

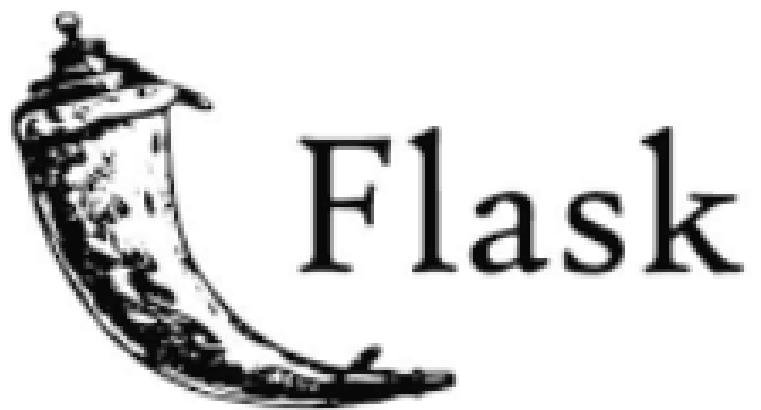
STUDENT MANAGEMENT SYSTEM



INTRODUCTION

Welcome to StudentHub – an Integrated Student Management System designed to modernize educational institutions' data handling. This project aims to revolutionize student data management by creating a streamlined, user-friendly database. Led by Urvashi Bhargava and Y Teresha at PES University, Bengaluru, under the guidance of Dr. Mannar Mannan J, this system prioritizes efficiency, data security, and accessibility, offering a scalable solution for educational institutions seeking to enhance their administrative processes.

