Dharmsinh Desai University, Nadiad

Faculty of Technology

Department of Computer Engineering

B.Tech (CE) Sem-IV
Subject :- Software Project

Project Report of : **CollagePortal**

Name : Chintan Gopalbhai Solanki

Roll No : CE125

ID : 19CEUBS162

Name : Kirtan Ghanshyambhai Soni

Roll No : CE127

ID : 19CEUON138

Termwork Prepared Under Guidance of :

Prof. Brijesh Bhatt

Prof. Pinkal Chauhan

Prof. Jigar Pandya

DHARMSINH DESAI UNIVERSITY
College Road, NADIAD-387001 (Gujarat)

Certificate

This is to certify that the project entitled as "**Collage-Portal**" is a bonafide report of the work carried out by

1) **Mr. Chintan Solanki (ID : 19CEUBS162)**
2) **Mr. Kirtan Soni (ID : 19CEUON138)**

of Department of Computer Engineering, semester VI, under the guidance and supervision of Prof. Jigar M. Pandya, Prof. Pinkal Chauhan, Prof. Brijesh Bhatt for the subject Semester Project-I during the academic year 2020-2021.

Prof. Brijesh S. Bhatt
(Project Guide and Associate Professor)
Faculty of Technology,
Dharmsinh Desai University,
Nadiad.

Dr. C.K. Bhensdadia
Head of CE Dept.,
Faculty of Technology,
Dharmsinh Desai University,
Nadiad.

INDEX

1. **Abstract**

The online education has became part of human life and we have been very well known to this situation since last year. In this kind of scenario online student-teacher (Professor) communication platforms play vary important role in the education system.

Collage-Portal is a platform that makes online communication and working much easier by providing a user-friendly environment to any institute. Giving tasks or lab works is and also taking the submission of the students work as easily as possible is the goal of our system.

2. Introduction

1.1 Brief Introduction

Collage-Portal is online classroom management system which provides ability to track, asses and give remark on students working and also keep track of the submissions to the professor/teacher. This application also provides accessibility control. Basically, a professor can create an assignment for the students and post in in the classroom of the subject. Students can see the assignment work on the assignment in their local machine and then submit their work on the classroom, the teachers can download their work. This way application provides safe and secure environment to the task of teaching.

1.2 Technologies And Platforms Used

- **Django :**

  Django is an open source framework which depends on the python libraries. Nowadays it is very popular and provides fast development, easy maintainability and clean design.

- **Git :**

  Git is a version control system which allows more than one user to work on a same project and share progress with each other concurrently.

- **Python Virtual Environment :**

  As Django is dependent on the python environment therefore most of the Django packages and tools are provided by python and Virtual Environment makes it easy to work on machines working on different operating systems.

3. Software Requirement Specificaions

Collage-Portal

Users :

(1) Students : Major group of users of the system.

(2) Professors : Users that are on the mirror side of Students.

(3) Admin : The user who manages the permissions.

Functional Requirements :

1. **Sign in/New Registration :**

- If user has an account then he/she can sign in using username and password.
- If user does not have an account then he/she can register.

INPUT :

User has to provide username and password in order to login or if the user is registering new then user has to create a unique username and password of their choice.

1) Username must be unique and password must be alphanumeric.

OUTPUT :

Registration successful message or home page if the user already had an account and was logging in.

2. **Profile/Permission control :**

- This area contains profile and all the details related to user.
- This module controls the rights of a user according to type of the account from which user has signed in.

INPUT :

NIL (Automatically decided according to user details)

OUTPUT :

Permission grant/rejected (nothing will be displayed).

3. **Courses manager :**

1) For teachers :

- Teachers can create and update courses for students using this module.

INPUT :

Provide a new course name and brief description of the course, pre-requisite of the course, etc.

OUTPUT :

A course code and link will be generated that teachers have to provide to the students.

2) For students :

- Students can enrol in any course using the link/code provided by the teacher.
- Students can unenroll from any course.

INPUT :

Enter the unique code/link provided by the teacher.

OUTPUT :

Successful enrolment message.

4. **Classroom (Home) :**

- This is also the very first page(Home page) opened after logging in.
- Students can see the list of the subjects that he/she is currently enrolled in and they can access those subjects to see the submission or the tasks provided by teacher.
- When a faculty accesses the home page they can manage the work they have given to students only on their subjects, but faculty member can only see their course they cannot see all the subjects which students have selected.

INPUT :

This is the first page opened after logging in but if user has switched to other tab they have to select the Home tab placed at top.

OUTPUT :

User will be redirected to their Classroom page.

5. **<u>Submission/file manager :</u>**

- Students will have to submit a file (File may be of any type such as pdf,word,ppt,etc).
- This system controls file management part of the program there will be submission tab provided in the subject task submission area.
- Students have to select the file from their storage or import from API mentioned above and submit from here.
- After a student uploads their work then the file is visible online and available for download on the teacher side.
- The file is accessible online so online grading is also possible.

INPUT :

The students have to upload the submission file. Them teachers have to open the submission tab provided on the top.

OUTPUT :

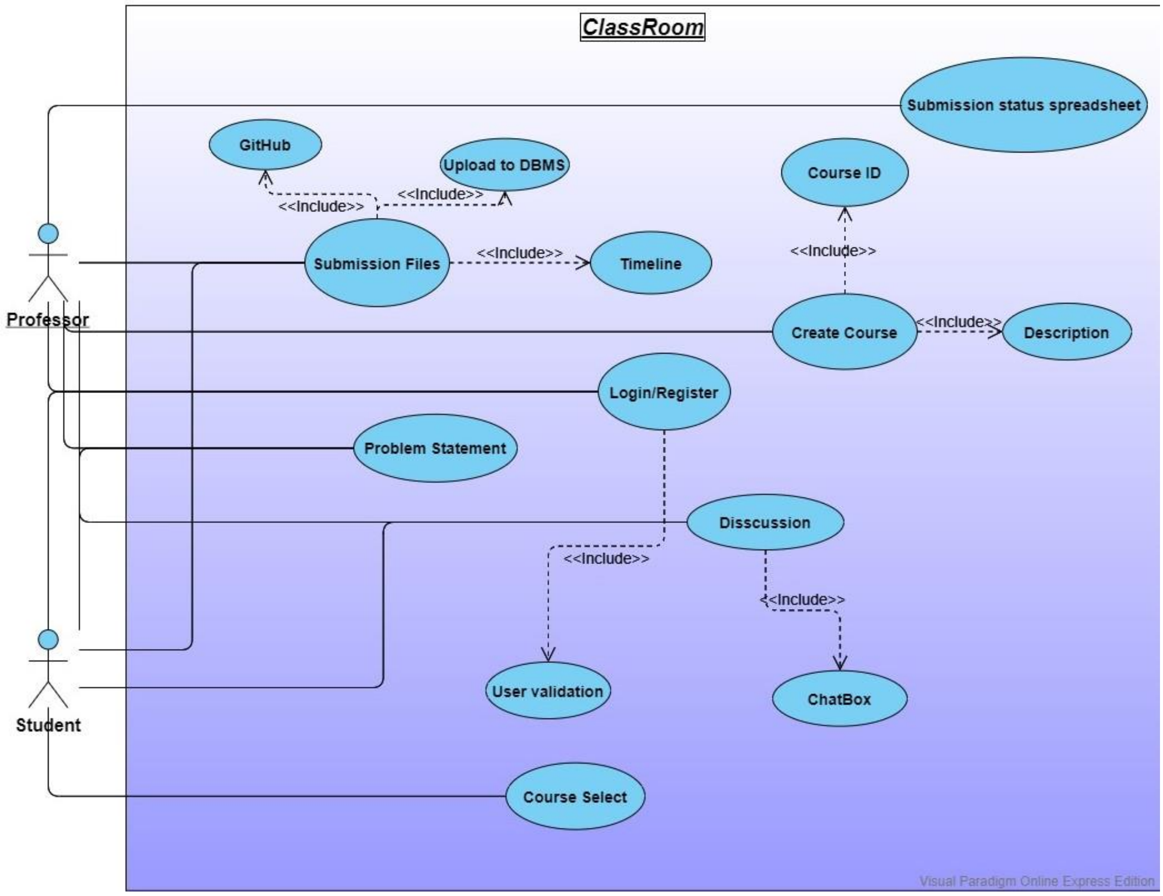
Teachers can see the list of all the students who have submitted and the students who have not. Files are available to download for teachers

