PES University, Bengaluru

(Established under Karnataka Act 16 of 2013)

Department of Computer Science & Engineering
Session: Jan - May 2022

Object Oriented Analysis and Design with Java - Laboratory
UE19CS353

Mini Project

Report on

Forum for Educational Institutions

By:

Gokul Karthik – PES1UG19CS170

Gurukiran S   – PES1UG19CS173

Gaurav J   – PES1UG19CS599

6th Semester section C

1. **Project Description**

   Forum for educational institutions where the students and teachers come together to learn and communicate with each other.

   Eduforum is a platform for students to discuss their doubts with fellow students and faculty.
   We have a homepage where the student and teacher and admin can choose as whom they want to login as.

   A teacher can login and create subjects and also manage topics / posts under the subjects they created. They can allow or disallow the access to students for the subjects created by them. Teachers can mark the topics as closed and can also upvote and downvote posts.

   Teachers create subjects, subjects contain topics (created by students/teachers), each topic can contain multiple posts. each post will contain the vote feature where the teacher or the student can upvote or downvote
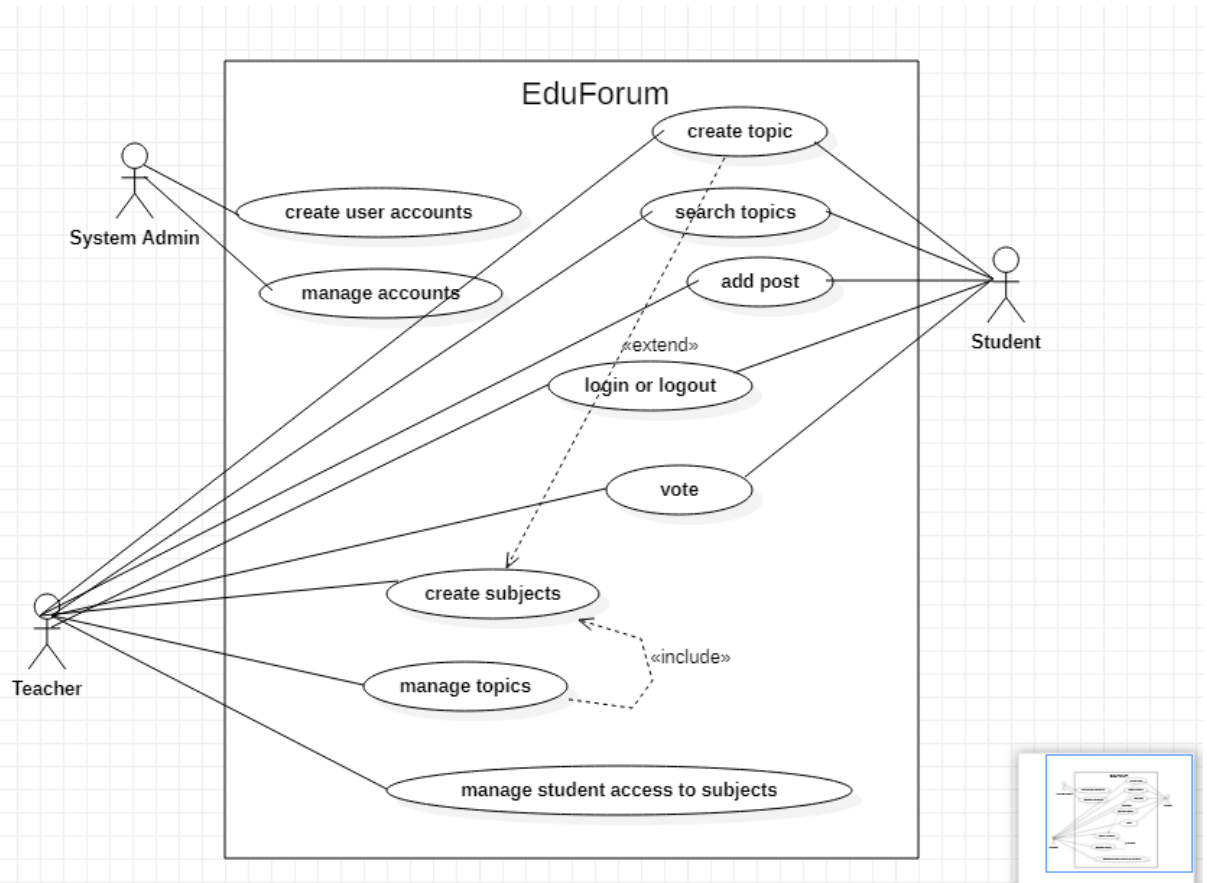
   Students can login and can join a subject using the code assigned by the system admin or the teacher and create topics related to the subjects. He can add posts under topics in the subjects which he has joined. Students can also upvote and downvote posts.

