PES UNIVERSITY
Department of Computer Science & Engineering

DBMS - UE20CS301

Mini Project

Students Club Management System

Submitted to :
Dr. Geetha D
Associate Professor

Submitted By :
Name : Sehag A
SRN : PES2UG20CS457
Semester : V
Section : G

Table of Contents

1. Short Description and Scope of the Project

Students clubs management system is an organised system for storing the details of many clubs belonging to an educational institution.It includes data such as :

- ❖ Database on all the students belonging to a club.
- ❖ Database on all the clubs that exist. It include details such as number of students belonging to that particular club, type of club, faculty coordinating that particular club and the student club head for each club.
- ❖ Database on the departments to which each student belongs to and which all clubs are maintained by each department.
- ❖ Database on faculties.This contains information related to the faculty members responsible for clubs and faculties who are head of each department.
- ❖ Database on all events organised by a particular club.

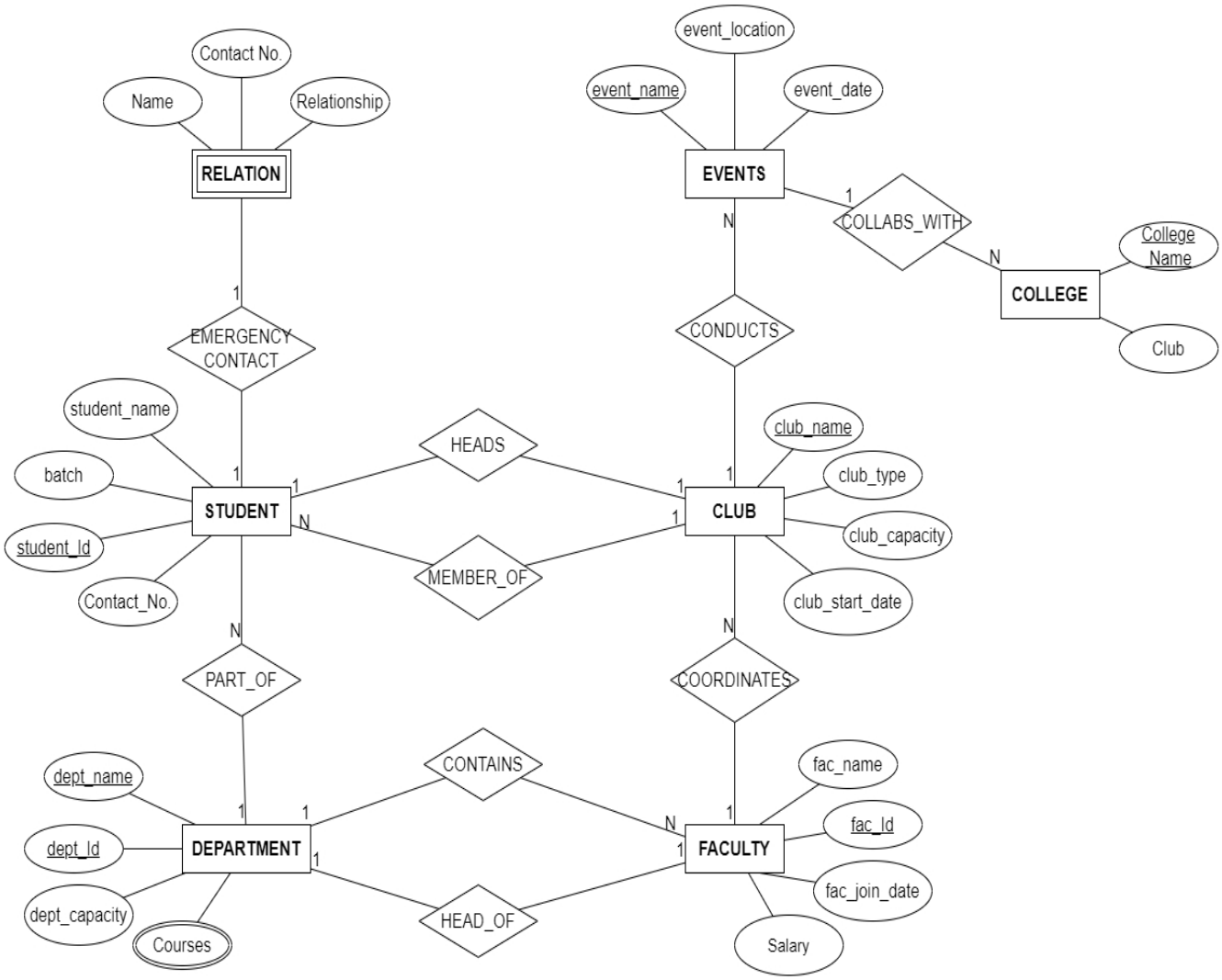
These databases coordinate with each other and store all the information regarding a students clubs management system.
With the help of these databases,we can have many functionalities such as:

1. The education institution can generate a complete report on every club and track their activities.
2. Students can look into events and enroll in their desired club based on their interests.
3. Faculty in-charge of a club can keep record of all the students details(such as name,DOB,SRN,contact.no) belonging to their club.
4. Keep track of all the departments managing the clubs.

