Dharmsinh Desai University, Nadiad

Faculty of Technology,

Department of Computer Engineering

B. Tech. CE Semester – VI

Subject: (CE – 620) Object Oriented Software Engineering

Project title: Assignment Management System

Made By:

Devanshi Sheth,

Roll No: CE-127

ID No: 18CEUON118

Guided by:

Prof. Mrudang T. Mehta

Asso. Professor, CE Dept., Dharmsinh Desai University

DHARMSINH DESAI UNIVERSITY
NADIAD-387001, GUJARAT

CERTIFICATE

This is to certify that the project entitled "**Assignment Management System**" is a bonafied report of the work carried out by:

Devanshi K Sheth, Student ID No: **18CEUON118**

of Department of Computer Engineering, semester VI, under the guidance and supervision for the subject Object Oriented Software Engineering. They were involved in Project training during academic year 2020-2021.

Prof. Mrudang T. Mehta	Dr. C.K Bhensdadia
Associate Professor	Prof. and Head,
Department of Computer Engineering,	Department of Computer Engineering,
Faculty of Technology,	Faculty of Technology,
Dharmsinh Desai University, Nadiad	Dharmsinh Desai University, Nadiad

Table of Contents

ABSTRACT

This document examines the design and implementation of an online assignment management system. Many reported failures in courses can be attributed to the carelessness of the students who failed to submit an assignment to the teacher for marking. Such carelessness could lead to a zero mark for the student whose paper was declared missing or simply not found. The manual method of submitting assignments directly to the teacher is simply not effective as these papers or hard copy could get damaged or get missing by the teacher or the student. Moreover, assignments that are large in terms of pages or volume could easily discourage a student from submitting it.

Moreover, The challenges faced by students in meeting assignment deadlines and cost associated with printing hard copies of paper, made a necessity to develop a user friendly system to tackle these challenges. The system allows students to submit assignments online to a particular course enrolled by them, and can get those assignments reviewed by the teachers who have access to the system for grading purpose.

INTRODUCTION

An online assignment management system extends the functionality of the standard assignment handling by catering for all the Department’s needs in terms of receiving assignments from students, making them available to teachers to mark, returning grades, comments and marked work to students and keeping records of those submission.

Universities, Polytechnics and colleges of education are considered the main provider of knowledge in various fields. Various courses of studies are taught in institutions, covering several fields. Most courses at universities consist of theoretical as well as practical subject matter. To evaluate the level of understanding and degree of comprehension among students, assignments are often given.

Assignments are submitted by students either individually or in groups. Assignment management involves collecting, marking, and redistributing to students. With traditional assignment management system, lots of problems arises. There may be problems due to distance, time, or format of the assignment (written or printed).

As the educational world is moving faster and becoming more competitive, almost every university started to use an online submission system, or newer technologies to facilitate their task, to have more time, and to be in pace with this fast moving IT world.

Technology Used:

- Django Web Framework
- SQLite Database

Platform Used:

- Visual Studio Code
- Selenium for Testing

SOFTWARE REQUIREMENTS SPECIFICATIONS (SRS)

1. Functional Requirements:

1.1 Manage Teacher:

1.1.1 Teacher Registration

Description: Registration of new Teacher

Input: Teacher Details like name, Email, Password etc.

Output: Teacher's account will be created and redirected to Login Page.

1.1.2 Login

Description: Teacher will login from his/her account.

Input: Teacher Details like Email, Password.

Output: Teacher will land on Home Page.

1.1.3 Add new Course

Description: Teacher can add new Course from his account.

Input: Course Details

Output: Course successfully added

1.1.4 Add new Assignment

Description: Teacher can add new assignment in a particular course.

Input: Assignment Details

Output: Assignment successfully added.

1.1.5 View Submission

Description: Teachers can view submitted assignments of particular course and as well as of particular student

Input: User input

Output: Submission list of assignment is shown.

1.1.6 Review Assignment

Description: Teacher can give marks to the submitted assignment of a student

Input: User input

Output: Marks added to the assignment.

1.1.7 Delete Assignment

Description: Teacher will be able to delete uploaded assignments

Input: User input

Output: Confirmation message

1.2 Manage Student:

1.2.1 Student Registration

Description: Registration of new Student

Input: Student Details like name, Email, Password etc

Output: Student account will be created and redirected to Login Page

1.2.2 Login

Description: Student will login from his account

Input: Student Details like Email, Password

Output: Student will land on Home Page

1.2.3 Enroll into Courses

Description: Students can enroll into courses of their branch

Input: user input

Output: Saved to Database

1.2.4 View Assignments

Description: Students can view pending assignments of their enrolled courses

Input: User login

Output: List of Assignment is shown

1.2.5 Upload Submissions

Description: Student can upload his/her assignment for Submission

Input: User input

Output: Submission successfully done

1.2.6 View Marks

Description: Student can view marks of their submitted assignment.

Input: User input

Output: List of Submission is shown.

2. Other Non-Functional Requirements

2.1 Performance Requirements

The performance of the system is better due to strong connectivity to database, adding deleting data and archiving database so the system gives less response time.

2.2 Safety Requirements

Humans are error-prone, but the negative effects of common errors should be limited

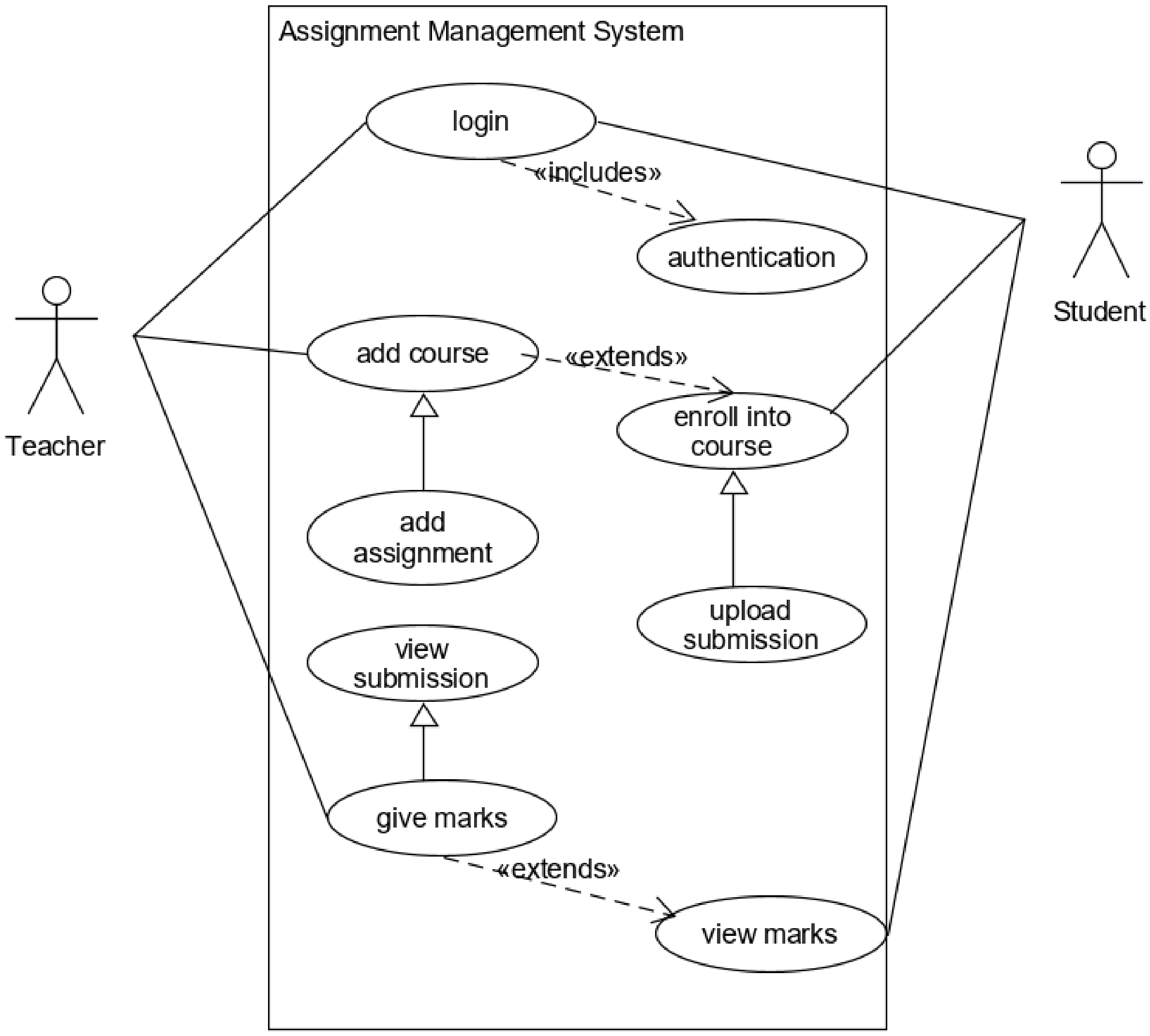
2.3 Security Requirements

The system requires the user to identity himself/herself using username or Email-id

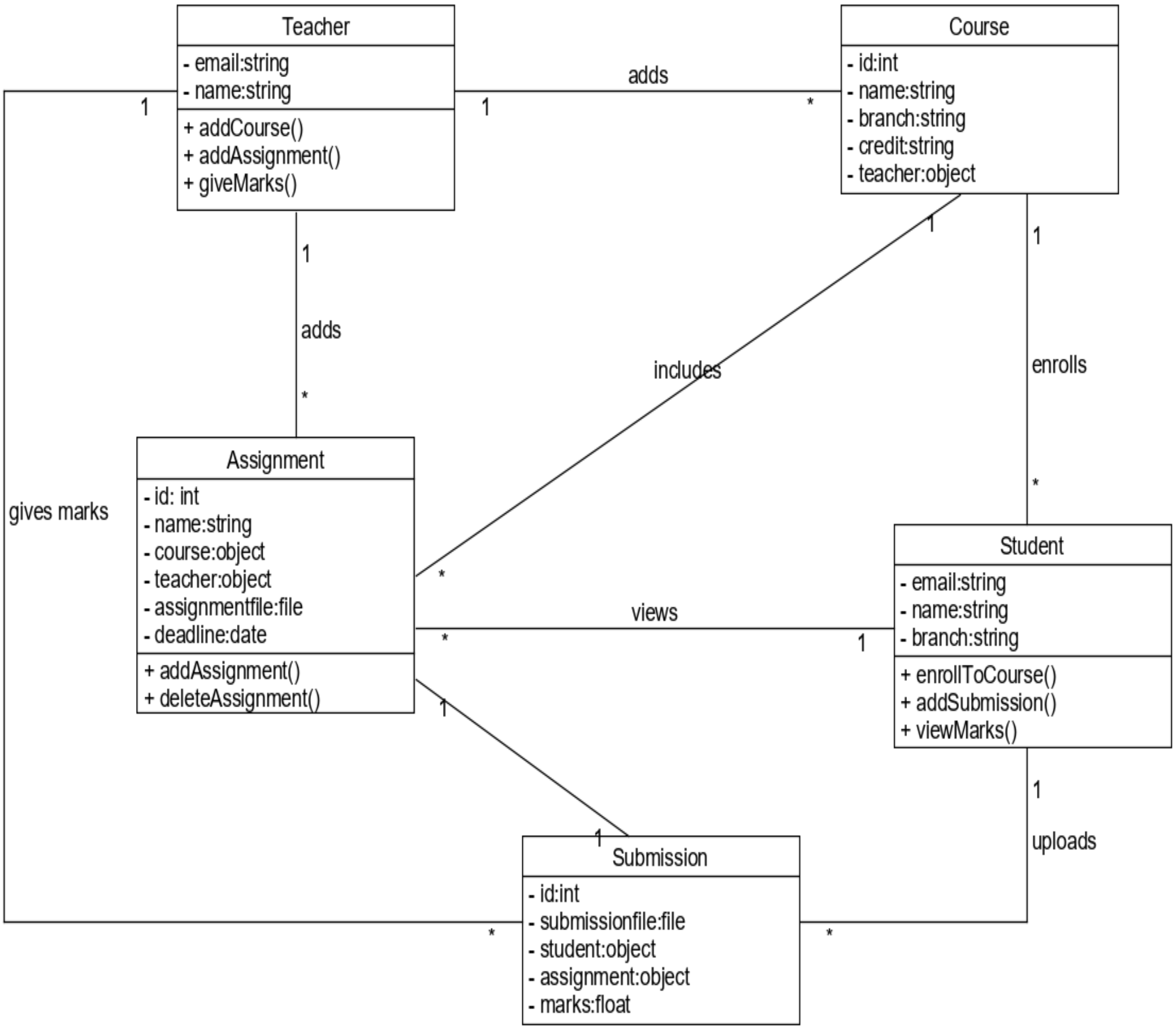
Any user who uses the system shall have a Unique User ID and Password