   Admin creates student/teacher accounts, He can manage all the accounts in the forum.
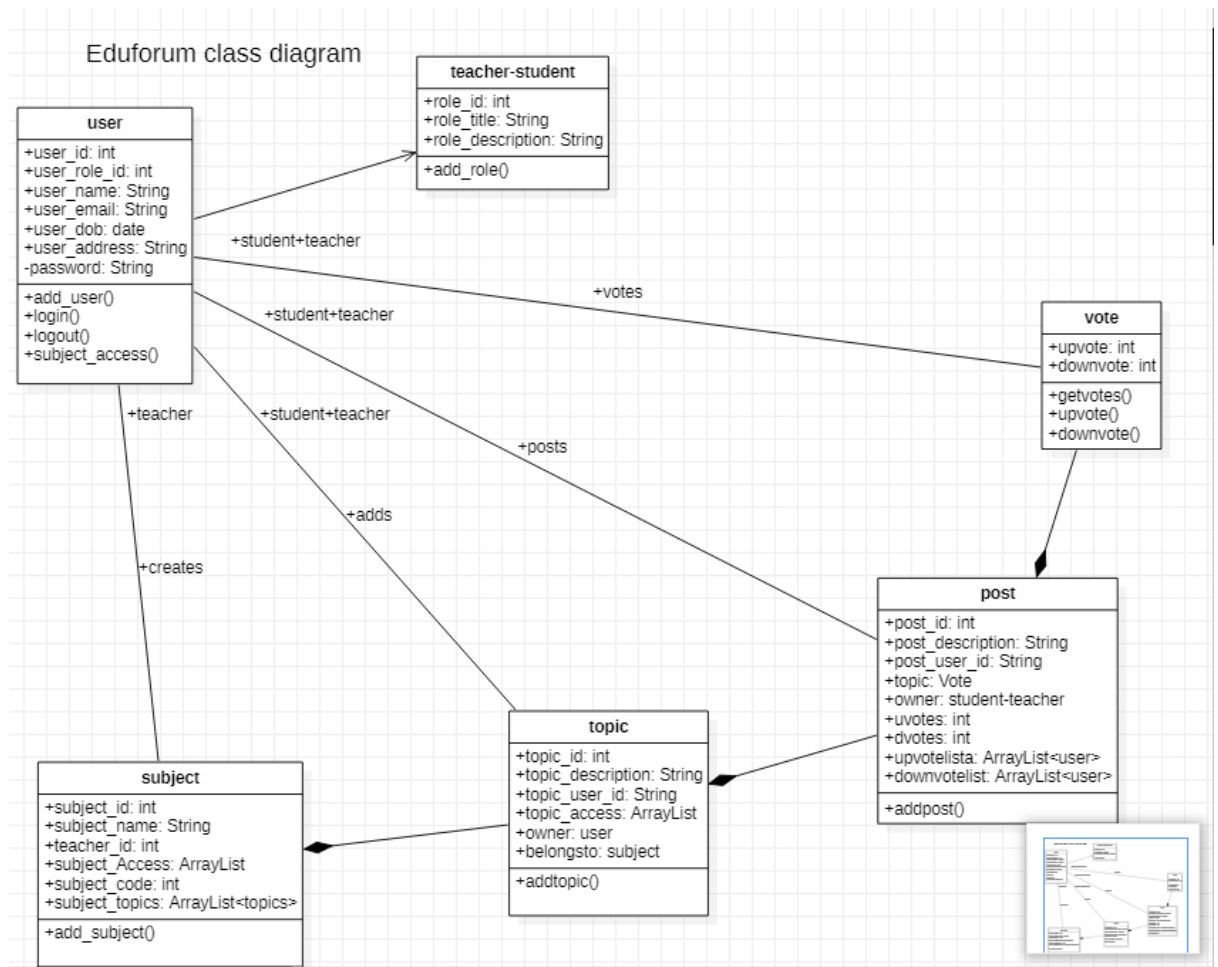