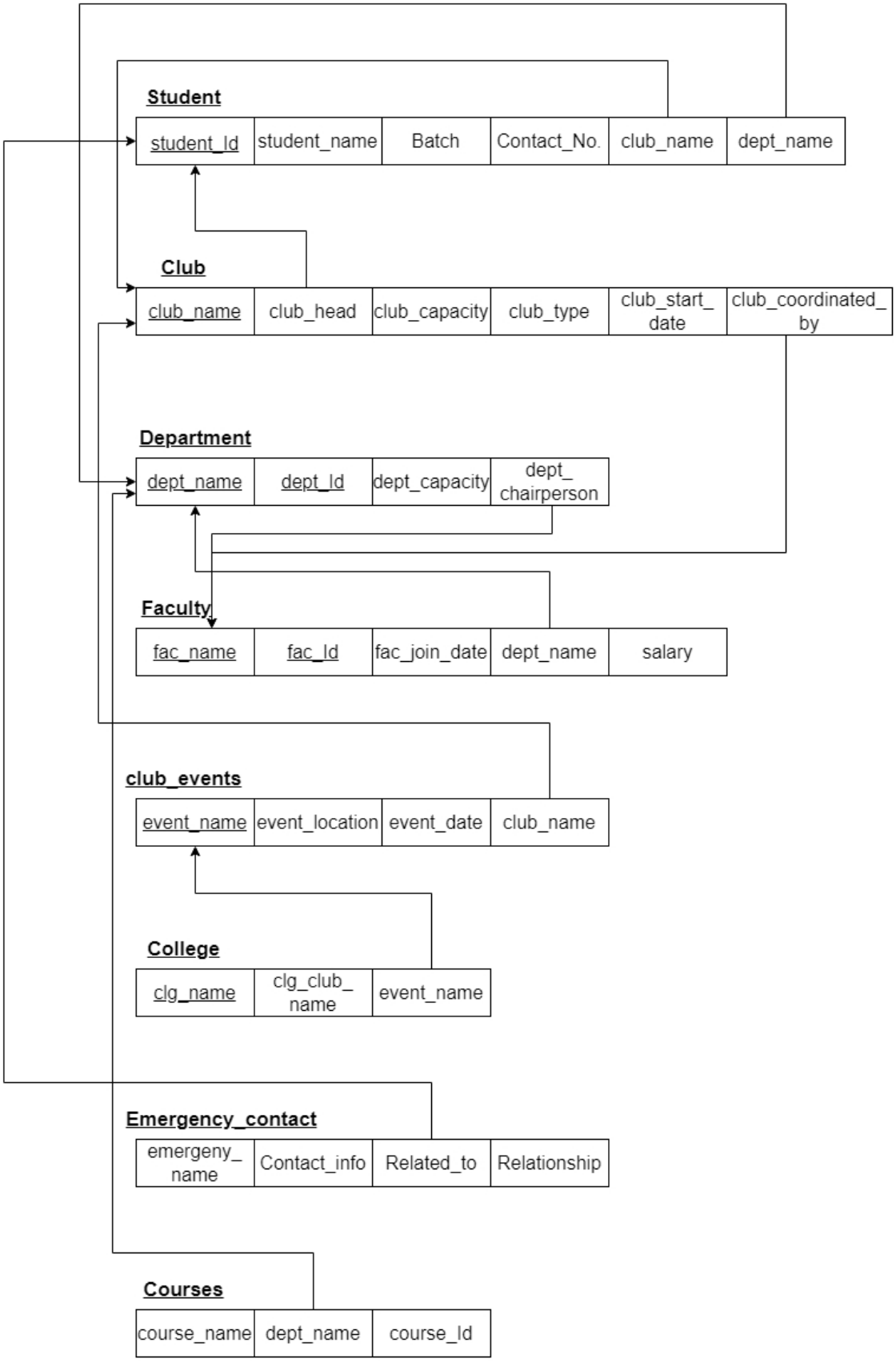
Tools being used:

- MySql for the database to store the data.
- Streamlit for front-end
- Xammp server
- Draw.io

2. ER Diagram


Contact No.
Name
Relationship
RELATION
1
EMERGENCY CONTACT
student_name
batch
student_Id
Contact_No.
STUDENT
1
1
N
HEADS
MEMBER_OF
N
PART_OF
dept_name
dept_Id
dept_capacity
Courses
DEPARTMENT
1
1
1
CONTAINS
HEAD_OF
event_location
event_name
event_date
EVENTS
N
1
CONDUCTS
1
COLLABS_WITH
N
COLLEGE
College Name
Club
club_name
club_type
club_capacity
club_start_date
CLUB
1
1
1
N
COORDINATES
fac_name
fac_Id
fac_join_date
Salary
FACULTY
N
1
1

3. Relational Schema



4. DDL statements – Building the database

i. To create table ‘club’ :

```
CREATE TABLE `club` (
  `club_name` varchar(255) NOT NULL,
  `club_head` varchar(255) DEFAULT NULL,
  `club_capacity` int(11) DEFAULT NULL,
  `club_type` varchar(255) DEFAULT NULL,
  `club_coordinated_by` varchar(255) DEFAULT NULL,
  `club_start_date` date DEFAULT NULL
);

ALTER TABLE `club`
  ADD PRIMARY KEY (`club_name`),
  ADD KEY `Club_head` (`club_head`),
  ADD KEY `Coordinated_by` (`club_coordinated_by`);

ALTER TABLE `club`
  ADD CONSTRAINT `club_ibfk_1` FOREIGN KEY (`Club_head`) REFERENCES `student` (`student_Id`),
  ADD CONSTRAINT `club_ibfk_2` FOREIGN KEY (`club_coordinated_by`) REFERENCES `faculty` (`fac_name`);
```

ii. To create table ‘club_events’ :

```
CREATE TABLE `club_events` (
  `event_name` varchar(255) NOT NULL,
  `event_location` varchar(255) DEFAULT NULL,
  `event_date` date DEFAULT NULL,
  `club_name` varchar(255) DEFAULT NULL
);

ALTER TABLE `club_events`
  ADD PRIMARY KEY (`event_name`),
  ADD KEY `club` (`club_name`);

ALTER TABLE `club_events`
  ADD CONSTRAINT `club_events_ibfk_1` FOREIGN KEY (`club_name`) REFERENCES `club` (`club_name`);
```

iii. To create table ‘college’ :

```
CREATE TABLE `college` (
  `clg_name` varchar(255) NOT NULL,
  `clg_club_name` varchar(255) DEFAULT NULL,
  `event_name` varchar(255) DEFAULT NULL
);

ALTER TABLE `college`
  ADD PRIMARY KEY (`clg_name`),
  ADD KEY `event_name` (`event_name`);

ALTER TABLE `college`
```

```
 ADD CONSTRAINT `college_ibfk_1` FOREIGN KEY (`event_name`) REFERENCES `club_events` (`event_name`);
```

iv. To create table ‘courses’ :

```
CREATE TABLE `courses` (
  `course_name` varchar(255) DEFAULT NULL,
  `course_Id` varchar(255) NOT NULL,
  `dept_name` varchar(255) DEFAULT NULL
);

ALTER TABLE `courses`
  ADD PRIMARY KEY (`course_Id`),
  ADD KEY `Dept` (`dept_name`);

ALTER TABLE `courses`
  ADD CONSTRAINT `courses_ibfk_1` FOREIGN KEY (`dept_name`) REFERENCES `department` (`dept_name`);
```

v. To create table ‘department’ :

```
CREATE TABLE `department` (
  `dept_name` varchar(255) NOT NULL,
  `dept_ID` varchar(11) DEFAULT NULL,
  `dept_capacity` int(11) DEFAULT NULL,
  `dept_chairperson` varchar(255) DEFAULT NULL
);

ALTER TABLE `department`
  ADD PRIMARY KEY (`dept_name`),
  ADD UNIQUE KEY `ID` (`dept_ID`),
  ADD KEY `Chairperson` (`dept_chairperson`);

ALTER TABLE `department`
  ADD CONSTRAINT `department_ibfk_1` FOREIGN KEY (`dept_chairperson`) REFERENCES `faculty` (`fac_name`);
```