ANALYSIS AND DESIGN

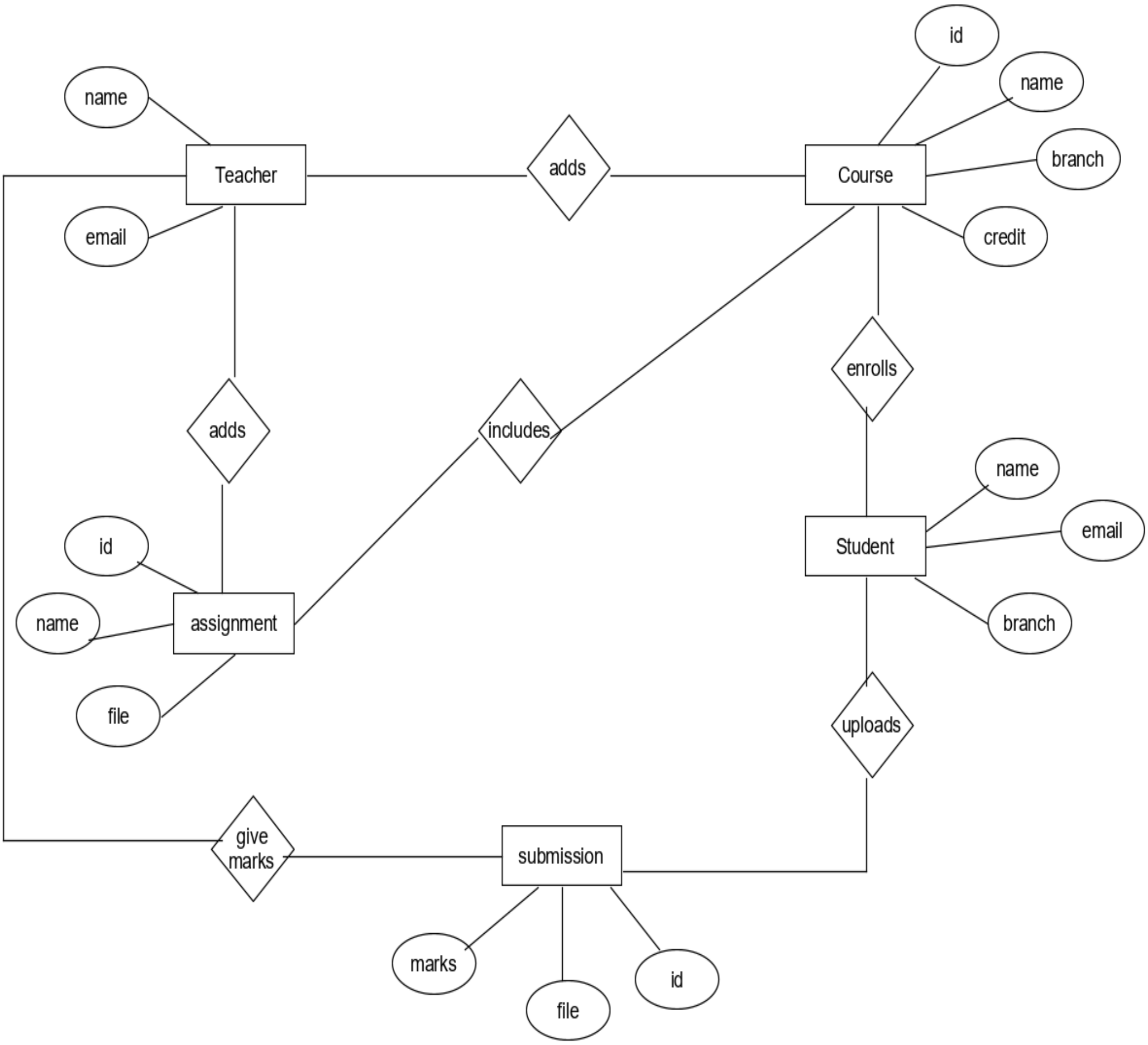
1. Use Case Diagram



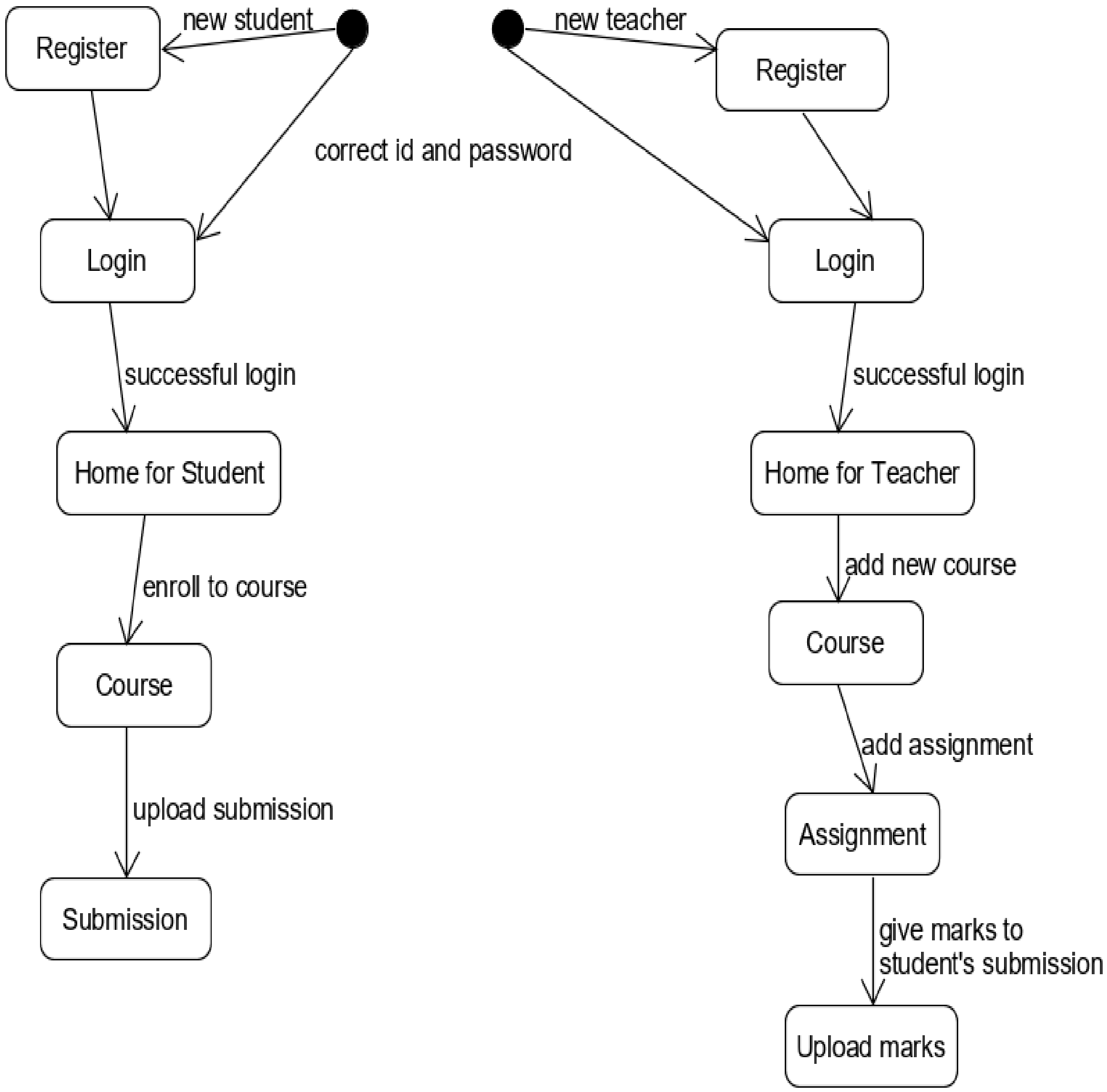
2. Class Diagram



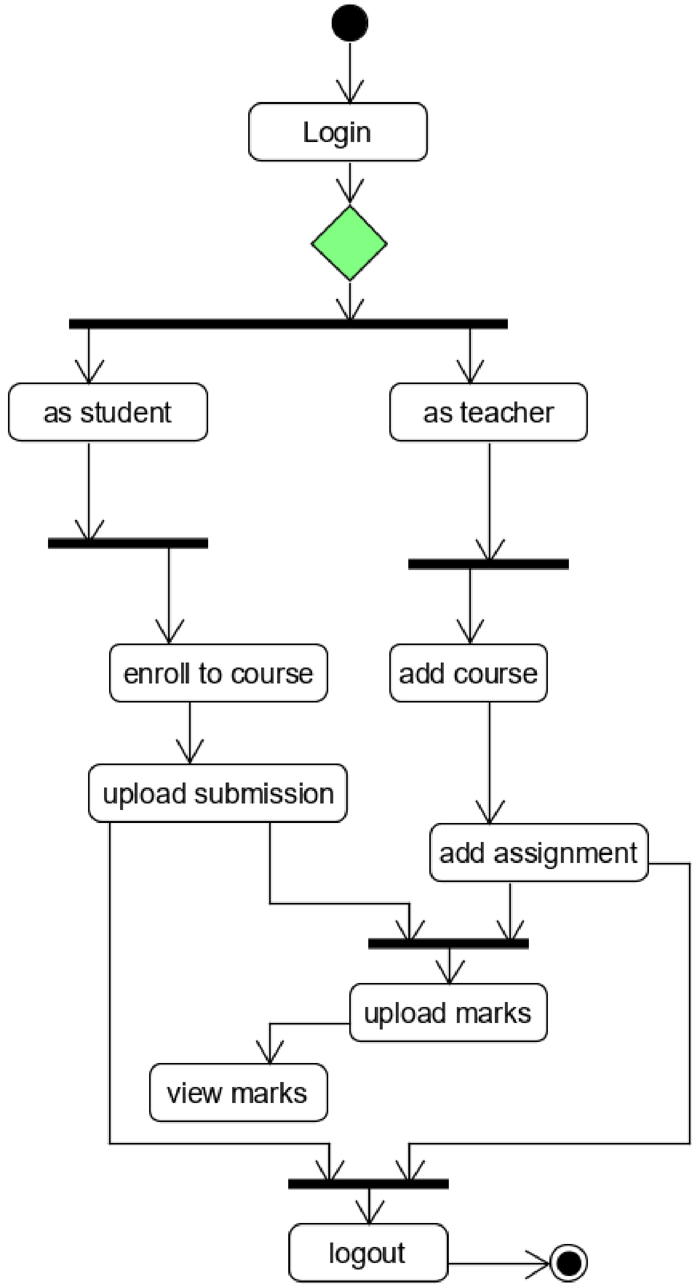
3. ER Diagram


id
name
Teacher
adds
Course
branch
email
credit
enrolls
adds
includes
name
id
Student
email
assignment
name
branch
file
uploads
give marks
submission
marks
id
file

