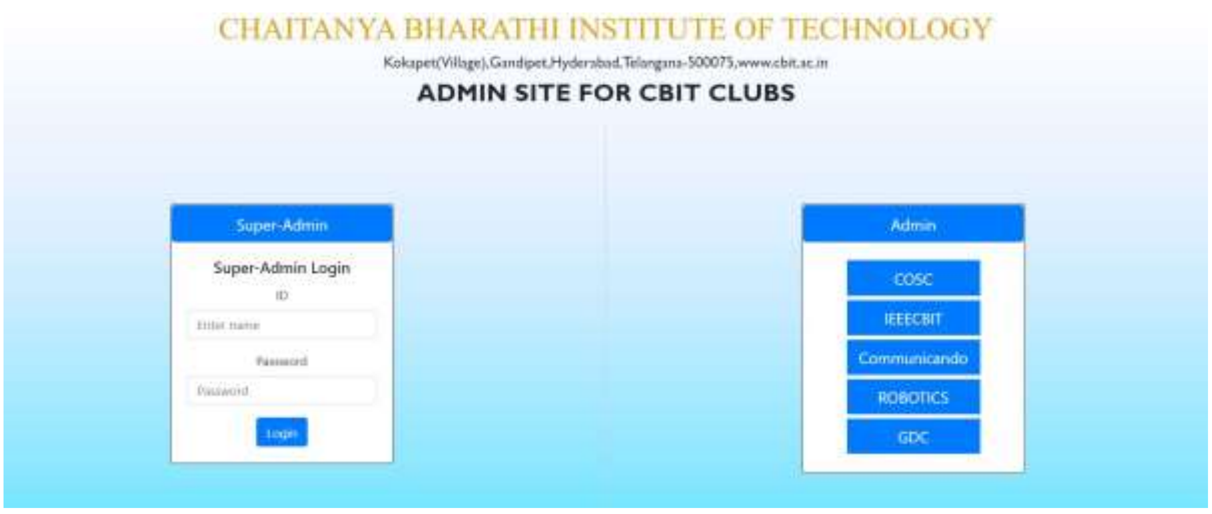
code to open activities of different clubs:

```
protected String doInBackground(String... params) {
    try {
        String tab = params[0];
        String can = params[1];
        int p = Integer.parseInt(can);
        Class.forName("com.mysql.jdbc.Driver");
        Connection con = DriverManager.getConnection(url, user, pass);
        String result = "Database Connection Successful\n";
        PreparedStatement st = con.prepareStatement("select eventname
from `"+tab+"` where stuid = ?");
        st.setInt(1, p);
        ResultSet rs = st.executeQuery();

        List<Map<String, String>> data = null;
        data = new ArrayList<Map<String, String>>();

        while (rs.next()) {
            Map<String, String> datanum = new HashMap<String,
String>();
            datanum.put("A", rs.getString(1).toString());
            data.add(datanum);
        }

        String[] fromwhere = {"A"};
        int[] viewswhere = {R.id.lblcountryname};
        ADAhere = new SimpleAdapter(ActiveActivity.this, data,
                R.layout.listtemplate, fromwhere, viewswhere);

        while (rs.next()) {
            result += rs.getString(1).toString() + "\n";
        }
        res = result;
    } catch (Exception e) {
        e.printStackTrace();
        res = e.toString();
    }
    return res;
}
```

7.5 Web Application

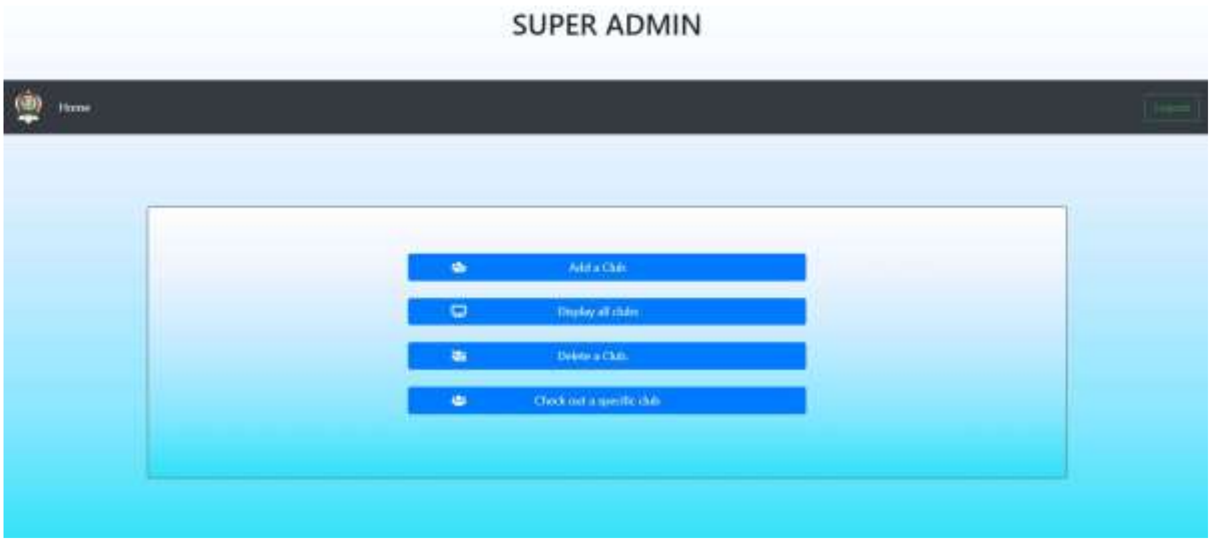
Web app designed have a Super Admin who controls all the Admin activities and Admin who controls respective Club activities and events and members of their respective Club.

