

UCS1617-MINI PROJECT SEMESTER PRACTICALS

Name: S.Nachammai Devi Pooja

Roll No: 185001096

Date:6/5/2021

Class & Sec: CSE-B

PROBLEM STATEMENT:

This implementation helps us to register courses . This course registration system is an interface between the course instructor (faculty) and the student who wants to enroll in the course. The faculty creates the courses and allows eligible student to get enrolled. The student can show their interest in the course by trying to enroll themselves in the course. If accepted by the faculty they can enlighten their knowledge in that course.

First both the faculty and student need to login by giving their login details. In case of a new student or faculty they can register. On authentication of their login details they will be taken to their respective pages . The Student can view their courses of interest and the details of the course are also displayed and they get to decide whether to enroll or not. The courses are of three types new , enrolled , pending . The details of each course such as the domain, faculty name, last date , pre requisite are mentioned here. The Faculty can create a new course and can also add a course with the details of the course. Even the courses already created can be viewed by the faculty. Once the Student has enrolled, the faculty can view the enrollment list of the students and can confirm or put the enrollment in pending based on their qualifications and the limit of the course. A table that is visible to both the faculty and the Student where that can view he/she is enrolled or not respectively.

IDENTIFICATION OF ACTORS :

USER:

- 1) Faculty
- 2) Student

Faculty : Faculty can create courses and accept students. They can view the enrolled list of students.

Student : The student can request to enroll in the course and check whether they are enrolled or not.

Both the users can register for new user.

IDENTIFICATION OF SCENARIOS:

- * Login :

In this page the based on the details given we find out whether the user is a student or faculty. Then they are taken to their respective pages.

- * Register :

In case of a new user student and faculty can register.

- * Creating a Course:

The faculty can create the courses and provide details.

- * Viewing course details:

Both the students and faculty can view the course details.

- * Enrolling the course :

The students can choose the courses they want to enroll here.

- * Confirmation of the enrollment :

The student can view if they have been enrolled .

- * Pending Confirmation of the enrollment:

The students can also view the courses in which they are still in waiting.

- * View enrolled list :

The faculty can view the enrolled students in the course created by them.

FULLY DRESSED USE CASE :

USE CASE : REGISTER FOR COURSE

Primary actor : Student

Stakeholder and interests :

Students : want to register for different courses easily and quickly.

Faculty : wants to complete the registration accurately and efficiently in less time, want to record the registered students with minimal effort, wants automatic and fast update of the inventory

Preconditions : Faculty is identified and authorized.

Success Guarantee/Post condition : Student is registered for the course successfully and the changes are reflected in the database.

Main Success Scenario :

The user fills a username and password and enter themselves as a student.

In case of a new user they can get registered.

The system validates the registration and updates the database.

The student views all available courses.

The student enters the course to be registered.

The system records the registration and updates.

The student is notified with approved course in enrolled courses table.

Alternative Flows :

* 1. At any time, system fails :

To support recovery and correct accounting, ensure all transaction sensitive state and events can be recovered from any step of the scenario.