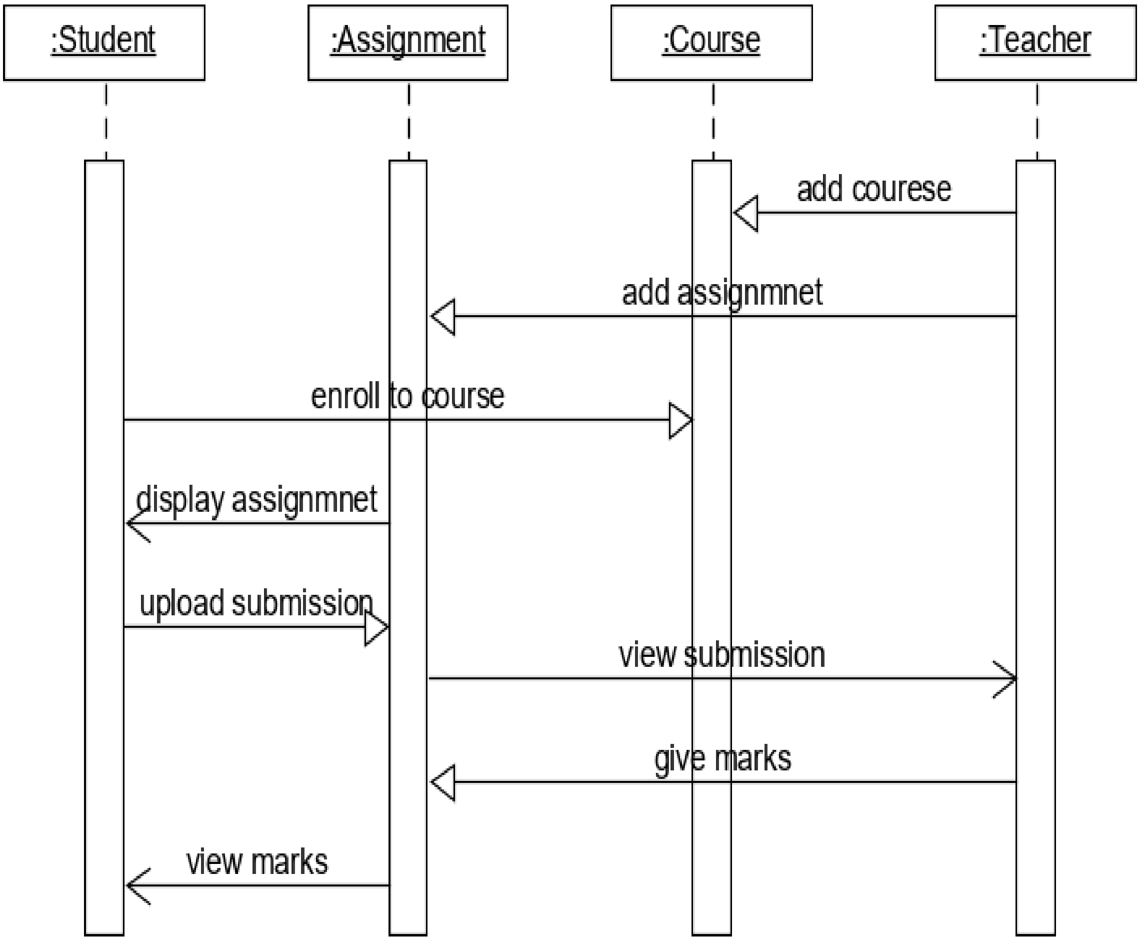
4. State Chart Diagram



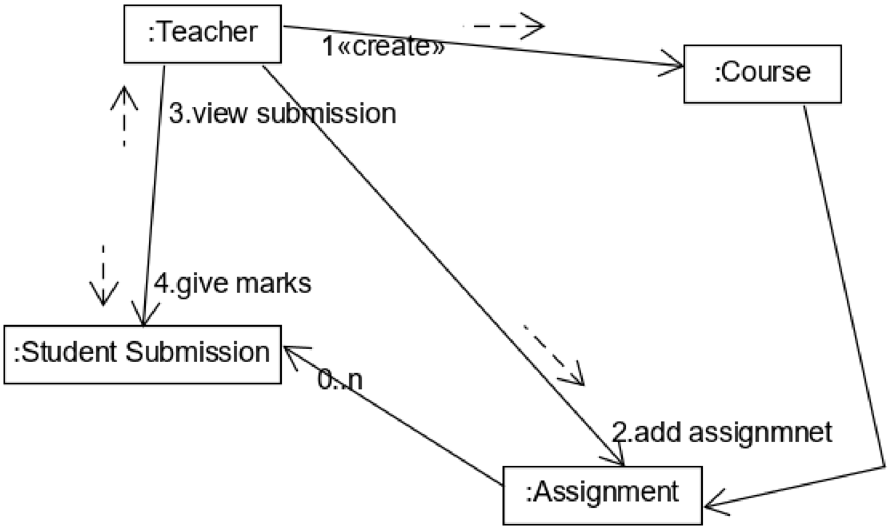
5. Activity Diagram

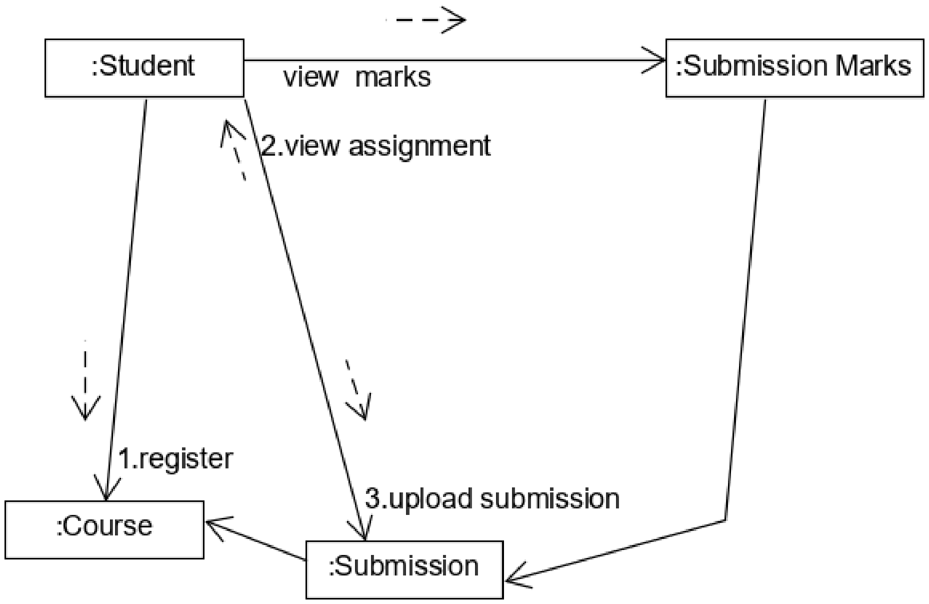


6. Sequence Diagram

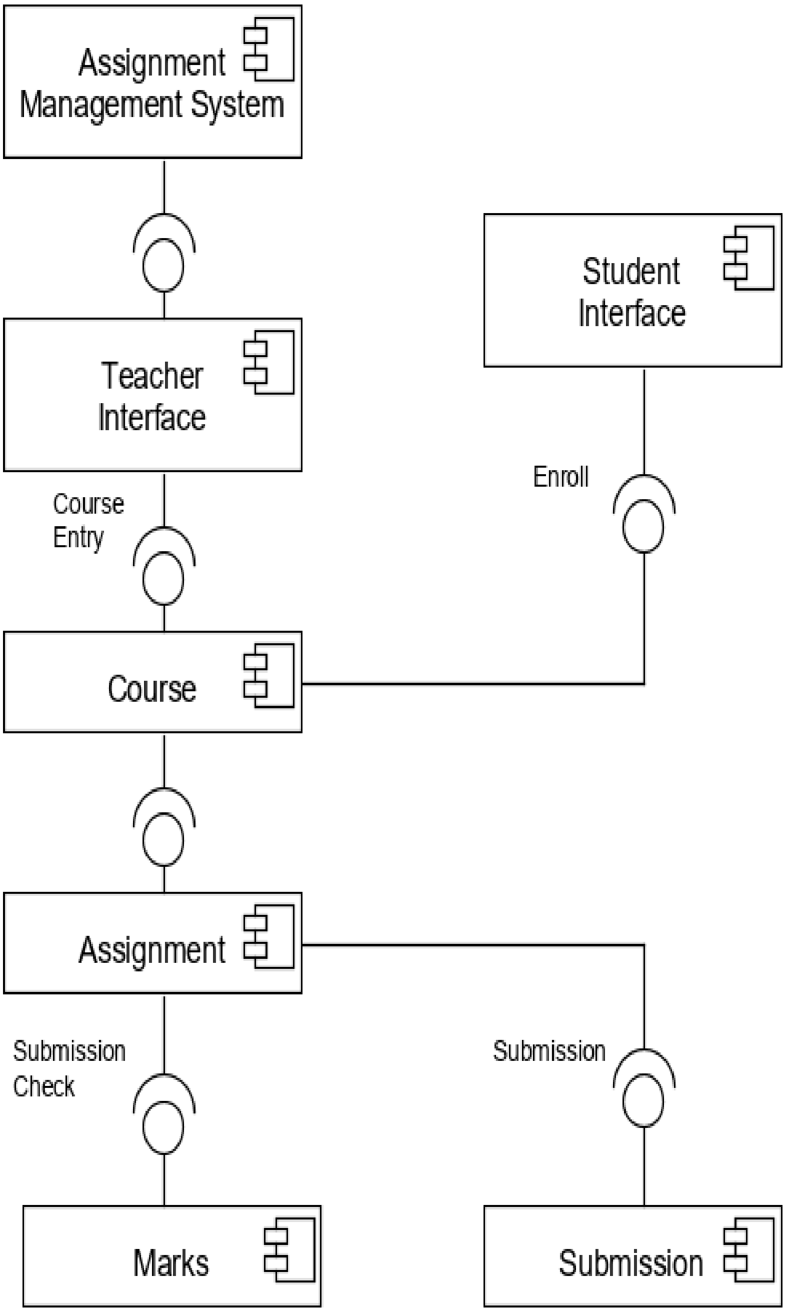


7. Collaboration Diagram

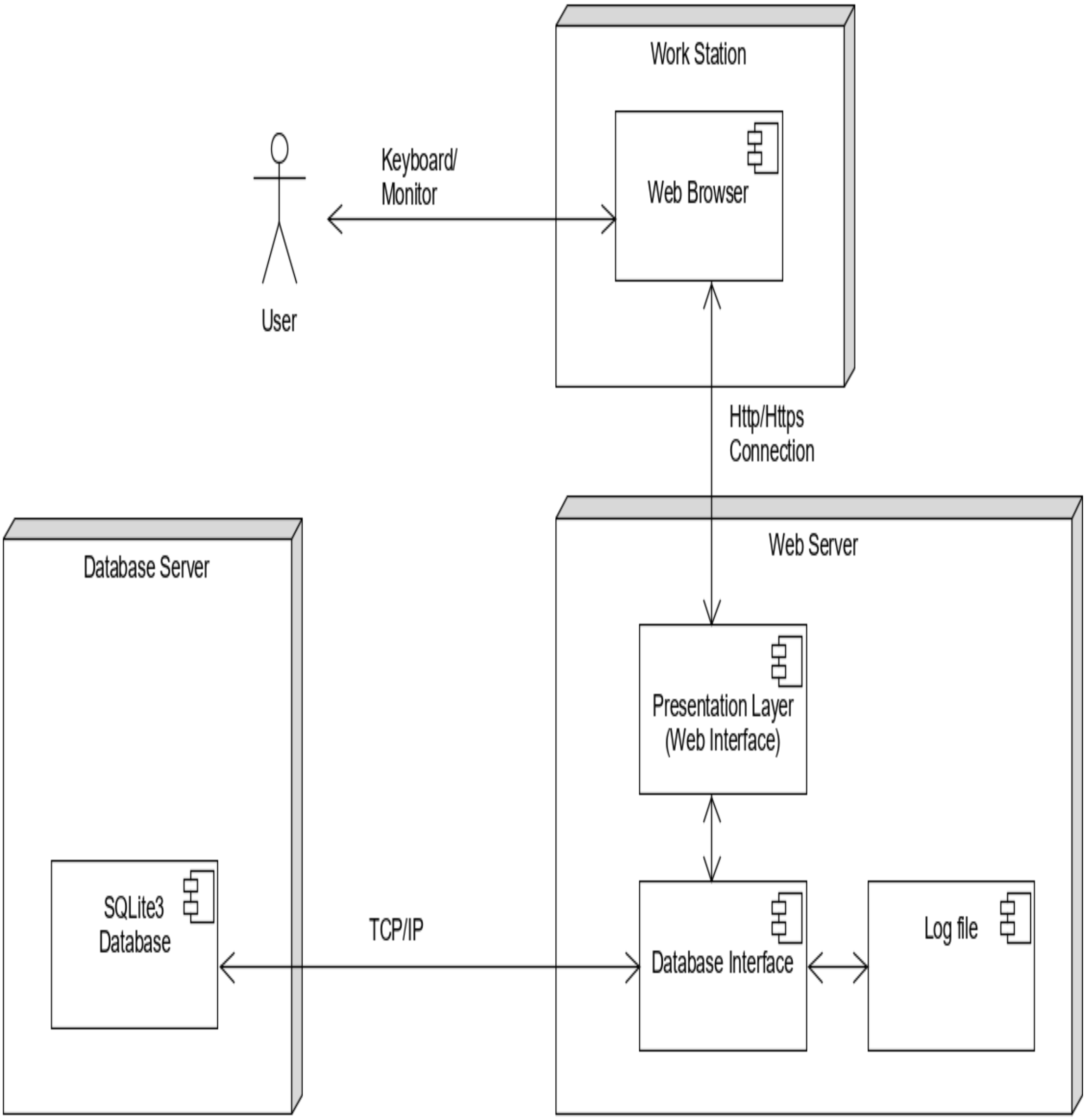




8. Component Diagram



9. Deployment Diagram



IMPLEMENTATION DETAILS

The System consists of 4 basic modules namely

1. User Module
2. Teacher Module
3. Student Module
4. Submission Module

Each module consists of several methods to implement the required functionality. Implementation is done using Django and database used in these modules is SQLite3.

User Module

There are two main users of the system, Teacher and Student. This module is used to handle authentication and authorization of the user. The user has to be logged in to access the functionalities that are provided by the system.

Teacher Module

This module handles the functionality related to Teacher. It allows the teacher to add new courses as well as to add assignments in particular course.

Student Module

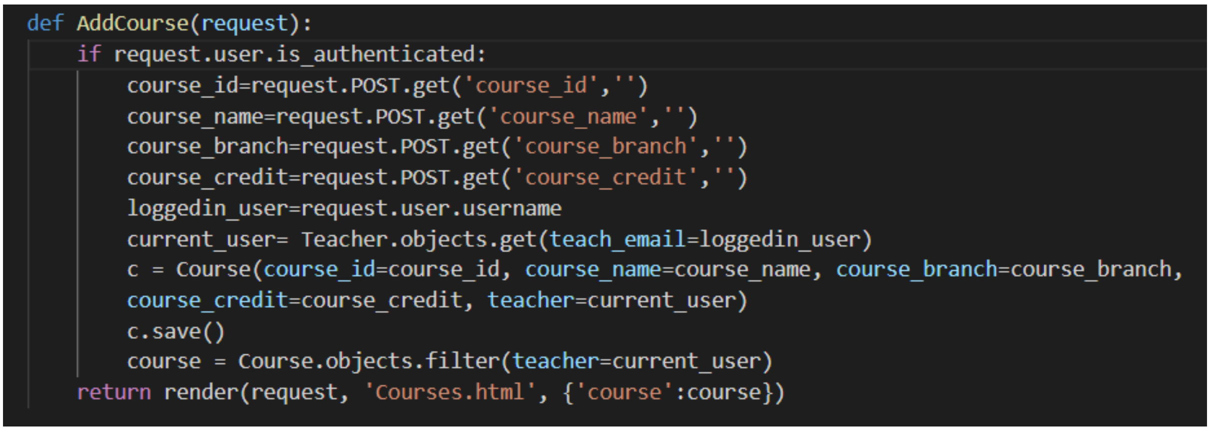
This module handles the functionality related to Student. It allows the student to enroll into courses added by the teacher as well as to upload their submission for assignments.

Submission Module

This module handles the functionality related to Submission. It allows the Teacher to view submissions of the student as well as to review & give marks to them. It allows the student to view the given marks by the teacher.