6. **<u>DBMS :</u>**

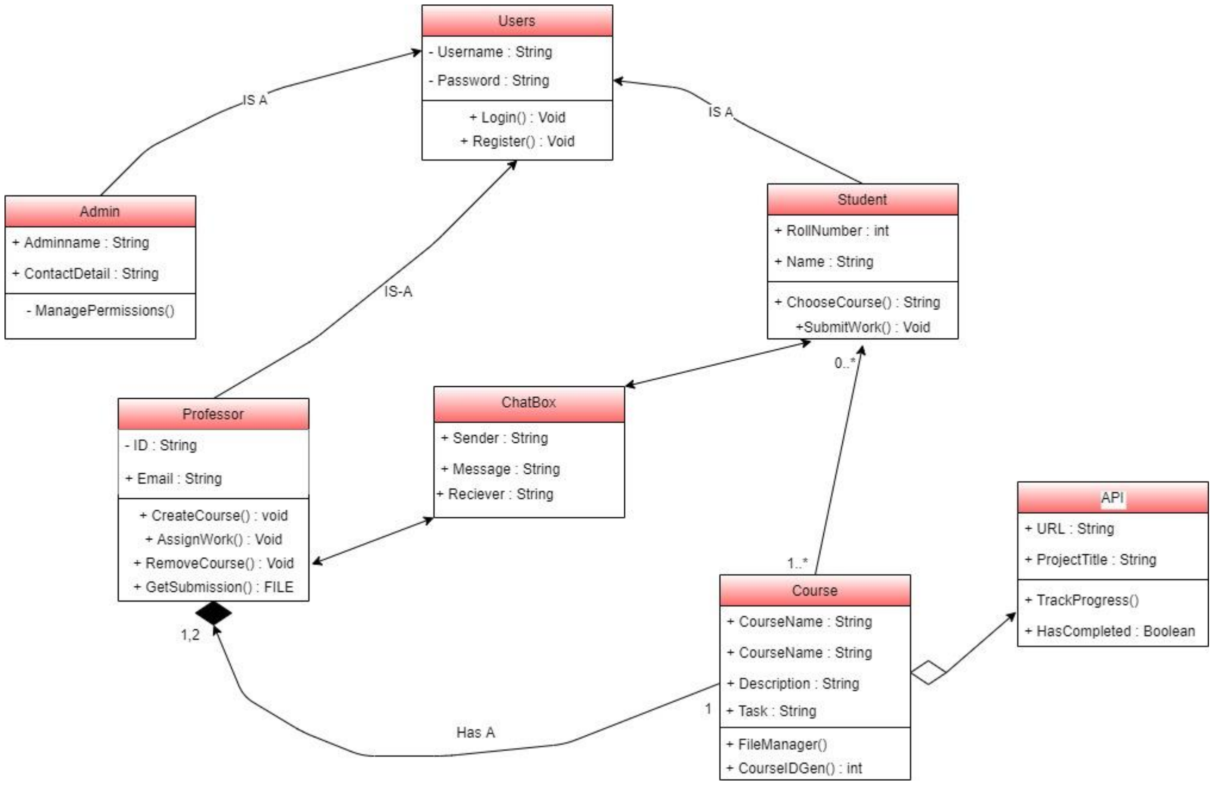
- A database module that will run in the back end to store all the submissions and uploads made by students in a FILE HANDLING SYSTEM.

4. Design Documents/Diagrams

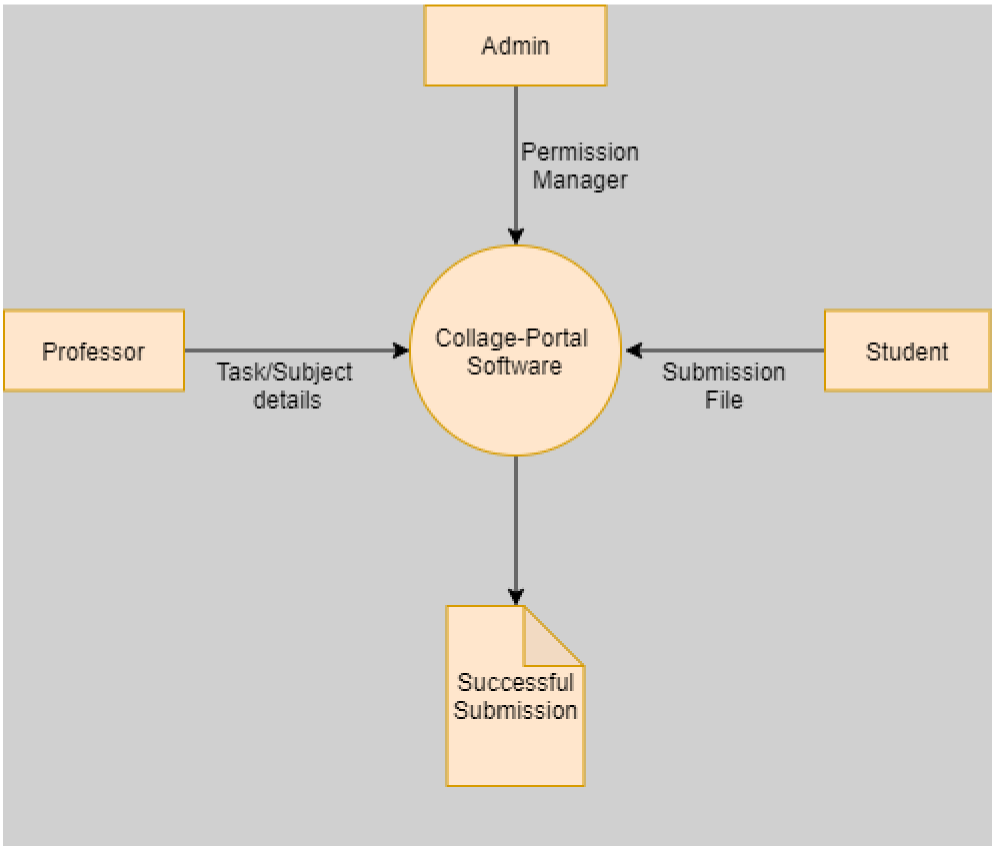
(1) Use Case Diagram :