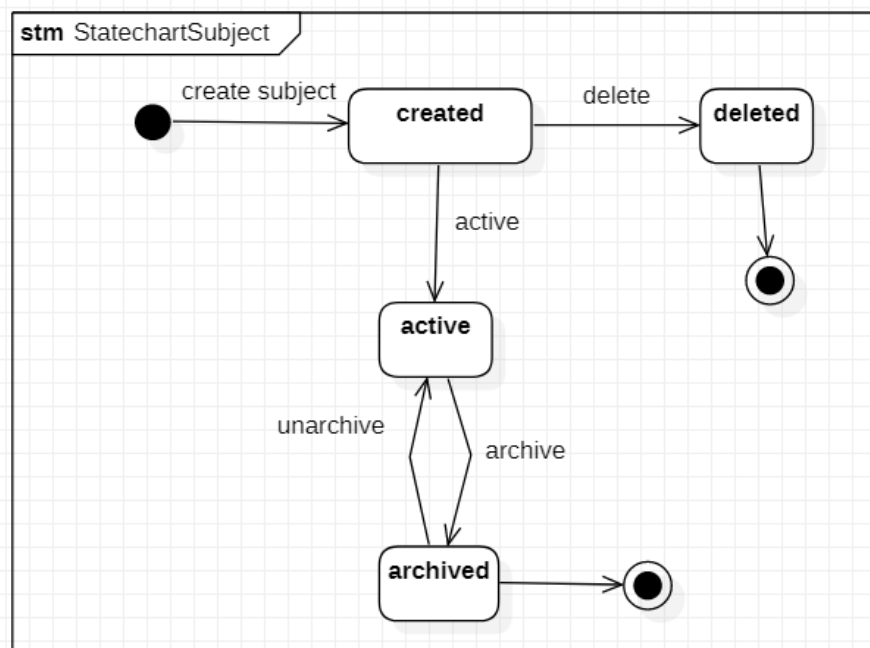   **Github repo for Eduforum**
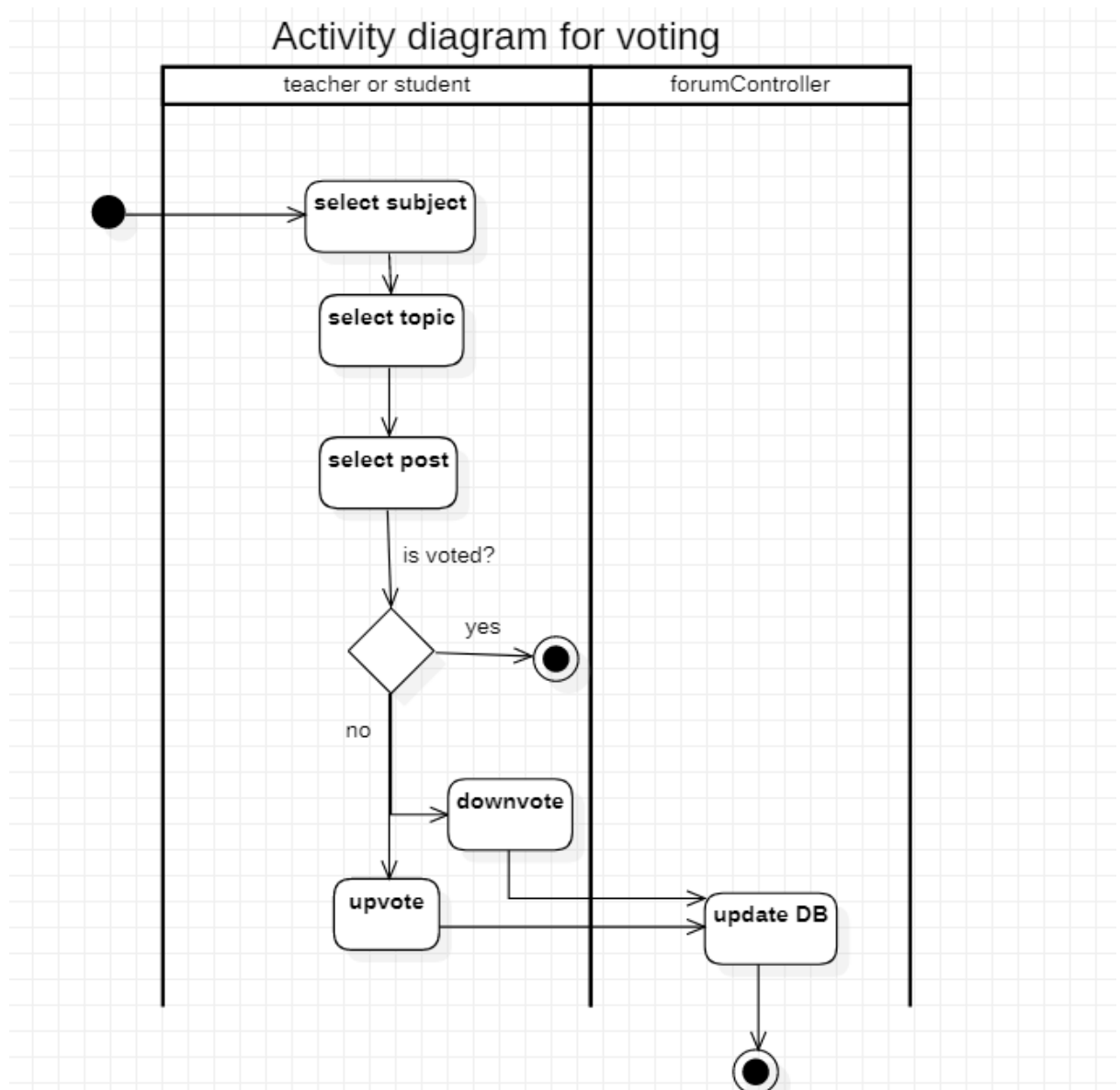
2. Analysis and Design Models