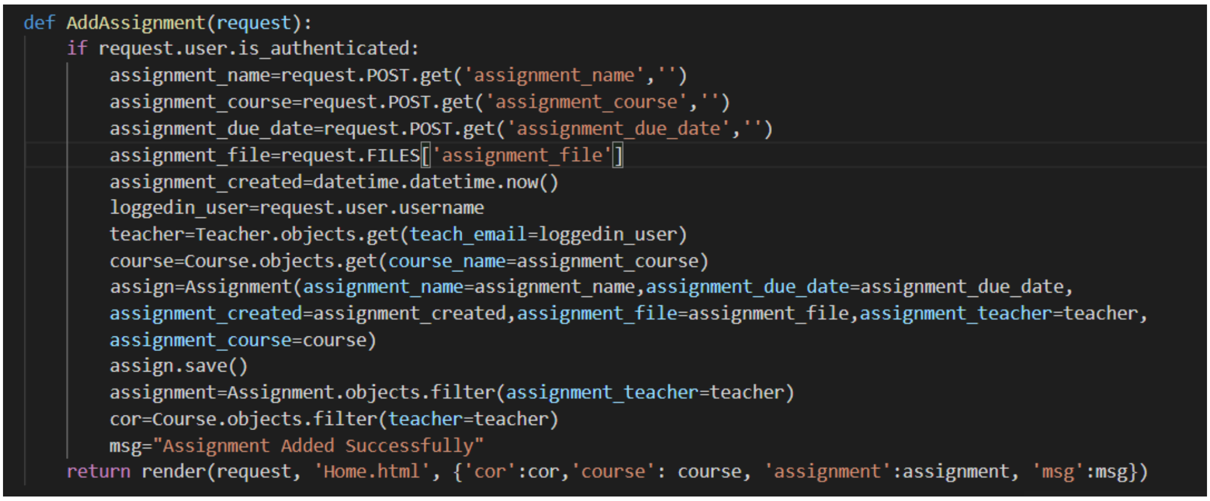
Major Functions Prototype:

1. Add Course



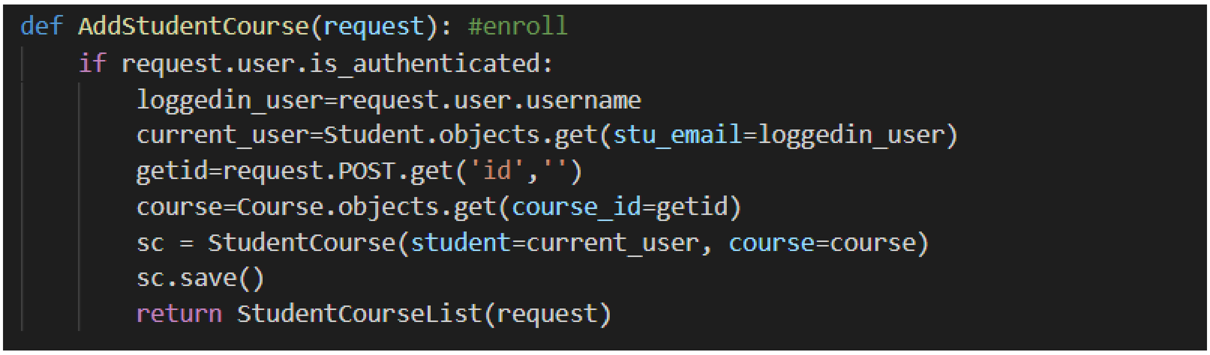
```
def AddCourse(request):
    if request.user.is_authenticated:
        course_id=request.POST.get('course_id','')
        course_name=request.POST.get('course_name','')
        course_branch=request.POST.get('course_branch','')
        course_credit=request.POST.get('course_credit','')
        loggedin_user=request.user.username
        current_user= Teacher.objects.get(teach_email=loggedin_user)
        c = Course(course_id=course_id, course_name=course_name, course_branch=course_branch,
        course_credit=course_credit, teacher=current_user)
        c.save()
        course = Course.objects.filter(teacher=current_user)
    return render(request, 'Courses.html', {'course':course})
```

2. Add Assignment



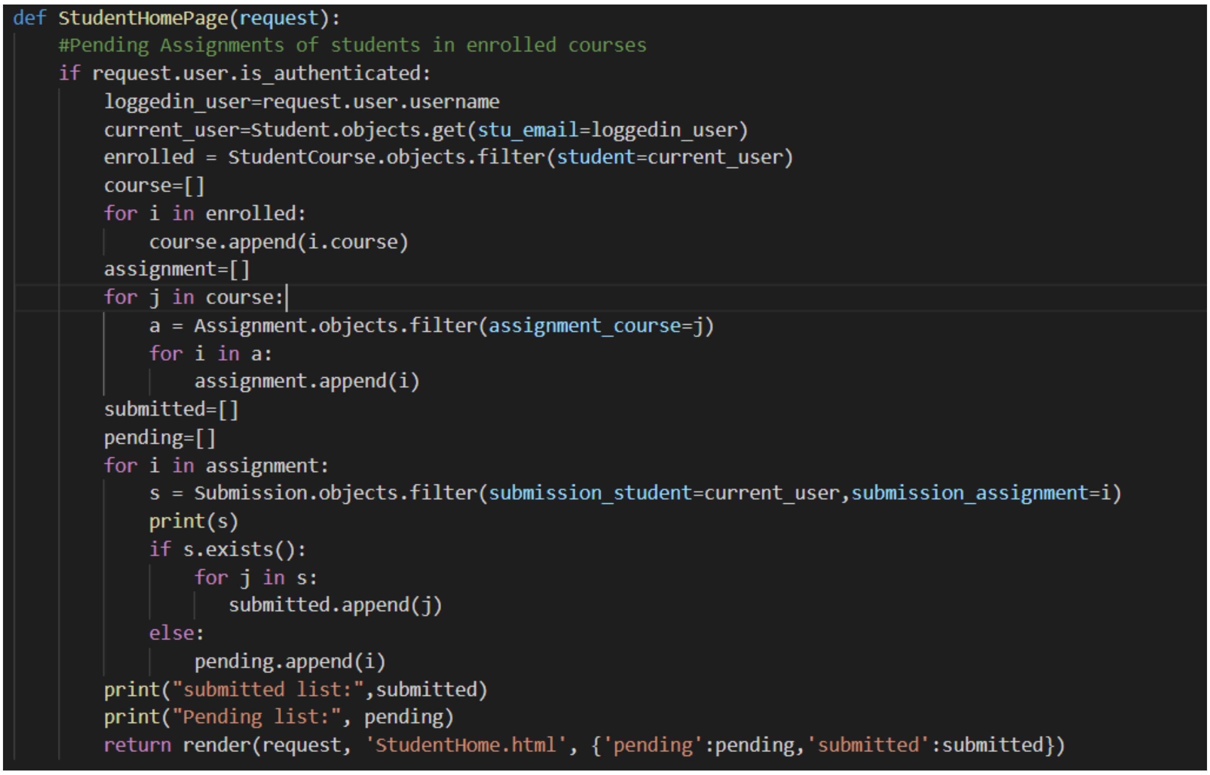
```
def AddAssignment(request):
    if request.user.is_authenticated:
        assignment_name=request.POST.get('assignment_name','')
        assignment_course=request.POST.get('assignment_course','')
        assignment_due_date=request.POST.get('assignment_due_date','')
        assignment_file=request.FILES['assignment_file']
        assignment_created=datetime.datetime.now()
        loggedin_user=request.user.username
        teacher=Teacher.objects.get(teach_email=loggedin_user)
        course=Course.objects.get(course_name=assignment_course)
        assign=Assignment(assignment_name=assignment_name,assignment_due_date=assignment_due_date,
        assignment_created=assignment_created,assignment_file=assignment_file,assignment_teacher=teacher,
        assignment_course=course)
        assign.save()
        assignment=Assignment.objects.filter(assignment_teacher=teacher)
        cor=Course.objects.filter(teacher=teacher)
        msg="Assignment Added Successfully"
    return render(request, 'Home.html', {'cor':cor,'course': course, 'assignment':assignment, 'msg':msg})
```

3. Enroll into Course



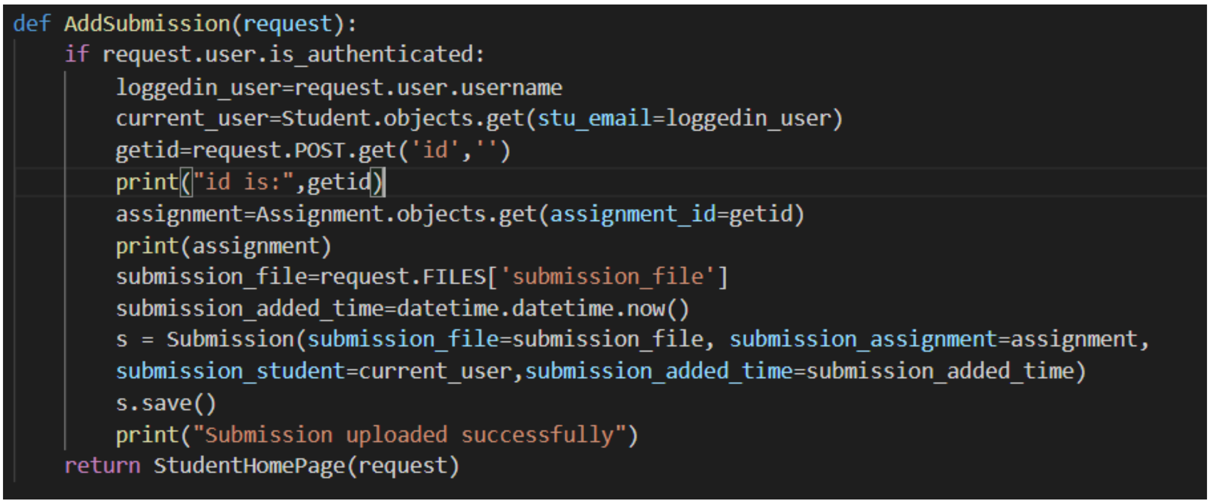
```
def AddStudentCourse(request): #enroll
    if request.user.is_authenticated:
        loggedin_user=request.user.username
        current_user=Student.objects.get(stu_email=loggedin_user)
        getid=request.POST.get('id','')
        course=Course.objects.get(course_id=getid)
        sc = StudentCourse(student=current_user, course=course)
        sc.save()
        return StudentCourseList(request)
```

4. View Assignments



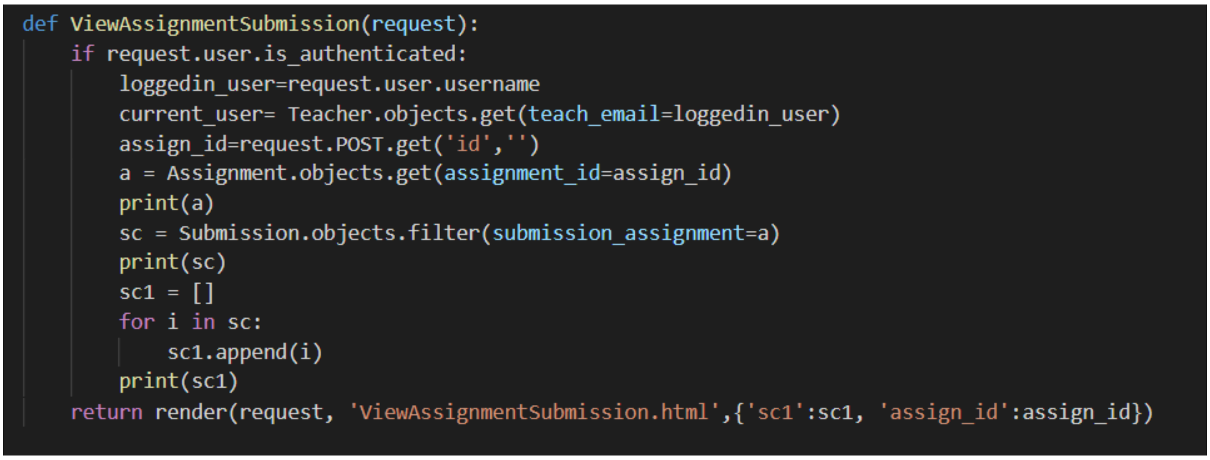
```
def StudentHomePage(request):
    #Pending Assignments of students in enrolled courses
    if request.user.is_authenticated:
        loggedin_user=request.user.username
        current_user=Student.objects.get(stu_email=loggedin_user)
        enrolled = StudentCourse.objects.filter(student=current_user)
        course=[]
        for i in enrolled:
            course.append(i.course)
        assignment=[]
        for j in course:
            a = Assignment.objects.filter(assignment_course=j)
            for i in a:
                assignment.append(i)
        submitted=[]
        pending=[]
        for i in assignment:
            s = Submission.objects.filter(submission_student=current_user,submission_assignment=i)
            print(s)
            if s.exists():
                for j in s:
                   submitted.append(j)
            else:
                pending.append(i)
        print("submitted list:",submitted)
        print("Pending list:", pending)
        return render(request, 'StudentHome.html', {'pending':pending,'submitted':submitted})
```