2. If the validation fails or the database update fails, alert the user. The user has to register again.

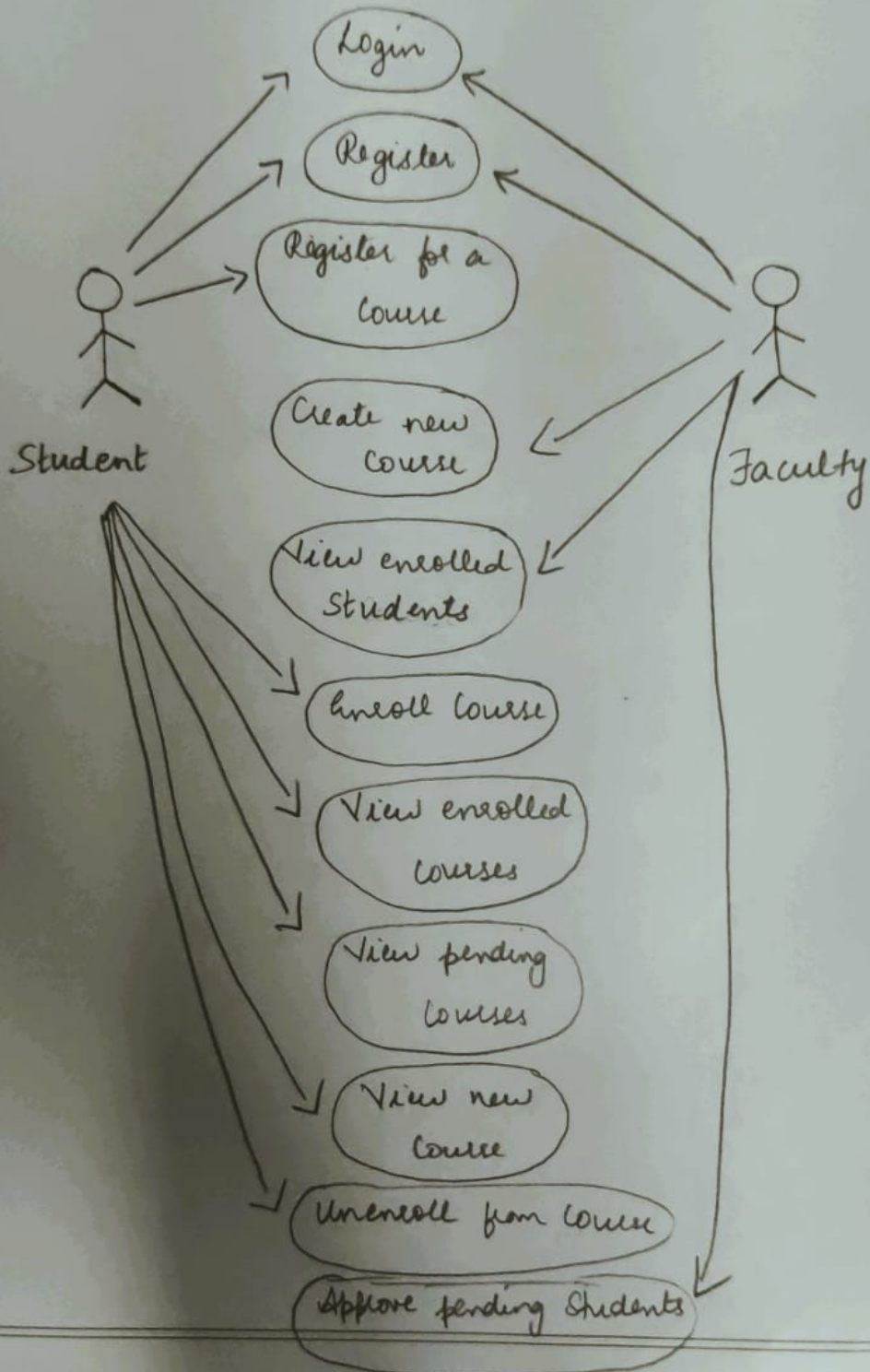
3. If the user enters invalid credentials, display an error message. The user has to login again

4. If no available courses in the database, display an appropriate message.

5. If the system update fails, return to the previous state.

USE CASE DIAGRAM:

Use - Case Diagram for Course Registration :-



IMPLEMENTATION:

app.py:

```
from operator import add
from re import A
from flask import Flask, render_template, url_for, request, redirect, session, g
from flask_sqlalchemy import SQLAlchemy
from sqlalchemy.orm import backref

app = Flask(__name__)
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///course.db'
app.static_folder = 'static'
app.config['SECRET_KEY'] = '185001112'
db = SQLAlchemy(app)

class Student(db.Model):
    __tablename__ = 'student'
    id = db.Column(db.Integer, primary_key=True)
    Name = db.Column(db.String(50), nullable=False)
    Password = db.Column(db.String(50), nullable=False)
    Email_id = db.Column(db.String(200), nullable=False)
    Contact = db.Column(db.String(15))
    Department = db.Column(db.String(30))

class Teacher(db.Model):
    __tablename__ = 'teacher'
    id = db.Column(db.Integer, primary_key=True)
    Name = db.Column(db.String(50), nullable=False)
    Password = db.Column(db.String(50), nullable=False)
    Email_id = db.Column(db.String(200), nullable=False)
    Contact = db.Column(db.String(15))
    Department = db.Column(db.String(30))
    courses = db.relationship('Course', backref=backref("teacher", uselist=False), lazy=True)

class Course(db.Model):
    __tablename__ = 'course'
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(50), nullable=False)
    last_date = db.Column(db.String(30))
    domain = db.Column(db.String(50), nullable=False)
    prerequisites = db.Column(db.String(100), nullable=False)
    teacher_id = db.Column(db.Integer, db.ForeignKey('teacher.id'), nullable=False)

applied = db.Table('applied',
    db.Column('course_id', db.Integer, db.ForeignKey('course.id'), primary_key=True),
    db.Column('student_id', db.Integer, db.ForeignKey('student.id'), primary_key=True),
    db.Column('appstatus', db.Integer)
)
```

```

@app.route('/', methods=['POST', 'GET'])
def login():
    if request.method == 'POST':
        username = request.form['uname']
        passwd = request.form['pass']
        selectstudent = Student.query.filter_by(Name=username).first()
        selectteacher = Teacher.query.filter_by(Name=username).first()
        if selectstudent and selectstudent.Password == passwd:
            session['type']=1
            session['user']=selectstudent.id
            g.user=selectstudent
            return redirect('/student')
        elif selectteacher and selectteacher.Password == passwd:
            session['type']=1
            session['user']=selectteacher.id
            g.user=selectteacher
            return redirect('/teacher')
        else:
            error = "User with Given Credentials not found"
            return render_template('login.html',error=error)
    else:
        session['type']=0
        session['user']=0
        return render_template('login.html',error=None)

@app.route('/student', methods=['POST', 'GET'])
def student():
    sel_id2=db.session.execute("SELECT course_id from 'applied' WHERE student_id= :s_id AND appstatus=1",{s_id:session["user"]}).fetchall()
    sel_id1=db.session.execute("SELECT course_id from 'applied' WHERE student_id= :s_id AND appstatus=0",{s_id:session["user"]}).fetchall()
    sel_id=db.session.execute("SELECT course_id from 'applied' WHERE student_id= :s_id",{s_id:session["user"]}).fetchall()
    sel=[]
    for s in sel_id:
        sel.append(s[0])
    allcourses=Course.query.filter(Course.id.notin_(sel)).all()
    sel=[]
    for s in sel_id1:
        sel.append(s[0])
    pendcourses=Course.query.filter(Course.id.in_(sel)).all()
    sel=[]
    for s in sel_id2:
        sel.append(s[0])
    appcourses=Course.query.filter(Course.id.in_(sel)).all()
    return
render_template('student.html',allcourses=allcourses,pendcourses=pendcourses,appcourses=appcourses)

@app.route('/teacher', methods=['POST', 'GET'])
def teacher():
    courses=Course.query.filter_by(teacher=Teacher.query.filter_by(id=session['user']).first()).all()

```

```

return render_template('faculty.html',courses=courses)

@app.route('/createcourse', methods=['GET','POST'])
def createcourse():
    if request.method == "POST":
        teacher = Teacher.query.filter_by(id=session['user']).first()
        course_name = request.form['subname']
        domain = request.form['domain']
        date = request.form['date']
        prereq = request.form['req']
        course =
Course(name=course_name,last_date=date,prerequisites=prereq,domain=domain,teacher=teacher)
        try:
            db.session.add(course)
            db.session.commit()
        except Exception as e:
            return("error")
        return redirect('/teacher')
    else:
        return redirect('/teacher')

@app.route('/studentsignup', methods=['POST', 'GET'])
def studentsignup():
    if request.method == 'POST':
        name = request.form['uname']
        passwd = request.form['pass']
        email = request.form['email']
        contact = request.form['phone']
        dept = request.form['dept']
        stud =
Student(Name=name>Password=passwd>Email_id=email>Contact=contact>Department=dept)
        try:
            db.session.add(stud)
            db.session.commit()
        except:
            return "error adding values to database"
            error = "Signed Up Successfully"
            return render_template('login.html',error=error)
    else:
        return render_template('studentsignup.html')

@app.route('/teachersignup', methods=['POST', 'GET'])
def teachersignup():
    if request.method == 'POST':
        name = request.form['uname']
        passwd = request.form['pass']
        email = request.form['email']
        contact = request.form['phone']
        dept = request.form['dept']
        teacher =
Teacher(Name=name>Password=passwd>Email_id=email>Contact=contact>Department=dept)
        try:

```

```

        db.session.add(teacher)
        db.session.commit()
    except:
        return "error adding values to database"
    error = "Signed Up Successfully"
    return render_template('login.html',error=error)