Use case diagram

class diagram for eduforum

Eduforum class diagram

teacher-student
+role_id: int
+role_title: String
+role_description: String

+add_role()

user
+user_id: int
+user_role_id: int
+user_name: String
+user_email: String
+user_dob: date
+user_address: String
-password: String

+add_user()
+login()
+logout()
+subject_access()

+student+teacher

+votes

vote
+upvote: int
+downvote: int

+getvotes()
+upvote()
+downvote()

+student+teacher

+teacher   +student+teacher

+posts

+adds

+creates

post
+post_id: int
+post_description: String
+post_user_id: String
+topic: Vote
+owner: student-teacher
+uvotes: int
+dvotes: int
+upvotelista: ArrayList<user>
+downvotelist: ArrayList<user>

+addpost()

topic
+topic_id: int
+topic_description: String
+topic_user_id: String
+topic_access: ArrayList
+owner: user
+belongsto: subject

+addtopic()

subject
+subject_id: int
+subject_name: String
+teacher_id: int
+subject_Access: ArrayList
+subject_code: int
+subject_topics: ArrayList<topics>

+add_subject()

State chart of the Subject

stm StatechartSubject

create subject → **created** → delete → **deleted**

active

active

unarchive   archive

archived

Activity diagram for voting

Activity diagram for voting

3. **Tools and Frameworks Used:**
   spring boot : used spring boot as the development framework for java web application
   Visual studio code : used to write the code
   Thymeleaf : used to easily access and manage database
   mongoDB : used mongoDB as the database for storing the data
   spring initializer : used to generate the spring boot project structure

   Description about the database – if applicable add ER diagram
   We used mongoDB database to store the database, since we don't have a relational structure to the data that we are trying to store so we decided to use mongoDB which is best to store the unstructured data. The ER diagram is not applicable in our database.