5. Upload Submission



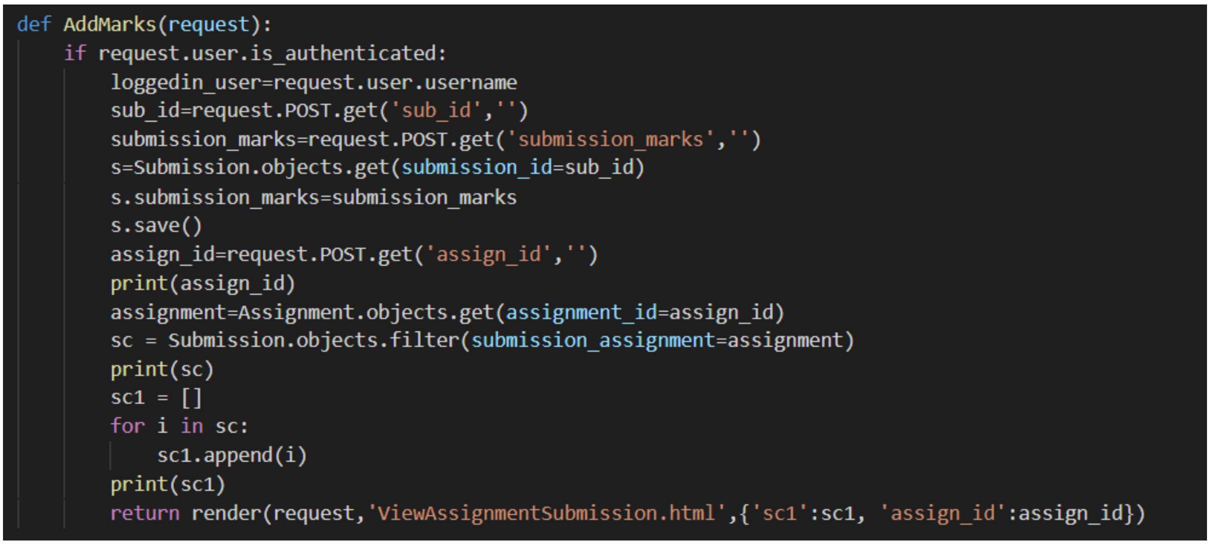
```
def AddSubmission(request):
    if request.user.is_authenticated:
        loggedin_user=request.user.username
        current_user=Student.objects.get(stu_email=loggedin_user)
        getid=request.POST.get('id','')
        print("id is:",getid)
        assignment=Assignment.objects.get(assignment_id=getid)
        print(assignment)
        submission_file=request.FILES['submission_file']
        submission_added_time=datetime.datetime.now()
        s = Submission(submission_file=submission_file, submission_assignment=assignment,
        submission_student=current_user,submission_added_time=submission_added_time)
        s.save()
        print("Submission uploaded successfully")
    return StudentHomePage(request)
```

6. View Assignment Submission



```python
def ViewAssignmentSubmission(request):
    if request.user.is_authenticated:
        loggedin_user=request.user.username
        current_user= Teacher.objects.get(teach_email=loggedin_user)
        assign_id=request.POST.get('id','')
        a = Assignment.objects.get(assignment_id=assign_id)
        print(a)
        sc = Submission.objects.filter(submission_assignment=a)
        print(sc)
        sc1 = []
        for i in sc:
            sc1.append(i)
        print(sc1)
    return render(request, 'ViewAssignmentSubmission.html',{'sc1':sc1, 'assign_id':assign_id})
```

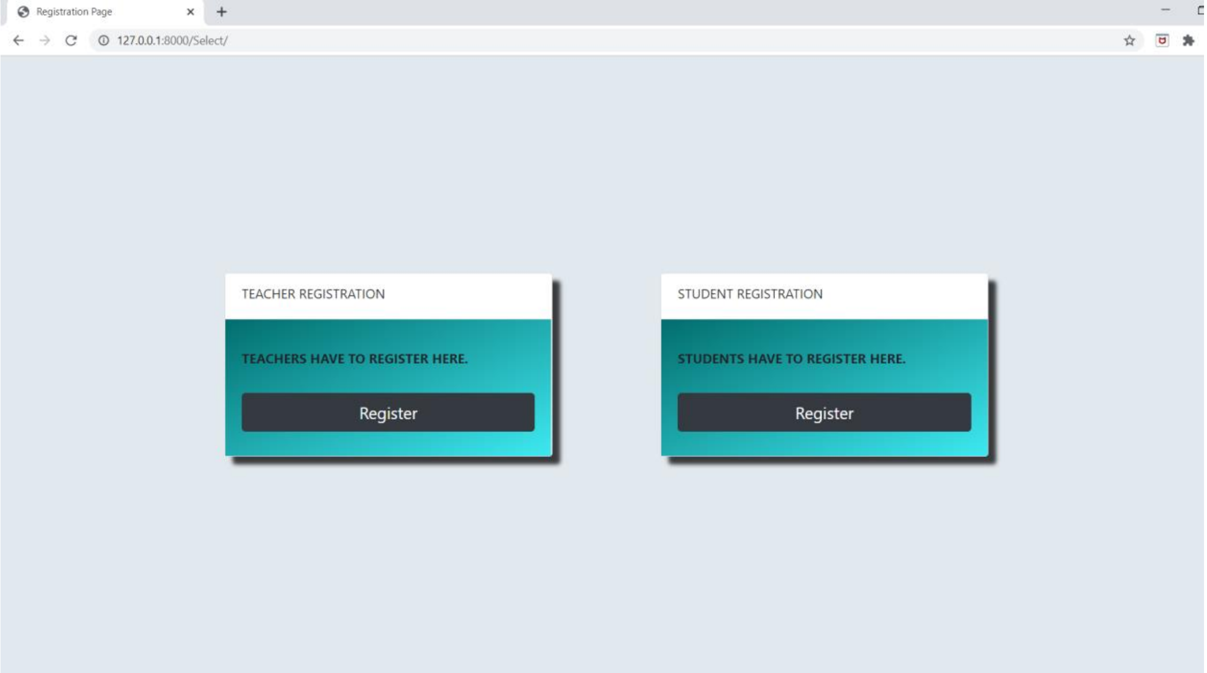
7. Give Marks

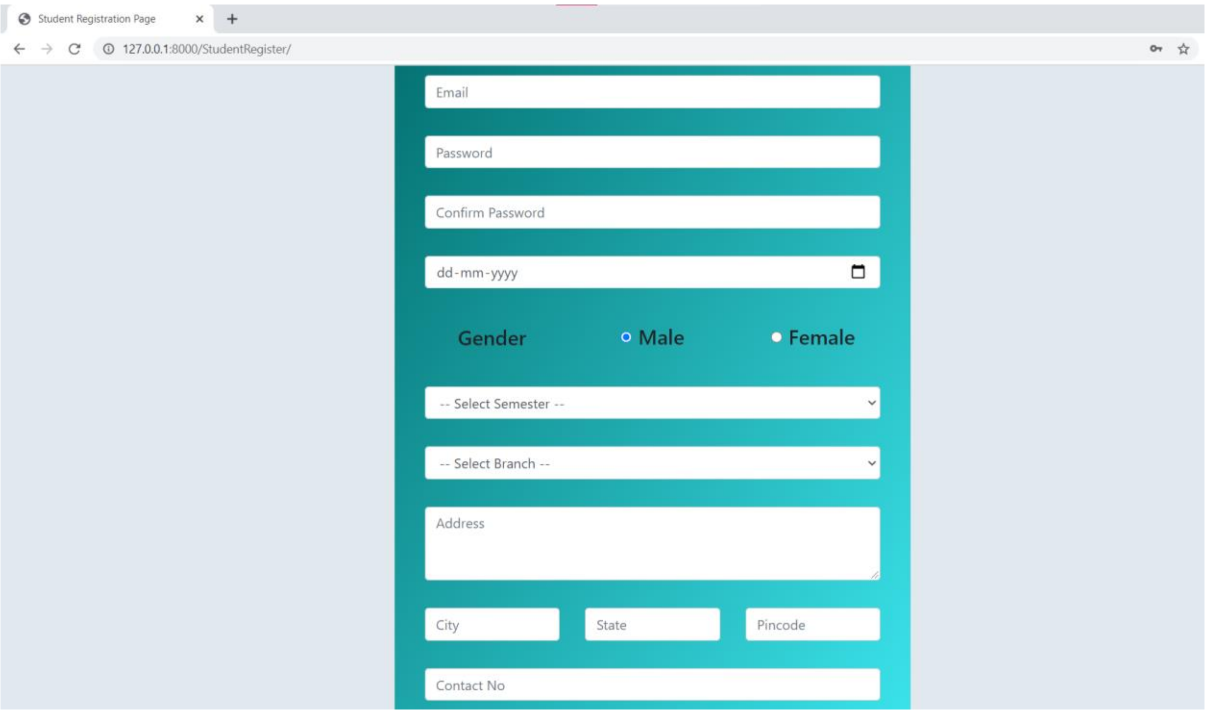


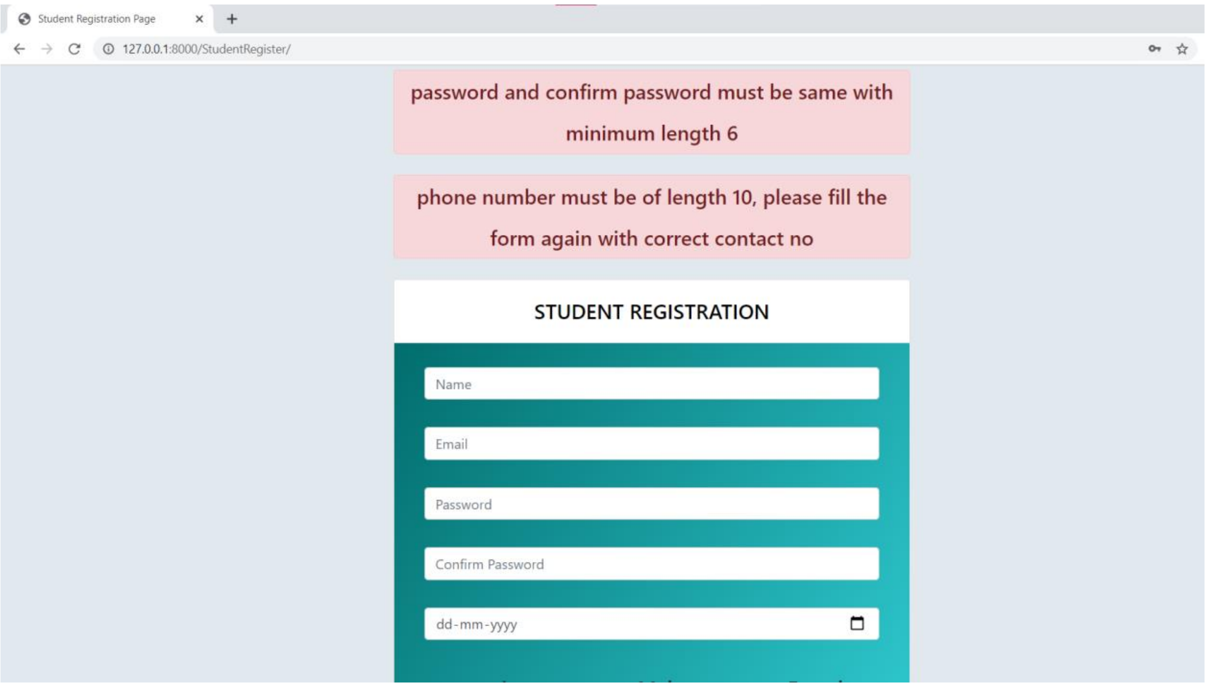
```python
def AddMarks(request):
    if request.user.is_authenticated:
        loggedin_user=request.user.username
        sub_id=request.POST.get('sub_id','')
        submission_marks=request.POST.get('submission_marks','')
        s=Submission.objects.get(submission_id=sub_id)
        s.submission_marks=submission_marks
        s.save()
        assign_id=request.POST.get('assign_id','')
        print(assign_id)
        assignment=Assignment.objects.get(assignment_id=assign_id)
        sc = Submission.objects.filter(submission_assignment=assignment)
        print(sc)
        sc1 = []
        for i in sc:
            sc1.append(i)
        print(sc1)
        return render(request,'ViewAssignmentSubmission.html',{'sc1':sc1, 'assign_id':assign_id})
```