vi. To create table ‘emergency_contact’ :

```
CREATE TABLE `emergency_contact` (
  `emergency_name` varchar(255) DEFAULT NULL,
  `contact_info` bigint(10) NOT NULL,
  `relationship` varchar(255) DEFAULT NULL,
  `related_to` varchar(13) DEFAULT NULL
);

ALTER TABLE `emergency_contact`
  ADD KEY `key` (`related_to`),
  ADD KEY `related_to` (`related_to`);

ALTER TABLE `emergency_contact`
  ADD CONSTRAINT `emergency_contact_ibfk_1` FOREIGN KEY (`related_to`) REFERENCES `student` (`student_Id`);
```

vii. To create table ‘faculty’ :

```
CREATE TABLE `faculty` (
  `fac_name` varchar(255) NOT NULL,
  `fac_Id` varchar(11) NOT NULL,
  `fac_join_date` date DEFAULT NULL,
  `dept_name` varchar(255) DEFAULT NULL,
  `salary` int(11) NOT NULL
);

ALTER TABLE `faculty`
  ADD PRIMARY KEY (`fac_Id`),
  ADD UNIQUE KEY `Name` (`fac_name`),
  ADD KEY `Department` (`dept_name`);

ALTER TABLE `faculty`
  ADD CONSTRAINT `faculty_ibfk_1` FOREIGN KEY (`dept_name`) REFERENCES `department` (`dept_name`);
```

viii. To create table 'student' :

```
CREATE TABLE `student` (
  `student_Id` varchar(13) NOT NULL,
  `student_name` varchar(255) DEFAULT NULL,
  `batch` int(4) DEFAULT NULL,
  `contact_no` bigint(10) DEFAULT NULL,
  `club_name` varchar(255) DEFAULT NULL,
  `dept_name` varchar(255) DEFAULT NULL
);

ALTER TABLE `student`
  ADD PRIMARY KEY (`student_Id`),
  ADD KEY `Club` (`club_name`),
  ADD KEY `Dept` (`dept_name`);

ALTER TABLE `student`
  ADD CONSTRAINT `student_ibfk_1` FOREIGN KEY (`club_name`) REFERENCES `club` (`club_name`),
  ADD CONSTRAINT `student_ibfk_2` FOREIGN KEY (`dept_name`) REFERENCES `department` (`dept_name`);
```

5. Populating the Database

i. To insert into table ‘club’ :

```
INSERT INTO `club` (`club_name`, `club_head`, `club_capacity`, `club_type`, `club_coordinated_by`,
`club_start_date`) VALUES
('HackerSpace', 'PESSTU009', 2, 'Technical', 'Ross Geller', '2018-06-11'),
('Hashtag', 'PESSTU004', 3, 'Cultural', 'Raymond Holt', '2014-08-13'),
('Pixelloid', 'PESSTU006', 2, 'Photography', 'Amy', '2019-12-25'),
('Shunya', 'PESSTU002', 1, 'Mathematical', 'Natasha', '2015-04-09'),
('Swarantraka', 'PESSTU007', 2, 'Singing', 'Tsunade', '2016-08-30'),
('Zerospace', 'PESSTU003', 1, 'Technical', 'Kakashi', '2020-10-18');
```

ii. To insert into table ‘club_events’ :

```
INSERT INTO `club_events` (`event_name`, `event_location`, `event_date`, `club_name`) VALUES
('Aatmatrisha', 'Ring road', '2022-04-16', 'Shunya'),
('Avions', 'Yelahanka', '2022-02-25', 'Zerospace'),
('In-genius', 'Ring road', '2022-09-10', 'HackerSpace'),
('Maaya', 'Electronic City', '2022-11-12', 'Hashtag'),
('Shaken and stirred', 'Jayanagar', '2022-07-17', 'Swarantraka'),
('What-a-shot', 'Lal bagh', '2022-12-25', 'Pixelloid');
```

iii. To insert into table ‘college’ :

```
INSERT INTO `college` (`clg_name`, `clg_club_name`, `event_name`) VALUES
('BMS', 'tech', 'In-genius'),
('DSCE', 'singing', 'Shaken and stirred'),
('JSS', 'math', 'Maaya'),
('MSR', 'space', 'Avions'),
('RNSIT', 'dance', 'Maaya'),
('RVCE', 'photography', 'What-a-shot'),
('RVITM', 'hacking', 'In-genius');
```

iv. To insert into table ‘courses’ :

```
INSERT INTO `college` (`clg_name`, `clg_club_name`, `event_name`) VALUES
('BMS', 'tech', 'In-genius'),
('DSCE', 'singing', 'Shaken and stirred'),
('JSS', 'math', 'Maaya'),
('MSR', 'space', 'Avions'),
('RNSIT', 'dance', 'Maaya'),
('RVCE', 'photography', 'What-a-shot'),
('RVITM', 'hacking', 'In-genius');
```

v. To insert into table ‘department’ :

```
INSERT INTO `department` (`dept_name`, `dept_ID`, `dept_capacity`, `dept_chairperson`) VALUES
('Aeronautic engineering', 'PES005', 2, 'Professor Proton'),
('Civil Engineerinng', 'PES004', 2, 'Amy'),
('Computer Science Engineering', 'PES001', 2, 'Walter'),
('Electronics and Communication Engineering', 'PES002', 2, 'Ross Geller'),
('Mechanical Engineering', 'PES003', 2, 'Dr.stone');
```

vi. To insert into table ‘emergency_contact’ :

```
INSERT INTO `emergency_contact` (`emergency_name`, `contact_info`, `relationship`, `related_to`) VALUES
('Steve Rogers', 9564210654, 'Friend', 'PESSTU001'),
('Daenarys', 6178161801, 'Aunt', 'PESSTU002'),
('Missy', 7113220045, 'Sister', 'PESSTU003'),
('Joey', 8245007441, 'Friend', 'PESSTU004'),
('Sasuke', 6874650400, 'Brother', 'PESSTU005'),
('Jane', 7984804564, 'Friend', 'PESSTU006'),
('Ace', 9780004411, 'Brother', 'PESSTU007'),
('Charles ', 9870048940, 'Brother', 'PESSTU008'),
('Kushina', 8797070701, 'Mother', 'PESSTU009'),
('Mikkel Kahnwald', 6456006451, 'Father', 'PESSTU010');
```