(2) **Class Diagram :**

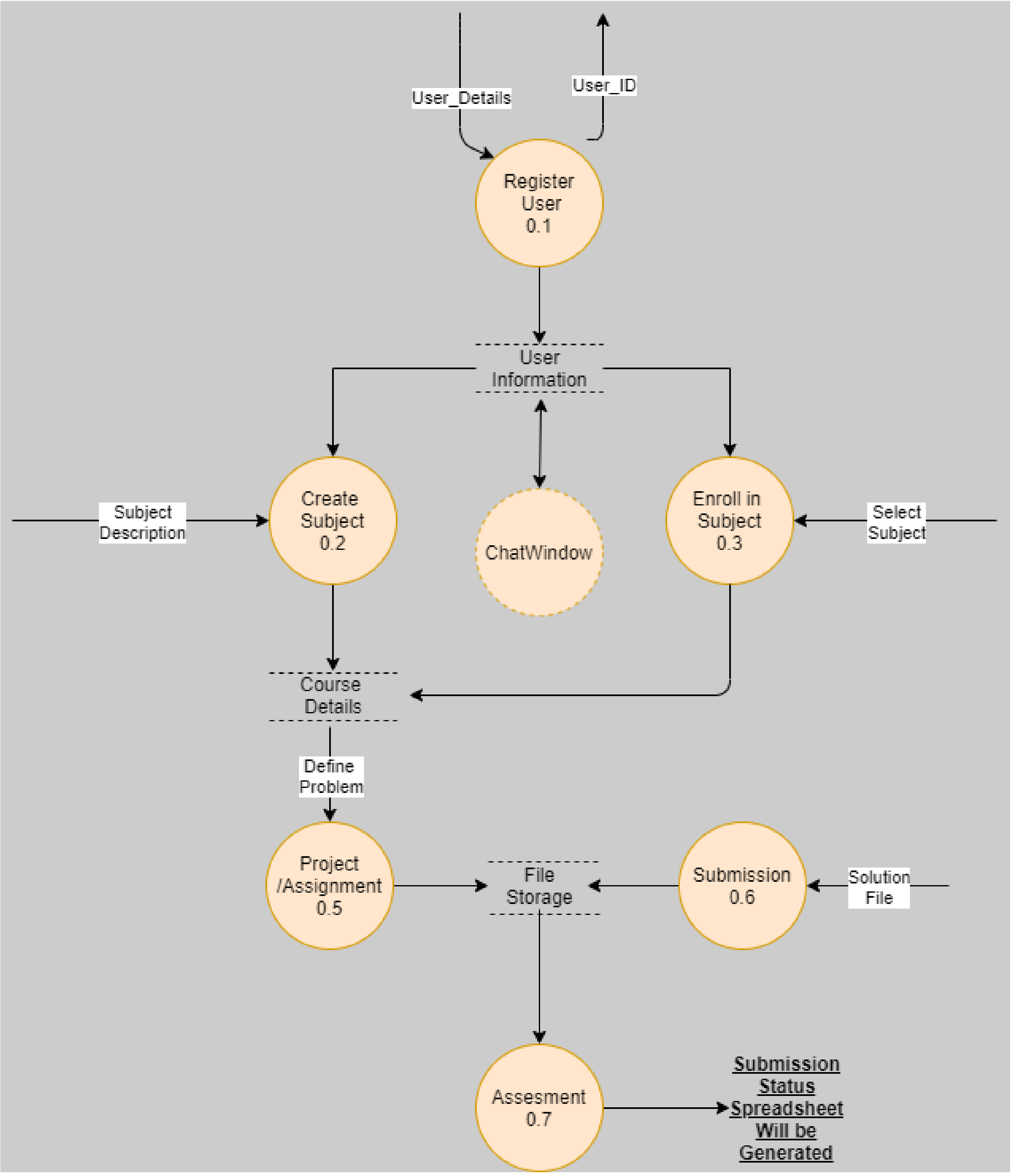


(3) **Context Diagram :**

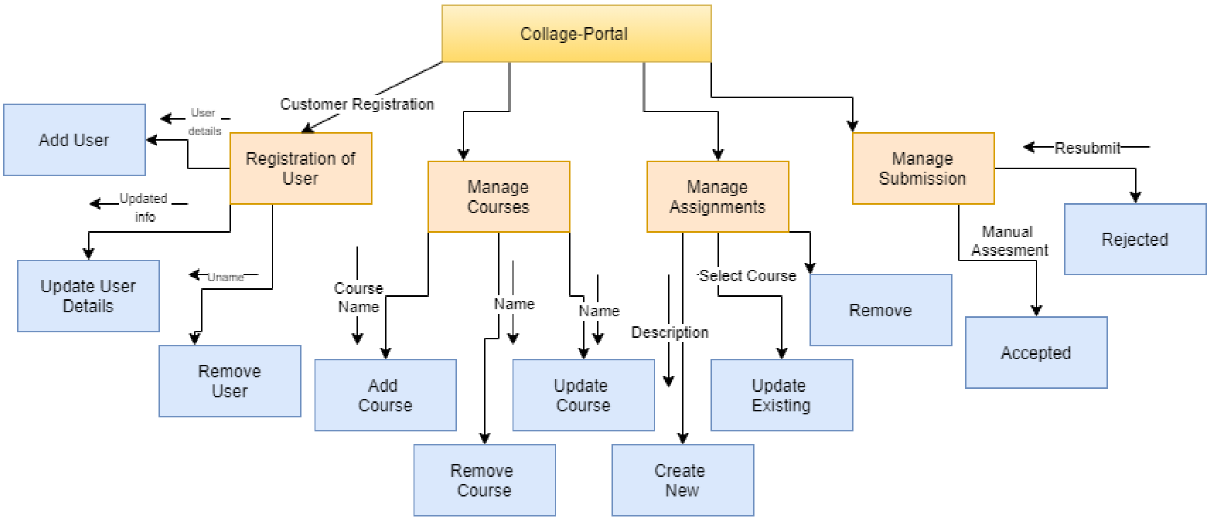


(A) Context Diagram

(4) Data Flow Diagram :

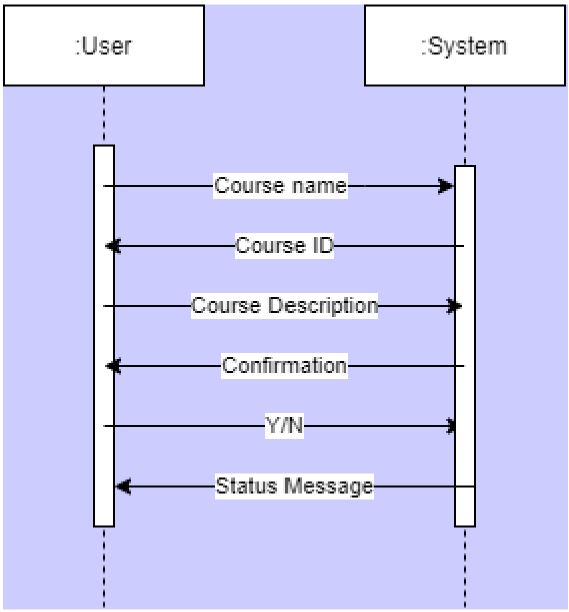


(5) Structure Chart :

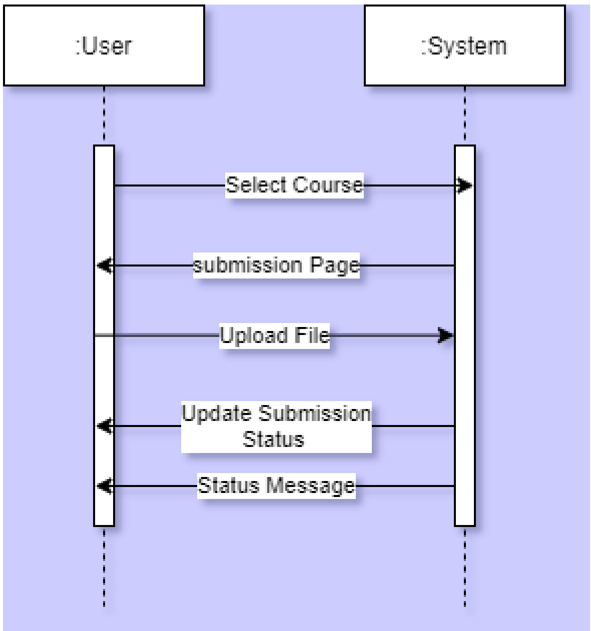


(6) Sequance Diagram :

Create Subject Sequance Diagram :