Login Page


CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY
Kokapet(Village),Gandipet,Hyderabad,Telangana-500075,www.cbit.ac.in
ADMIN SITE FOR CBIT CLUBS
Super-Admin
Super-Admin Login
ID
Enter name
Password
Password
Login
Admin
COSC
IEEECBIT
Communicando
ROBOTICS
GDC

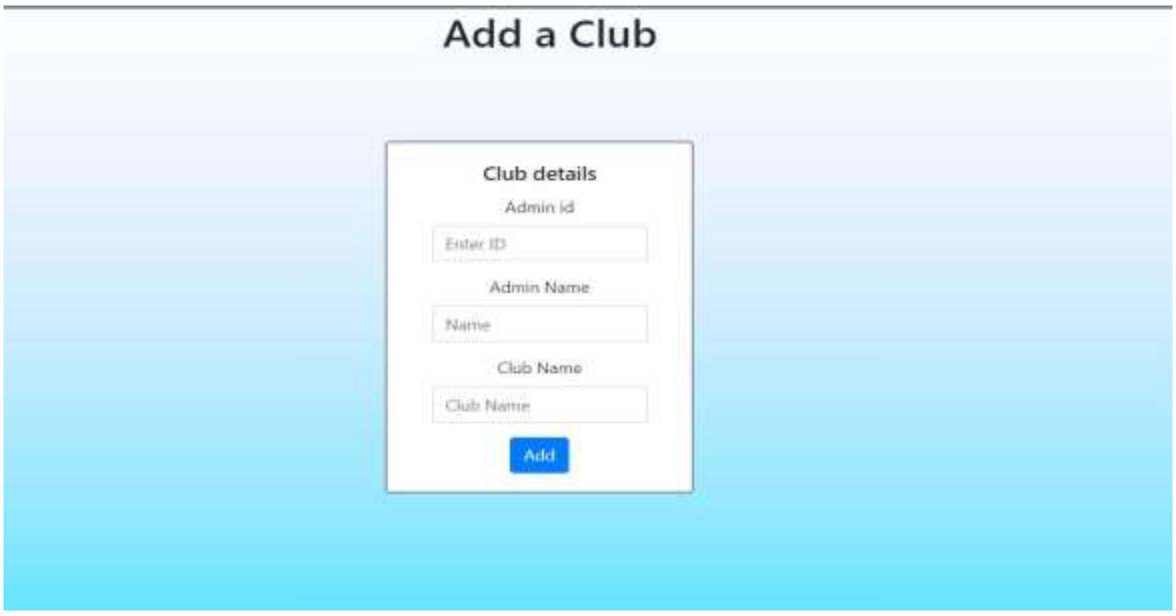
Login page includes Super Admin Login, Clubs by which respective admin can get to their Admin Login Page.

1) Super Admin



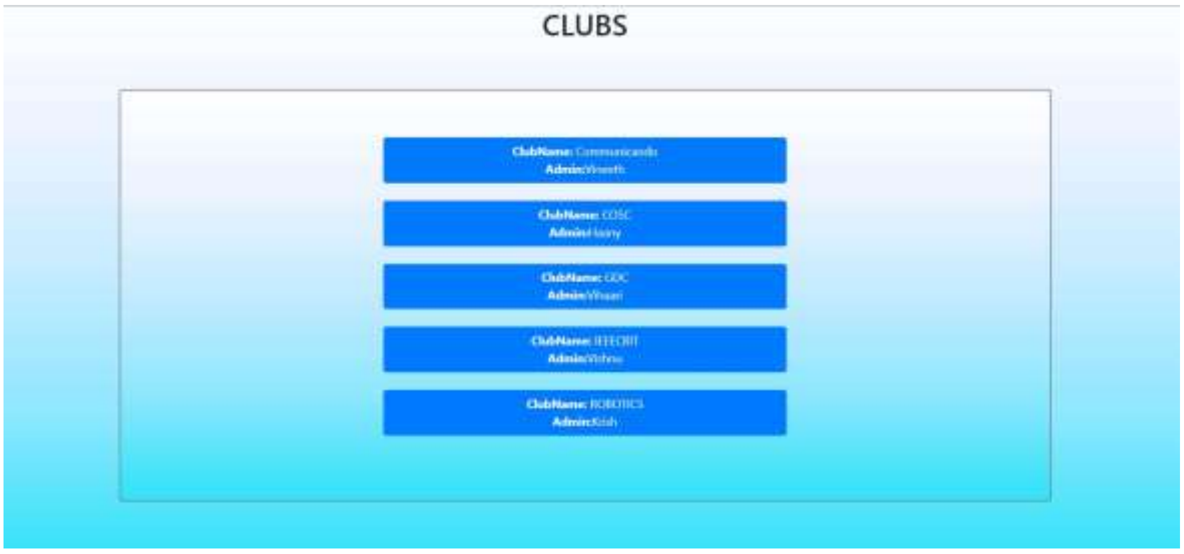
After Super Admin logins with correct credentials, Super Admin will be redirected to Super Admin Page where he can Add a Club, View Admins, Delete a Club, Checkout a Club.