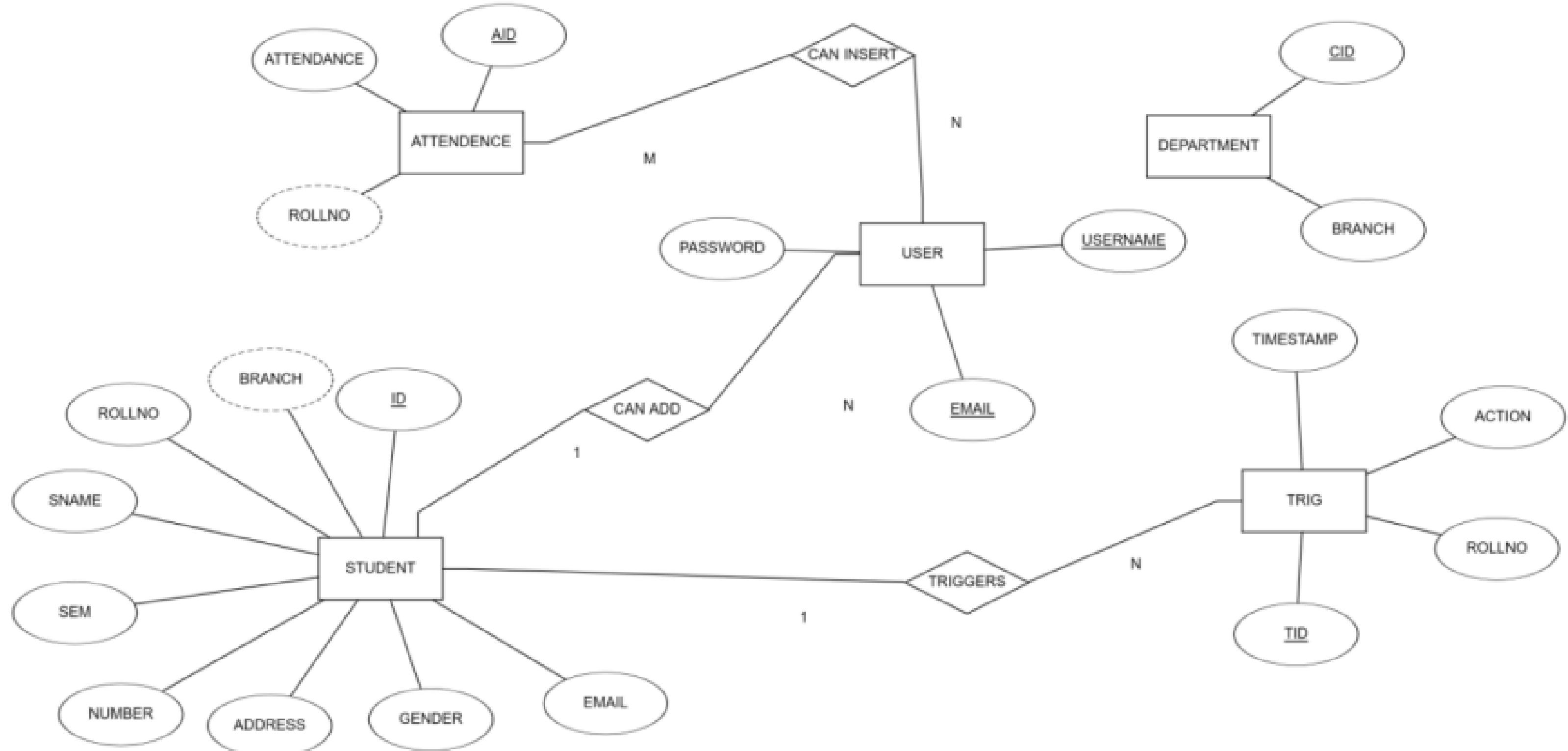
TECH STACK



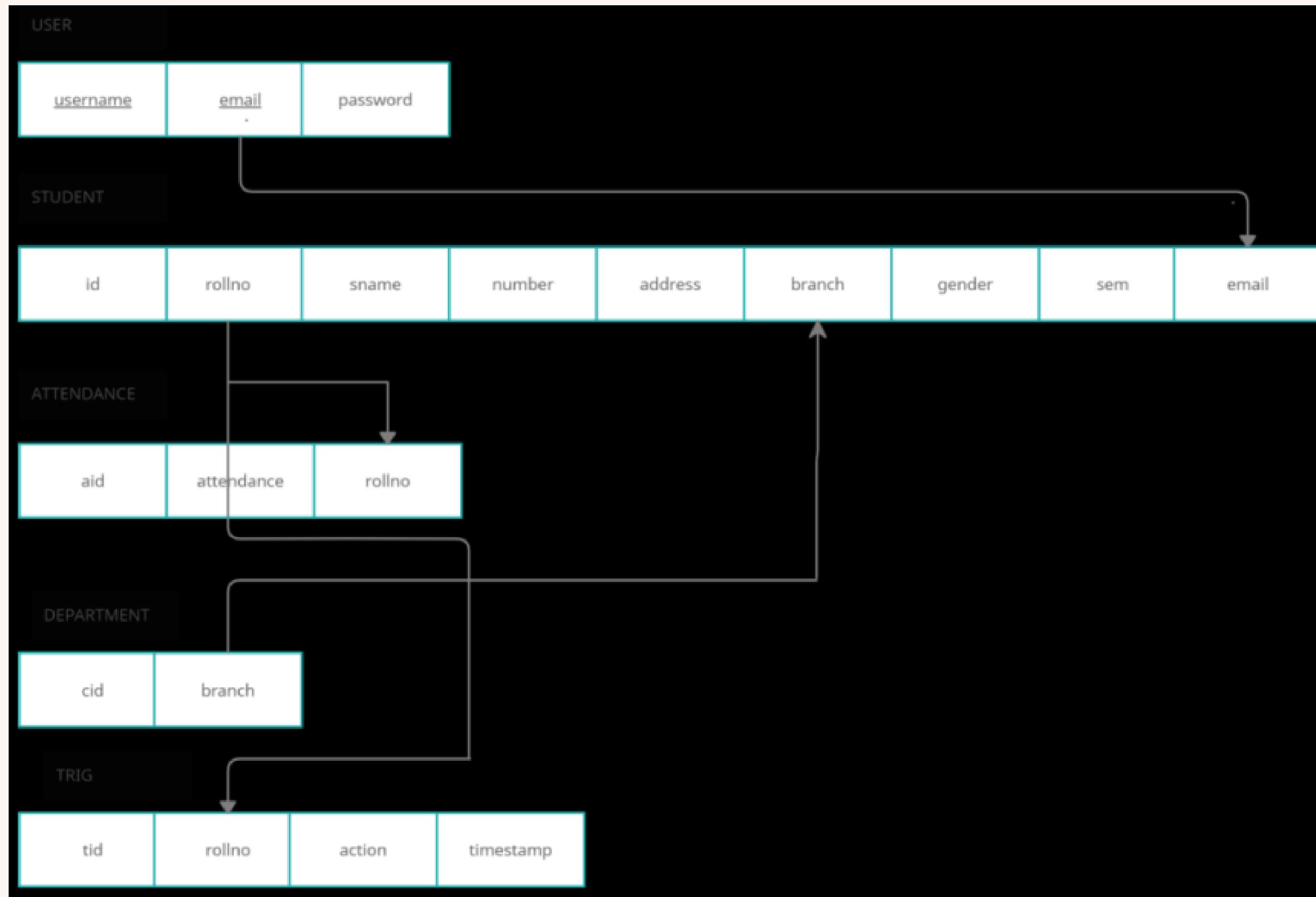
TABLES

Table	Action	Rows	Type	Collation	Size	Overhead
attendance	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	16.0 KiB	-
department	Browse Structure Search Insert Empty Drop	6	InnoDB	utf8mb4_general_ci	16.0 KiB	-
student	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	16.0 KiB	-
test	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	16.0 KiB	-
trig	Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_general_ci	16.0 KiB	-
user	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	16.0 KiB	-
6 tables	Sum		15	InnoDB	utf8mb4_general_ci	96.0 KiB
<input type="checkbox"/>	<input type="checkbox"/> Check all	With selected:				0 B