SCREENSHOTS

1. Student and Teachers Registration with proper validation.





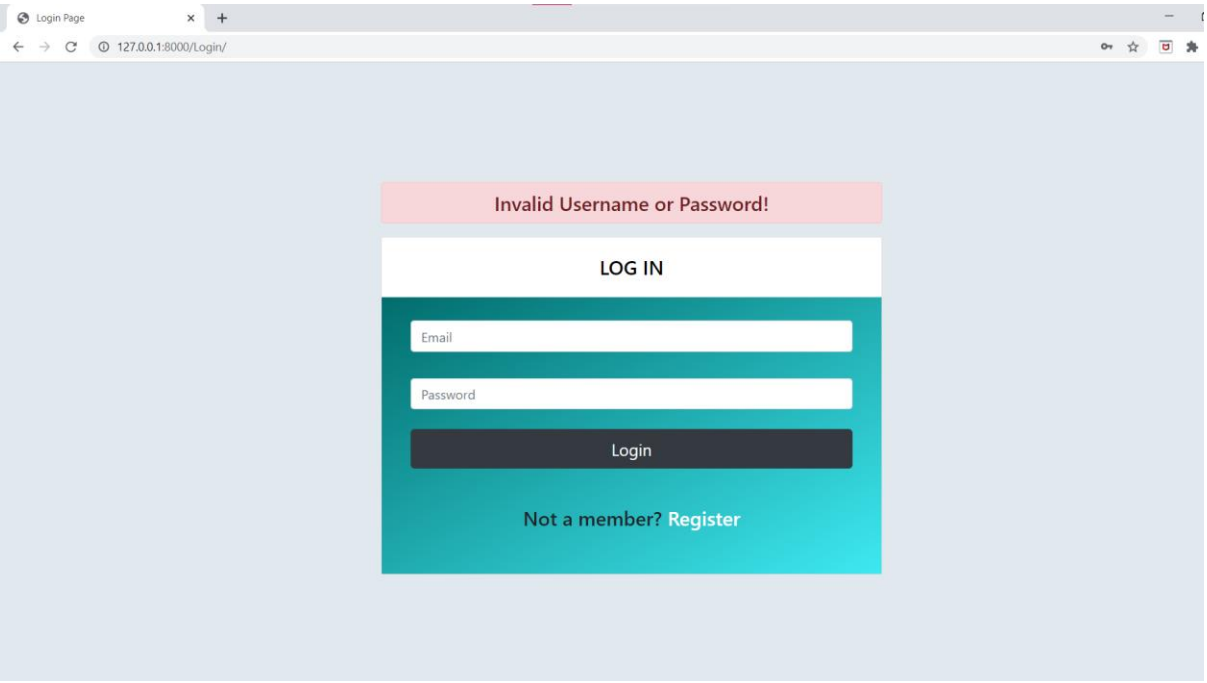


password and confirm password must be same with minimum length 6

phone number must be of length 10, please fill the form again with correct contact no

STUDENT REGISTRATION

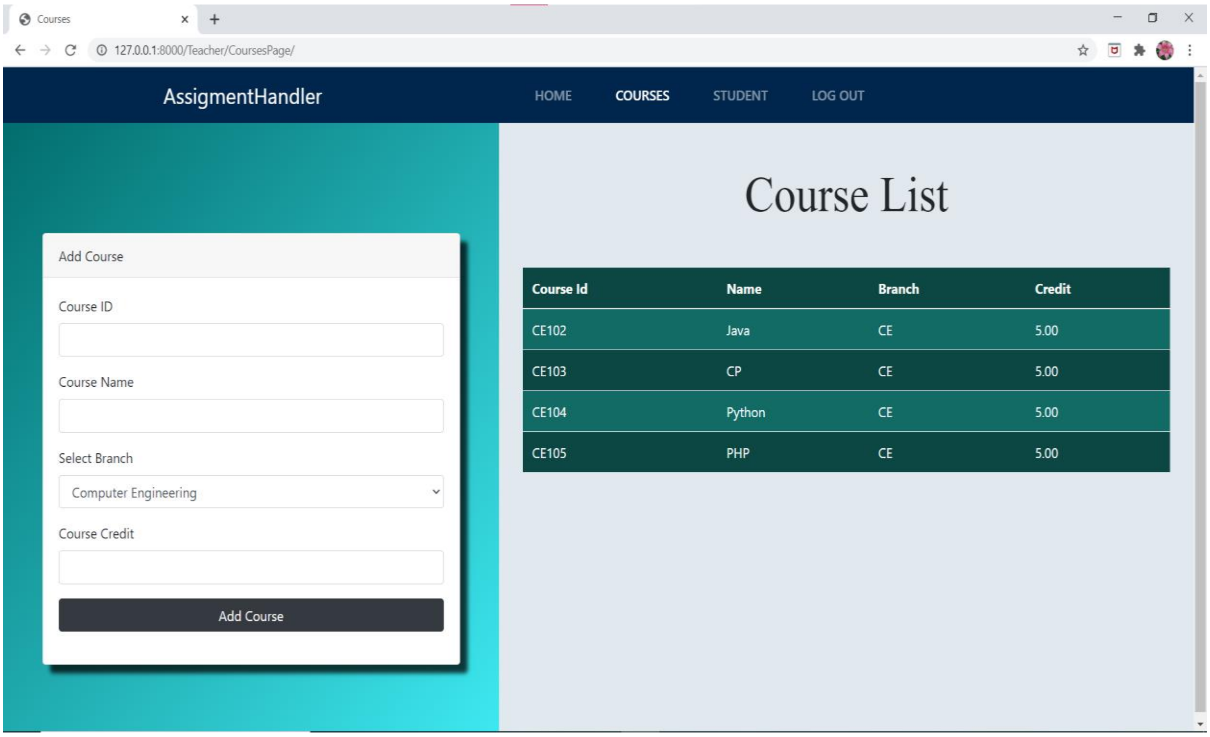
2. Login Page with Validation



Invalid Username or Password!

LOG IN

3. Teachers can add course and view course List



Course List

Course Id	Name	Branch	Credit
CE102	Java	CE	5.00
CE103	CP	CE	5.00
CE104	Python	CE	5.00
CE105	PHP	CE	5.00

Add Course

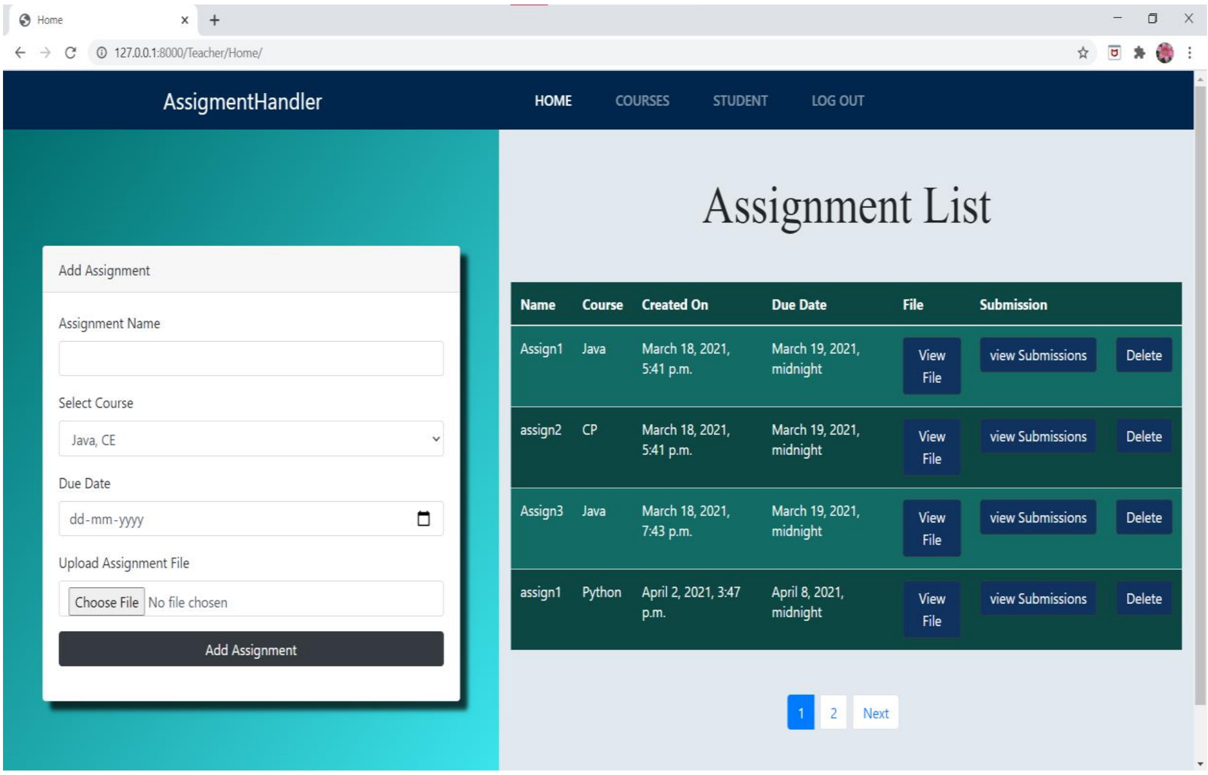
Course ID

Course Name

Select Branch: Computer Engineering

Course Credit

4. Teachers can add assignments and view Assignment List



Assignment List

Name	Course	Created On	Due Date	File	Submission	
Assign1	Java	March 18, 2021, 5:41 p.m.	March 19, 2021, midnight	View File	view Submissions	Delete
assign2	CP	March 18, 2021, 5:41 p.m.	March 19, 2021, midnight	View File	view Submissions	Delete
Assign3	Java	March 18, 2021, 7:43 p.m.	March 19, 2021, midnight	View File	view Submissions	Delete
assign1	Python	April 2, 2021, 3:47 p.m.	April 8, 2021, midnight	View File	view Submissions	Delete

Add Assignment

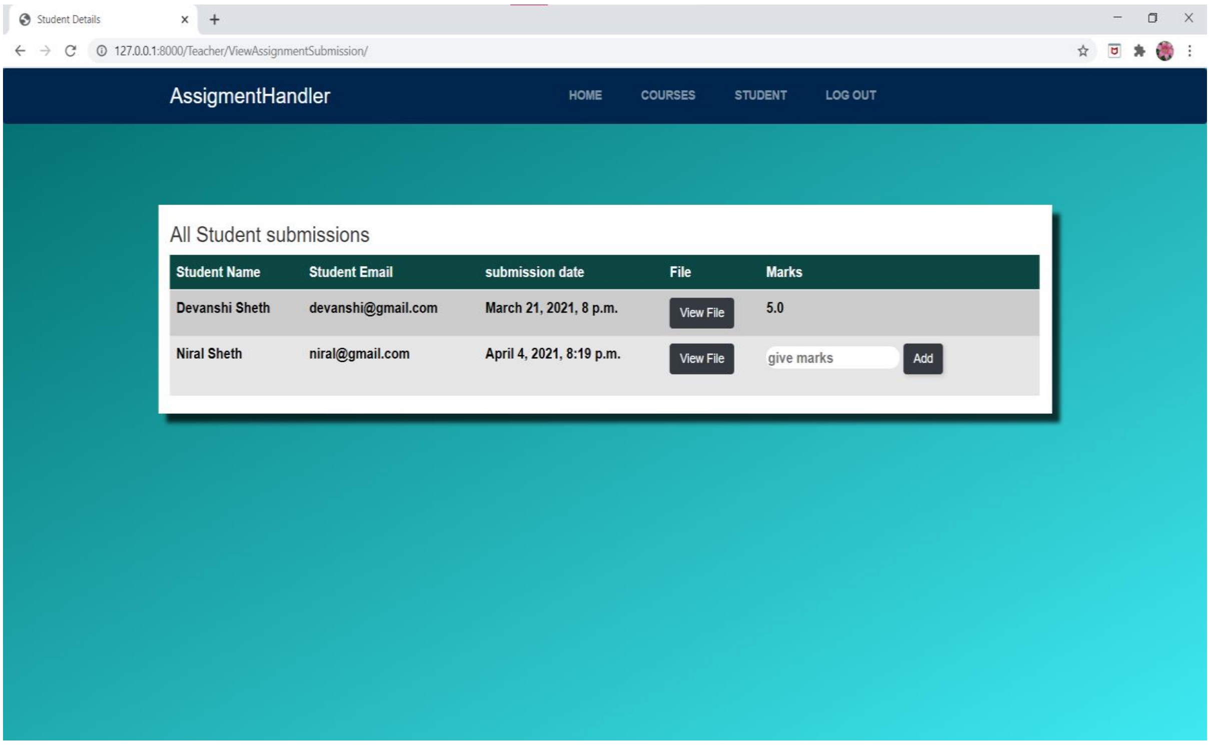
Assignment Name

Select Course: Java, CE

Due Date: dd-mm-yyyy

Upload Assignment File: No file chosen

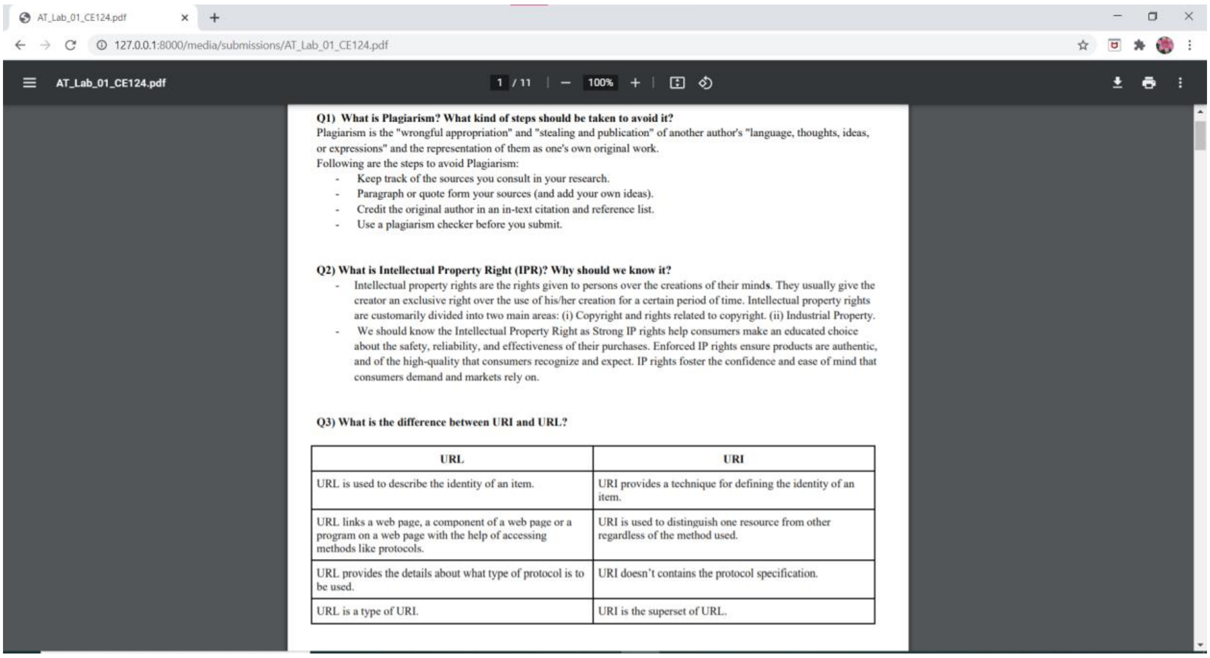
5. Teacher can view submission of assignments of a particular course and can give marks to student’s submission



All Student submissions

Student Name	Student Email	submission date	File	Marks
Devanshi Sheth	devanshi@gmail.com	March 21, 2021, 8 p.m.	View File	5.0
Niral Sheth	niral@gmail.com	April 4, 2021, 8:19 p.m.	View File	give marks Add

Teachers can view the submitted file of a student.



