PES
UNIVERSITY

Mini project report on

Student Database Management System – StudentHub

Submitted in partial fulfilment of the requirements for the award of degree of

Bachelor of Technology

in

Computer Science & Engineering

UE21CS351 – DBMS Project

Submitted by:

Urvashi Bhargava	**PES2UG21CS579**
Y Teresha	**PES2UG21CS618**

Under the guidance of

Dr. Mannar Mannan J

PES University

AUG - DEC 2023

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

FACULTY OF ENGINEERING

PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)

Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, India

PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)
Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, India

CERTIFICATE

This is to certify that the mini project entitled

Disasters data Management System

is a bonafide work carried out by

Urvashi Bhargava	**PES2UG21CS579**
Y Teresha	**PES2UG21CS618**

In partial fulfilment for the completion of fifth semester DBMS Project (UE20CSS301) in the Program of Study -Bachelor of Technology in Computer Science and Engineering under rules and regulations of PES University, Bengaluru during the period AUG. 2022 – DEC. 2022. It is certified that all corrections / suggestions indicated for internal assessment have been incorporated in the report. The project has been approved as it satisfies the 5th semester academic requirements in respect of project work.

Signature
Dr. Mannar Mannan J

DECLARATION

We hereby declare that the DBMS Project entitled **Student Database Management System** has been carried out by us under the guidance of **Dr. Mannar Mannan J** and submitted in partial fulfilment of the course requirements for the award of degree of **Bachelor of Technology** in **Computer Science and Engineering** of **PES University, Bengaluru** during the academic semester AUG – DEC 2023.

Urvashi Bhargava	**PES2UG21CS579**	
Y Teresha	**PES2UG21CS618**	Y. Teresha

ACKNOWLEDGEMENT

I would like to express my gratitude to Dr. Mannar Mannan, Department of Computer Science and Engineering, PES University, for his continuous guidance, assistance, and encouragement throughout the development of this UE21CS351 - DBMS Project.

I am deeply grateful to Dr. M. R. Doreswamy, Chancellor, PES University, Prof. Jawahar Doreswamy, Pro Chancellor – PES University, Dr. Suryaprasad J, Vice-Chancellor, PES University for providing to me various opportunities and enlightenment every step of the way. Finally, this DBMS Project could not have been completed without the continual support and encouragement I have received from my family and friends.

ABSTRACT

This document introduces the design and implementation of an Integrated Student Management System (ISMS) tailored for educational institutions. The ISMS serves as a comprehensive software solution aimed at replacing manual processes with an efficient and scalable digital infrastructure. The system is equipped with user roles, including students and administrators, each with specific privileges to manage and access pertinent information.

Administrators wield the power to add new users, edit student details, and oversee department-wise attendance records. The system facilitates the seamless addition and management of student records, ensuring that accurate and up-to-date information is readily available. Furthermore, attendance tracking capabilities provide administrators with a systematic overview of student participation.

This document outlines the development and implementation of an Integrated Student Management System (ISMS) tailored for educational institutions, presenting a solution to replace manual processes with a robust digital infrastructure. Designed to accommodate students and administrators with distinct privileges, the ISMS empowers administrators to efficiently manage user accounts, edit student details, and oversee department-wise attendance records. It streamlines the addition and management of student records, ensuring real-time accuracy and accessibility. Key features include user-friendly registration, scalable architecture, and a comprehensive search function for students to retrieve personal details and attendance records effortlessly. The ISMS promises to optimize time efficiency, enhance data accuracy, and provide a user-friendly interface, offering a transformative approach to student management within educational institutions.

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

1.1 Purpose of project

The primary objective of this project is to create an intuitive and user-friendly student database management system that effectively eliminates data redundancy, offering a synchronized and centralized repository of student information. Our system will prioritize data security by implementing login and password methods, minimizing the chances of information leakage. Additionally, it will facilitate swift storage and retrieval of data, streamlining administrative tasks and improving coordination among students. Ultimately, our goal is to significantly reduce paperwork and create a more efficient, modernized student management solution.

1.2 Scope of project

The scope of this project encompasses the development of a centralized and userfriendly student database management system that will streamline data entry processes, enhance data security through login and password methods, facilitate efficient report generation, and significantly reduce paperwork in educational institutions. The system's scalability and potential for integration with other educational systems will ensure adaptability to evolving needs. Additionally, comprehensive training and support will be provided to ensure seamless adoption and continued efficient operation within the educational institution. StudentHub will be extremely useful for students.

1.3 Limitations

●Time-consuming data entry due to manual record maintenance, imposing a significant burden on faculties.

● Extensive paperwork involved in record-keeping, with data stored in physical files and registers.

● Increased storage requirements as files and registers accumulate, taking up physical space.

● Low reliability, as using paper for storing valuable data information is not a secure or dependable method.

2. PROBLEM DEFINITION

The current educational system relies heavily on manual record-keeping and administrative processes, leading to inefficiencies, data inaccuracies, and a lack of scalability. Educational institutions face challenges in managing student information, attendance records, and user access control. The absence of a centralized database management system hampers the seamless flow of information and impedes the institution's ability to adapt to a growing student population. Consequently, there is a critical need for a comprehensive DBMS solution that can address these challenges by automating data management, ensuring data integrity, and providing a scalable platform for efficient educational administration. This project aims to develop and implement a DBMS solution that can revolutionize the way educational institutions handle student data, ultimately improving overall operational efficiency and decision-making processes.