ER DIAGRAM



RELATIONSHIP SCHEMA



TRIGGERS

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())

QUERIES

1) SIMPLE QUERY WITH GROUP BY, AGRREGATE

```
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)
```

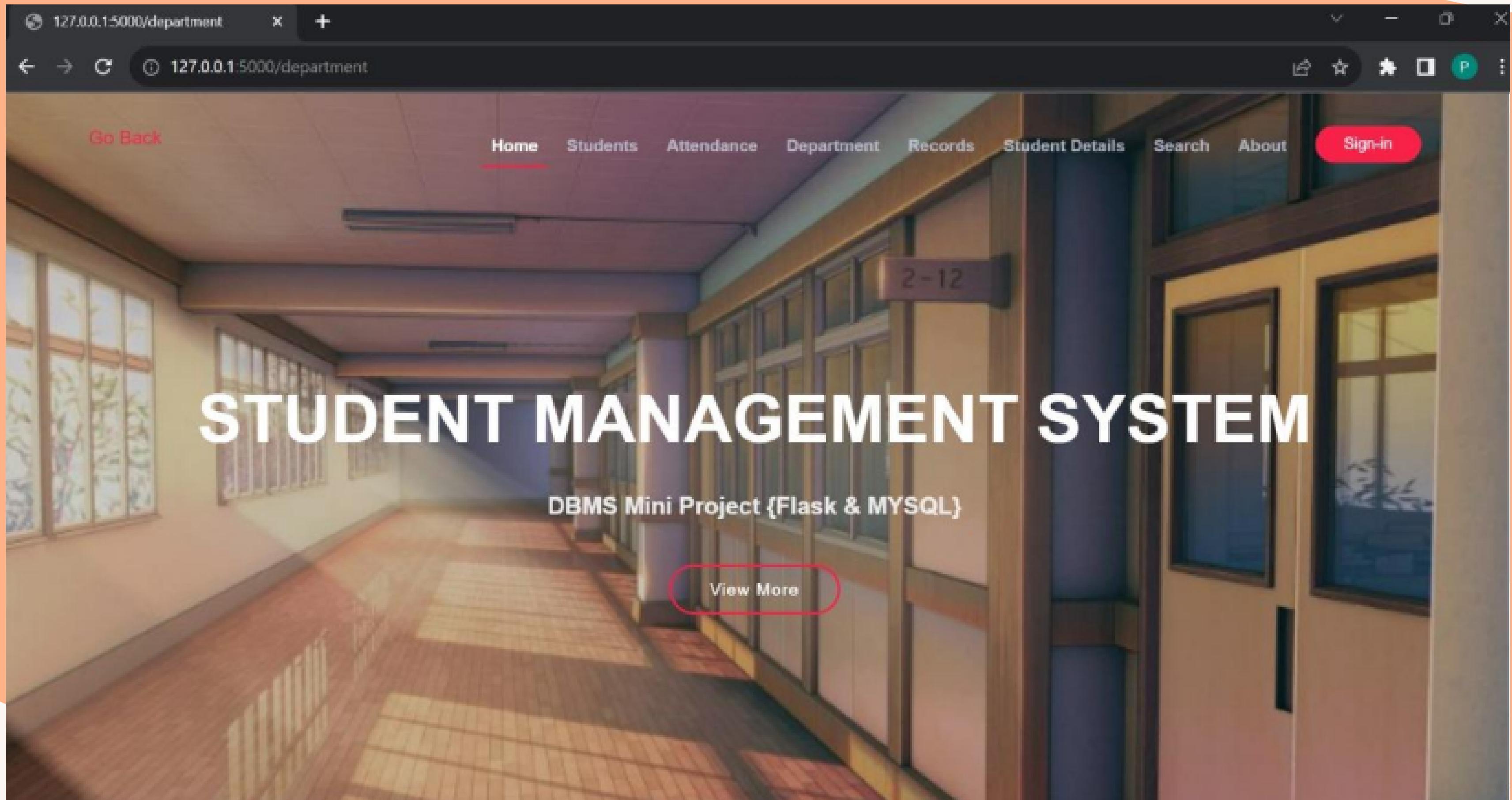
2) UPDATE OPERATION

```
@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)
```

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')
```

FRONT END



S.M.S.

LOGIN

Please log in to access this page.

Email address

Password

Login

New User?

Signup

[Go Back](#)[Home](#) [Students](#) [Attendance](#) [Department](#) [Records](#) [Student Details](#) [Search](#) [About](#)

Welcome Unashi

[Logout](#)[View More](#)

Add Attendance Details

Select RollNoAttendance Percentage

[Go Back](#)[Home](#)[Students](#)[Attendance](#)[Department](#)[Records](#)[Student Details](#)[Search](#)[About](#)

Welcome Urvashi

[Logout](#)

DBMS Mini Project {Flask & MYSQL}

[View More](#)

2-12

Add Department

Enter Department

[Add Department](#)

**THANK YOU
VERY MUCH!**