```

```

else:
    return render_template('teachersignup.html')

```

```

@app.route('/apply/<int:id>', methods=['POST', 'GET'])
def apply(id):
    d={"jid":id,"aid":session['user']}
    db.session.execute('INSERT INTO "applied" ("course_id", "student_id", "appstatus") VALUES
( :jid , :aid ,0)',d)
    db.session.commit()
    return redirect('/student')

```

```

@app.route('/seestudents/<int:id>', methods=['POST', 'GET'])
def seestudents(id):
    sel_id=db.session.execute("SELECT student_id from 'applied' WHERE course_id= :course_id
AND appstatus=0",{ 'course_id':id}).fetchall()
    sel=[]
    course=Course.query.filter_by(id=id).first()
    for sid in sel_id:
        sel.append(sid[0])
    pendstudents=Student.query.filter(Student.id.in_(sel)).all()
    sel_id=db.session.execute("SELECT student_id from 'applied' WHERE course_id= :course_id
AND appstatus=1",{ 'course_id':id}).fetchall()
    sel=[]
    for sid in sel_id:
        sel.append(sid[0])
    students=Student.query.filter(Student.id.in_(sel)).all()
    return
    render_template('viewdetails.html',pendstudents=pendstudents,course=course,students=students)

```

```

@app.route('/approve/<int:id>/<int:stid>', methods=['POST', 'GET'])
def approve(id,stid):
    db.session.execute("UPDATE 'applied' set 'appstatus'=1 WHERE course_id= :id AND
student_id= :stid",{ 'id':id,'stid':stid})
    db.session.commit()
    return redirect(url_for("seestudents",id=id))

```

```

@app.route('/delete/<int:id>', methods=['POST', 'GET'])
def delete(id):
    stid=session['user']
    print(id,stid)
    db.session.execute("DELETE FROM 'applied' WHERE course_id= :id AND student_id= :stid",
{'id':id,'stid':stid})
    db.session.commit()
    return redirect('/student')

```



```
@app.route('/view', methods=['POST', 'GET'])
def view():
    return render_template('view.html')
```

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

login.html:

```
<html>
  <head>
    <title>
      Course Registration System
    </title>
  </head>
  <body>
    <center>
      <h2 style="padding-top:200px;">NPTEL Course Registrations</h2><br>
      <form action="/" method="POST">
        <label for="uname">Username:</label><br>
        <input type="text" id="uname" name="uname"><br>
        <label for="pass">Password:</label><br>
        <input type="text" id="pass" name="pass"><br><br>
        <input type="submit" value="Submit">
      </form>
      <div>
        {%if error%}
          <h4 style="margin-left: 4.5rem;">{{error}}</h4>
        {% endif%}
      </div><br>
      <a href="/studentsignup"> <button>Signup as Student</button> </a>
      <a href="/teachersignup"> <button>Signup as Faculty</button> </a>
    </center>
  </body>
</html>
```

student.html:

```
<html>
<head>

</head>
<style>
  table {
    font-family: arial, sans-serif;
    border-collapse: collapse;
    width: 100%;
  }
```

```

td, th {
    border: 1px solid #dddddd;
    text-align: left;
    padding: 8px;
}

tr:nth-child(even) {
    background-color: #dddddd;
}
</style>
<body>
    <h2 style="padding-top:200px;">Welcome Student !!!</h2><br>
    <a href="/"><button>Logout</button></a>
<h3>List of New Courses:</h3>
<table>
    <tr>
        <th>Subject Name</th>
        <th>Domain</th>
        <th>Teacher Name</th>
        <th>LastDate</th>
        <th>PreRequisite</th>
        <th>Apply</th>
    </tr>
    {%for course in allcourses%}
    <tr>
        <td>{{ course.name }}</td>
        <td>{{ course.domain }}</td>
        <td>{{ course.teacher.Name }}</td>
        <td>{{ course.last_date }}</td>
        <td>{{ course.prerequisites }}</td>
        <td><a href="/apply/{{ course.id }}"><button>Apply</button></a></td>
    </tr>
    {%endfor%}

</table>
<h3>List of Enrolled Courses:</h3>
<table>
    <tr>
        <th>Subject Name</th>
        <th>Domain</th>
        <th>Teacher Name</th>
        <th>PreRequisite</th>
        <th>View Details</th>
    </tr>
    {%for course in appcourses%}
    <tr>
        <td>{{ course.name }}</td>
        <td>{{ course.domain }}</td>
        <td>{{ course.teacher.Name }}</td>
        <td>{{ course.prerequisites }}</td>
        <td><a href="/view"><button>View</button></a></td>
    </tr>