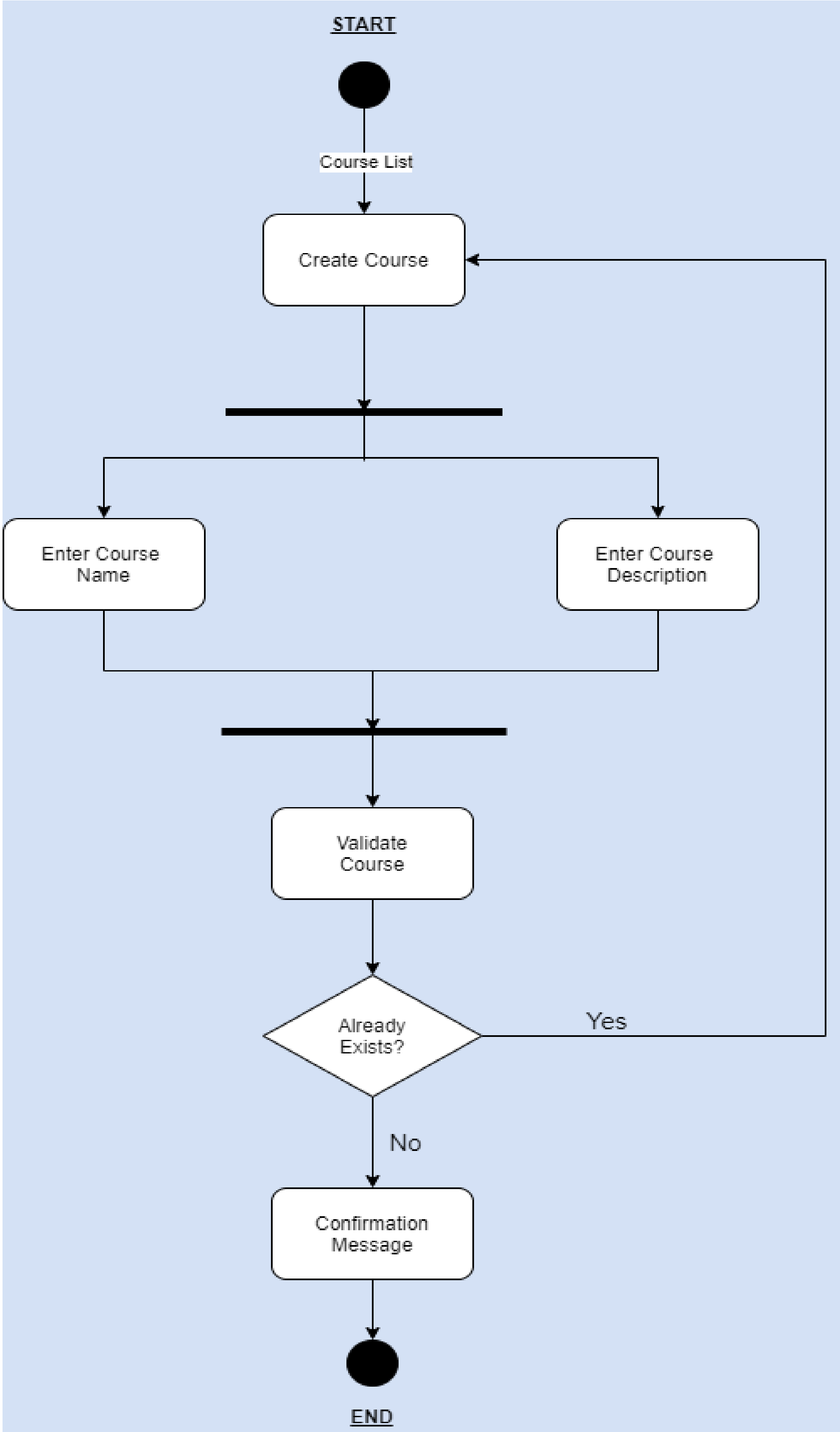


Submission Sequence Diagram :

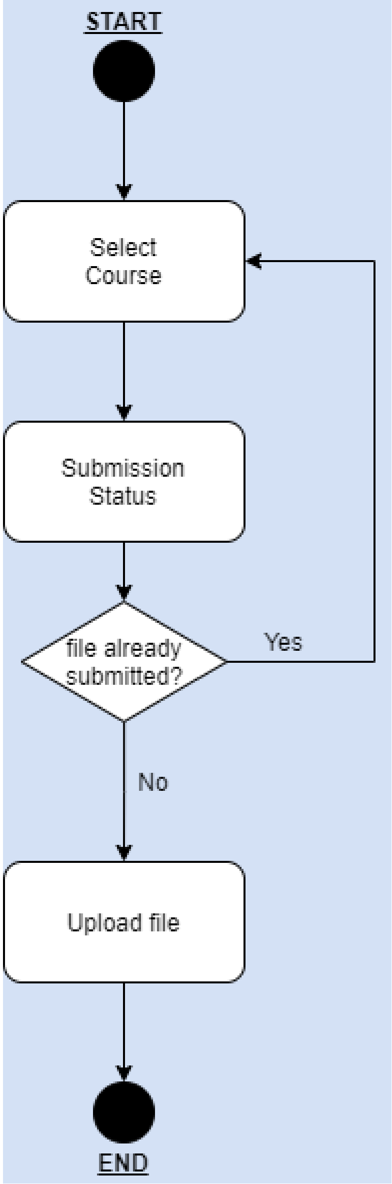


(7) Activity Diagram :

Create/Add subject Activity Diagram :



Submission Activity Diagram :



5. Implementation Details :

List of Required Modules :

Login/Authentication Module :

This modules allow new users to create accounts to use the system and also provides user authentication system for the already registered clients to log in to their account. This module also decides rights and permissions given to the specified user when the user logs in.
E.g. A user who is logged in to the system using professor account only has right to create or post a new assignment.

Create Assignment Module :

This module allows only the professor accounts to add or create a new assignment for their subject whenever a professor tries to create a new assignment the system will ask for the title of the assignments and also a description of the assignment.

File management Module :

While creating a new assignment professor can also add any file to explain the problem statement in full detail this module manages these file from the professor as well as student will submit their work on some kind of file so file management will take care of that too basically file manager will interact with the database when any transaction that contain file transfer takes place.

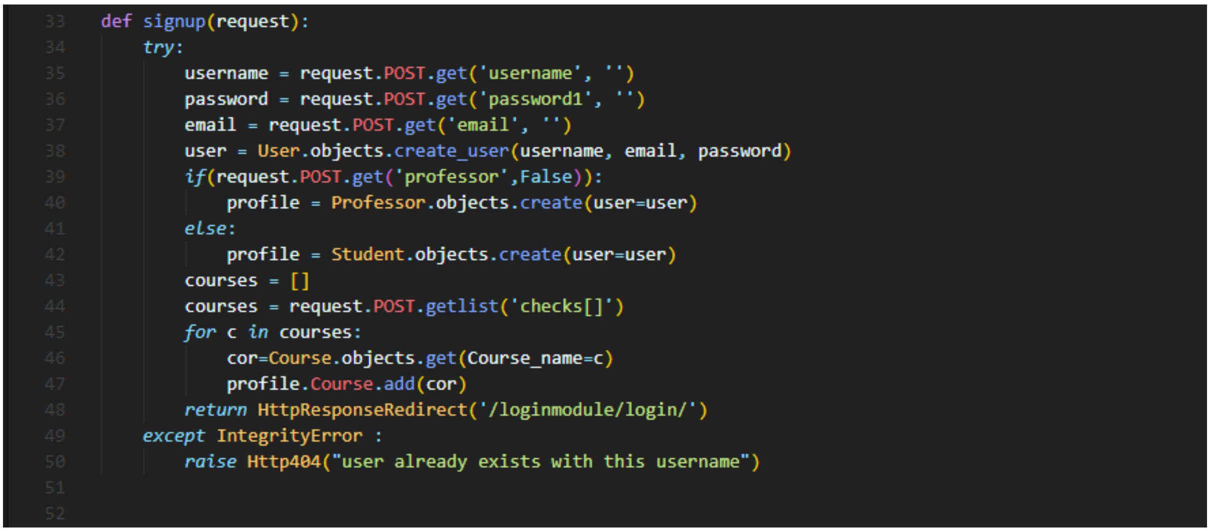
Course Management Module :

This module lets professors create new courses in their respective branches and also allows students to enrol in those courses. Not all students can join every class and not all professors can create a new subjects for any branches.

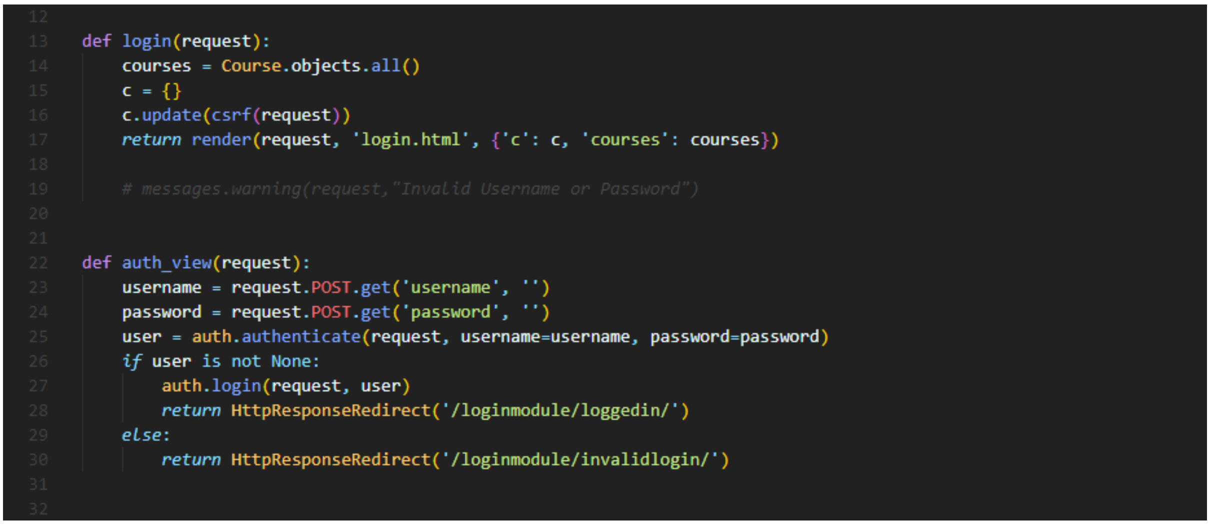
Major Functionalities :