vii. To insert into table ‘faculty’ :

```
INSERT INTO `faculty` (`fac_name`, `fac_Id`, `fac_join_date`, `dept_name`, `salary`) VALUES
('Dr.stone', 'PESFAC001', '2012-10-01', 'Mechanical Engineering', 100000),
('Natasha', 'PESFAC002', '2019-07-24', 'Electronics and Communication Engineering', 90000),
('Kakashi', 'PESFAC003', '2012-01-17', 'Computer Science Engineering', 120000),
('Amy', 'PESFAC004', '2021-12-15', 'Civil Engineerinng', 85000),
('Walter', 'PESFAC005', '2010-11-24', 'Computer Science Engineering', 150000),
('Professor Proton', 'PESFAC006', '2018-09-12', 'Aeronautic engineering', 105000),
('Raymond Holt', 'PESFAC007', '2021-12-15', 'Mechanical Engineering', 94000),
('Ross Geller', 'PESFAC008', '2020-08-10', 'Electronics and Communication Engineering', 90000),
('Jiraiya', 'PESFAC009', '2019-02-20', 'Aeronautic engineering', 98000),
('Tsunade', 'PESFAC010', '2022-06-11', 'Civil Engineerinng', 120106);
```

viii. To insert into table ‘student’ :

```
INSERT INTO `student` (`student_Id`, `student_name`, `batch`, `contact_no`, `club_name`, `dept_name`) VALUES
('PESSTU001', 'Tony stark', 2018, 9554234955, 'Hashtag', 'Computer Science Engineering'),
('PESSTU002', 'Jon snow', 2019, 8322111156, 'Shunya', 'Civil Engineerinng'),
('PESSTU003', 'Sheldon', 2019, 7156483213, 'Zerospace', 'Electronics and Communication Engineering'),
('PESSTU004', 'Chandler', 2019, 7645981652, 'Hashtag', 'Electronics and Communication Engineering'),
('PESSTU005', 'Itachi', 2019, 9659883213, 'Hashtag', 'Mechanical Engineering'),
('PESSTU006', 'Jesse Pinkman', 2020, 6760974501, 'Pixelloid', 'Computer Science Engineering'),
('PESSTU007', 'Luffy', 2020, 9814146544, 'Swarantraka', 'Civil Engineerinng'),
('PESSTU008', 'Jake ', 2020, 7168618682, 'Swarantraka', 'Electronics and Communication Engineering'),
('PESSTU009', 'Naruto', 2020, 6575768450, 'HackerSpace', 'Aeronautic engineering'),
('PESSTU010', 'Jonas Kahnwald', 2020, 7050365600, 'Pixelloid', 'Aeronautic engineering'),
('PESSTU011', 'Chota Bheem', 2019, 6624040461, 'HackerSpace', 'Aeronautic engineering');
```

6. Join Queries

i. Find the emergency contact of students :

```
SELECT student.student_name, emergency_contact.emergency_name, emergency_contact.relation
ship, emergency_contact.contact_info
FROM student
INNER JOIN emergency_contact ON emergency_contact.related_to = student.student_Id;
```

student_name	emergency_name	relationship	contact_info
Tony stark	Steve Rogers	Friend	9564210654
Jon snow	Daenarys	Aunt	6178161801
Sheldon	Missy	Sister	7113220045
Chandler	Joey	Friend	8245007441
Itachi	Sasuke	Brother	6874650400
Jesse Pinkman	Jane	Friend	7984804564
Luffy	Ace	Brother	9780004411
Jake	Charles	Brother	9870048940
Naruto	Kushina	Mother	8797070701
Jonas Kahnwald	Mikkel Kahnwald	Father	6456006451

ii. Find colleges collabrating with the events :

```
SELECT club_events.event_name,college.clg_name
FROM club_events NATURAL JOIN college;
```

event_name	clg_name
In-genius	BMS
In-genius	RVITM
Maaya	JSS
Maaya	RNSIT
What-a-shot	RVCE
Shaken and stirred	DSCE
Avions	MSR

iii. Find salary of each chairperson of a department :

```
SELECT department.dept_name,faculty.salary as chairperson_salary
FROM department
LEFT JOIN faculty on faculty.fac_name = department.dept_chairperson;
```

dept_name	chairperson_salary
Civil Engineerinng	85000
Mechanical Engineering	100000
Aeronautic engineering	105000
Electronics and Communication Engineering	90000
Computer Science Engineering	150000

iv. Find the coordinator of each event :

```
SELECT club_events.event_name,club.club_coordinated_by AS 'Coordinator'
FROM club_events NATURAL JOIN club;
```

event_name	Coordinator
In-genius	Ross Geller
Maaya	Raymond Holt
What-a-shot	Amy
Aatmatrisha	Natasha
Shaken and stirred	Tsunade
Avions	Kakashi

7. Aggregate Functions

i. Find avg salary of each dept :

```
SELECT faculty.dept_name,AVG(salary) as 'Average salary'
FROM faculty GROUP BY faculty.dept_name;
```

	dept_name	Average salary
Edit Copy Delete	Aeronautic engineering	101500.0000
Edit Copy Delete	Civil Engineerinng	102553.0000
Edit Copy Delete	Computer Science Engineering	135000.0000
Edit Copy Delete	Electronics and Communication Engineering	90000.0000
Edit Copy Delete	Mechanical Engineering	97000.0000

ii. Find no. of events by each club :

```
SELECT club_name, COUNT(*) as 'No. of events'
FROM club_events
GROUP BY club_events.club_name;
```

club_name	No. of events
HackerSpace	1
Hashtag	1
Pixelloid	1
Shunya	1
Swarantraka	1
Zerospace	1

iii. Find faculty with most experience :

```
SELECT *
FROM faculty
WHERE fac_join_date = (SELECT MIN(faculty.fac_join_date) FROM faculty);
```

	fac_name	fac_Id	fac_join_date	dept_name	salary
Edit Copy Delete	Walter	PESFAC005	2010-11-24	Computer Science Engineering	150000

iv. Calculate number of students in each club and arrange in descending order :

```
SELECT club_name,COUNT(*) as 'No. of students'
FROM student
GROUP BY student.club_name ORDER BY COUNT(*) DESC;
```

club_name	No. of students
Hashtag	3
HackerSpace	2
Pixelloid	2
Swarantraka	2
Shunya	1
Zerospace	1

8. Set Operations

i. Find all faculty who are not coordinating any club :

```
SELECT faculty.fac_name
FROM faculty
WHERE faculty.fac_name NOT IN(SELECT club.club_coordinated_by FROM club);
```