3. ER MODEL

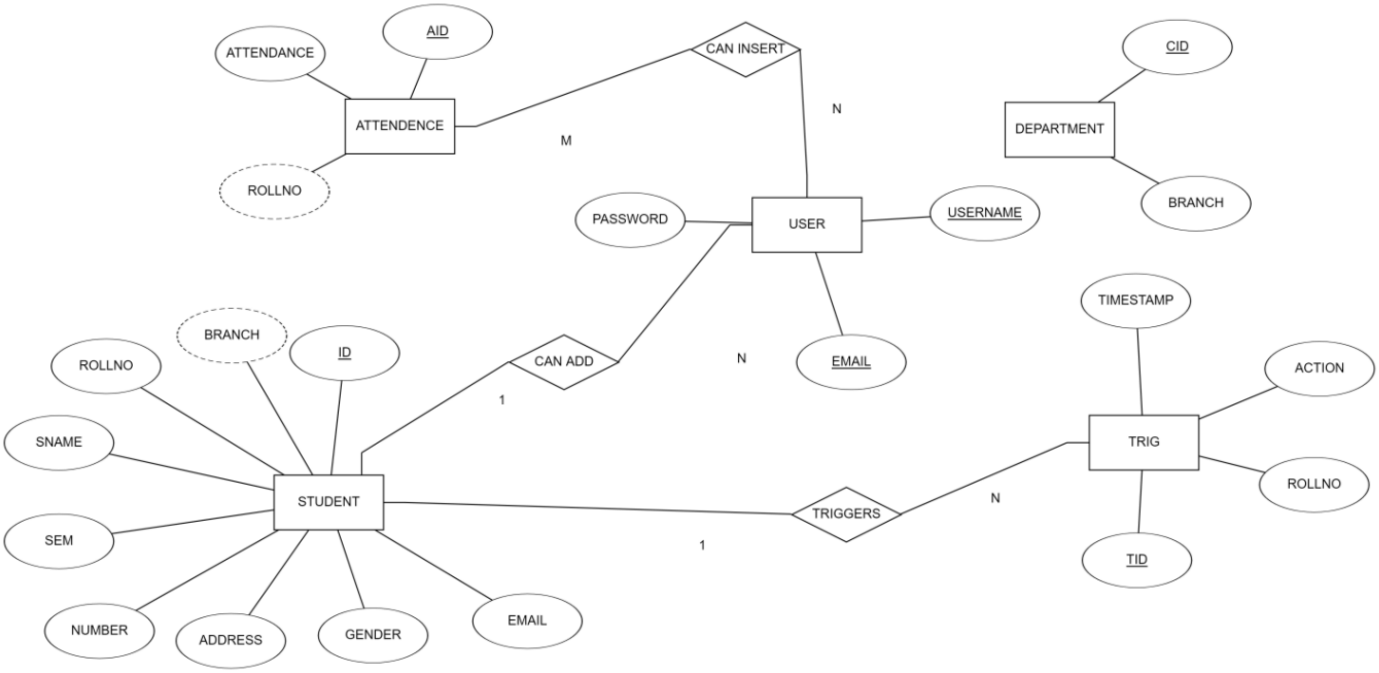


Fig 3.1 ER Diagram

4. ER TO RELATIONAL MAPPING

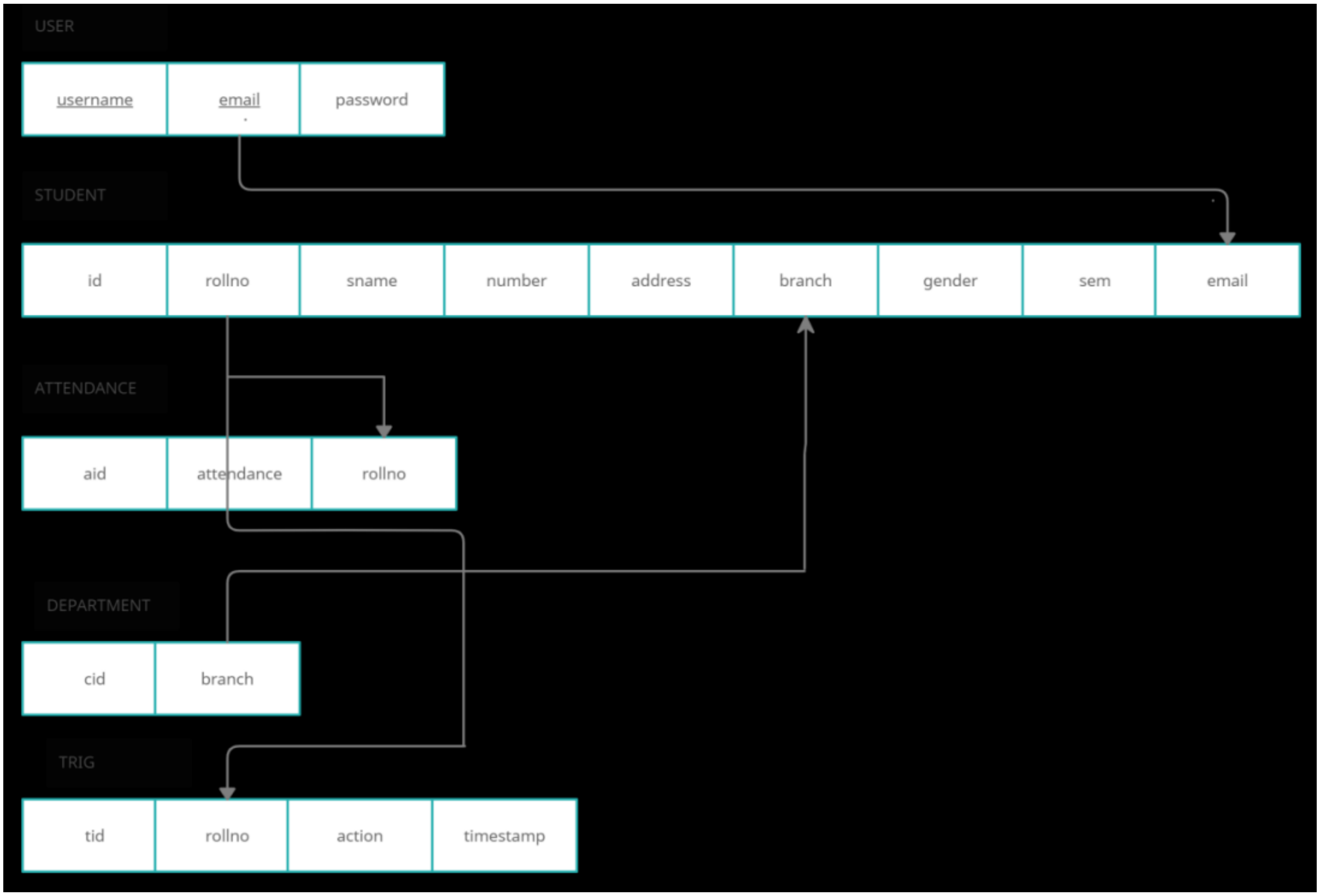
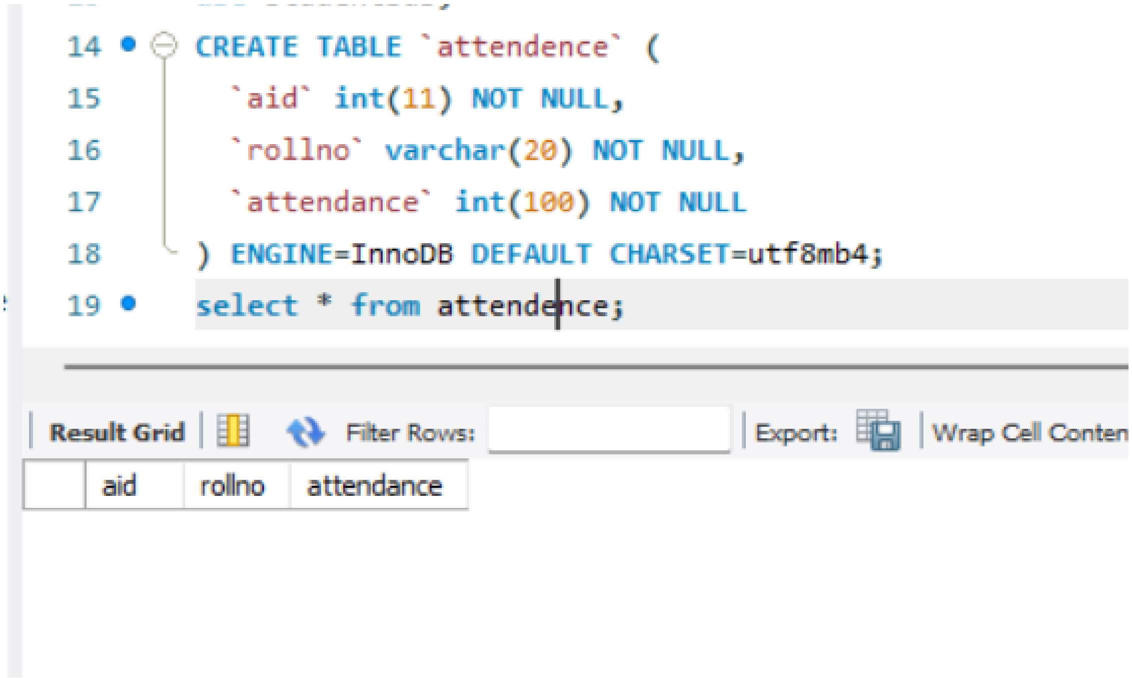


Fig 3.2 Relational Schema

5. DDL STATEMENTS

STATEMENTS WITH SCREEN SHOTS OF THE TABLE CREATION

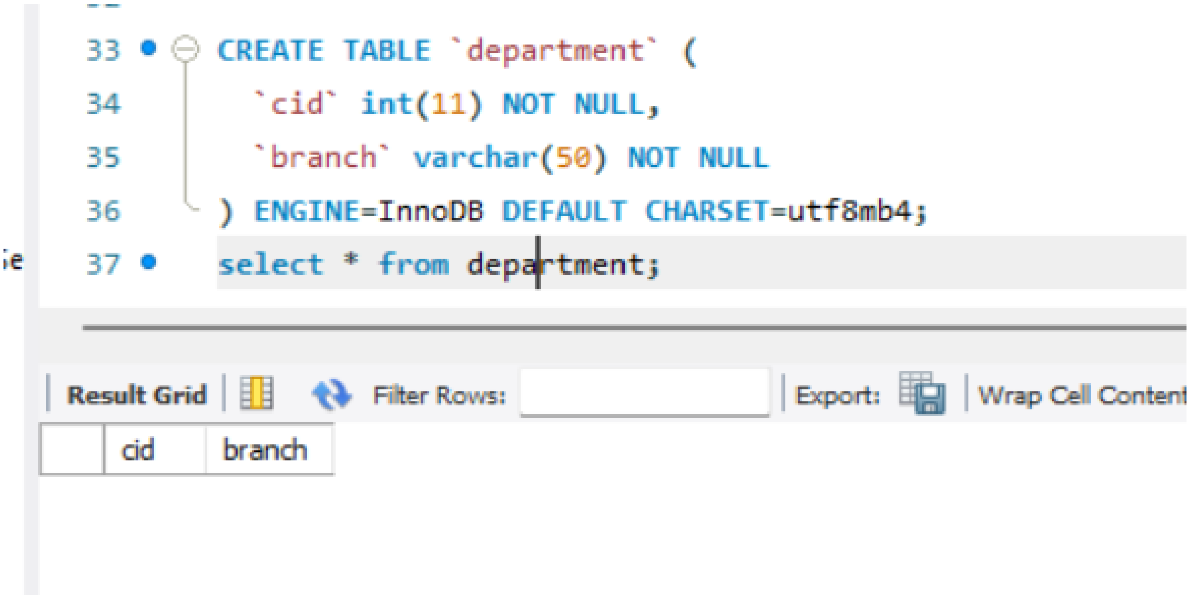
TABLE 1: ATTENDANCE



```
CREATE TABLE `attendence` (
  `aid` int(11) NOT NULL,
  `rollno` varchar(20) NOT NULL,
  `attendance` int(100) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
select * from attendence;
```

	aid	rollno	attendance

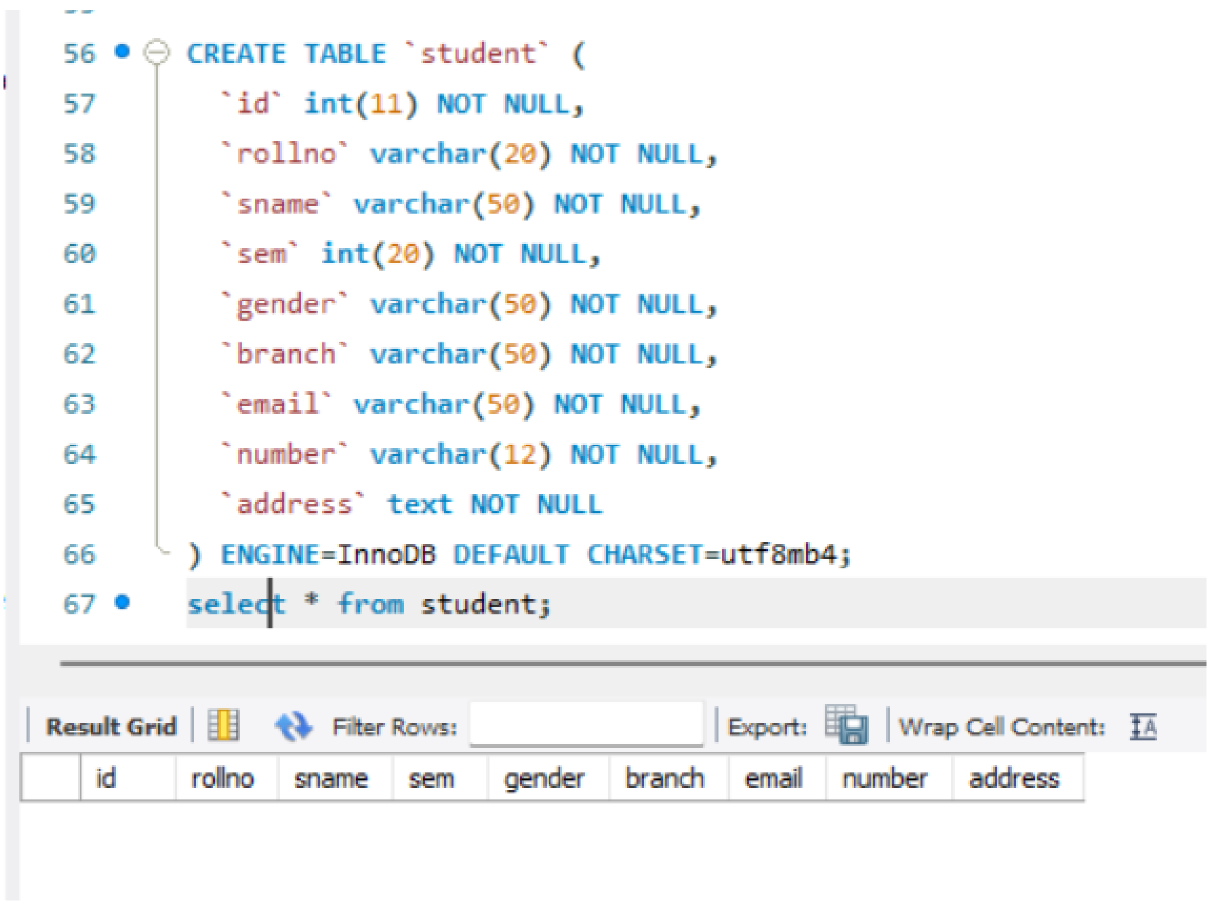
TABLE 2: DEPARTMENT



```
CREATE TABLE `department` (
  `cid` int(11) NOT NULL,
  `branch` varchar(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
select * from department;
```

	cid	branch

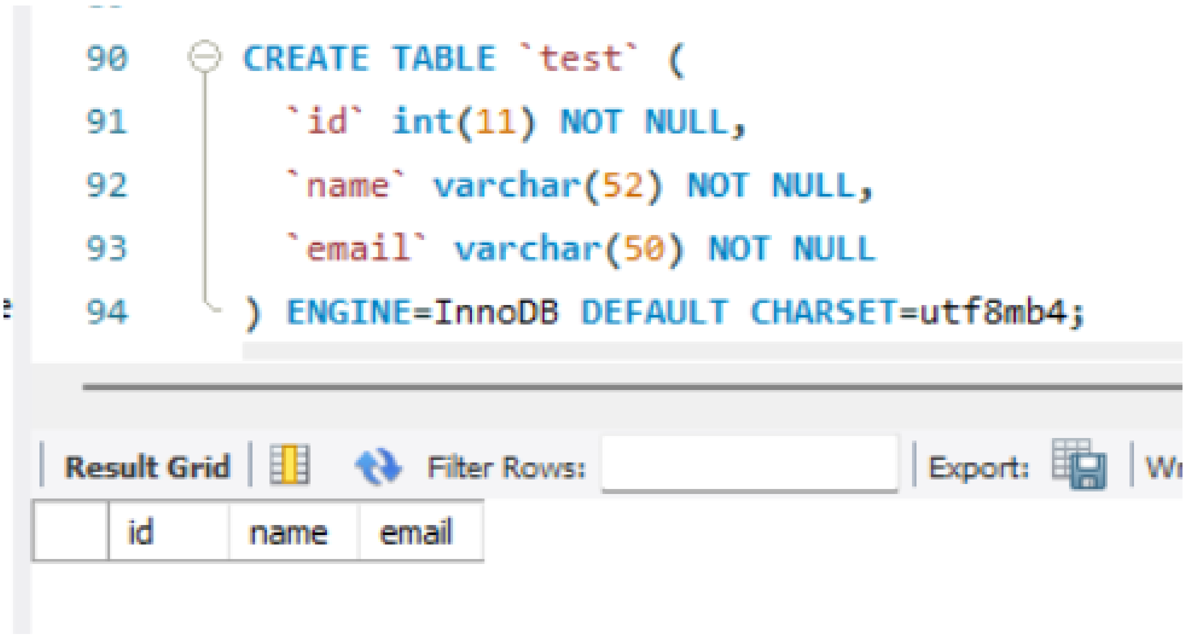
TABLE 3 : STUDENT



```sql
CREATE TABLE `student` (
  `id` int(11) NOT NULL,
  `rollno` varchar(20) NOT NULL,
  `sname` varchar(50) NOT NULL,
  `sem` int(20) NOT NULL,
  `gender` varchar(50) NOT NULL,
  `branch` varchar(50) NOT NULL,
  `email` varchar(50) NOT NULL,
  `number` varchar(12) NOT NULL,
  `address` text NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
select * from student;
```