```

```

    {%endfor%}

</table>

<h3>List of Pending Courses:</h3>
<table>
  <tr>
    <th>Subject Name</th>
    <th>Domain</th>
    <th>Teacher Name</th>
    <th>PreRequiste</th>
    <th>Unenroll</th>
  </tr>
  {%for course in pendcourses%}
  <tr>
    <td>{{ course.name }}</td>
    <td>{{ course.domain }}</td>
    <td>{{ course.teacher.Name }}</td>
    <td>{{ course.prerequisites }}</td>
    <td><a href="/delete/{{ course.id }}"><button>Delete</button></a></td>
  </tr>
  {%endfor%}

</table>
</body>
</html>

```

faculty.html

```

<html>
  <head>

  </head>
  <body>
    <a href="/"><button>Logout</button></a>
    <h3>Create New Course:</h3>
    <form action="createcourse" method="POST">
      <label for="subname">Subject Name:</label><br>
      <input type="text" id="uname" name="subname"><br>
      <label for="domain">Domain:</label><br>
      <input type="text" id="domain" name="domain"><br>
      <label for="req">Prerequisite:</label><br>
      <input type="text" id="request" name="req"><br>
      <label for="date">Last Date to Apply:</label><br>
      <input type="text" id="date" name="date"><br><br>
      <input type="submit" value="Submit">
    </form>
    <style>
      table {
        font-family: arial, sans-serif;
        border-collapse: collapse;
        width: 100%;

```

```

    }

    td, th {
        border: 1px solid #dddddd;
        text-align: left;
        padding: 8px;
    }

    tr:nth-child(even) {
        background-color: #dddddd;
    }
</style>

<h3>List of Created Courses:</h3>
<table>
  <tr>
    <th>Course Name</th>
    <th>View Deatils</th>

  </tr>
  {%for course in courses%}
  <tr>
    <td>{{ course.name }}</td>

    <td><a href="/seestudents/{{ course.id }}"><button>View</button></a></td>
  </tr>
  {%endfor%}

</table>
</body>
</html>

```

studentsignup.html

```

<html>
  <head>

  </head>
  <body>
    <center>
      <h2 style="padding-top:200px;">Student Signup</h2><br>
      <h3>Enter the following details:</h3>
      <form action="/studentsignup" method='POST'>
        <label for="uname">Username:</label><br>
        <input type="text" id="uname" name="uname"><br>
        <label for="pass">Password:</label><br>
        <input type="text" id="pass" name="pass"><br>
        <label for="dept">Department:</label><br>
        <input type="text" id="dept" name="dept"><br>
        <label for="email">EmailId:</label><br>
        <input type="email" id="email" name="email"><br>
        <label for="phone">Phone Number:</label><br>

```

```

        <input type="text" id="phone" name="phone"><br>
        <input type="submit" value="Submit">
    </form>
</center>
</body>
</html>

```

teachersignup.html

```

<html>
  <head>

  </head>
  <body>
    <center>
      <h2 style="padding-top:200px;">Faculty Signup</h2><br>
      <h3>Enter the following details:</h3>
      <form action="/teachersignup" method='POST'>
        <label for="uname">Username:</label><br>
        <input type="text" id="uname" name="uname"><br>
        <label for="pass">Password:</label><br>
        <input type="text" id="pass" name="pass"><br>
        <label for="dept">Department:</label><br>
        <input type="text" id="dept" name="dept"><br>
        <label for="email">EmailId:</label><br>
        <input type="email" id="email" name="email"><br>
        <label for="phone">Phone Number:</label><br>
        <input type="text" id="phone" name="phone"><br>
        <input type="submit" value="Submit">
      </form>
    </center>
  </body>
</html>

```

view.html:

```

<html>
  <head>

  </head>
  <body>
    <a href="/student"><button>Back</button></a>
    <p> You have been successfully enrolled in this course.<br>
    Please wait for the faculty to post more contents. </p>
    <h3> Enjoy Learning </h3>
  </body>
</html>