←T→	fac_name
Edit Copy Delete	Dr.stone
Edit Copy Delete	Jiraiya
Edit Copy Delete	Professor Proton
Edit Copy Delete	Walter

ii. Find all faculties who are chairperson of a dept and also coordinate a club :

```
SELECT department.dept_chairperson AS 'Faculty'
FROM department
WHERE EXISTS(SELECT club.club_coordinated_by FROM club WHERE club.club_coordinated_by=de
partment.dept_chairperson);
```

←T→	Faculty
Edit Copy Delete	Amy
Edit Copy Delete	Ross Geller

iii. Find all students who are club heads and belong to ECE department :

```
SELECT club.club_head
FROM club
WHERE club.club_head IN (SELECT student.student_Id FROM student WHERE student.dept_name =
'Electronics and Communication Engineering');
```

←T→	club_head
Edit Copy Delete	PESSTU003
Edit Copy Delete	PESSTU004

iv. Find all events which do not collabrate with any other colleges :

```
SELECT club_events.event_name
FROM club_events
LEFT OUTER JOIN college USING(event_name) WHERE college.event_name IS NUll;
```

event_name
Aatmatrisha

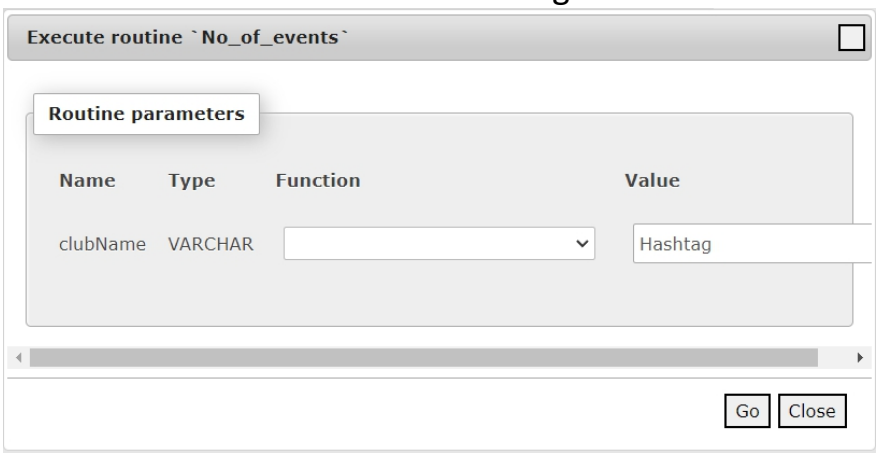
9. Functions and Procedures

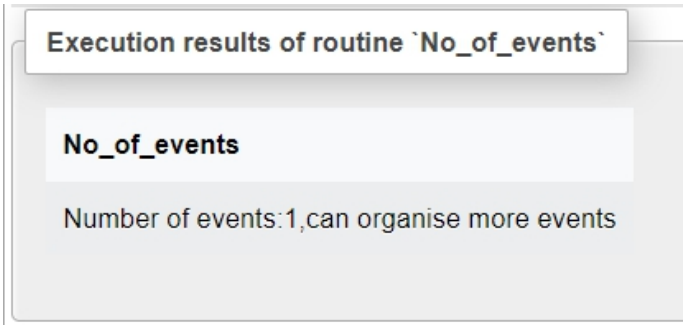
i. Function :

Check the number of events organised by a club and alert if the number of events exceeds 2.

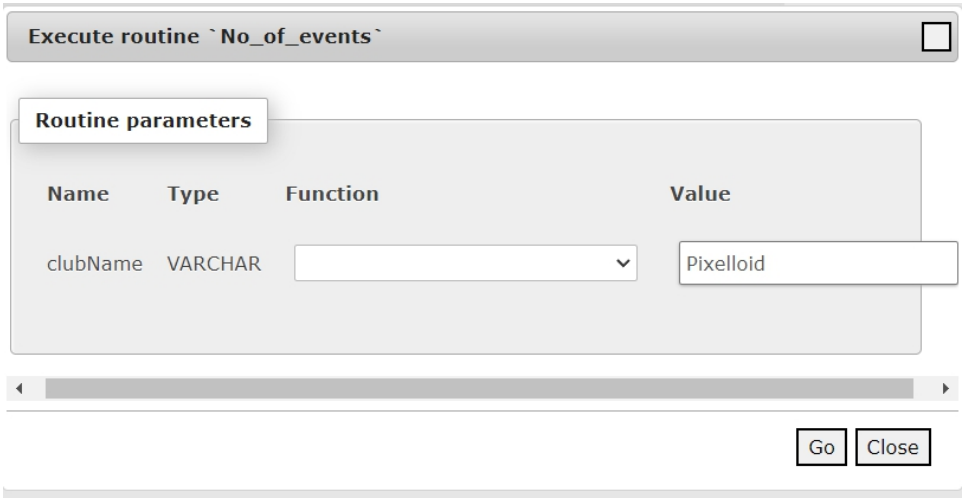
```
DELIMITER $$
CREATE FUNCTION No_of_events(clubName varchar(255))
RETURNS varchar(255)
DETERMINISTIC
BEGIN
        DECLARE message varchar(255) ;
        DECLARE no_of_events int;
        SET no_of_events = (SELECT COUNT(*) from club_events where
clubName=club_events.club_name);
        IF no_of_events>2 THEN
                SET message = Concat('Number of events:',convert(no_of_events,char),',cannot
organise more events');
        ELSE
        SET message = Concat('Number of events:',convert(no_of_events,char),',can organise more
events');
        END IF;
        RETURN message;
END $$
```

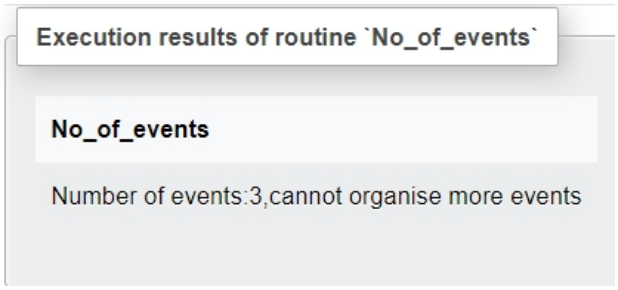
❖ Let's check for a club who has organised 1 event -