	id	rollno	sname	sem	gender	branch	email	number	address

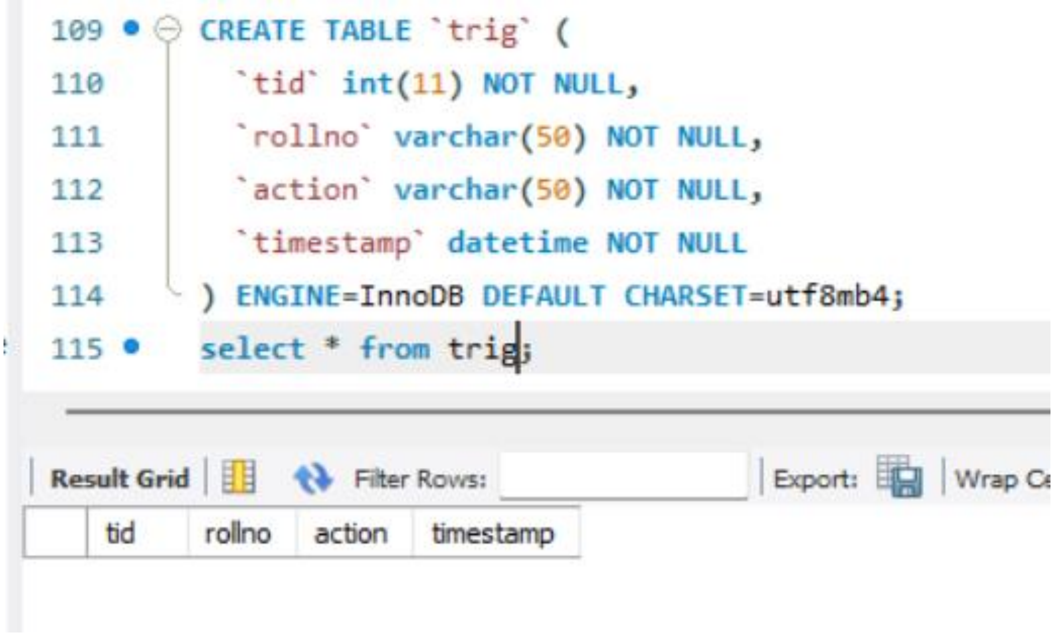
TABLE 4: TEST



```sql
CREATE TABLE `test` (
  `id` int(11) NOT NULL,
  `name` varchar(52) NOT NULL,
  `email` varchar(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

	id	name	email

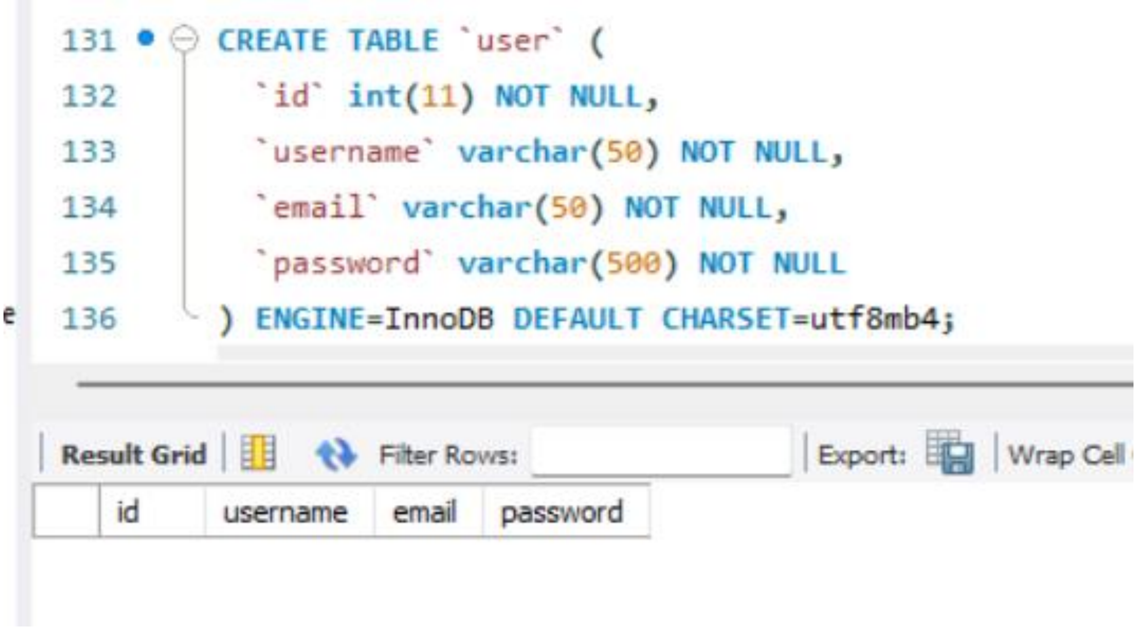
TABLE 5: TRIG



```
CREATE TABLE `trig` (
  `tid` int(11) NOT NULL,
  `rollno` varchar(50) NOT NULL,
  `action` varchar(50) NOT NULL,
  `timestamp` datetime NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
select * from trig;
```

	tid	rollno	action	timestamp

TABLE 6 : USER



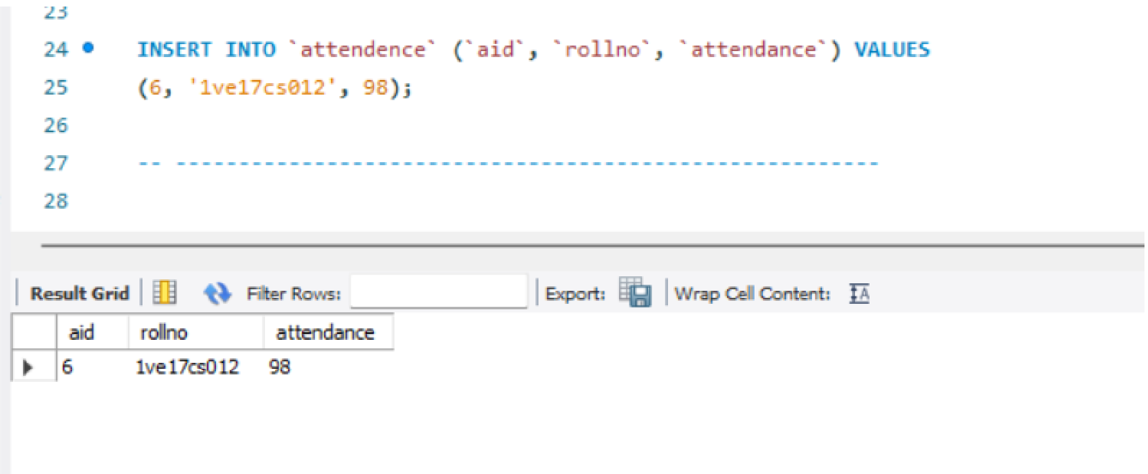
```
CREATE TABLE `user` (
  `id` int(11) NOT NULL,
  `username` varchar(50) NOT NULL,
  `email` varchar(50) NOT NULL,
  `password` varchar(500) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

	id	username	email	password

6. DML STATEMENTS

STATEMENTS WITH SCREEN SHOTS OF THE TABLE WITH INSERTED VALUES

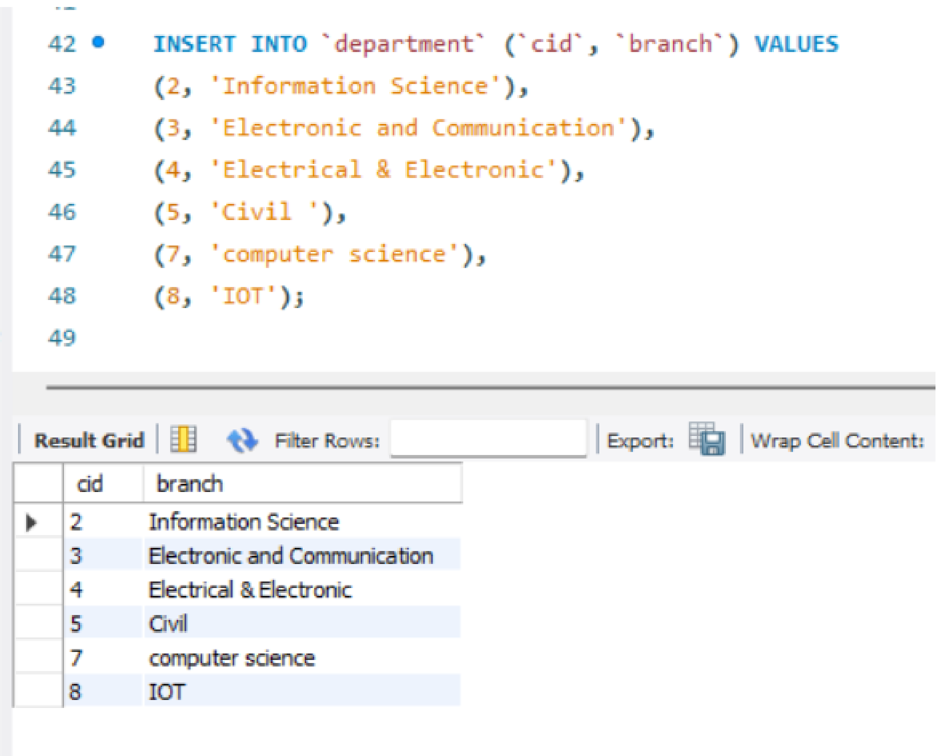
TABLE 1: ATTENDANCE



```
INSERT INTO `attendence` (`aid`, `rollno`, `attendance`) VALUES
(6, '1ve17cs012', 98);


-- ---------------------------------------------------------
```