Q1) What is Plagiarism? What kind of steps should be taken to avoid it?

Plagiarism is the "wrongful appropriation" and "stealing and publication" of another author's "language, thoughts, ideas, or expressions" and the representation of them as one's own original work.

Following are the steps to avoid Plagiarism:

- Keep track of the sources you consult in your research.
- Paragraph or quote form your sources (and add your own ideas).
- Credit the original author in an in-text citation and reference list.
- Use a plagiarism checker before you submit.

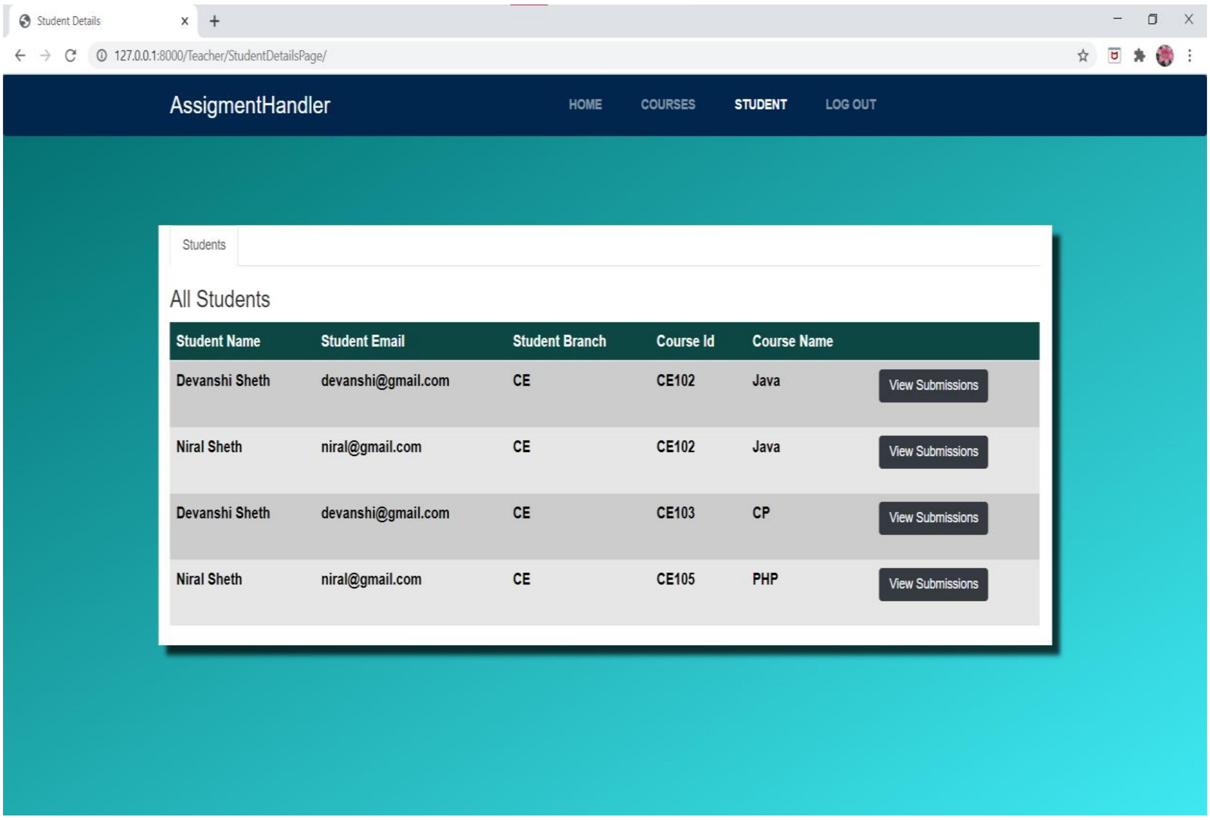
Q2) What is Intellectual Property Right (IPR)? Why should we know it?

- Intellectual property rights are the rights given to persons over the creations of their minds. They usually give the creator an exclusive right over the use of his/her creation for a certain period of time. Intellectual property rights are customarily divided into two main areas: (i) Copyright and rights related to copyright. (ii) Industrial Property.
- We should know the Intellectual Property Right as Strong IP rights help consumers make an educated choice about the safety, reliability, and effectiveness of their purchases. Enforced IP rights ensure products are authentic, and of the high-quality that consumers recognize and expect. IP rights foster the confidence and ease of mind that consumers demand and markets rely on.

Q3) What is the difference between URI and URL?

URL	URI
URL is used to describe the identity of an item.	URI provides a technique for defining the identity of an item.
URL links a web page, a component of a web page or a program on a web page with the help of accessing methods like protocols.	URI is used to distinguish one resource from other regardless of the method used.
URL provides the details about what type of protocol is to be used.	URI doesn’t contains the protocol specification.
URL is a type of URI.	URI is the superset of URL.

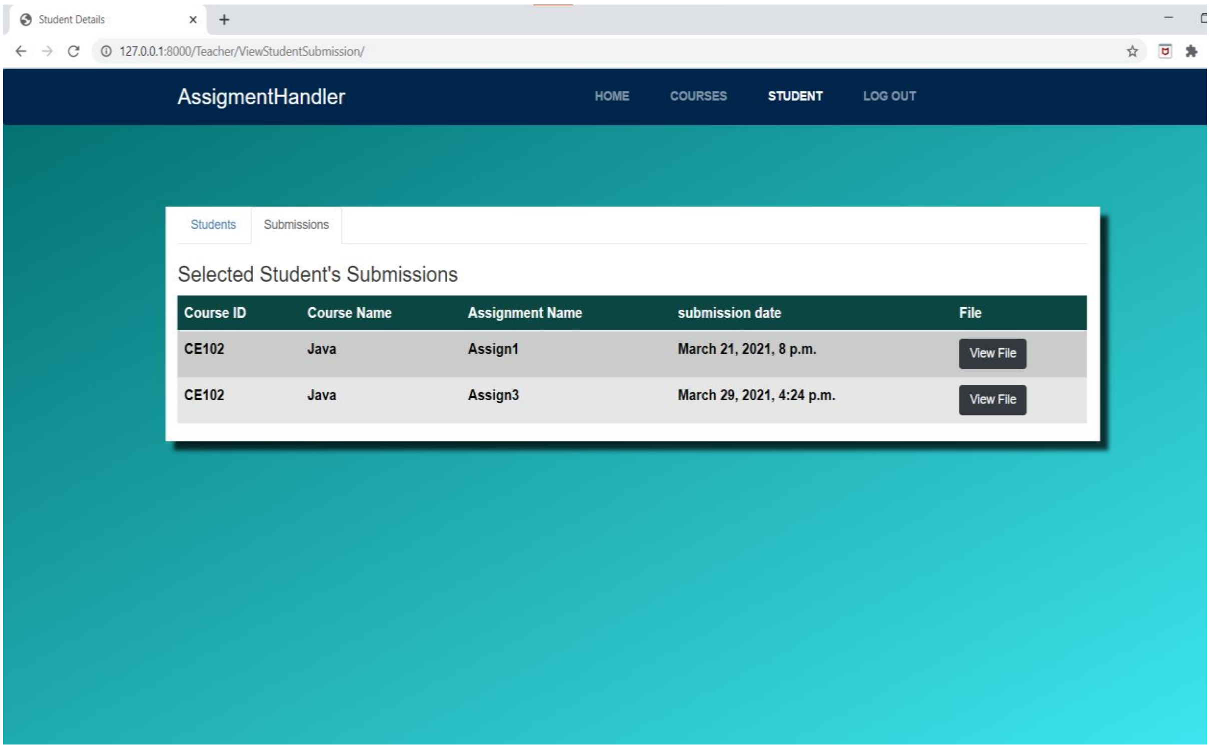
6. **Teachers can view the details of all students enrolled in a course which are added by them.**



All Students

Student Name	Student Email	Student Branch	Course Id	Course Name	
Devanshi Sheth	devanshi@gmail.com	CE	CE102	Java	View Submissions
Niral Sheth	niral@gmail.com	CE	CE102	Java	View Submissions
Devanshi Sheth	devanshi@gmail.com	CE	CE103	CP	View Submissions
Niral Sheth	niral@gmail.com	CE	CE105	PHP	View Submissions

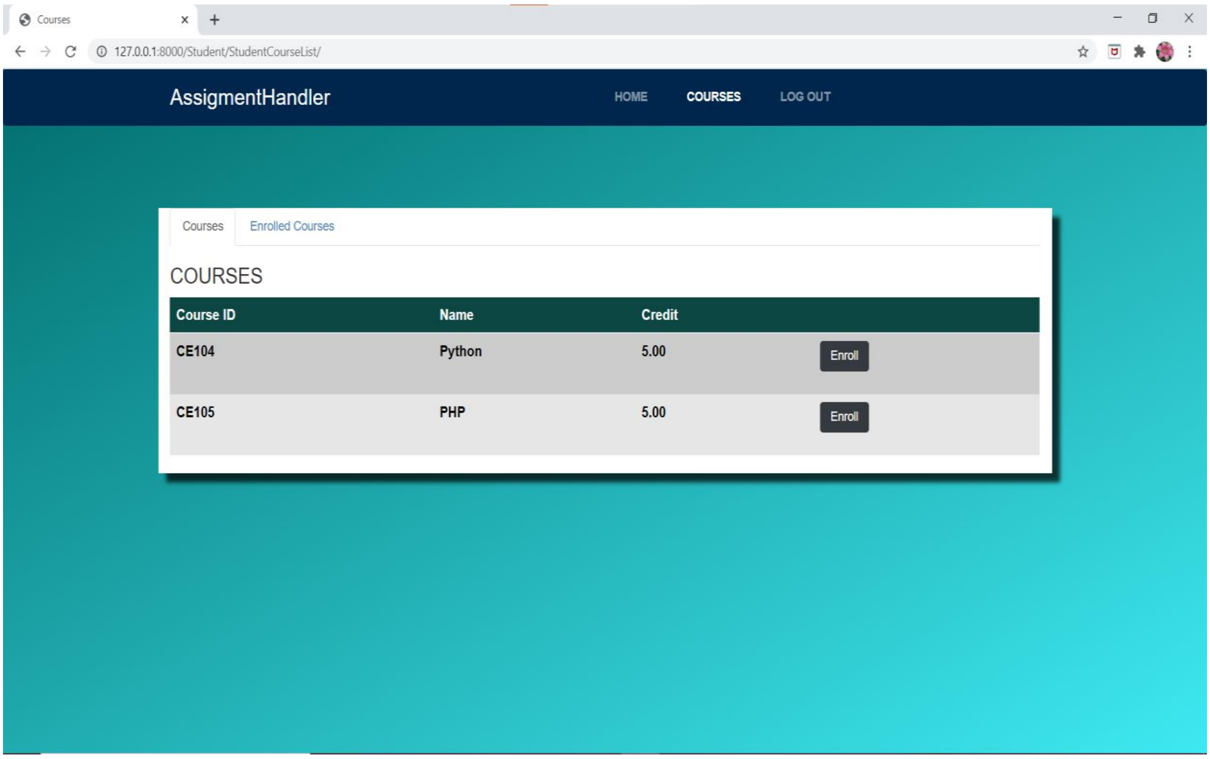
7. **Teachers can view all submissions of a particular student.**



