

# MAD LA 5

```
from flask import Flask, render_template, request, redirect
from flask_sqlalchemy import SQLAlchemy

app=Flask(__name__)
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///database.sqlite3'
db=SQLAlchemy(app)
```

```
class Student(db.Model):
    student_id=db.Column(db.Integer,primary_key=True,autoincrement=True)
    roll_number=db.Column(db.String(200),nullable=False,unique=True)
    first_name=db.Column(db.String(200),nullable=False)
    last_name=db.Column(db.String(200))

class Course(db.Model):
    courses = {'course_1':1, 'course_2':2, 'course_3':3, 'course_4':4}
    course_id=db.Column(db.Integer,primary_key=True,autoincrement=True)
    course_code=db.Column(db.String(200),nullable=False)
    course_name=db.Column(db.String(200),nullable=False)
    course_description=db.Column(db.String(200))

class Enrollments(db.Model):

    enrollment_id=db.Column(db.Integer,primary_key=True,autoincrement=True)
    estudent_id=db.Column(db.Integer,
db.ForeignKey('student.student_id'),nullable=False)
    ecourse_id=db.Column(db.Integer, db.ForeignKey('course.course_id'),
nullable=False)

with app.app_context():
    db.create_all()
```

```
@app.route('/',methods=['GET','POST'])
def home():
    students = Student.query.all()
    return render_template('home.html', students=students)
```

```

@app.route('/student/<int:student_id>', methods=['GET'])
def view(student_id):
    details_student =
Student.query.filter_by(student_id=student_id).first()
    enrollments =
Enrollments.query.filter_by(estudent_id=student_id).all()
    courses = []
    for enrollment in enrollments:
        course =
Course.query.filter_by(course_id=enrollment.ecourse_id).first()
        if course:
            courses.append(course)
    return render_template('about.html', student=details_student,
courses=courses)

```

```

@app.route('/student/create', methods=['GET', 'POST'])
def add():
    if request.method=='GET':
        return render_template('add.html')
    elif request.method=='POST':
        roll_number=request.form.get('roll')
        first_name=request.form.get('f_name')
        last_name=request.form.get('l_name')
        scourses=request.form.getlist('courses')

exist=Student.query.filter_by(roll_number=request.form['roll']).first()
    if exist is None:

db.session.add(Student(roll_number=request.form['roll'],first_name=request
.form['f_name'],last_name=request.form['l_name']))
        db.session.commit()
        courses=request.form.getlist('courses')
        for course in courses:

db.session.add(Enrollments(estudent_id=Student.query.filter_by(roll_number
=request.form['roll']).first().student_id,ecourse_id=Course.courses[course
]))
# OR USE

```

```

student =
Student.query.filter_by(roll_number=request.form['roll']).first()
student_id = student.student_id if student else None
course_id = Course.courses.get(course)
if student_id and course_id:
    enrollment = Enrollments(estudent_id=student_id, ecourse_id=course_id)
    db.session.add(enrollment)
    db.session.commit()

    db.session.commit()
    return redirect('/')
return render_template('exists.html')

```

```

@app.route('/student/<int:student_id>/delete', methods=['GET'])
def delete(student_id):
    Student.query.filter_by(student_id=student_id).delete()
    Enrollments.query.filter_by(estudent_id=student_id).delete()
    db.session.commit()
    return redirect('/')

```

```

@app.route('/student/<int:student_id>/update', methods=['GET', 'POST'])
def update(student_id):
    if request.method == 'GET':
        row = Student.query.filter_by(student_id=student_id).first()
        enrolls =
Enrollments.query.filter_by(estudent_id=student_id).all()
        cid = [enroll.ecourse_id for enroll in enrolls]
        print(cid)
        return render_template('update.html', row=row, cid=cid)
    elif request.method == 'POST':
        stud = Student.query.filter_by(student_id=student_id).first()
        stud.first_name = request.form['f_name']
        stud.last_name = request.form['l_name']
        Enrollments.query.filter_by(estudent_id=student_id).delete()
        for course in request.form.getlist('courses'):
            db.session.add(Enrollments(estudent_id=student_id,ecourse_id=Course.courses[course]))

```

```
db.session.commit()
return redirect('/')
```

```
if __name__ == '__main__':
    app.run(debug=True)
```