aid	rollno	attendance
6	1ve17cs012	98

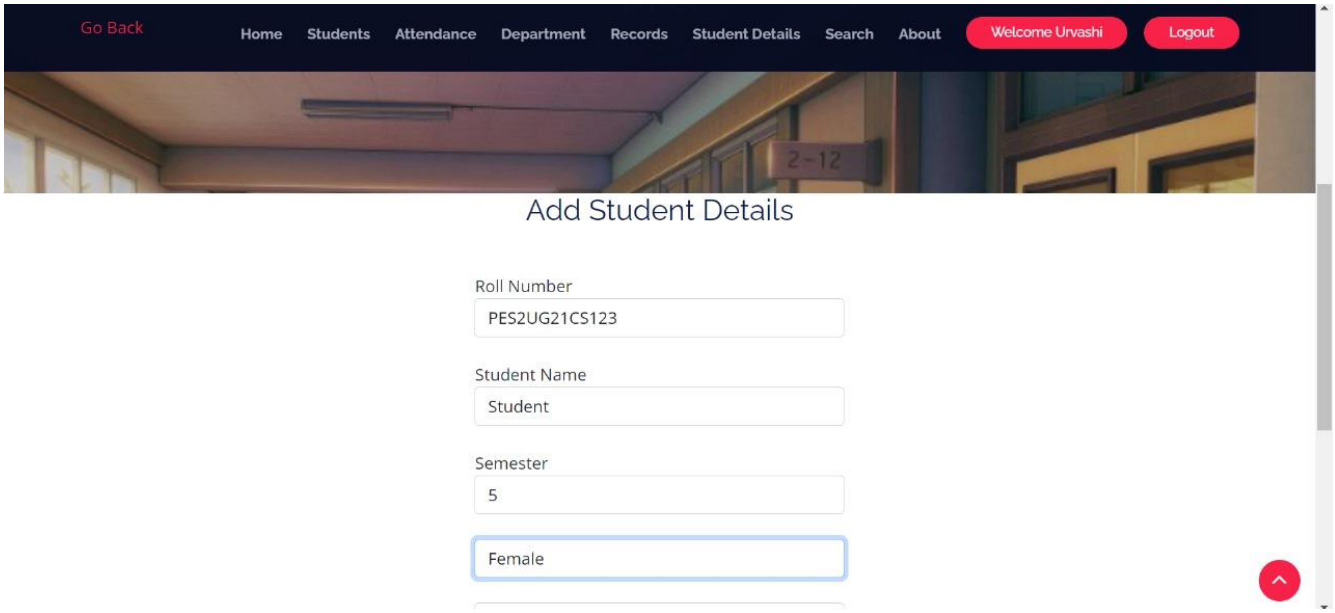
TABLE 2: DEPARTMENT



```
INSERT INTO `department` (`cid`, `branch`) VALUES
(2, 'Information Science'),
(3, 'Electronic and Communication'),
(4, 'Electrical & Electronic'),
(5, 'Civil '),
(7, 'computer science'),
(8, 'IOT');
```

cid	branch
2	Information Science
3	Electronic and Communication
4	Electrical & Electronic
5	Civil
7	computer science
8	IOT

TABLE 3 :STUDENT



Add Student Details

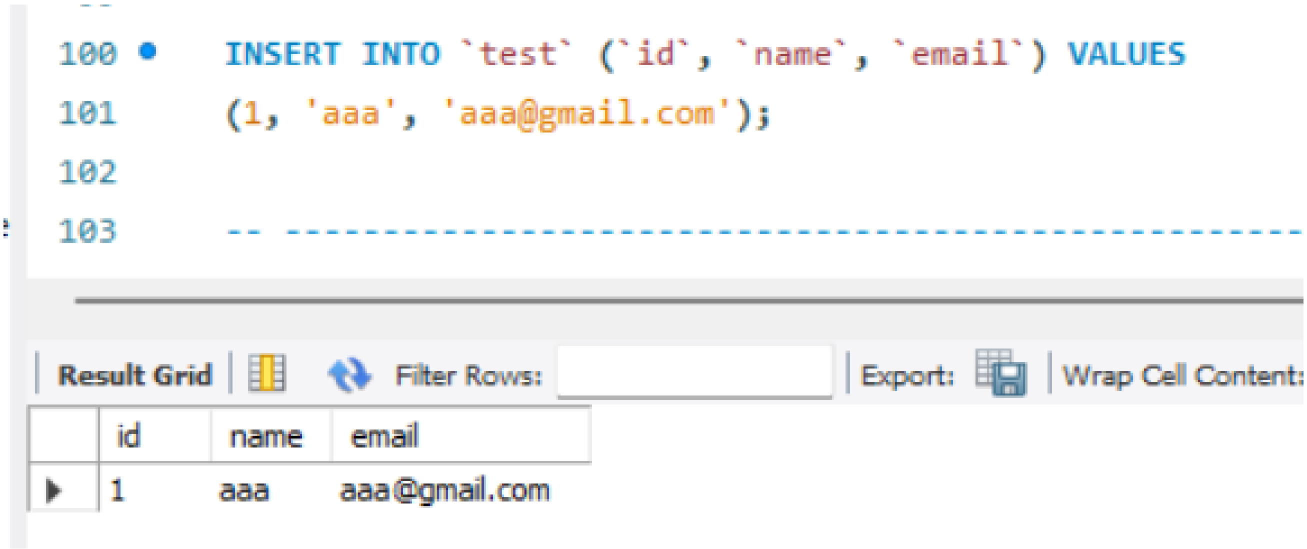
Roll Number
PES2UG21CS123

Student Name
Student

Semester
5

Female

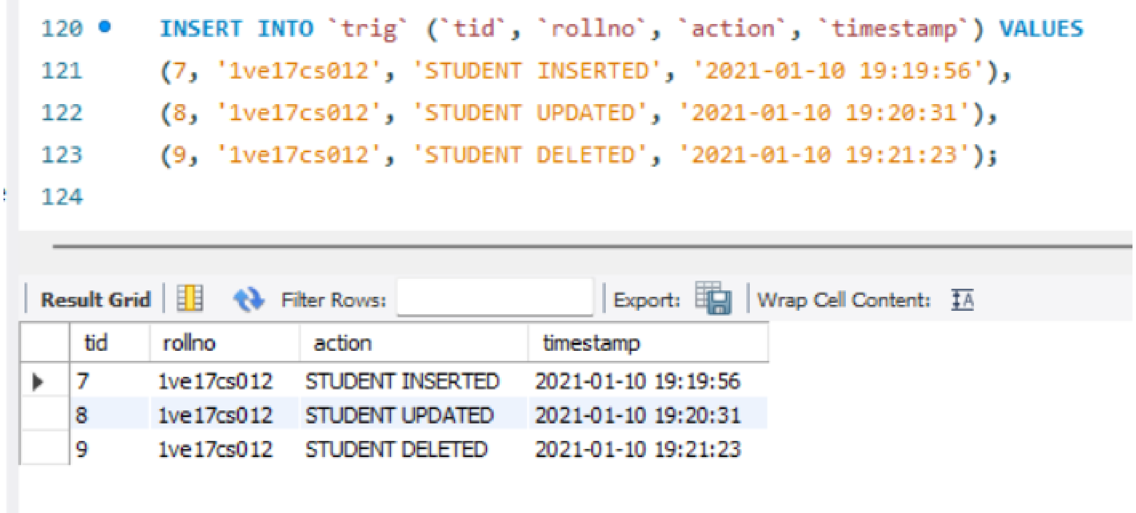
TABLE 4: TEST



```
INSERT INTO `test` (`id`, `name`, `email`) VALUES
(1, 'aaa', 'aaa@gmail.com');

-- ---------------------------------------------------------
```

id	name	email
1	aaa	aaa@gmail.com

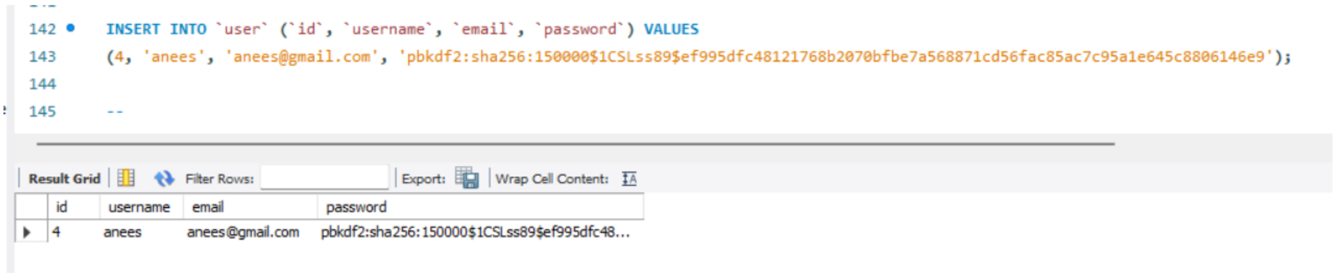
TABLE 5: TRIG



```sql
INSERT INTO `trig` (`tid`, `rollno`, `action`, `timestamp`) VALUES
(7, '1ve17cs012', 'STUDENT INSERTED', '2021-01-10 19:19:56'),
(8, '1ve17cs012', 'STUDENT UPDATED', '2021-01-10 19:20:31'),
(9, '1ve17cs012', 'STUDENT DELETED', '2021-01-10 19:21:23');
```

tid	rollno	action	timestamp
7	1ve17cs012	STUDENT INSERTED	2021-01-10 19:19:56
8	1ve17cs012	STUDENT UPDATED	2021-01-10 19:20:31
9	1ve17cs012	STUDENT DELETED	2021-01-10 19:21:23

TABLE 6: USER

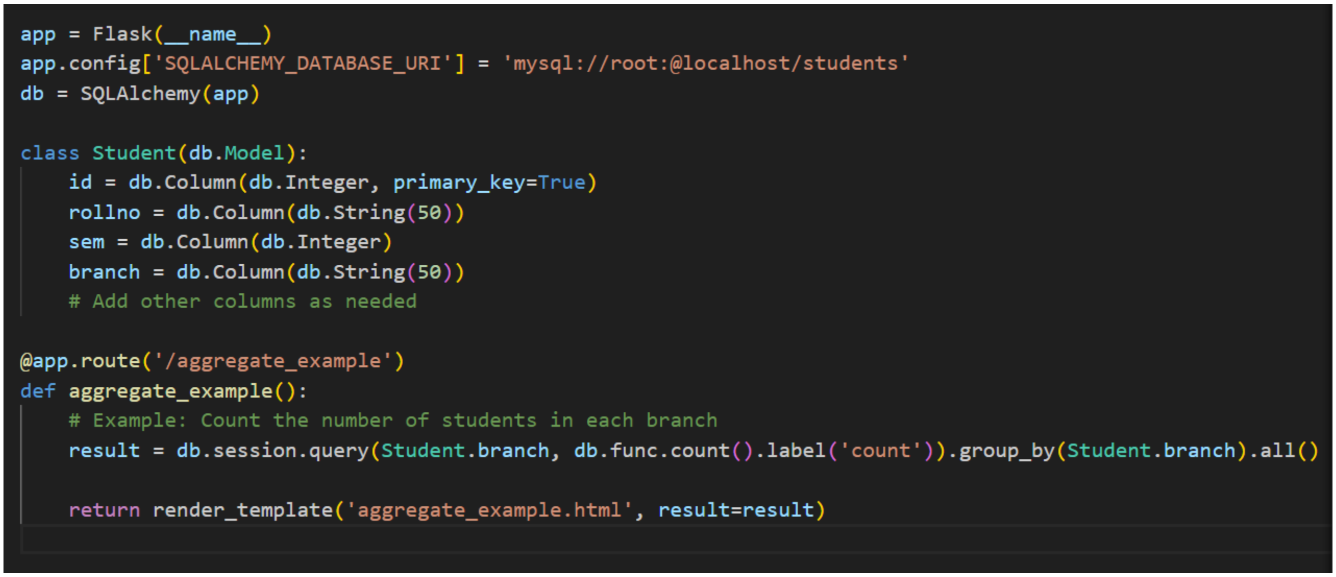


```sql
INSERT INTO `user` (`id`, `username`, `email`, `password`) VALUES
(4, 'anees', 'anees@gmail.com', 'pbkdf2:sha256:150000$1CSLss89$ef995dfc48121768b2070bfbe7a568871cd56fac85ac7c95a1e645c8806146e9');

--
```

id	username	email	password
4	anees	anees@gmail.com	pbkdf2:sha256:150000$1CSLss89$ef995dfc48...