Add a Club



Super Admin can Add a Club by giving Student id, Admin name, Club name. A new Club with the given details will be created. Admin will get a notification to his personal mail, that he was appointed as Club Admin here the admin e-mail id is taken from the Database. E-mail sent to admin contains a Random Password for his Login and he can change password later.

Display all Admins


CLUBS

ClubName: Communicando
Admin: Vineeth

ClubName: COSC
Admin: Harry

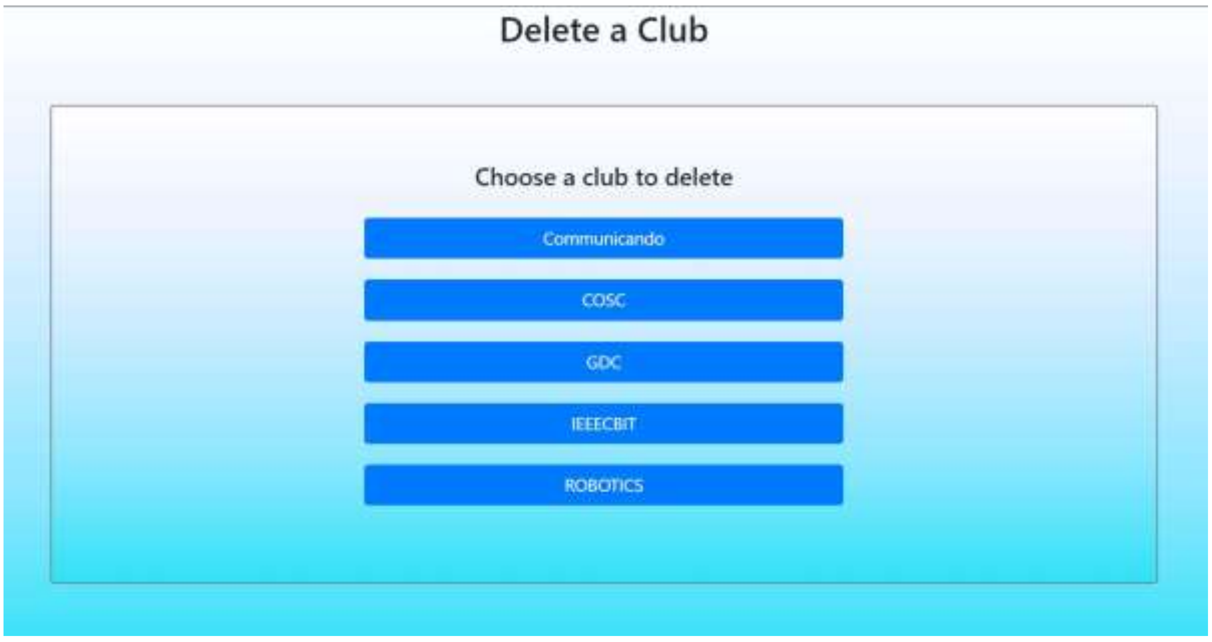
ClubName: GDC
Admin: Vihaari

ClubName: IEEECBIT
Admin: Vishnu

ClubName: ROBOTICS
Admin: Krish

Super Admin can have Club name and Admin name of all the clubs by the Display all Admins functionality.