Registration/login :

Django default user management system is used for authentication.



```python
def signup(request):
    try:
        username = request.POST.get('username', '')
        password = request.POST.get('password1', '')
        email = request.POST.get('email', '')
        user = User.objects.create_user(username, email, password)
        if(request.POST.get('professor',False)):
            profile = Professor.objects.create(user=user)
        else:
            profile = Student.objects.create(user=user)
        courses = []
        courses = request.POST.getlist('checks[]')
        for c in courses:
            cor=Course.objects.get(Course_name=c)
            profile.Course.add(cor)
        return HttpResponseRedirect('/loginmodule/login/')
    except IntegrityError :
        raise Http404("user already exists with this username")
```



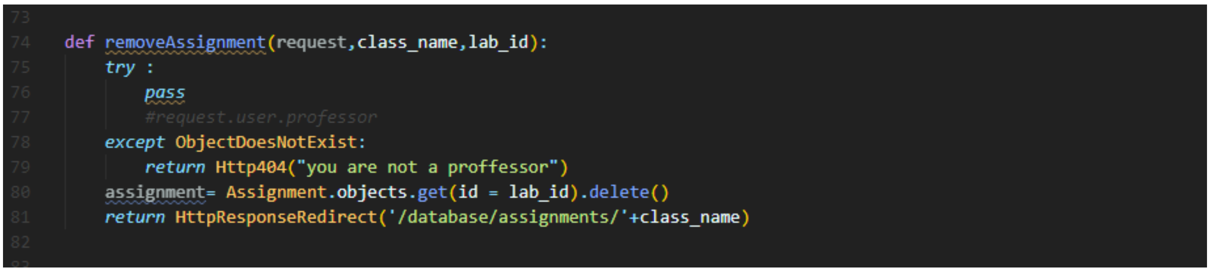
```python
def login(request):
    courses = Course.objects.all()
    c = {}
    c.update(csrf(request))
    return render(request, 'login.html', {'c': c, 'courses': courses})

    # messages.warning(request,"Invalid Username or Password")


def auth_view(request):
    username = request.POST.get('username', '')
    password = request.POST.get('password', '')
    user = auth.authenticate(request, username=username, password=password)
    if user is not None:
        auth.login(request, user)
        return HttpResponseRedirect('/loginmodule/loggedin/')
    else:
        return HttpResponseRedirect('/loginmodule/invalidlogin/')
```

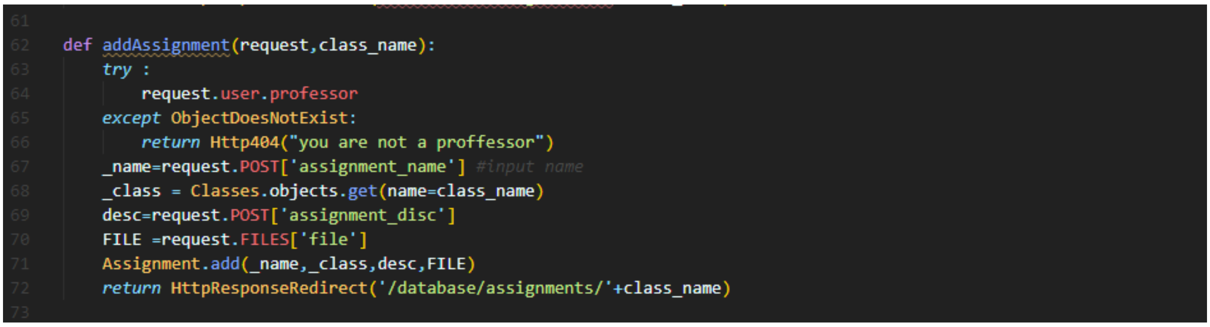
Add/Remove assignment :

Add :



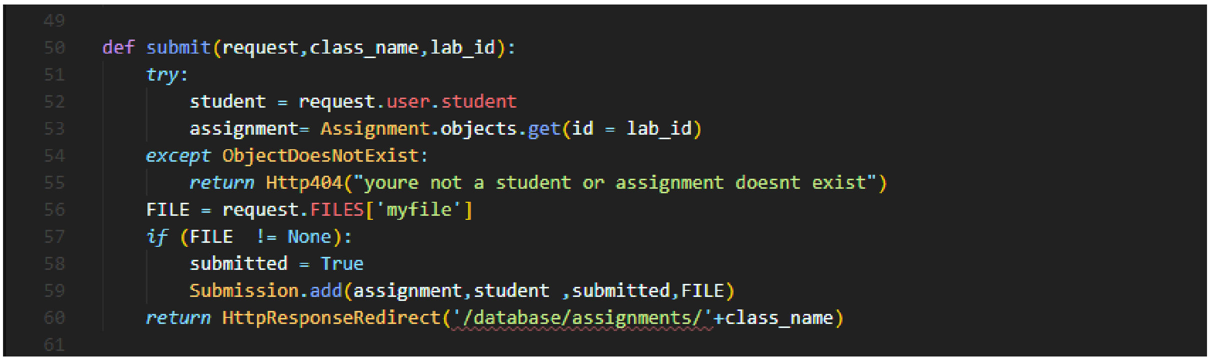
```python
def removeAssignment(request,class_name,lab_id):
    try :
        pass
        #request.user.professor
    except ObjectDoesNotExist:
        return Http404("you are not a proffessor")
    assignment= Assignment.objects.get(id = lab_id).delete()
    return HttpResponseRedirect('/database/assignments/'+class_name)
```

Remove :



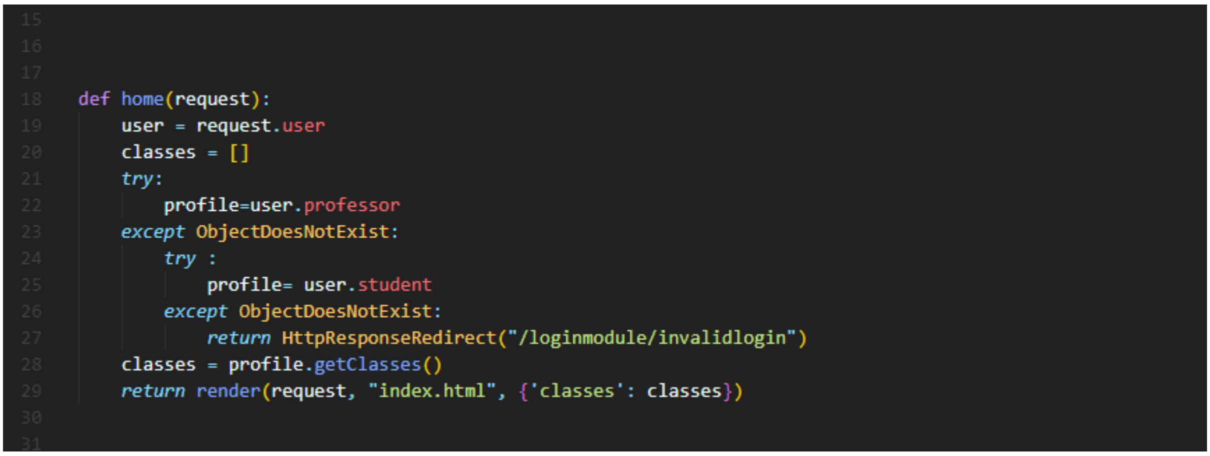
```python
def addAssignment(request,class_name):
    try :
        request.user.professor
    except ObjectDoesNotExist:
        return Http404("you are not a proffessor")
    _name=request.POST['assignment_name'] #input name
    _class = Classes.objects.get(name=class_name)
    desc=request.POST['assignment_disc']
    FILE =request.FILES['file']
    Assignment.add(_name,_class,desc,FILE)
    return HttpResponseRedirect('/database/assignments/'+class_name)
```