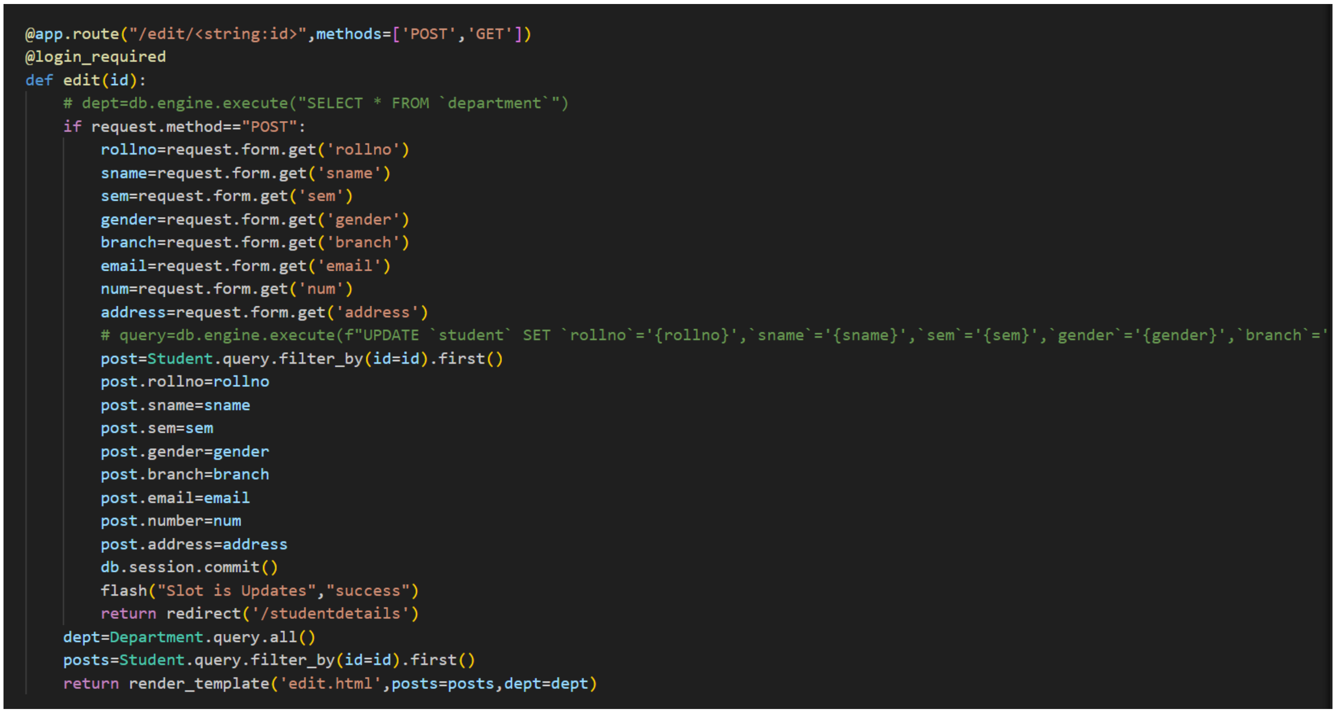
7. QUERIES

7.1 SIMPLE QUERY WITH GROUP BY, AGRREGATE



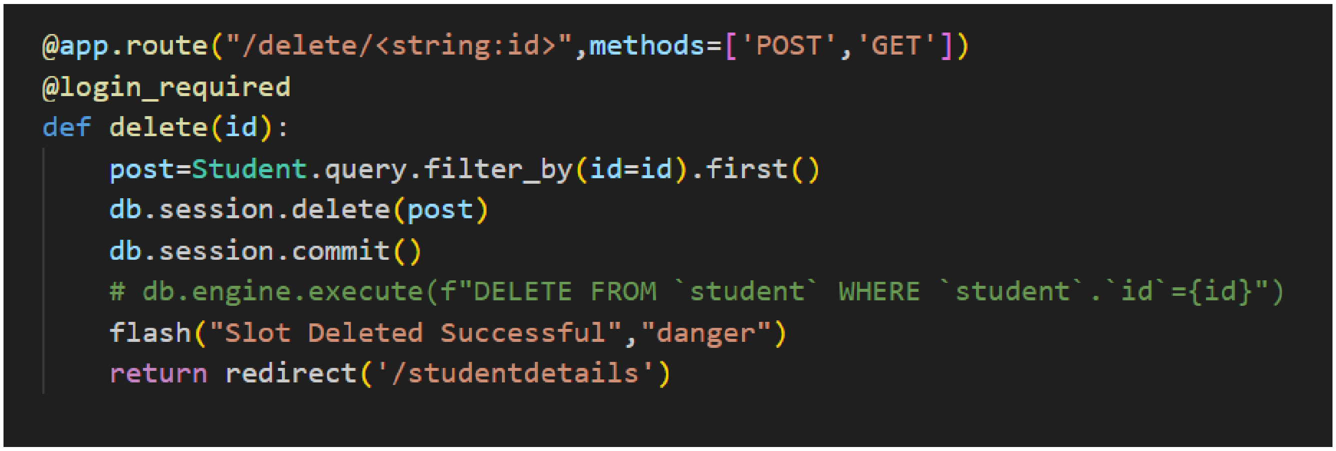
```python
app = Flask(__name__)
app.config['SQLALCHEMY_DATABASE_URI'] = 'mysql://root:@localhost/students'
db = SQLAlchemy(app)

class Student(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    rollno = db.Column(db.String(50))
    sem = db.Column(db.Integer)
    branch = db.Column(db.String(50))
    # Add other columns as needed

@app.route('/aggregate_example')
def aggregate_example():
    # Example: Count the number of students in each branch
    result = db.session.query(Student.branch, db.func.count().label('count')).group_by(Student.branch).all()

    return render_template('aggregate_example.html', result=result)
```

7.2 UPDATE OPERATION



```python
@app.route("/edit/<string:id>",methods=['POST','GET'])
@login_required
def edit(id):
    # dept=db.engine.execute("SELECT * FROM `department`")
    if request.method=="POST":
        rollno=request.form.get('rollno')
        sname=request.form.get('sname')
        sem=request.form.get('sem')
        gender=request.form.get('gender')
        branch=request.form.get('branch')
        email=request.form.get('email')
        num=request.form.get('num')
        address=request.form.get('address')
        # query=db.engine.execute(f"UPDATE `student` SET `rollno`='{rollno}',`sname`='{sname}',`sem`='{sem}',`gender`='{gender}',`branch`='
        post=Student.query.filter_by(id=id).first()
        post.rollno=rollno
        post.sname=sname
        post.sem=sem
        post.gender=gender
        post.branch=branch
        post.email=email
        post.number=num
        post.address=address
        db.session.commit()
        flash("Slot is Updates","success")
        return redirect('/studentdetails')
    dept=Department.query.all()
    posts=Student.query.filter_by(id=id).first()
    return render_template('edit.html',posts=posts,dept=dept)
```

7.3 DELETE OPERATION



```
@app.route("/delete/<string:id>",methods=['POST','GET'])
@login_required
def delete(id):
    post=Student.query.filter_by(id=id).first()
    db.session.delete(post)
    db.session.commit()
    # db.engine.execute(f"DELETE FROM `student` WHERE `student`.`id`={id}")
    flash("Slot Deleted Successful","danger")
    return redirect('/studentdetails')
```

8. STORED PROCEDURES, FUCNTIONS AND TRIGGERS

8.1 STORED PROCEDURES OR FUNCTIONS

Routine name: proc

Type: procedure

Definition: Select * from register;

8.2 TRIGGERS

Triggers used :

1: Trigger name: on insert

Table: register

Time: after

Event: insert

INSERT INTO trig VALUES(null,NEW.rid,'Farmer Inserted',NOW())

2: Trigger name: on delete

Table: register

Time: after

Event: delete

Definition: INSERT INTO trig VALUES(null,OLD.rid,'FARMER DELETED',NOW())