Selected Student's Submissions

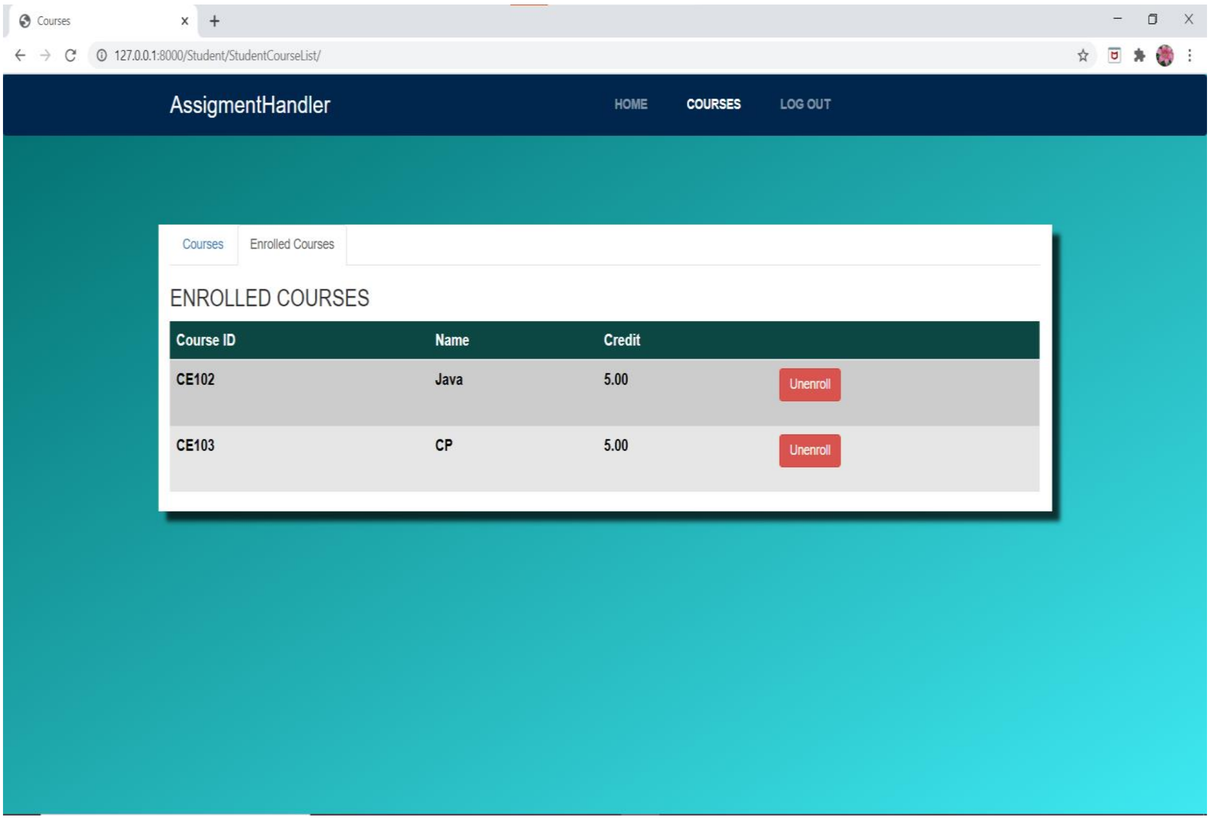
Course ID	Course Name	Assignment Name	submission date	File
CE102	Java	Assign1	March 21, 2021, 8 p.m.	View File
CE102	Java	Assign3	March 29, 2021, 4:24 p.m.	View File

8. **Students can enroll into the courses of their branch. They can view their branch's unenrolled courses as well as their enrolled courses.**



COURSES

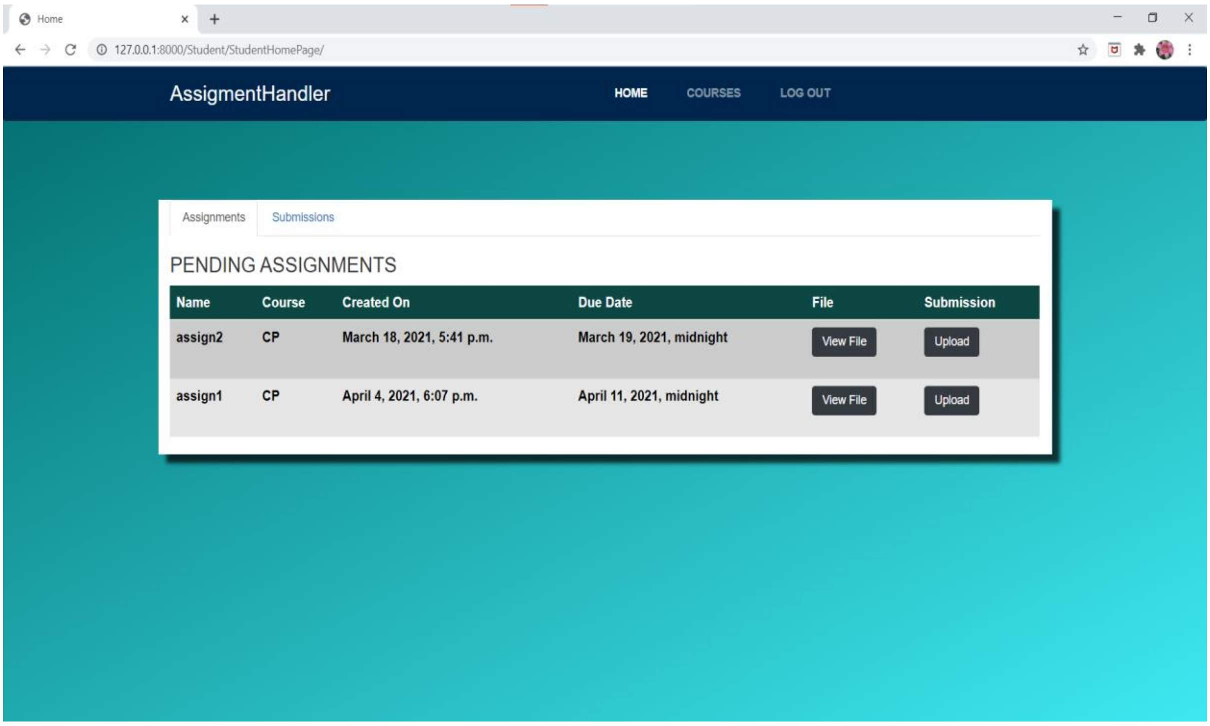
Course ID	Name	Credit	
CE104	Python	5.00	Enroll
CE105	PHP	5.00	Enroll



ENROLLED COURSES

Course ID	Name	Credit	
CE102	Java	5.00	Unenroll
CE103	CP	5.00	Unenroll

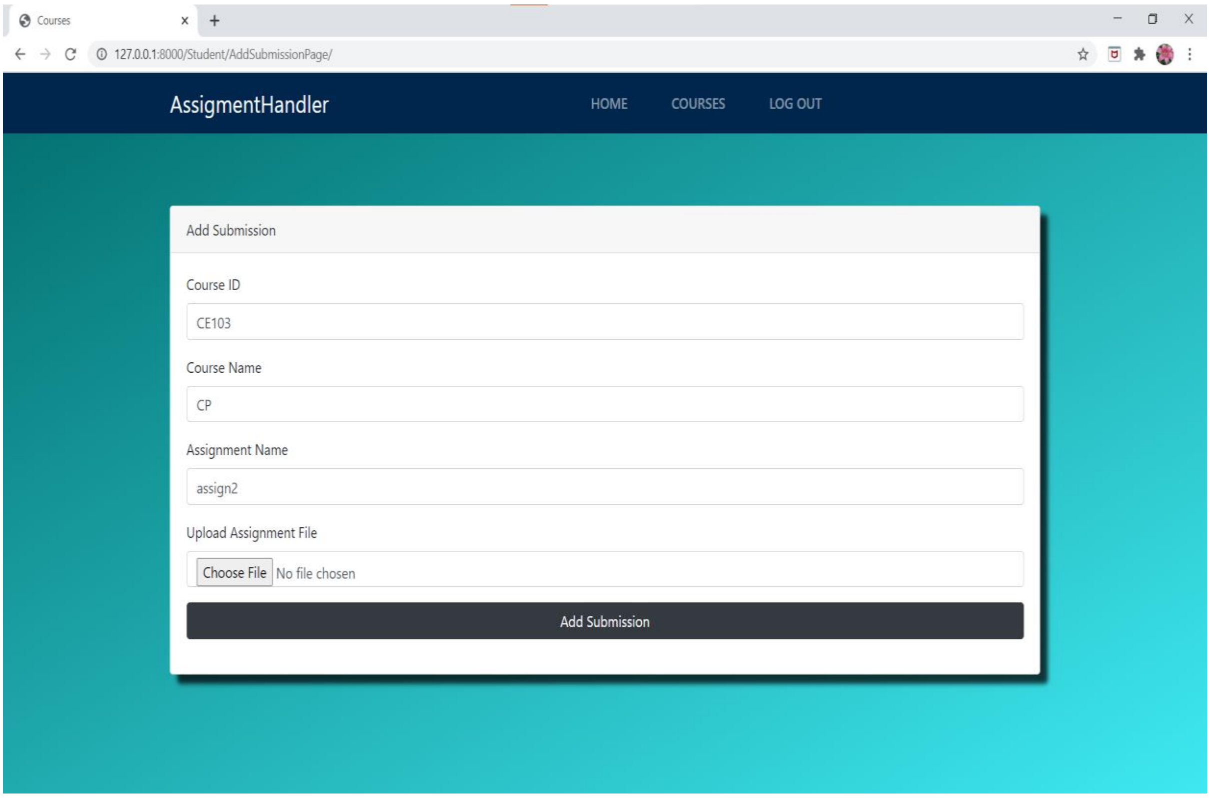
9. On Home Page Student Can view their Pending Assignments of the courses in which they are enrolled



PENDING ASSIGNMENTS

Name	Course	Created On	Due Date	File	Submission
assign2	CP	March 18, 2021, 5:41 p.m.	March 19, 2021, midnight	View File	Upload
assign1	CP	April 4, 2021, 6:07 p.m.	April 11, 2021, midnight	View File	Upload

10. Student can upload submission of a particular assignment



Add Submission

Course ID

CE103

Course Name

CP

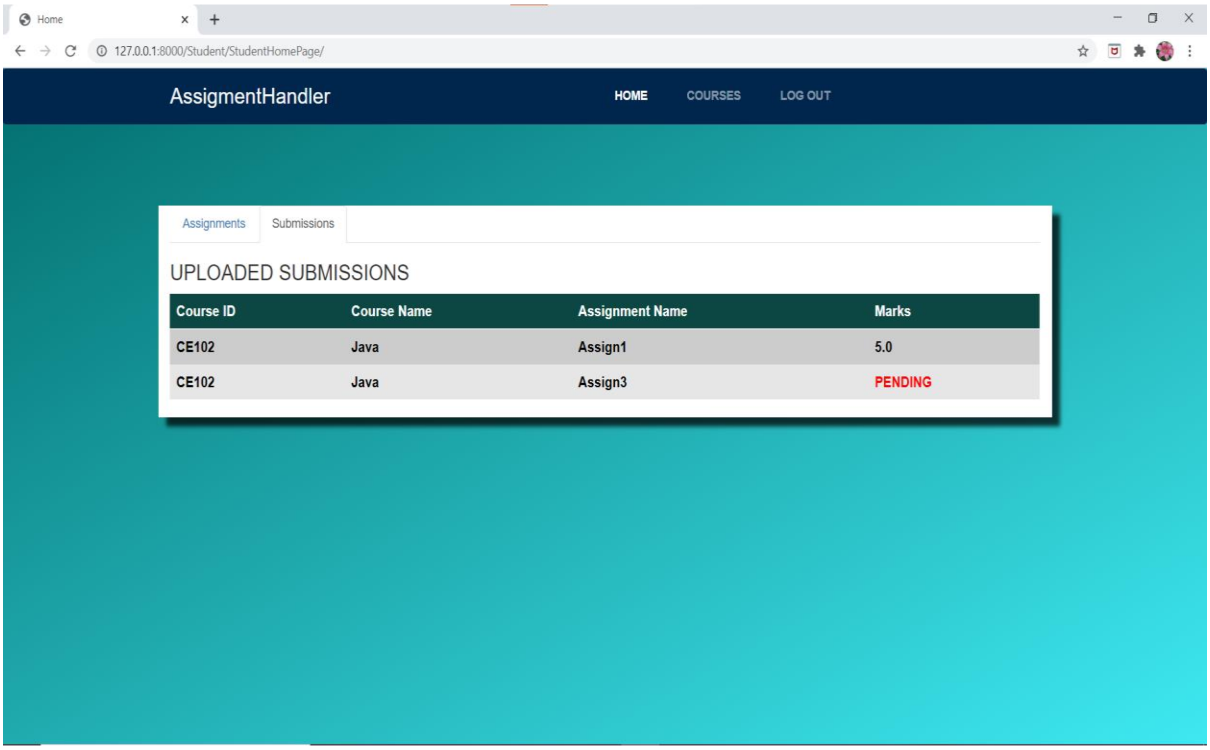
Assignment Name

assign2

Upload Assignment File

Choose File No file chosen

11. Students can view their submitted assignments with given marks



UPLOADED SUBMISSIONS

Course ID	Course Name	Assignment Name	Marks
CE102	Java	Assign1	5.0
CE102	Java	Assign3	PENDING

TESTING

For testing, Selenium is used.

Selenium is a tool that allows us to create tests that are run in the browser and interact with your UI in the same way as if you were manually testing your website or app.

CONCLUSION

The functionalities are implemented in system after understanding all the system modules according to the requirements. Functionalities that are successfully implemented in the system are:

- Teacher and Student Registration
- Login
- User authentication
- Logout
- Teacher can add course
- Teacher can add assignment in particular course
- Student can enroll in course
- Student can upload submission of particular assignment
- Teacher can view all submissions of particular assignment as well as of particular student
- Teacher give marks to student's submission
- Students can view their list of submission with given marks.

FUTURE WORK

- Google Authentication can be added for the registration of students and teacher.
- Currently in the system, teacher can only add specific date for deadline. Further we can enable them to add specific time with specific date.
- Further teacher can also comment and give their reviews for the submission along with the marks.

REFERENCES

https://docs.djangoproject.com/en/3.1/