Delete a Club


Delete a Club

Choose a club to delete

Communicando

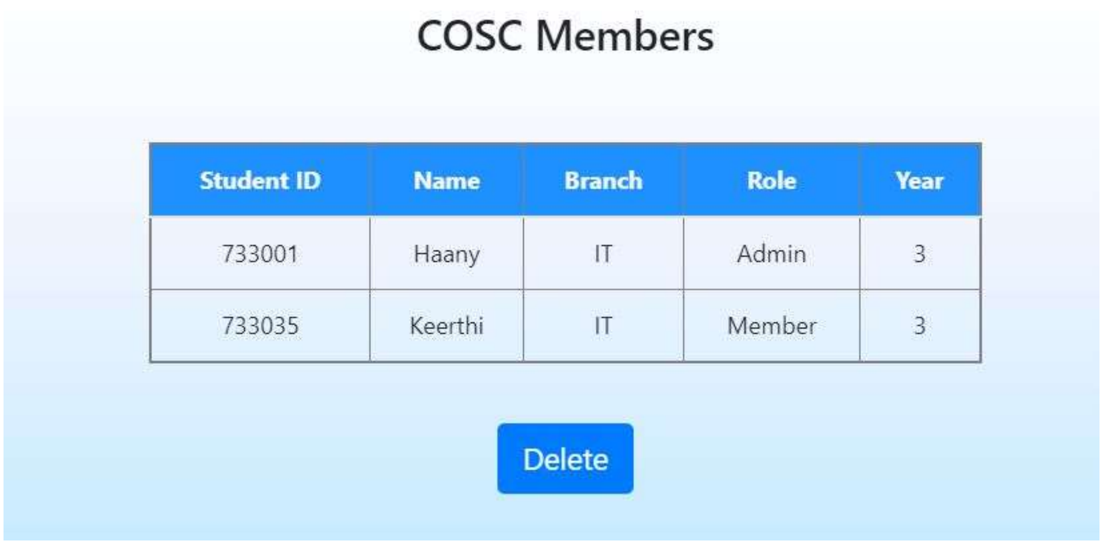
COSC

GDC

IEEECBIT

ROBOTICS

Super Admin can Dismiss a Club by clicking on Delete a Club he will be redirected to a page where the Super Admin need to select a Club to delete.

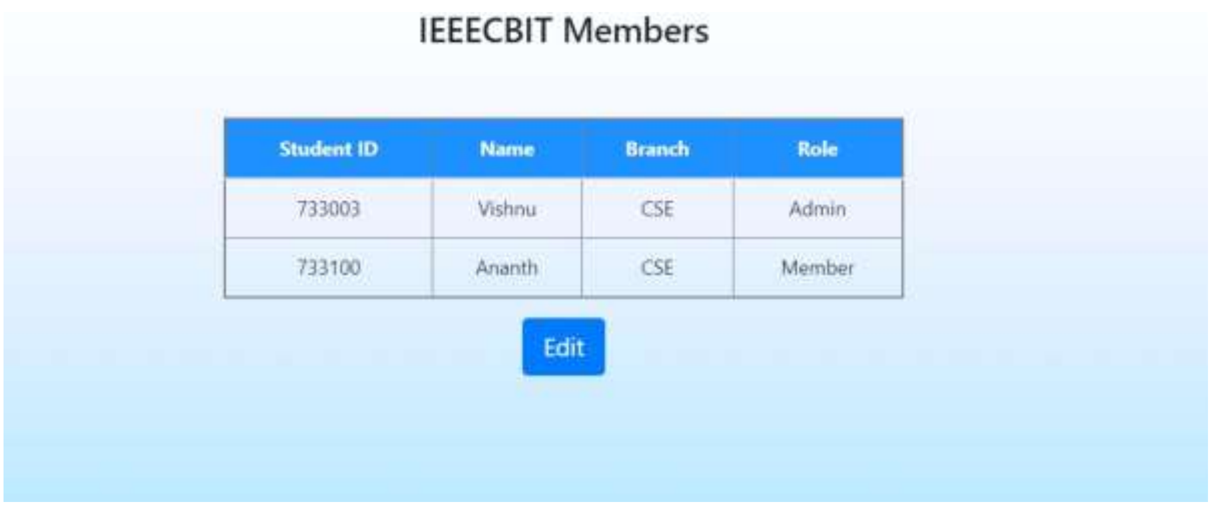


COSC Members

Student ID	Name	Branch	Role	Year
733001	Haany	IT	Admin	3
733035	Keerthi	IT	Member	3

The Confirmation Page appears where it has all the members of Club displayed and then Super Admin can click Confirm which Deletes a Club.
After the Club gets Deleted the respective Club Admin gets an e-mail regarding the removal of Club.\