3: Trigger name: on update

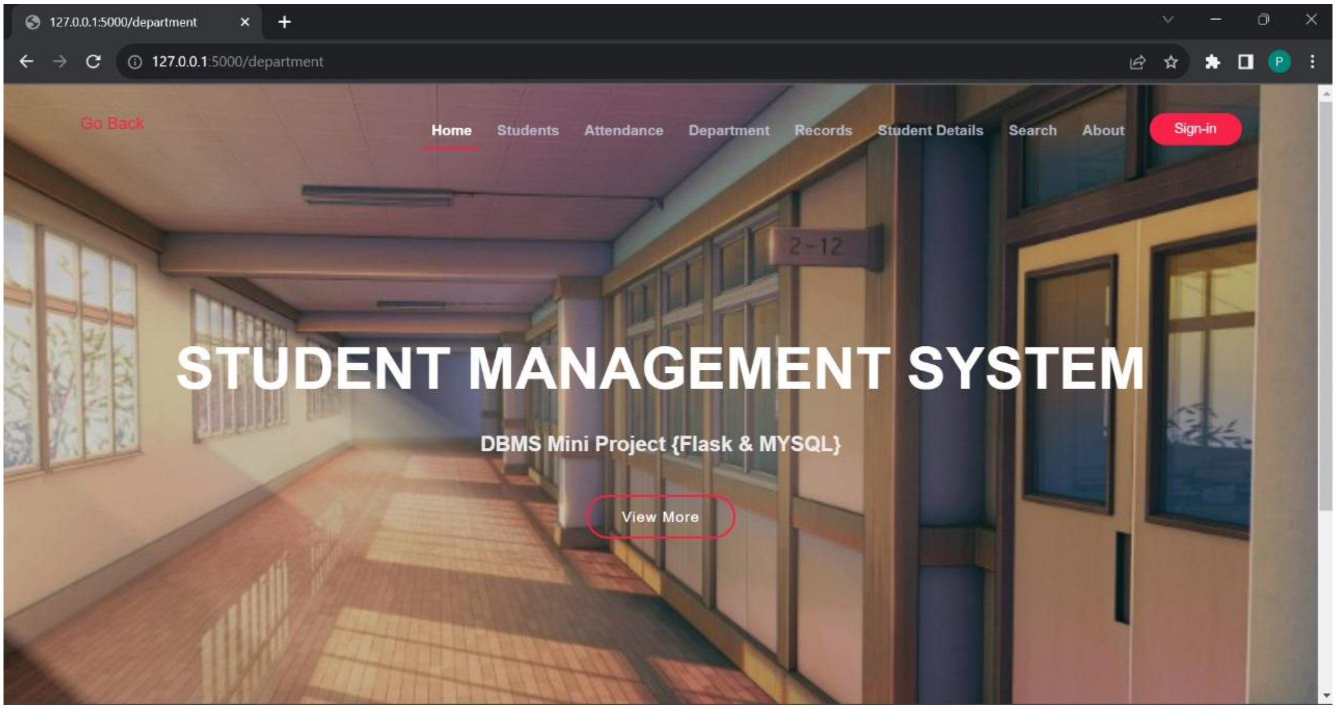
Table: register

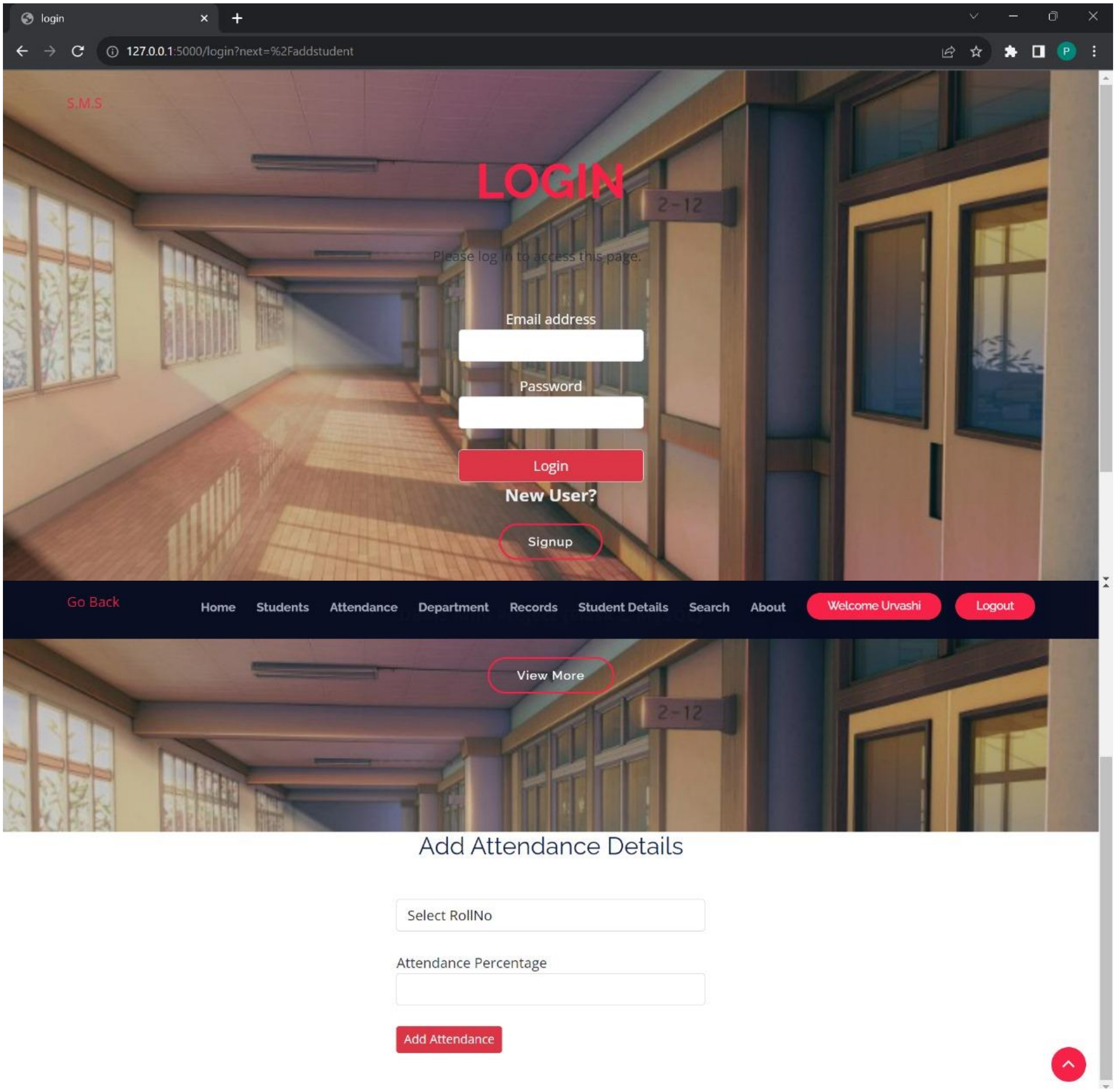
Time: after

Event: update

Definition: INSERT INTO trig VALUES(null,NEW.rid,'FARMER UPDATED',NOW())

9. FRONT END DEVELOPMENT


127.0.0.1:5000/department
127.0.0.1:5000/department
Go Back
Home
Students
Attendance
Department
Records
Student Details
Search
About
Sign-in
2-12
STUDENT MANAGEMENT SYSTEM
DBMS Mini Project {Flask & MYSQL}
View More



LOGIN

Please log in to access this page.

Email address

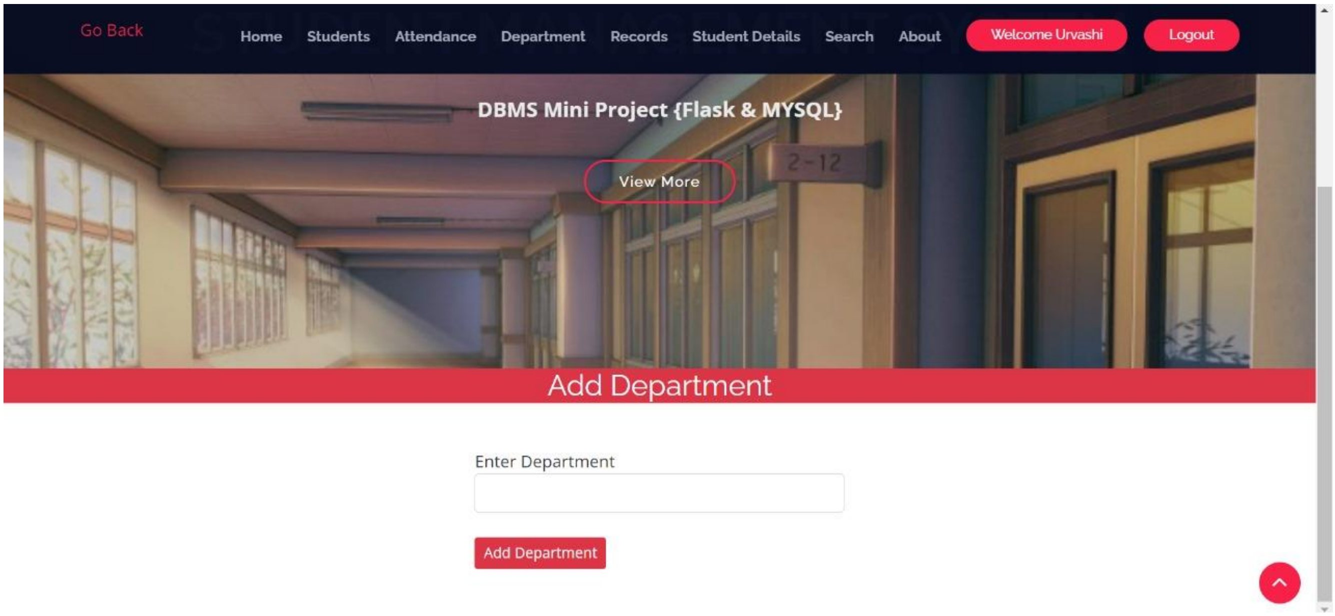
Password

New User?