Submit :



```python
def submit(request,class_name,lab_id):
    try:
        student = request.user.student
        assignment= Assignment.objects.get(id = lab_id)
    except ObjectDoesNotExist:
        return Http404("youre not a student or assignment doesnt exist")
    FILE = request.FILES['myfile']
    if (FILE  != None):
        submitted = True
        Submission.add(assignment,student ,submitted,FILE)
    return HttpResponseRedirect('/database/assignments/'+class_name)
```

Classroom Home :



```python
def home(request):
    user = request.user
    classes = []
    try:
        profile=user.professor
    except ObjectDoesNotExist:
        try :
            profile= user.student
        except ObjectDoesNotExist:
            return HttpResponseRedirect("/loginmodule/invalidlogin")
    classes = profile.getClasses()
    return render(request, "index.html", {'classes': classes})
```

6. Data Dictionary :

Auth-User :

Name	Data Type	Length	Description
Id	Int	11	PK,Auto-Inc
Password	Varchar	6	
Last_login	Datetime	-	
Is_superuser	Tinyint	1	
Username	varchar	150	Index,unique
First_name	Varchar	150	
Last_name	Varchar	150	
Email	Varchar	254	
Is_staff	Tinyint	1	
Is_active	Tinyint	1	
Date_joined	Datetime	6	

Database_classes :

Name	Data Type	Length	Description
Id	Int	11	PK
Name	Varchar	100	
Description	Varchar	50	
Course_id	Int	11	
Professor_id	Int	11	

Database_course :

Name	Data Type	Length	Description
Id	Int	11	PK
Course_name	Varchar	100	
Department	Varchar	20	

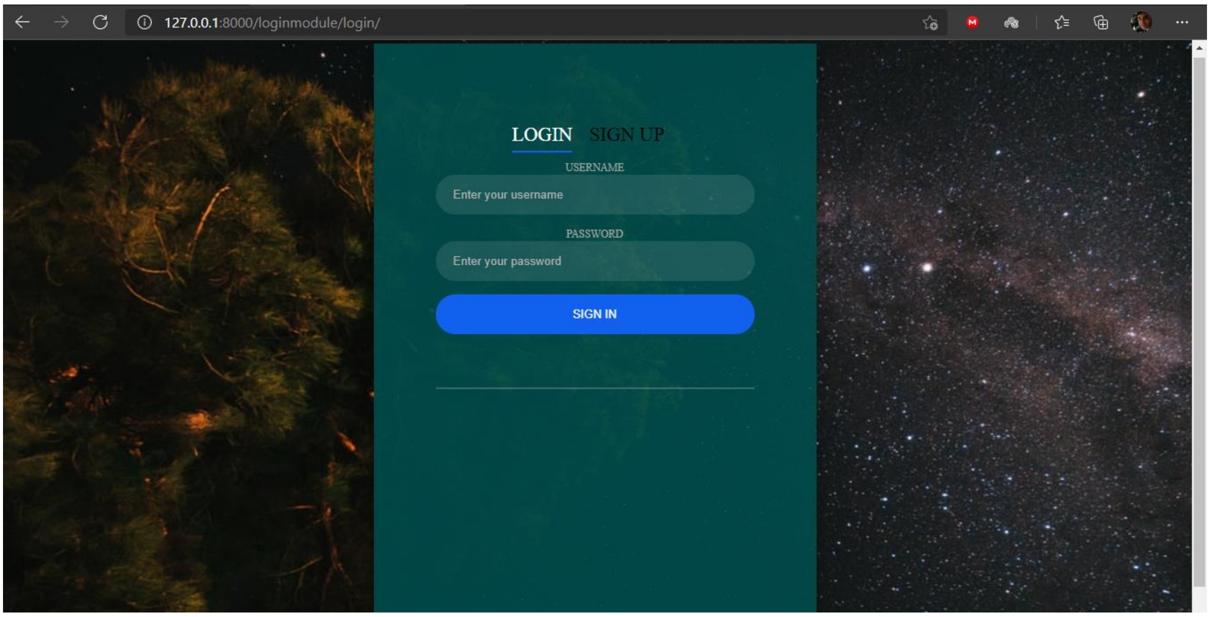
Database_submission :

Name	**Data Type**	**Length**	**Description**
Id	Int	11	PK,AI
Submitted	Tinyint	1	
File	Varchar	100	
Remarks	Longtext		
Assignment_id	Int	11	
Student_id	Int	11	
Date	Datetime	6	

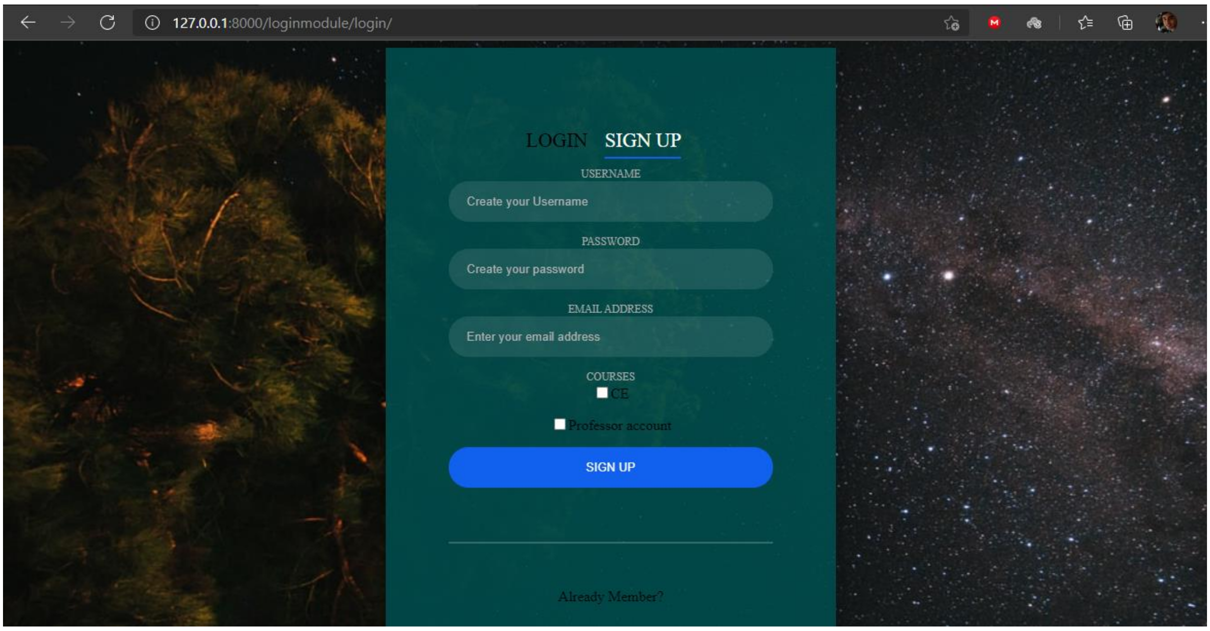
7. Work Flow/Layouts :

Screenshots :

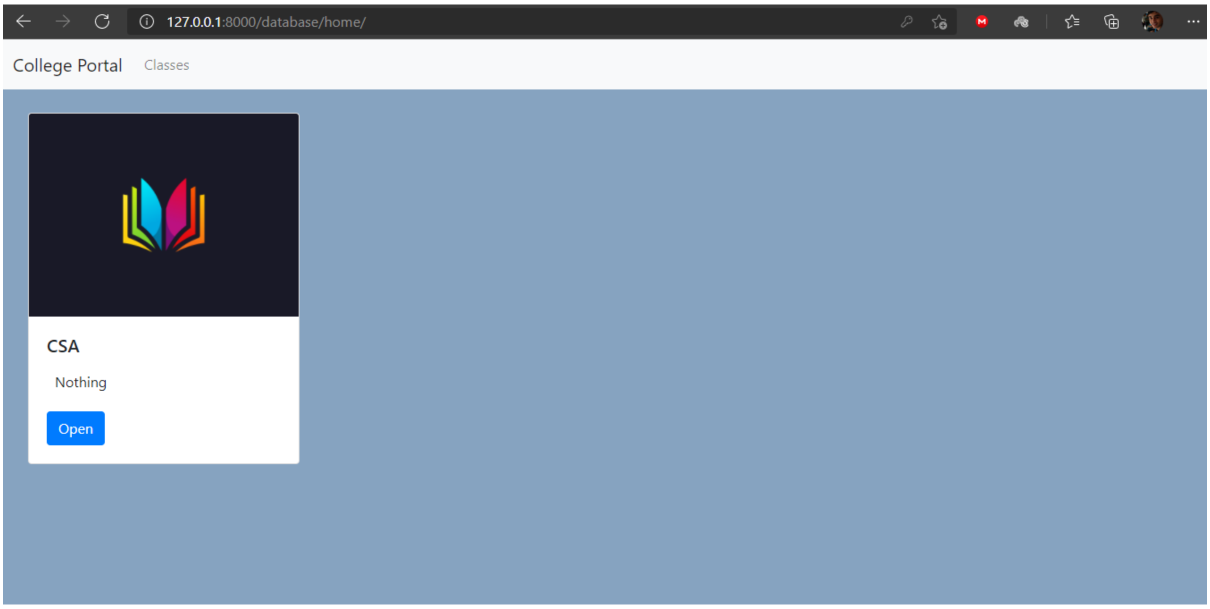
Login :