```

viewdetails.html:

```

<html>
  <head>

```

```

</head>
<style>
    table {
        font-family: arial, sans-serif;
        border-collapse: collapse;
        width: 100%;
    }

    td, th {
        border: 1px solid #dddddd;
        text-align: left;
        padding: 8px;
    }

    tr:nth-child(even) {
        background-color: #dddddd;
    }
</style>
<body>
    <a href="/teacher"><button>Back</button></a>
<h3>List of Approved Candidates:</h3>
<table>
    <tr>
        <th>Name</th>
        <th>Email</th>
        <th>Department</th>

    </tr>
    {%for student in students%}
    <tr>
        <td>{{student.Name}}</td>
        <td>{{student.Email_id}}</td>
        <td>{{student.Department}}</td>
    </tr>
    {%endfor%}
</table>
<h3>List of Enrolled Courses:</h3>
<table>
    <tr>
        <th>Name</th>
        <th>Email</th>
        <th>Department</th>
        <th>Approve Student</th>
    </tr>
    {%for student in pendstudents%}
    <tr>
        <td>{{student.Name}}</td>
        <td>{{student.Email_id}}</td>
        <td>{{student.Department}}</td>
        <td><a href="/approve/{{course.id}}/{{student.id}}"><button>Approve</button></a></td>
    </tr>
    {%endfor%}

```

```
</table>
</body>
</html>
```

OUTPUT:-

Login.html

Course Registration System

127.0.0.1:5000

096_Nachamm...

NPTEL Course Registrations

Username:

Password:

Student Login:

Course Registration System

127.0.0.1:5000

096_Nachamm...

NPTEL Course Registrations

Username:

Password:

127.0.0.1:5000/student

127.0.0.1:5000/student

096_Nachamm...

Welcome Student !!!

Logout

List of New Courses:

Subject Name	Domain	Teacher Name	LastDate	PreRequisite	Apply
IP	CSE	Nachammai	2021-06-20	HTML,CSS,JS	<div>Apply</div>

List of Enrolled Courses:

Subject Name	Domain	Teacher Name	PreRequisite	View Details
OOAD	CSE	Manisha	Software Engineering	<div>View</div>
Miniproject	CSE	Manisha	OOAD	<div>View</div>

List of Pending Courses:

Subject Name	Domain	Teacher Name	PreRequisite	Unenroll
CD	CSE	Nachammai	TOC	<div>Delete</div>

127.0.0.1:5000/view

127.0.0.1:5000/view

Back

You have been successfully enrolled in this course.
Please wait for the faculty to post more contents.

Enjoy Learning

096_Nachamm...

Teacher Login:

Course Registration System

127.0.0.1:5000

096_Nachamm...

NPTEL Course Registrations

Username:

Manisha

Password:

5678

Submit

Signup as Student

Signup as Faculty

127.0.0.1:5000/teacher

127.0.0.1:5000/teacher

096_Nachamm...

Logout

Create New Course:

Subject Name:

OOAD

Domain:

CSE

Prerequisite:

Software Engineering

Last Date to Apply:

2021-07-22

Submit

List of Created Courses:

Course Name	View Deatils
OOAD	<div>View</div>
Miniproject	<div>View</div>

127.0.0.1:5000/seestudent

127.0.0.1:5000/seestudents/1

Back

List of Approved Candidates:

Name	Email	Department
Pooja	ndpooja0305@gmail.com	CSE
Devi	ndpooja0305@gmail.com	CSE

List of Enrolled Courses:

Name	Email	Department	Approve Student
------	-------	------------	-----------------

New Registration:

096_Nachamm...

Student Signup

Enter the following details:

Username:

Prabha

Password:

1234

Department:

EEE

EmailId:

ramaprabha@gmail.com

Phone Number:

9952092232

Submit

127.0.0.1:5000/teachersignup

096_Nachamm...