Checkout a Specific Club



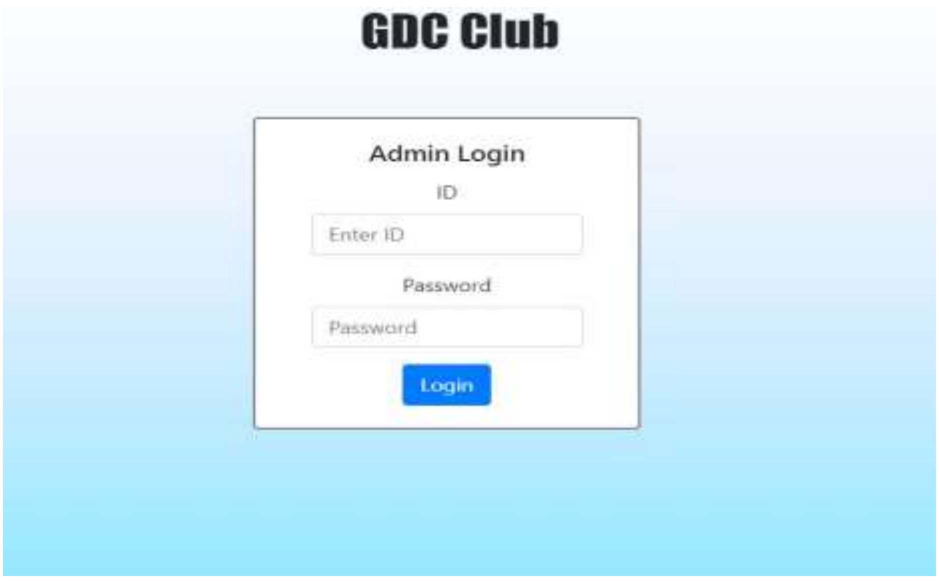
IEEECBIT Members

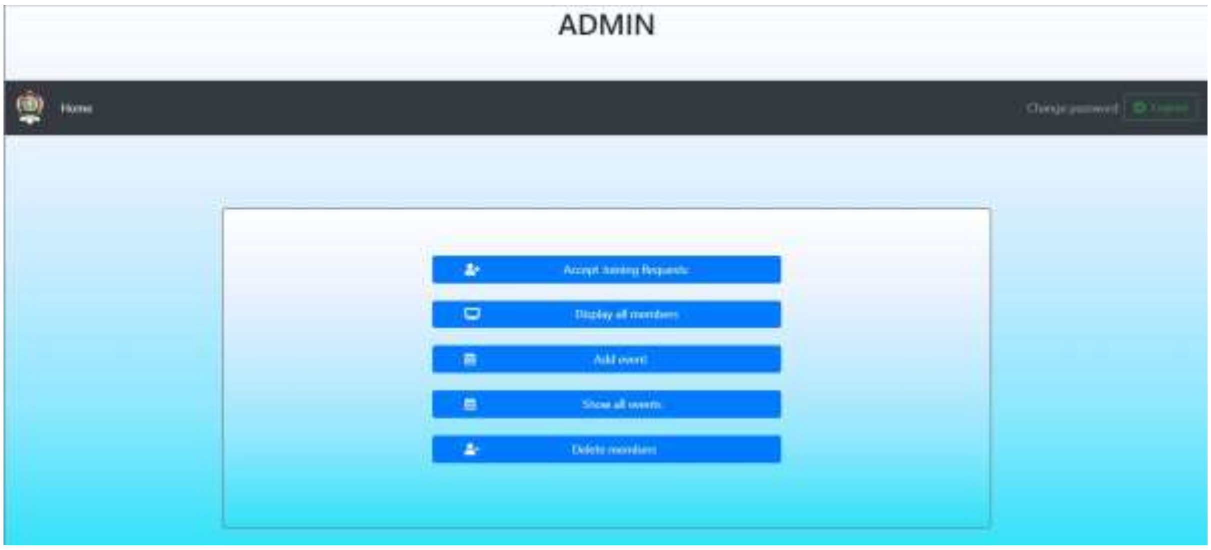
Student ID	Name	Branch	Role
733003	Vishnu	CSE	Admin
733100	Ananth	CSE	Member

Super Admin can click on Checkout a Specific Club where he needs to select a Club to View members of club or Super Admin can make a member of Club as Admin and previous Admin gets demoted to member automatically. For promoting a member to Admin, Super admin need to click on Edit button then he is redirected to Edit Admin page where he needs to give the member Student id to make him Admin.

2) Admin

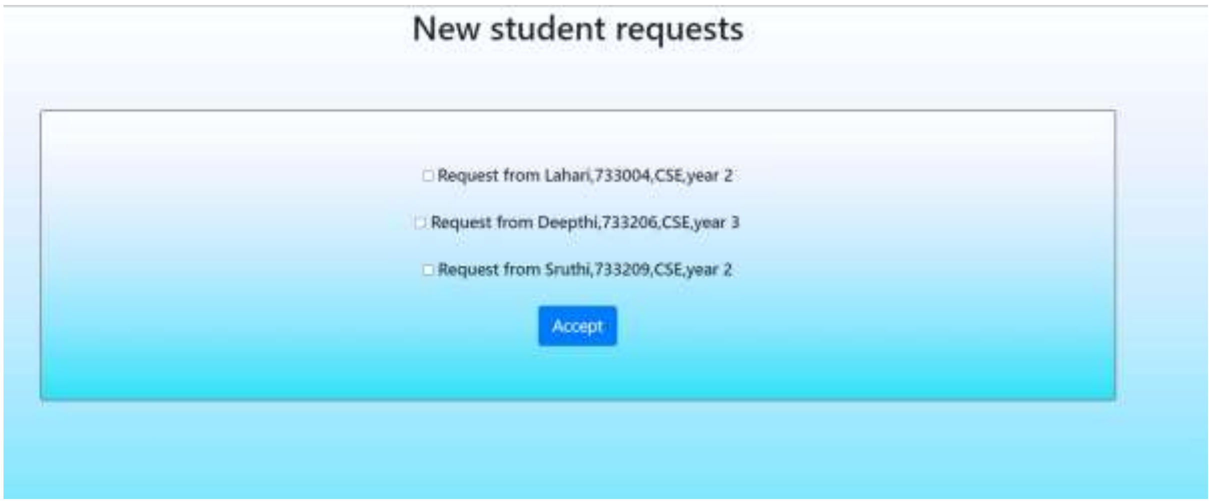
Login





After Admin logins with correct credentials, Admin will be redirected to Admin Page where he can Accept Joining Requests to the Club, View Members of Club, Add an Event, View Events of the Club, View Participants of Club, Delete Members from Club, Change Password.