Sign Up :



Home Page

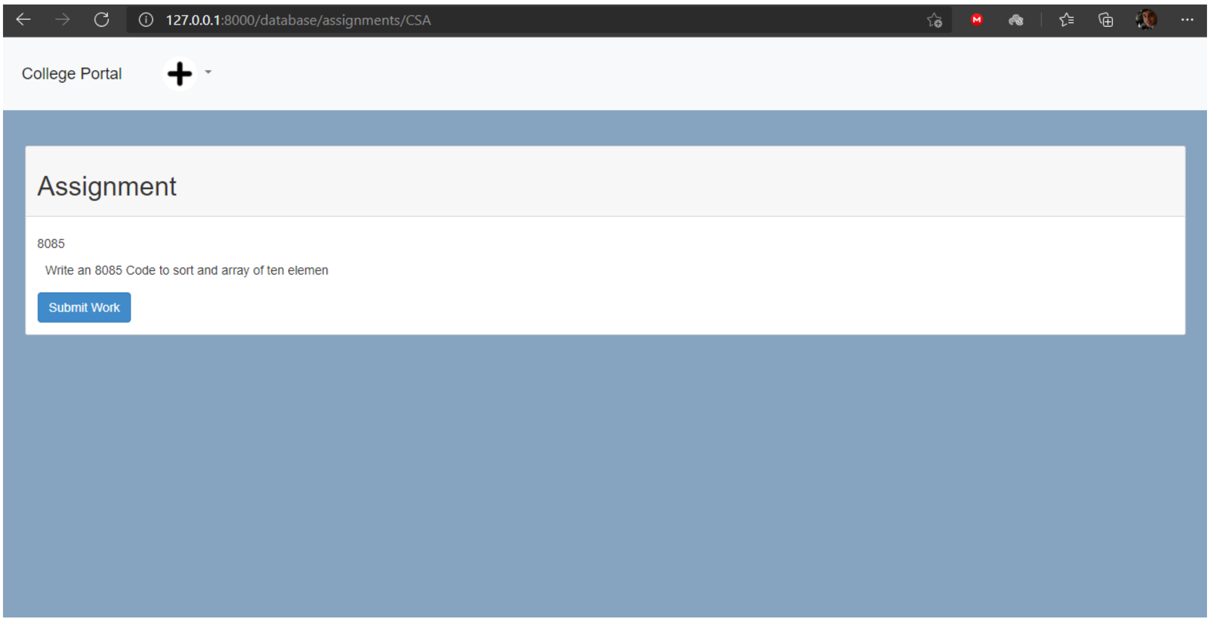


127.0.0.1:8000/database/home/

CSA

Nothing

Assignment page(StudentSide)



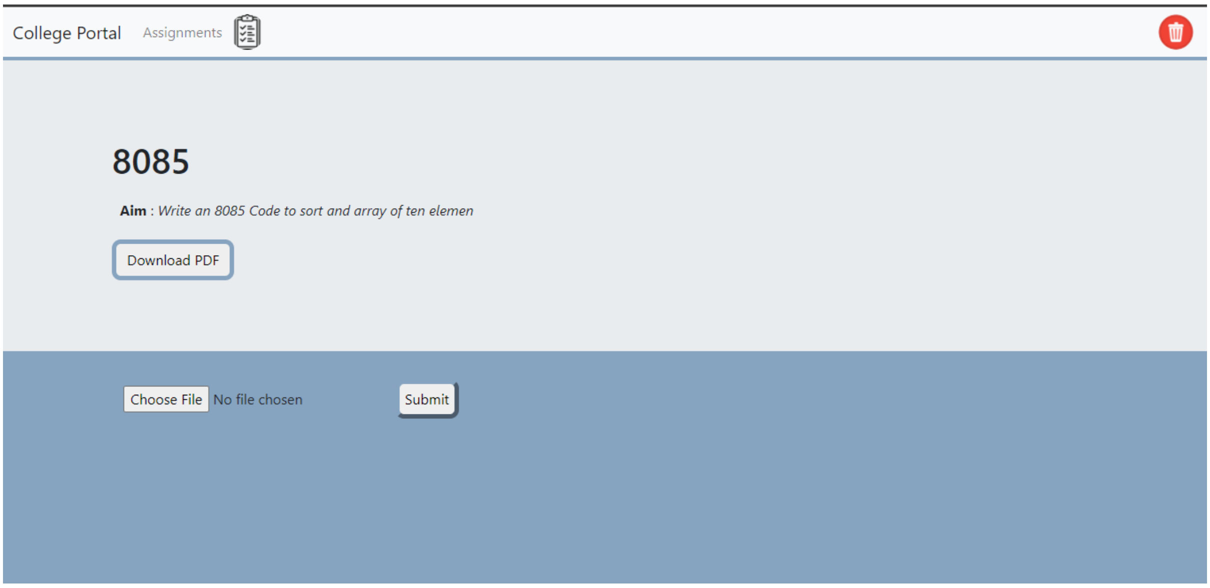
127.0.0.1:8000/database/assignments/CSA

Assignment

8085

Write an 8085 Code to sort and array of ten elemen

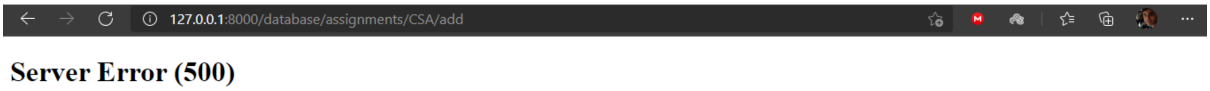
Submission Page(ProfessorSide)



8085

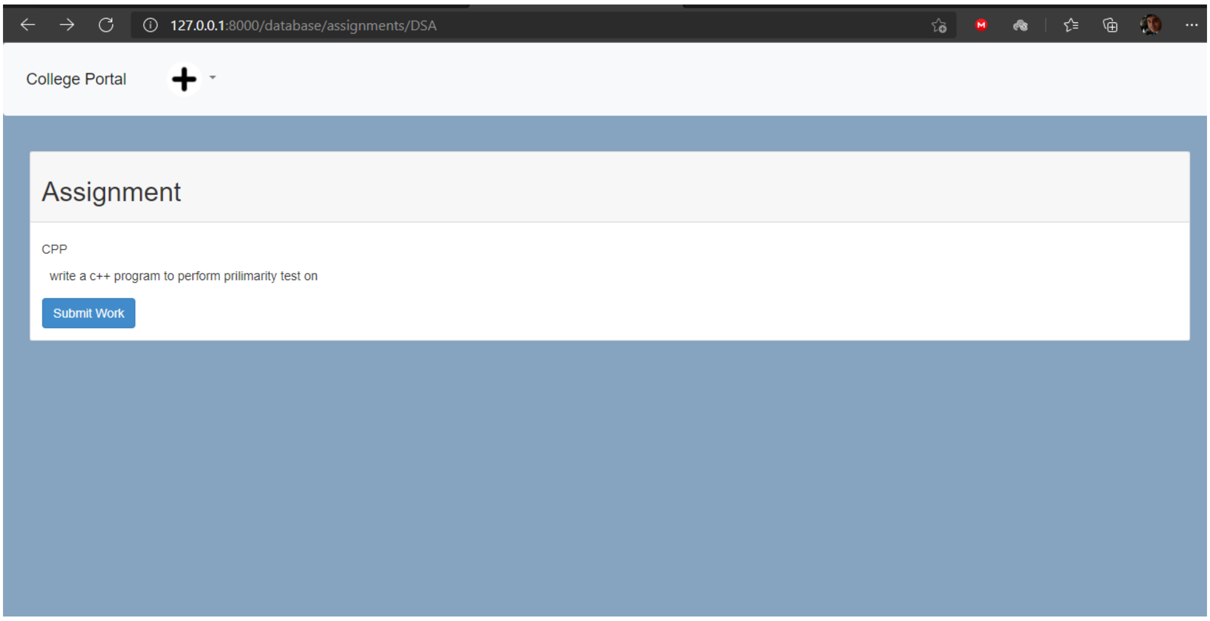
Aim : *Write an 8085 Code to sort and array of ten elemen*