Faculty Signup

Enter the following details:

Username:

Nachammai

Password:

5678

Department:

CSE

EmailId:

ndpooja0305@gmail.com

Phone Number:

9941330536

Submit

DATABASE:

File Edit View Help

New Database Open Database Write Changes Revert Changes

Database Structure Browse Data Edit Pragmas Execute SQL

Table: teacher

	id	Name	Password	Email_id	Contact	Department
Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	Nachammai	5678	ndpooja030...	9941330536	CSE
2	2	Manisha	5678	manishas@...	9876543210	CSE
3	3	Praveen	5678	vpraveenpl...	9894440536	ECE

New Record Delete Record

File Edit View Help

New Database Open Database Write Changes Revert Changes

Database Structure Browse Data Edit Pragmas Execute SQL

Table: student

	id	Name	Password	Email_id	Contact	Department
Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	Pooja	1234	ndpooja030...	9941330536	CSE
2	2	Devi	1234	ndpooja030...	9941330536	CSE
3	3	Prathyush	1234	prathyushs...	6382072914	ECE
4	4	Prabha	1234	ramaprabha...	9952092232	EEE
5	5	Bharath	1234	bharathkum...	8820827192	CSE

New Record Delete Record

File Edit View Help					
New Database Open Database Write Changes Revert Changes					
Database Structure Browse Data Edit Pragmas Execute SQL					
Table: course					
	id	name	last_date	domain	prerequisites
	Filter	Filter	Filter	Filter	Filter
1	1	OOAD	2021-07-22	CSE	Software En...
2	2	Miniproject	2021-06-05	CSE	OOAD
3	3	IP	2021-06-20	CSE	HTML.CSS,JS
4	4	CD	2020-06-06	CSE	TOC

IDENTIFICATION OF TEST CASE SCENARIO :-

- * Teacher Login
- * Student Login
- * New Registration
- * Creating a course
- * Enrolling a course
- * Course confirmation (or) Pending Details

Test Cases:

Test_ID	Test Scenario	Test Step	Test Data	Predicted Result	Actual Result	Pass/Fail
T01	Teacher Login	Enter the Username & Password	Username: Pooja Password:1234	Logged in	Logged in	Pass
T02	Teacher Login	Enter the Username & Password	Username: Raja Password:7654	Invalid, Try Again	Logged in	Fail
T03	Student Login	Enter Username & Password	Username: Swetha Password:1234	Logged in	Logged in	Pass
T04	Student Login	Enter Username & Password	Username: Suresh Password: 1234	Invalid, Try again	Logged in	Fail
T05	New Registration	Enter new Username and Password	Username: Roja Password:1234	Registered	Registered	Pass
T06	Creating a Course	After the Teacher is logged in to the system where the create resume	Name: OOAD Teacher: Manisha Last Date of Enrolling: 08/05/2021 Button clicked: Create Course	Course created Successfully	Course created Successfully	Pass
T07	Creating a Course	After the Teacher is logged in to the system where the create course option is present	Name: OOAD Teacher: Manisha Last Date of Enrolling: Button clicked: Create Course	Fill up the required field	Course created Successfully	Fail
T08	Enrolling a Course	After the Student is logged in to the System where the enroll course option is present	Select the Course: OOAD Information about the course: The course is handled by Manisha. The last date is 08/05/2021 Button clicked: Enroll course	Enrolled Successfully	Enrolled Successfully	Pass
T09	Enrolling a Course	After the Student is logged in to the System where the enroll course option is present	Select the Course: Information about the course: The course is handled by Manisha. The last date is 08/05/2021 Button clicked: Enroll	Fill up the required Field	Enrolled Successfully	Fail

			course			
T10	Confirmation of course by the Teacher	After the Student selects the course and enrolls, the teacher confirms the course.	Student Name: Pooja Enrolled Course: OOAD Button clicked: Confirm Enrollment	Enrollment Confirmed and added to the table	Enrollment Confirmed and added to the table	Pass
T11	Confirmation of course by the Teacher	After the Student selects the course and enrolls, the teacher confirms the course.	Student Name: Pooja Enrolled Course: OOAD Button clicked: Confirm Enrollment	Enrollment Pending	Enrollment Pending	Fail

Improvements:

With the help of test cases I ensured that if different features within an application are working as expected. It validates whether the implementation is free of defects and is working as per the expectations of the end users. The improvements we can make in this project are:

- We can have an attribute like number of seats available. Thus students will be able to see number of seats available. This will help the teacher also know how many students can get enrolled and give the opportunity to eligible students.
- We can have an admin who takes care of all registration instead of burdening the teacher.
- We can include an option to pay fee after getting accepted for the requested course.
- We can keep a deadline for enrolling the course.
- We can allow the teacher to view the academic details to be able to view based on their merits.