Accept Joining Requests

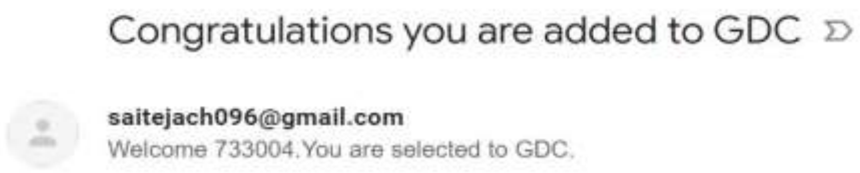

New student requests

☐ Request from Lahari,733004,CSE,year 2

☐ Request from Deepthi,733206,CSE,year 3

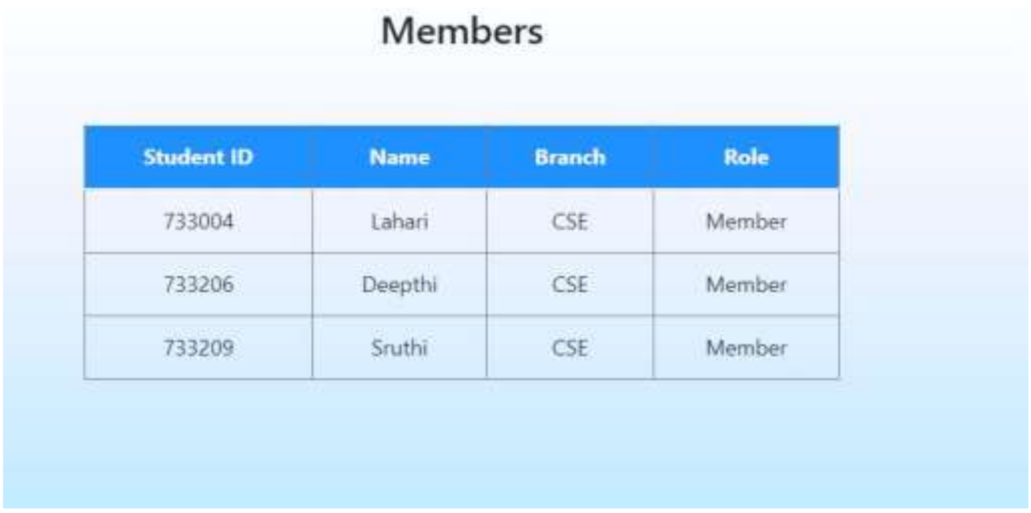
☐ Request from Sruthi,733209,CSE,year 2

Admin of a Club will receive the Joining Requests from students. Admin can accept the requests then students will be notified by an e-mail.


Congratulations you are added to GDC

saitejach096@gmail.com
Welcome 733004.You are selected to GDC.

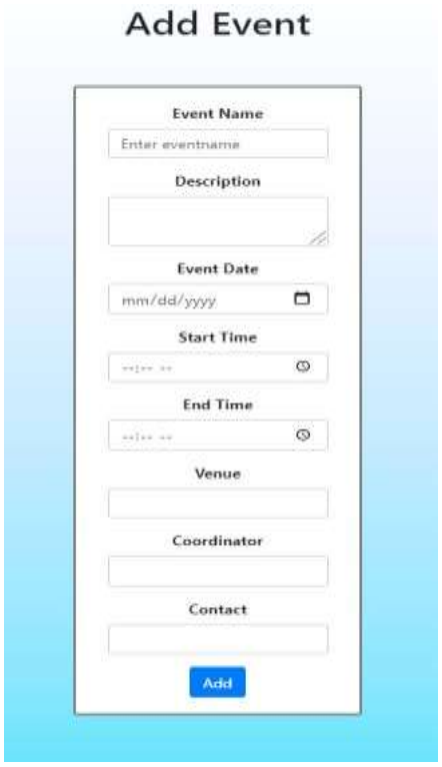
Display All Members


Members

Student ID	Name	Branch	Role
733004	Lahari	CSE	Member
733206	Deepthi	CSE	Member
733209	Sruthi	CSE	Member

Admin can view all the members of the respective Club.