❖ Let's check for a club who has organised 3 events -



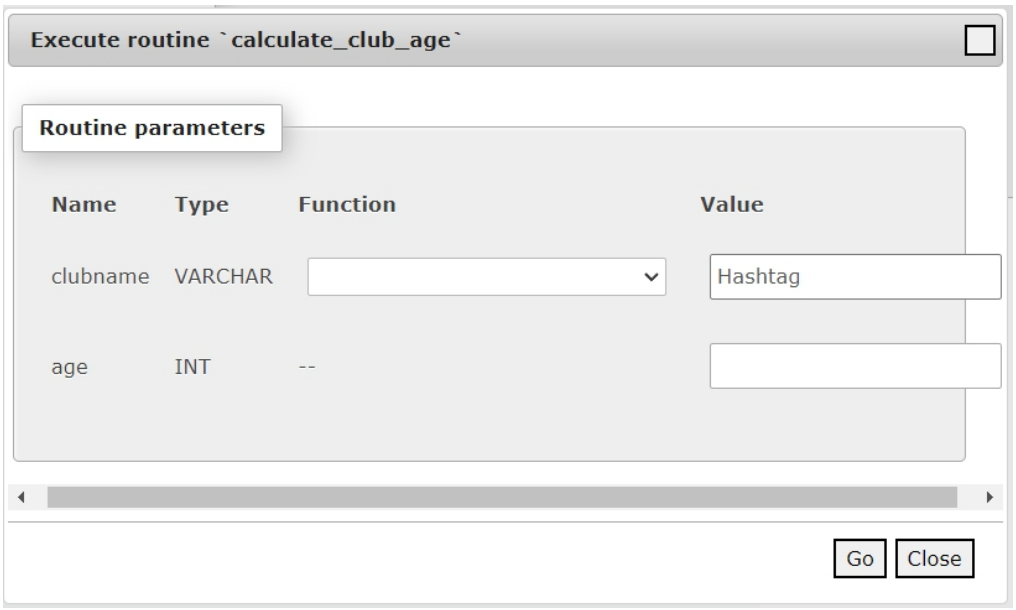


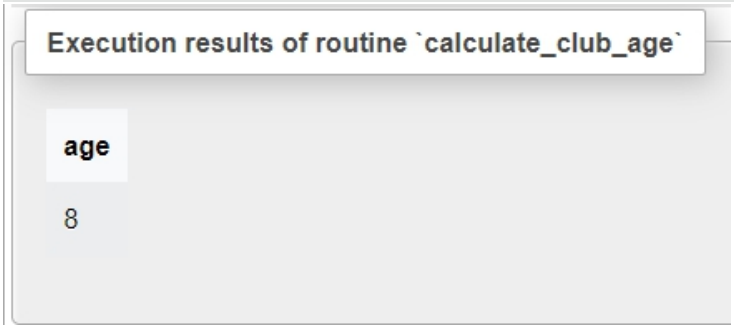
ii. Stored procedure :

Calculate the age of a club.

```
DELIMITER $$
CREATE PROCEDURE calculate_club_age(IN clubname varchar(255),out age int)
BEGIN
        DECLARE uid varchar(255);
  DECLARE start_date date;
  SET uid = (SELECT club_name FROM club WHERE clubname = club.club_name);
  SET start_date = (SELECT club_start_date FROM club WHERE club.club_name=uid);
  SET age = (SELECT datediff(CURRENT_DATE(),start_date)/365);
  IF uid != NUll THEN
```

```
        UPDATE club
    SET Age = @age WHERE club_name=uid;
        END IF;
END;$$
DELIMITER ;
```





10. Triggers and Cursors

i. Trigger

Update the capacity of the club on adding a new student to the club.

```
CREATE TRIGGER updateClubCapacity
AFTER INSERT ON student
FOR EACH ROW BEGIN UPDATE club
SET club.club_capacity = club.club_capacity + 1
WHERE club.club_name = New.club_name;
END;
```

Before adding a student to the club :

club_name	club_head	club_capacity	club_type	club_coordinated_by	club_start_date
HackerSpace	PESSTU009	2	Technical	Ross Geller	2018-06-11

Adding a new student to club 'HackerSpace' :

```
INSERT INTO `student` (`student_Id`, `student_name`, `batch`, `contact_no`, `club_name`, `dept_name`) VALUES ('PESSTU012', 'Itadori', '2019', '7860401210', 'HackerSpace', 'Computer Science Engineering');
```

After insertion :

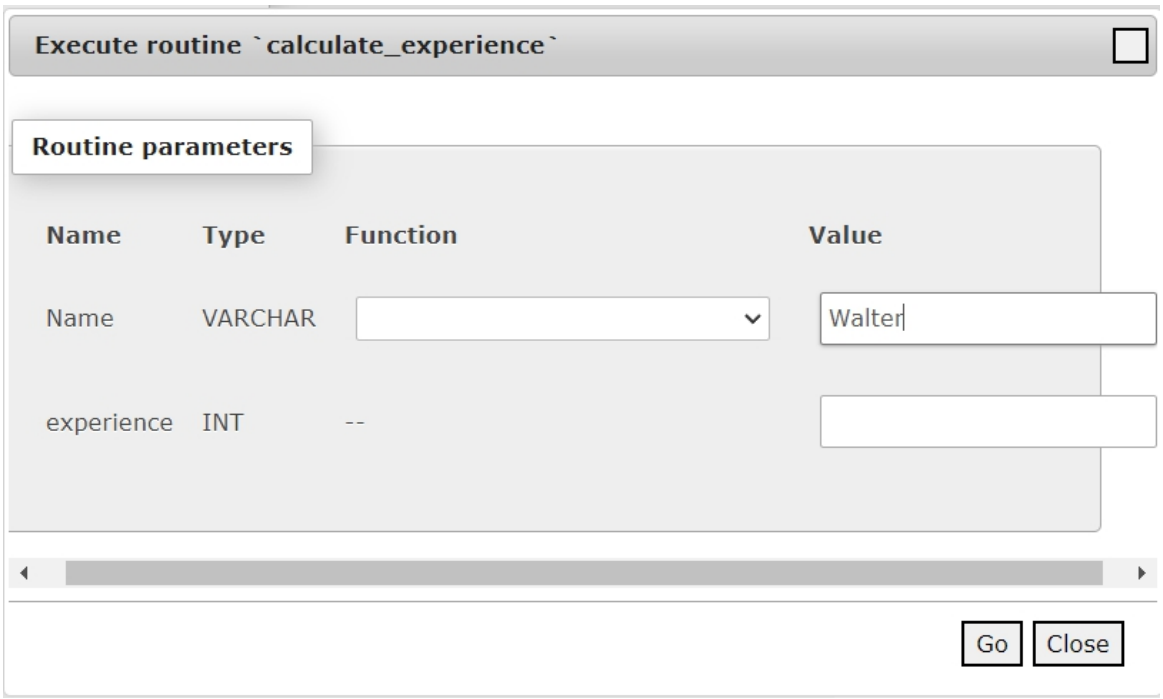
club_name	club_head	club_capacity	club_type	club_coordinated_by	club_start_date
HackerSpace	PESSTU009	3	Technical	Ross Geller	2018-06-11