When student tries to create assignment



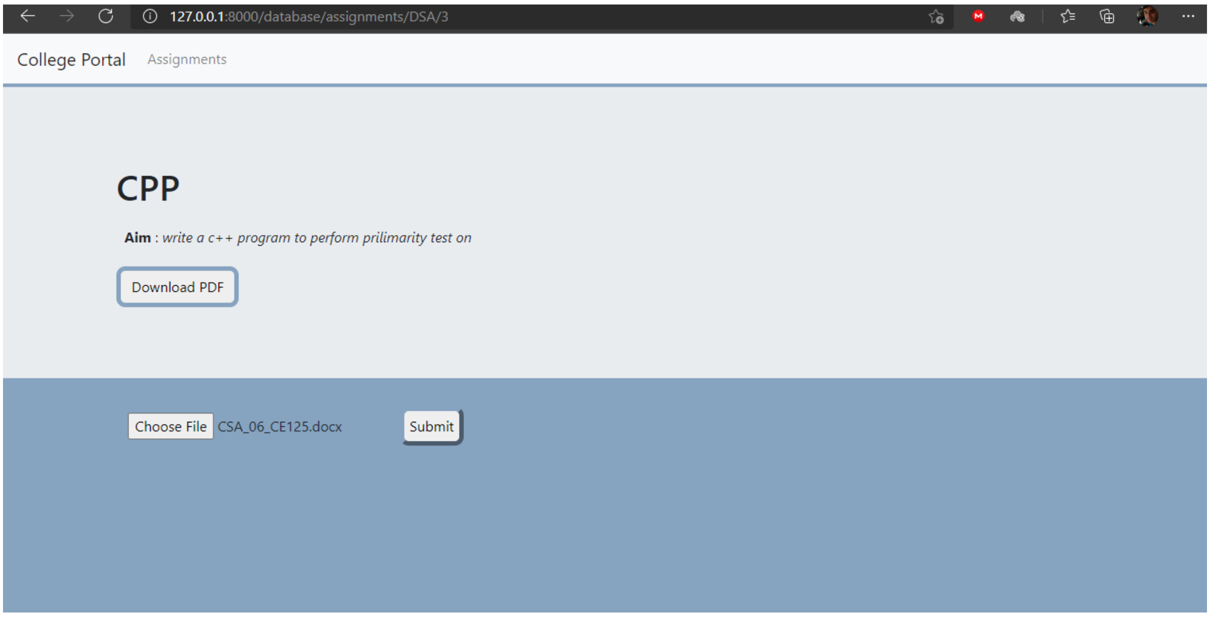
Server Error (500)

Assignment Page (StudentSide)



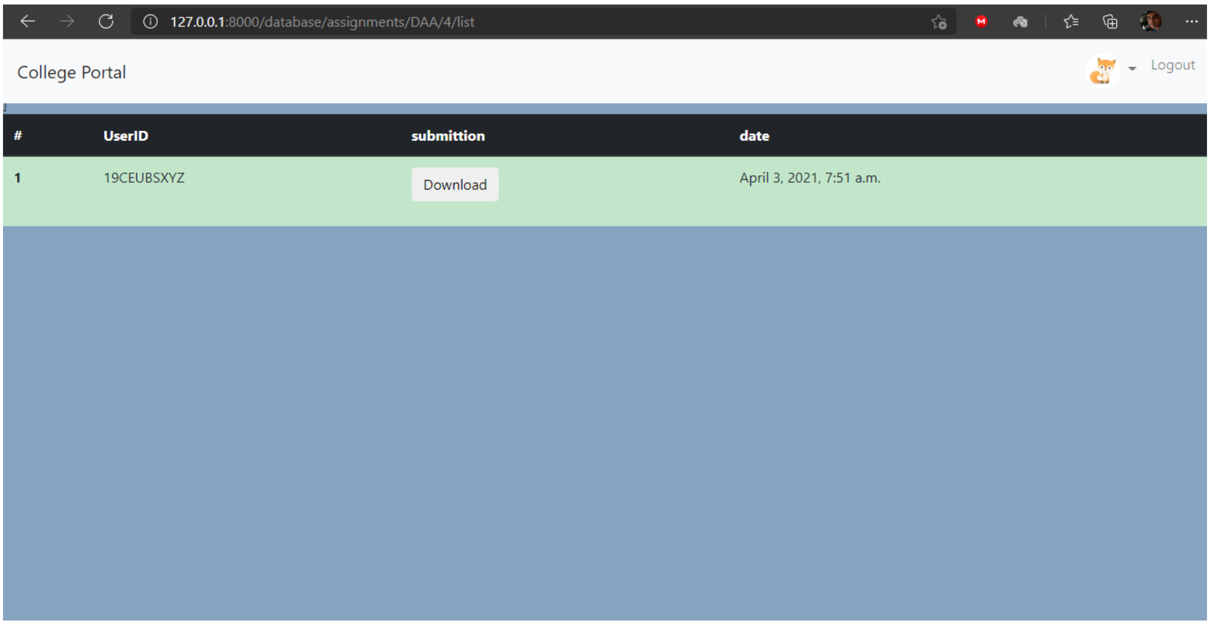
127.0.0.1:8000/database/assignments/DSA
College Portal
+
Assignment
CPP
write a c++ program to perform prilimarity test on
Submit Work

Submission Page(StudentSide)



127.0.0.1:8000/database/assignments/DSA/3
College Portal
Assignments
CPP
Aim : write a c++ program to perform prilimarity test on
Download PDF
Choose File
CSA_06_CE125.docx
Submit

Submission List (Only accessible by students)



#	UserID	submittion	date
1	19CEUBSXYZ	Download	April 3, 2021, 7:51 a.m.

8. Conclusion :

We can conclude that the system is reliable and user-friendly for both students and professors. And also provides good environment to work together and better way of communication than handling assignments and lab works physically. Also platform can be used to conduct online examinations.

Also CollagePortal application provides a nice way of managing the permissions. Students can only work on assignments that are provided to them by the teacher and submit their work. Only teachers have right to modify or create new subjects and assignments.

Not only the visible front end components but all the transactions with the database at backend are also well thought and carried out efficiently making retrieving of data from database easy and convenient. All the integrations of the RDBMS tables and structures are well designed so that interchange of data can easily be done.

9. Limitations and future extensions :

Limitations :

- Forgot password functionality not provided.
- Professors can create subjects and assignments for pre-defined hard coded subjects only.
- All types of submission files cannot be opened online.
- We tried our best to keep most of the functionalities available to the main users of the system but still admin has a little bit of interference in the working.
- Deletion of users can only be done by the Admin.

Future-Extensions :

- All the limitations that are mentioned above can be implemented in the future extensions of this web application.
- Control of the program can be made more biased towards students and teachers.
- Better authentication system can be implemented
- To make it easy to join classes we can generate class code of a scannable QR code to join the class.

10.Bibliography :

Reference materials used for this project :

Stackoverflow : https://stackoverflow.com/
DJango official documentation :- https://docs.djangoproject.com/en/3.1/
W3Schools :- https://www.w3schools.com/
Bootstrap libraries, and many more….

Tools used in making of the project :

DJango framework : https://www.djangoproject.com/
Visual Studio Code : https://code.visualstudio.com/
GitHub : https://github.com/
Xampp as database deployment
MySql as database