Add Attendance Details

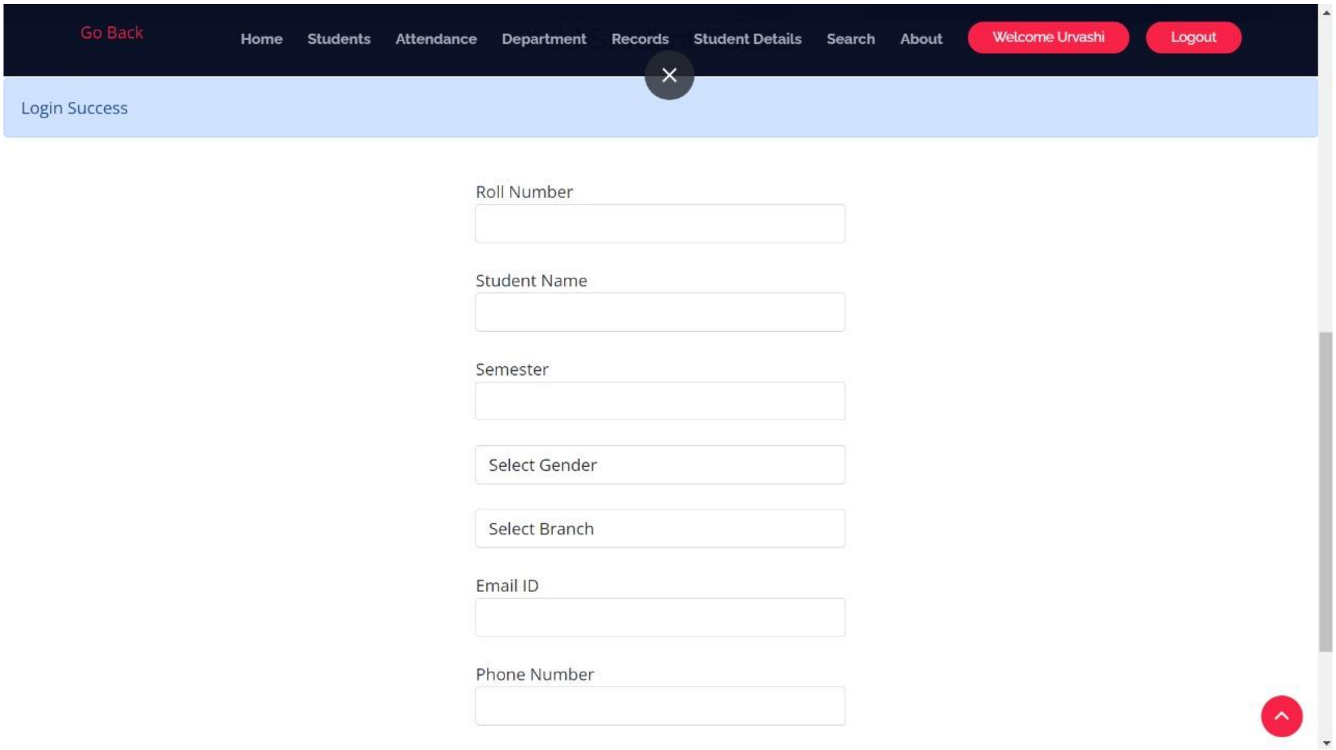
Select RollNo

Attendance Percentage



DBMS Mini Project {Flask & MYSQL}

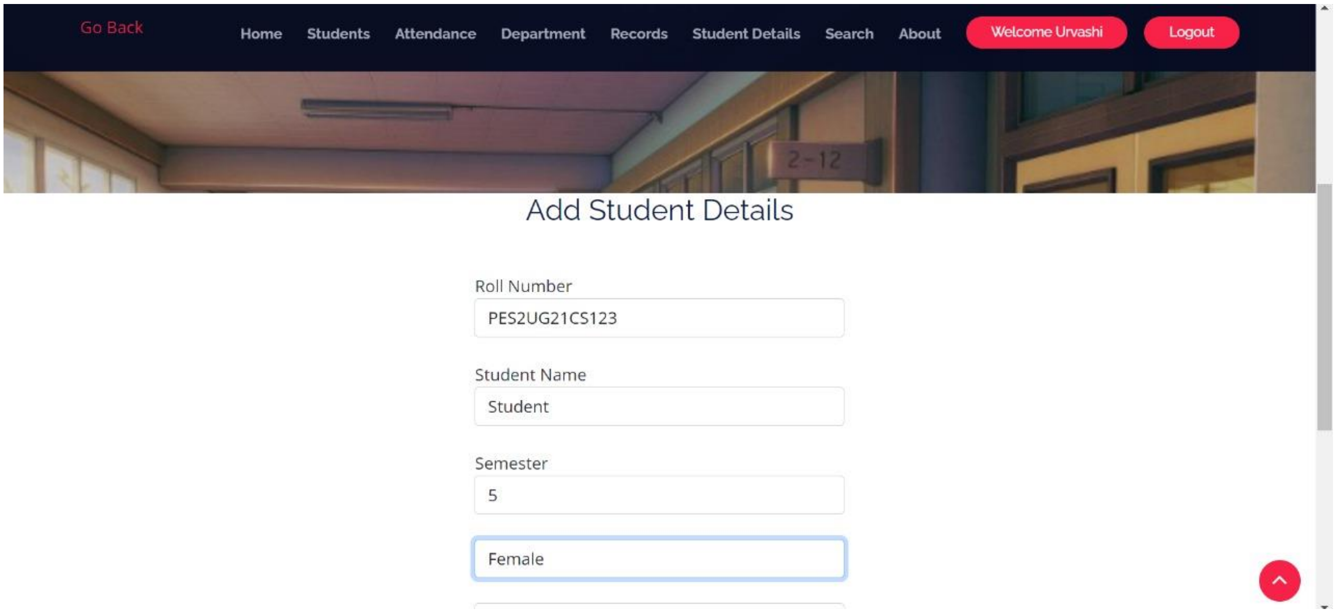
Add Department



Login Success

CRUD OPERATIONS

1) CREATE



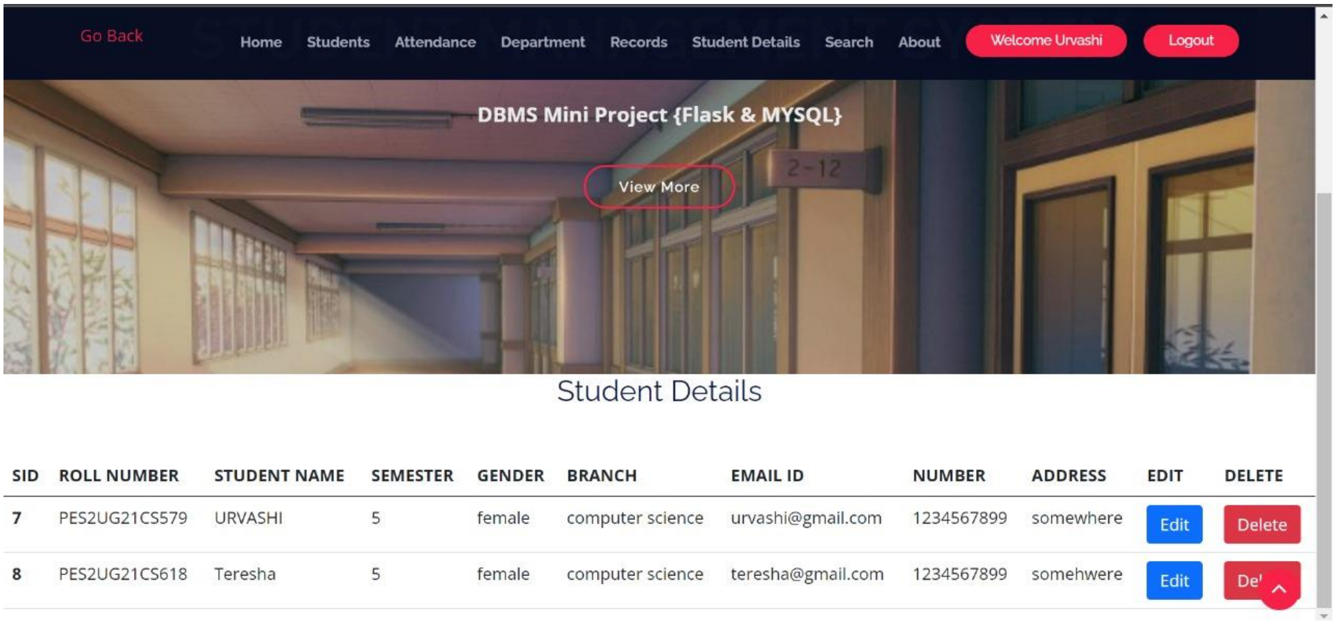
Add Student Details

Roll Number: PES2UG21CS123

Student Name: Student

Semester: 5

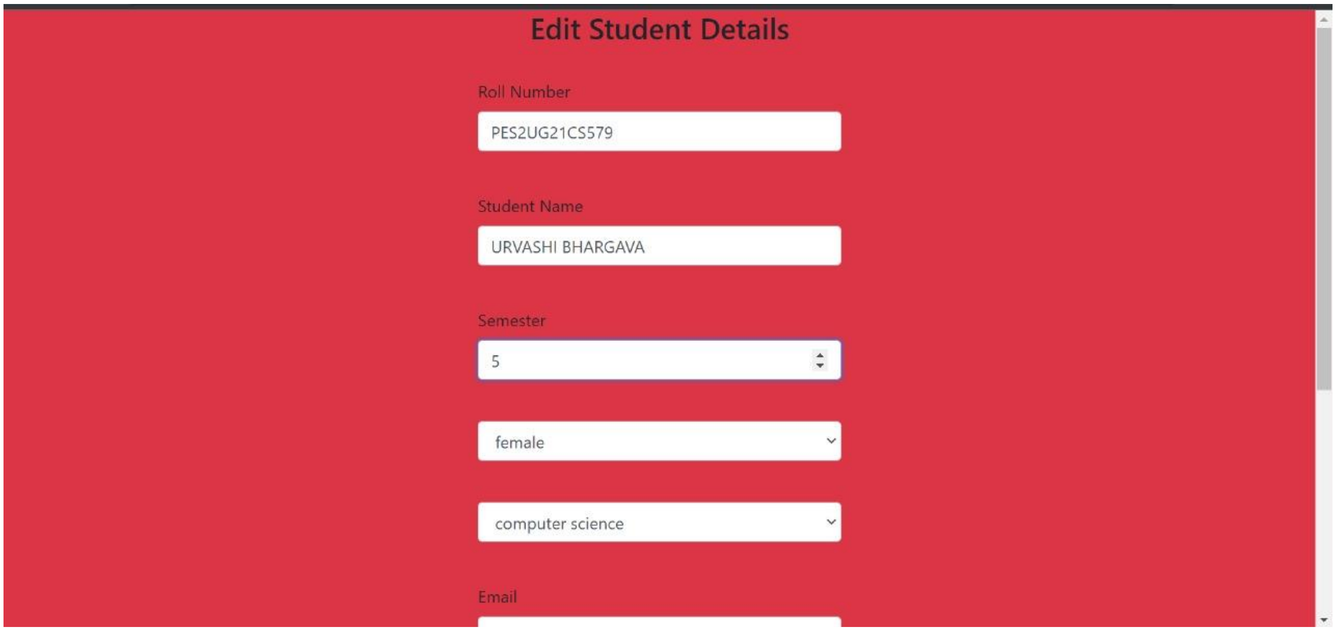
Female



Student Details

SID	ROLL NUMBER	STUDENT NAME	SEMESTER	GENDER	BRANCH	EMAIL ID	NUMBER	ADDRESS	EDIT	DELETE
7	PES2UG21CS579	URVASHI	5	female	computer science	urvashi@gmail.com	1234567899	somewhere	Edit	Delete
8	PES2UG21CS618	Teresha	5	female	computer science	teresha@gmail.com	1234567899	somehwere	Edit	De

2) UPDATE



Edit Student Details

Roll Number: PES2UG21CS579

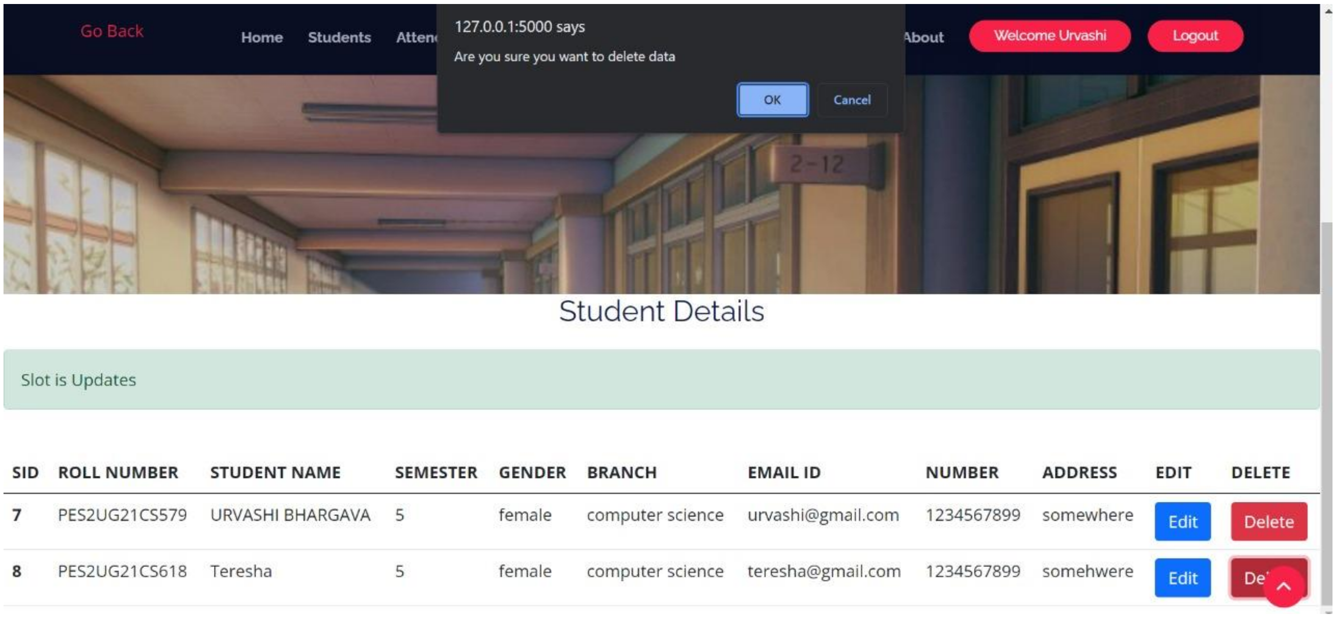
Student Name: URVASHI BHARGAVA

Semester: 5

female

computer science

Email



127.0.0.1:5000 says

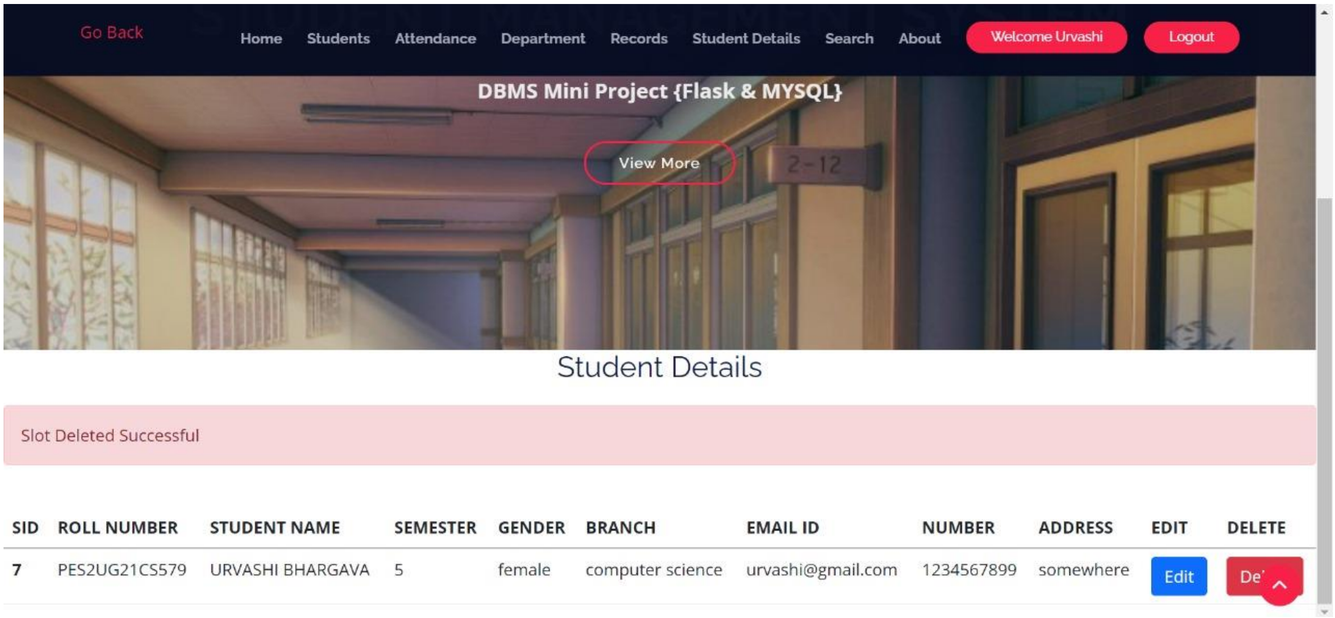
Are you sure you want to delete data

Student Details

Slot is Updates

SID	ROLL NUMBER	STUDENT NAME	SEMESTER	GENDER	BRANCH	EMAIL ID	NUMBER	ADDRESS	EDIT	DELETE
7	PES2UG21CS579	URVASHI BHARGAVA	5	female	computer science	urvashi@gmail.com	1234567899	somewhere	Edit	Delete
8	PES2UG21CS618	Teresha	5	female	computer science	teresha@gmail.com	1234567899	somehwere	Edit	De