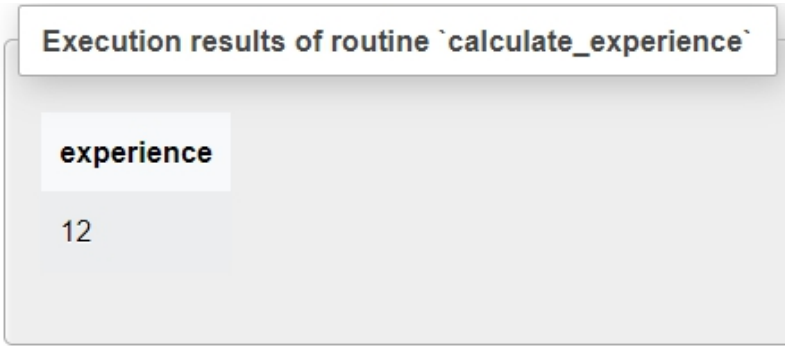
ii. Cursor

Calculate experience of a faculty using their join date. Create a procedure and use cursors to execute the operation.

```
DELIMITER $$
CREATE PROCEDURE calculate_experience(IN Name varchar(255),out experience int)
BEGIN
        DECLARE uid varchar(255);
        DECLARE join_date date;
        DECLARE new_cursor CURSOR FOR SELECT faculty.fac_name FROM faculty  WHERE
        Name=faculty.fac_name;
        OPEN new_cursor;
        FETCH new_cursor INTO uid;
        SET join_date = (SELECT fac_join_date FROM faculty WHERE          faculty.fac_name=uid);
        SET experience = (SELECT datediff(CURRENT_DATE(),join_date)/365);
        IF uid != NUll THEN
                UPDATE faculty
```

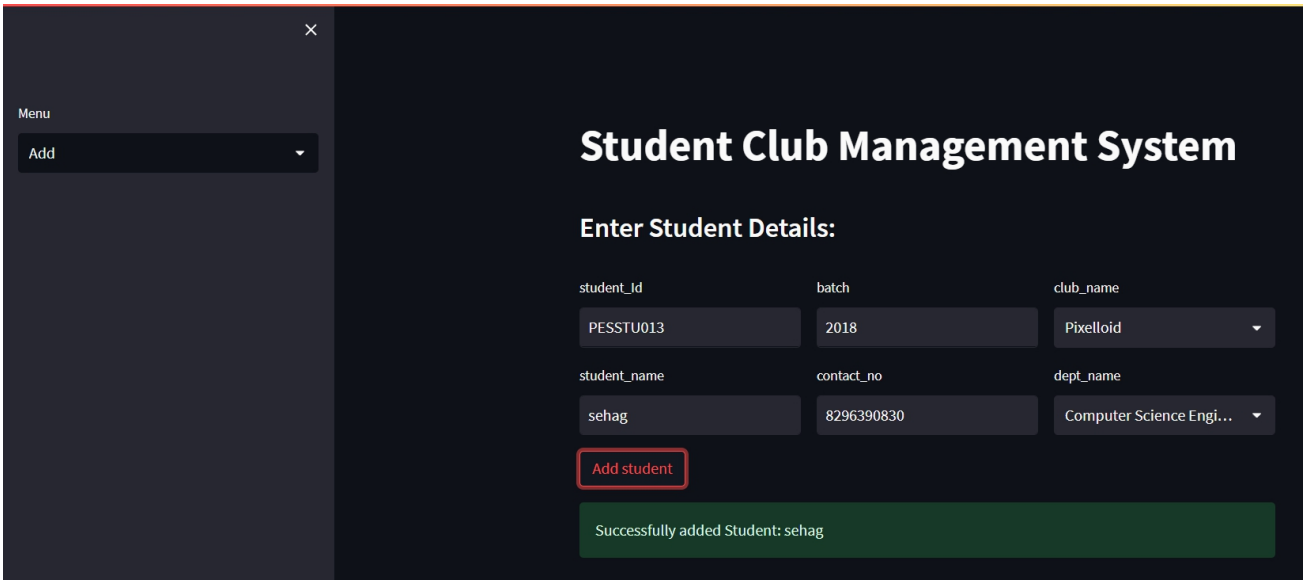
```
            SET fac_experience = @experience WHERE fac_name=uid;
        CLOSE new_cursor;
        END IF;
END;$$
DELIMITER ;
```