Add an Event


Add Event

Event Name

Enter eventname

Description

Event Date

mm/dd/yyyy

Start Time

--:-- --

End Time

--:-- --

Venue

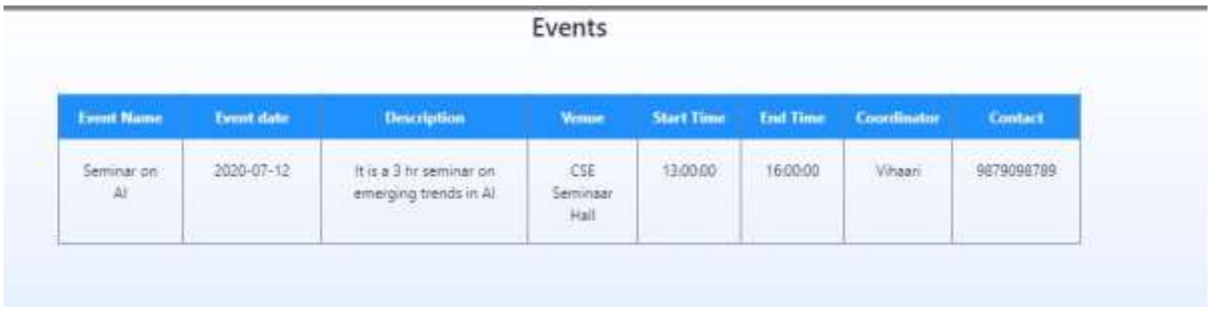
Coordinator

Contact

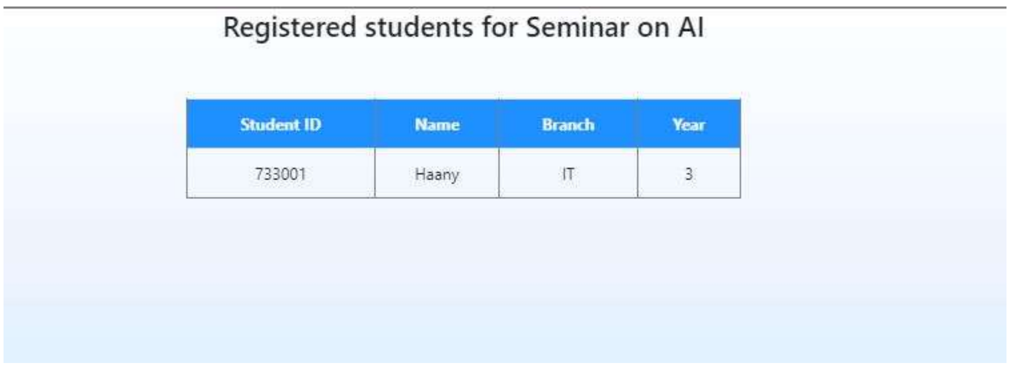
Add

Admin can Add an Event by giving Event name, Venue, Start time, End time, Description of the event, Coordinator, Contact no of the Coordinator.

Show all Events


Events

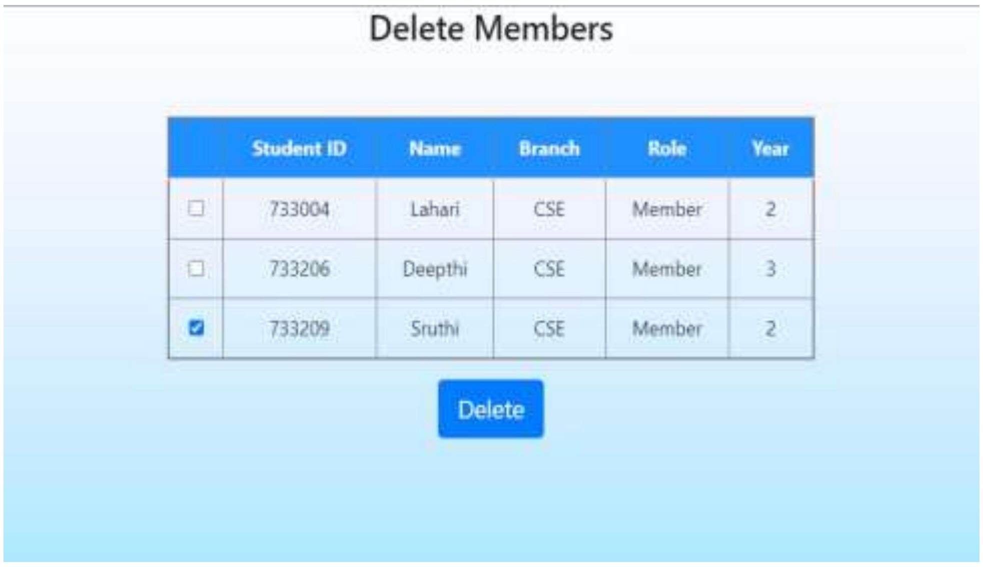
Event Name	Event date	Description	Venue	Start Time	End Time	Coordinator	Contact
Seminar on AI	2020-07-12	It is a 3 hr seminar on emerging trends in AI	CSE Seminaar Hall	13:00:00	16:00:00	Vihaari	9879098789


Registered students for Seminar on AI

Student ID	Name	Branch	Year
733001	Haany	IT	3

Admin can view all Events of the Club, Participants of an Event by clicking on any Event.

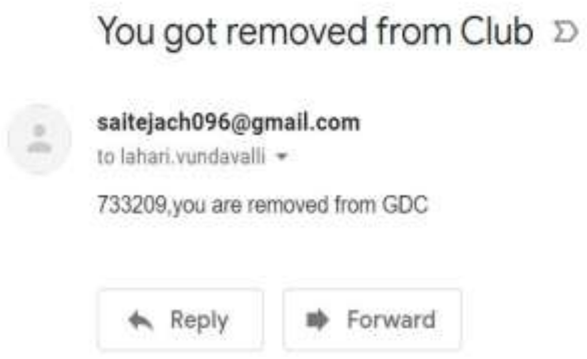
Delete Members



Delete Members

	Student ID	Name	Branch	Role	Year
☐	733004	Lahari	CSE	Member	2
☐	733206	Deepthi	CSE	Member	3
☑	733209	Sruthi	CSE	Member	2

Admin can delete members from the club and an e-mail notification will be sent to them.