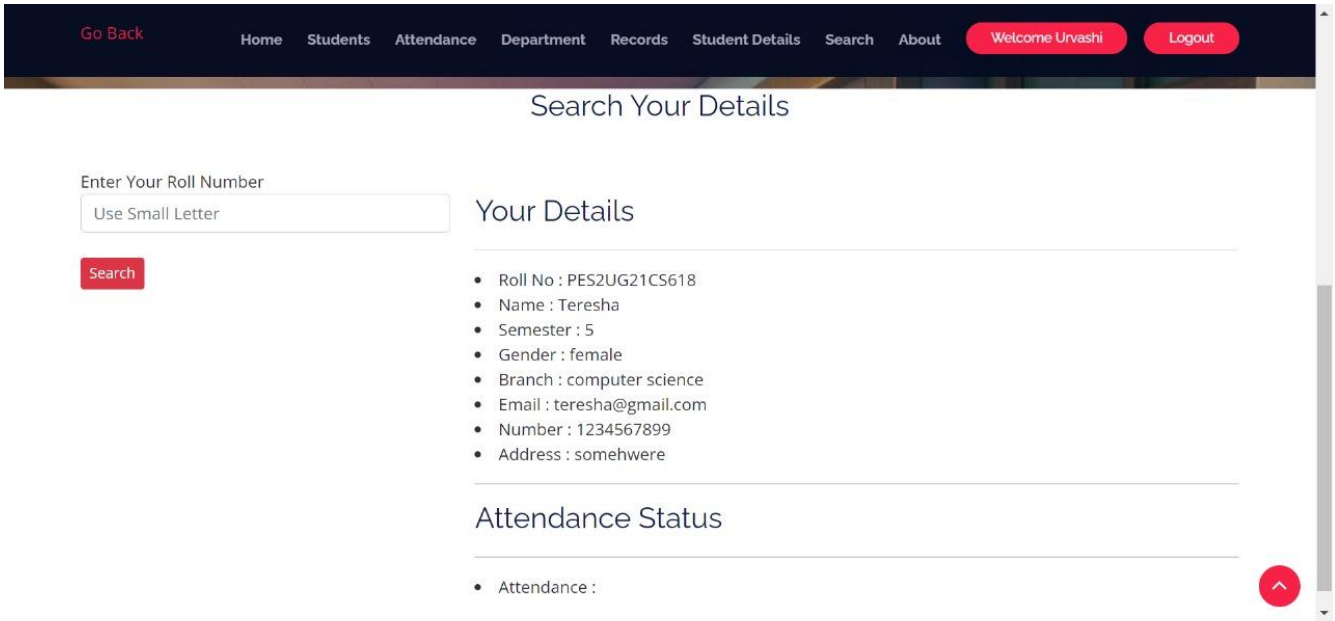
3) DELETE



Student Details

Slot Deleted Successful

SID	ROLL NUMBER	STUDENT NAME	SEMESTER	GENDER	BRANCH	EMAIL ID	NUMBER	ADDRESS	EDIT	DELETE
7	PES2UG21CS579	URVASHI BHARGAVA	5	female	computer science	urvashi@gmail.com	1234567899	somewhere	Edit	De'



Search Your Details

Enter Your Roll Number

Use Small Letter

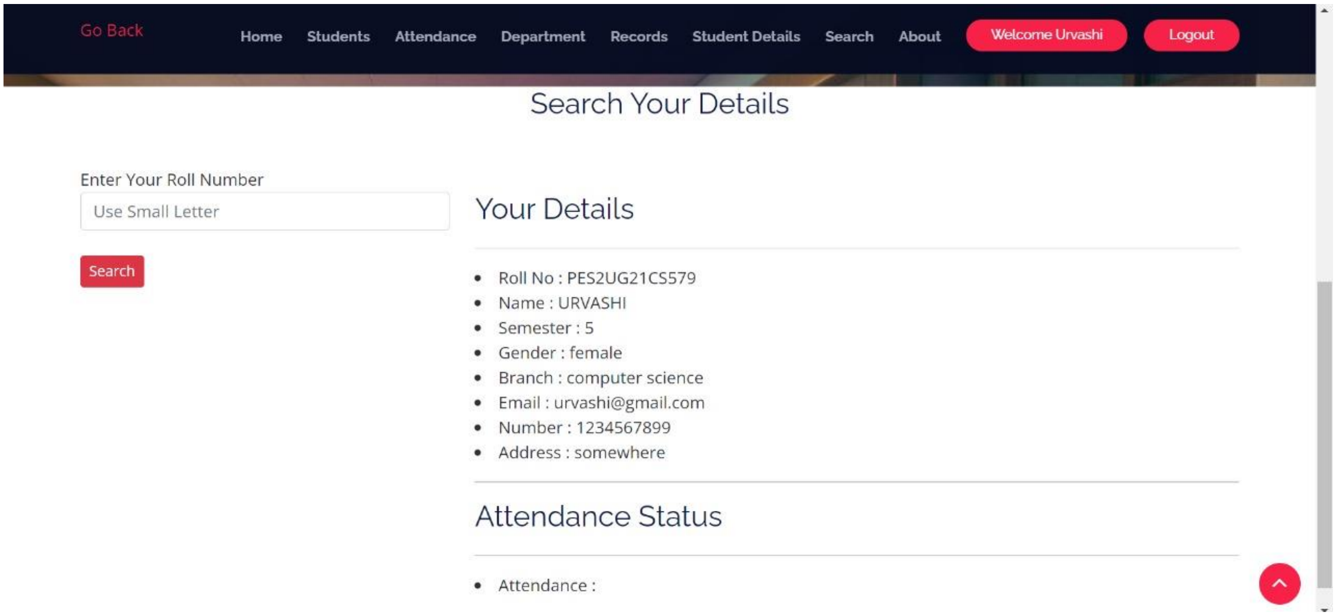
Search

Your Details

- Roll No : PES2UG21CS618
- Name : Teresha
- Semester : 5
- Gender : female
- Branch : computer science
- Email : teresha@gmail.com
- Number : 1234567899
- Address : somehwere

Attendance Status

- Attendance :



Search Your Details

Enter Your Roll Number

Use Small Letter

Search

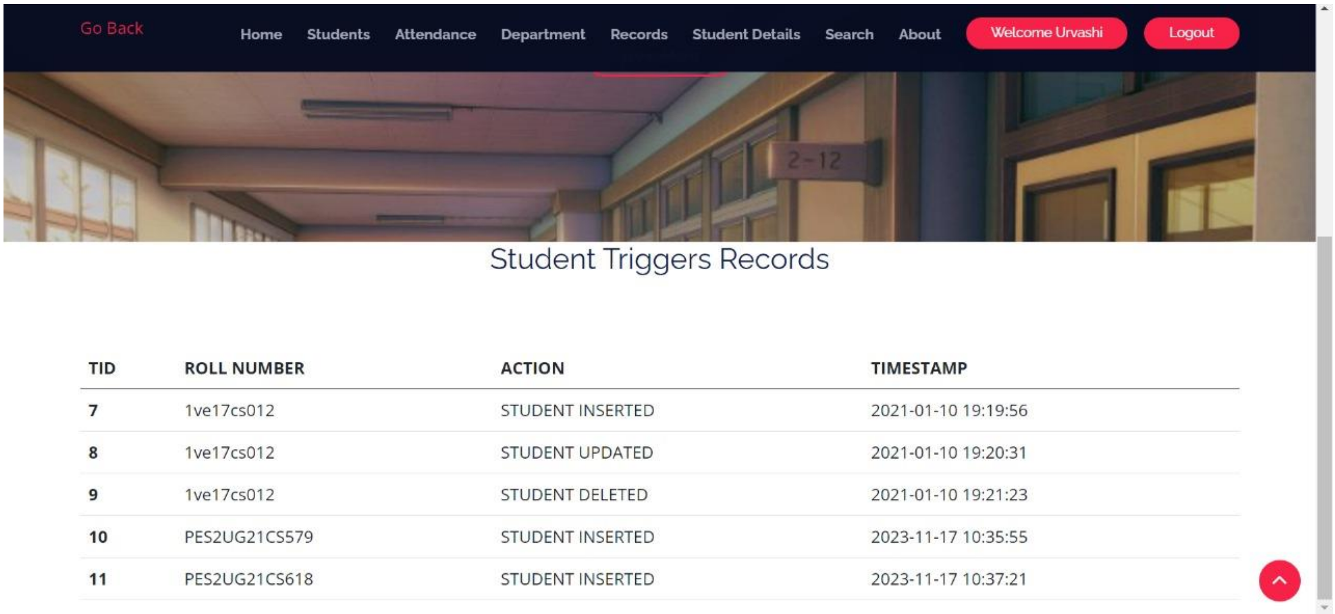
Your Details

- Roll No : PES2UG21CS579
- Name : URVASHI
- Semester : 5
- Gender : female
- Branch : computer science
- Email : urvashi@gmail.com
- Number : 1234567899
- Address : somewhere

Attendance Status

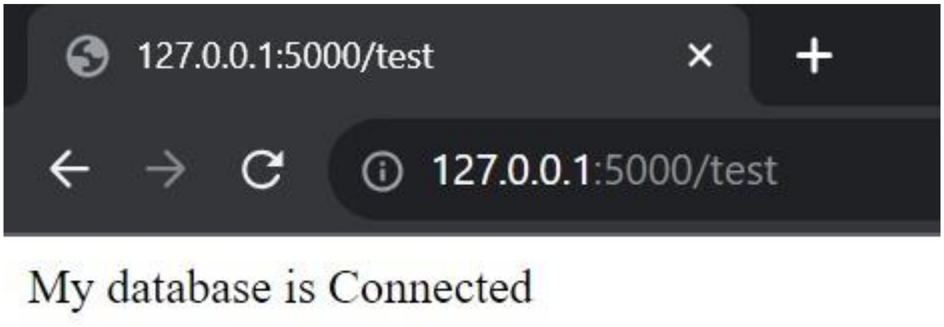
- Attendance :

TRIGGERS



Student Triggers Records

TID	ROLL NUMBER	ACTION	TIMESTAMP
7	1ve17cs012	STUDENT INSERTED	2021-01-10 19:19:56
8	1ve17cs012	STUDENT UPDATED	2021-01-10 19:20:31
9	1ve17cs012	STUDENT DELETED	2021-01-10 19:21:23
10	PES2UG21CS579	STUDENT INSERTED	2023-11-17 10:35:55
11	PES2UG21CS618	STUDENT INSERTED	2023-11-17 10:37:21



My database is Connected

REFERENCES

1) https://www.youtube.com
2) https://www.google.com
3) http://www.getbootstrap.com