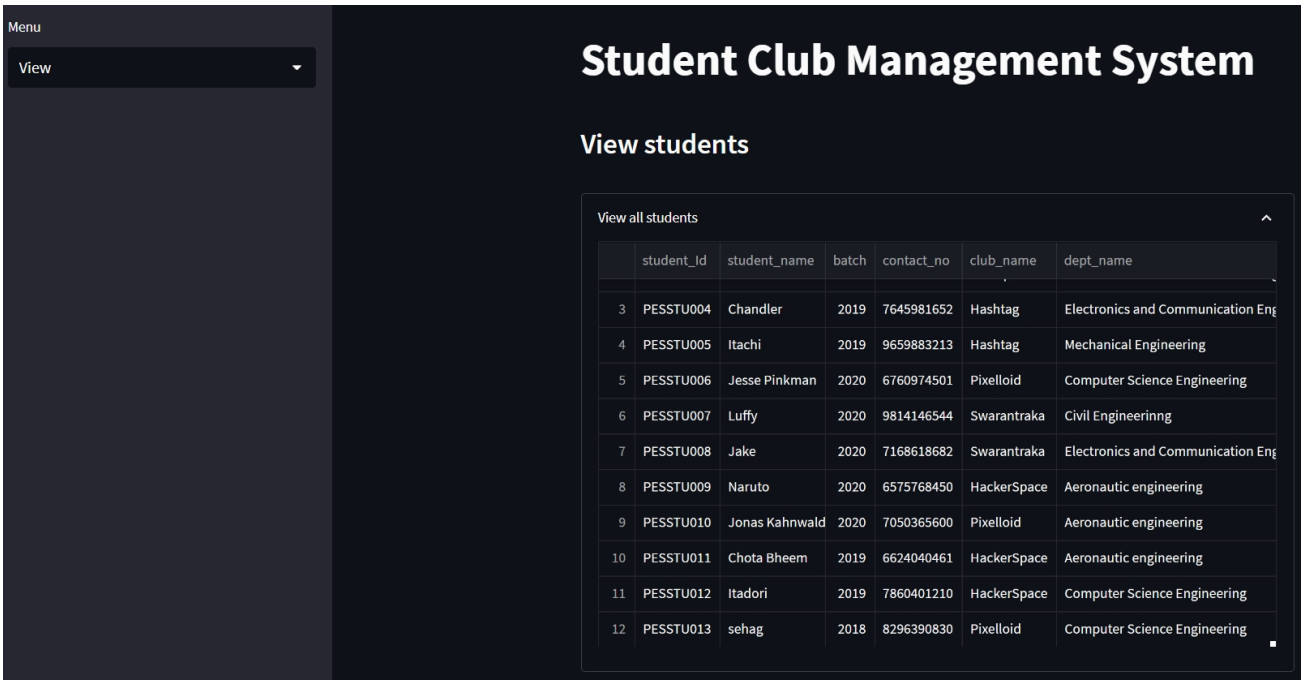


11. Developing a Frontend

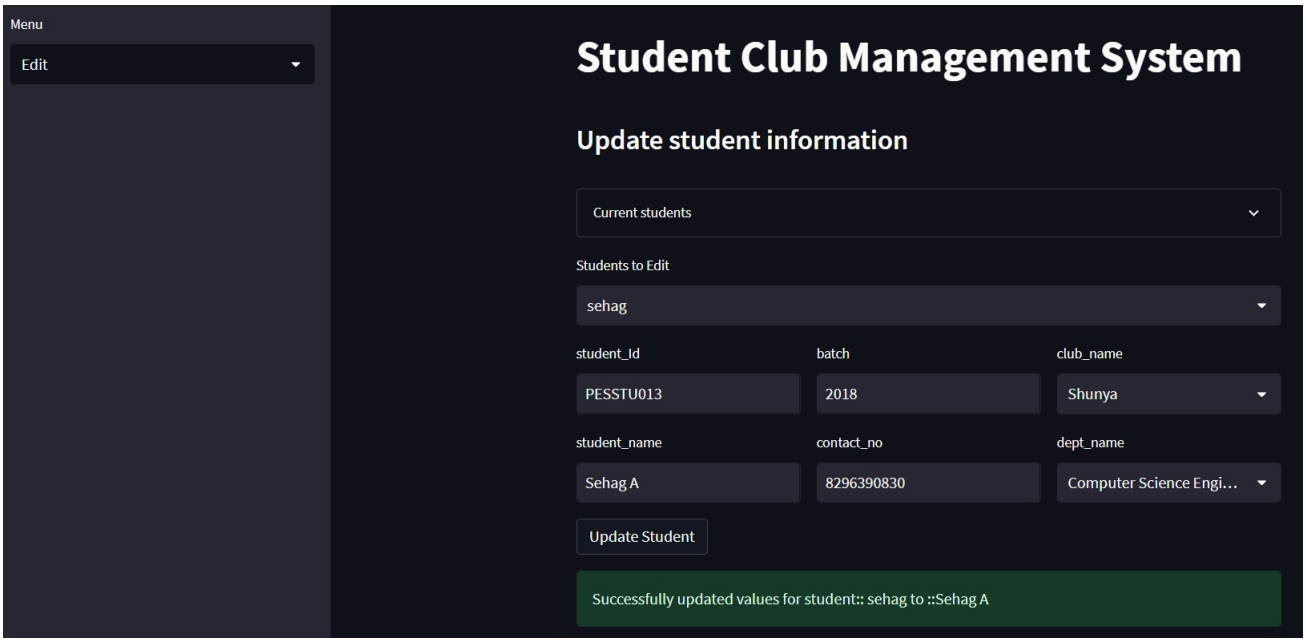
i. Adding a student to the database :



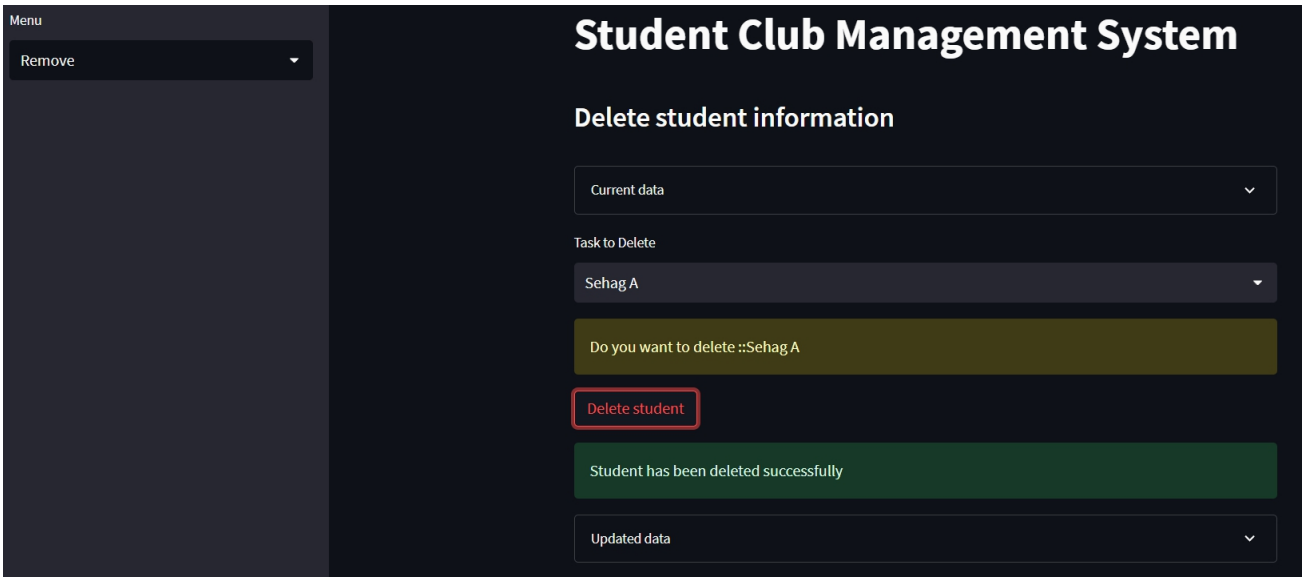
ii. View all the students in the database :



iii. Update a student information :



iv. Remove a student from the database :



12. Conclusion

This database can store all the information regarding a students,clubs,club events,faculties and many more information which are essential for managing all the clubs in an organisation.With the help of these databases,we can have many functionalities such as the education institution can generate a complete report on every club and track their events, students can look into events and enroll in their desired club based on their interests, faculty in-charge of a club can keep record of all the students details(such as name,DOB,SRN,contact.no) belonging to their club, keep track of all the departments managing the clubs.

The front-end so developed makes it user-friendly for the users to manage their database.