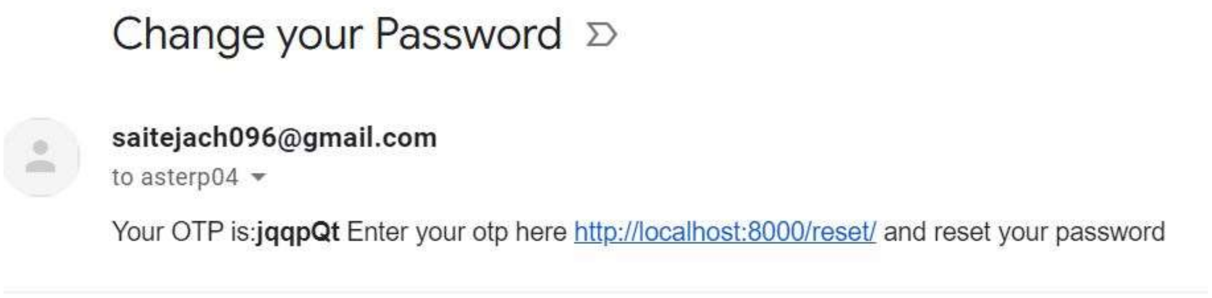


You got removed from Club

saitejach096@gmail.com
to lahari.vundavalli

733209,you are removed from GDC

Change Password

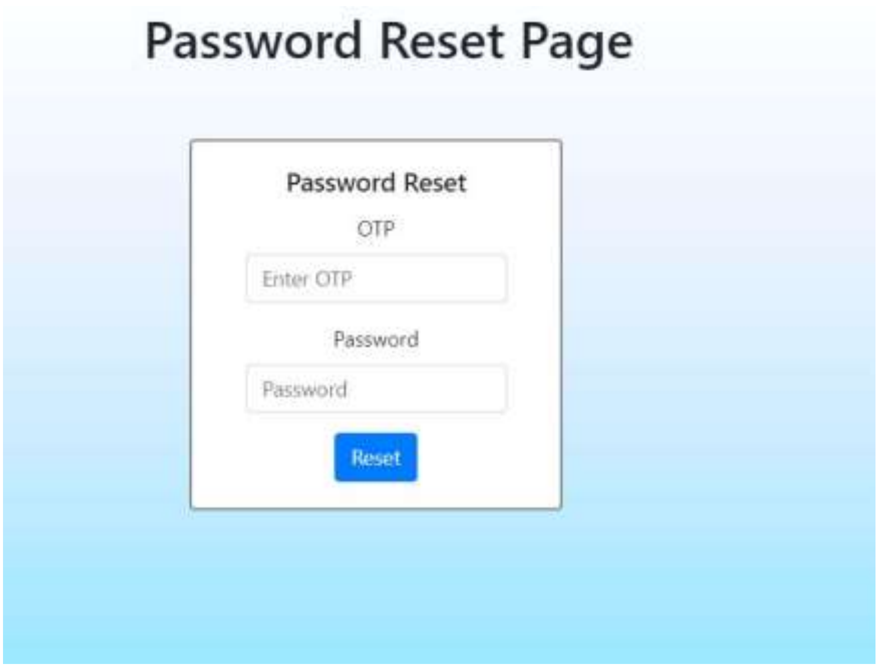


Change your Password

saitejach096@gmail.com
to asterp04

Your OTP is:**jqqpQt** Enter your otp here http://localhost:8000/reset/ and reset your password

Admin can change their Password an e-mail with an OTP, reset password link will be sent to their e-mailid. Admin must enter OTP, New Password where the OTP will be used for Verification.


Password Reset Page
Password Reset
OTP
Enter OTP
Password
Password
Reset

8.Conclusion

This project helps students to view all the clubs and their activities very easily all at a place and take part in the club. This will help students save their time and increase the participation in the clubs. Searching for events of all clubs is very easy by using this instead of searching posters in notice boards or searching in whatsapp. This project helps admins to spread the news of events very easier instead of making announcements in all classes which kills most of their time. The club management becomes much easier because all the details are stored at one place and modifications can be done very easily.

9.Appendices

Section 9.1. Resources

Visit the documentation pages of the technologies used from the links provided below.

- Android documentation: https://developer.android.com/docs
- Mysql documentation: https://dev.mysql.com/doc/
- Django documentation: https://docs.djangoproject.com/en/3.0/
- Flask RESTful documentation: https://flask-restful.readthedocs.io/en/latest/
- Flask JWT extended documentation: https://flask-jwtextended.readthedocs.io/en/stable/
- JWT introduction: https://jwt.io/introduction/