4. **Design Principles and Design Patterns Applied**

   Highlight the classes which form the structure of the Design pattern applied.
   Adapter pattern:
   The **adapter pattern** is a software design pattern (also known as wrapper, an alternative naming shared with the decorator pattern) that allows the interface of an existing class to be used as another interface. It is often used to make existing classes work with others without modifying their source code.

   Authentication checker for the system that the user is logged in or not. The classes which use the adapter pattern which helps in authenticating the system are
   - student class
   - teacher class
   - admin class

Highlight any 2 design principles followed in your design

we followed and used 2 Solid design principles

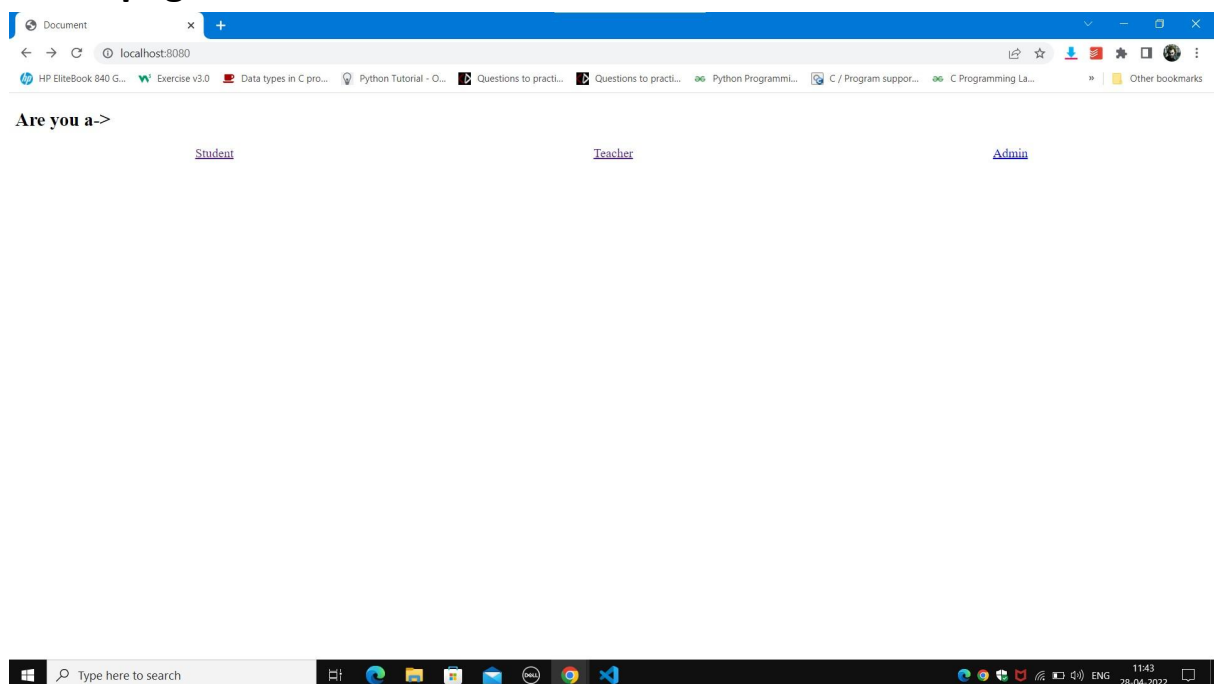**Single responsibility principle**

Following this principle means that each class only does one thing and every class or module only has responsibility for one part of the software's functionality. We have classes called service such as StudentService.java TeacherService.java SubjectService.java etc that only has one responsibility for functionality of the application such as student, teacher, Subject. Which makes the application better.

Liskov Substitution Principle

The classes that use the liskov substitution principle are user class, teacher class, student class where the user class acts as the base class for the teacher and student class, so whenever we want to refer to the student class or teacher class we can directly use the reference of the user class.

5. Application Screenshots (3-4 important pages)
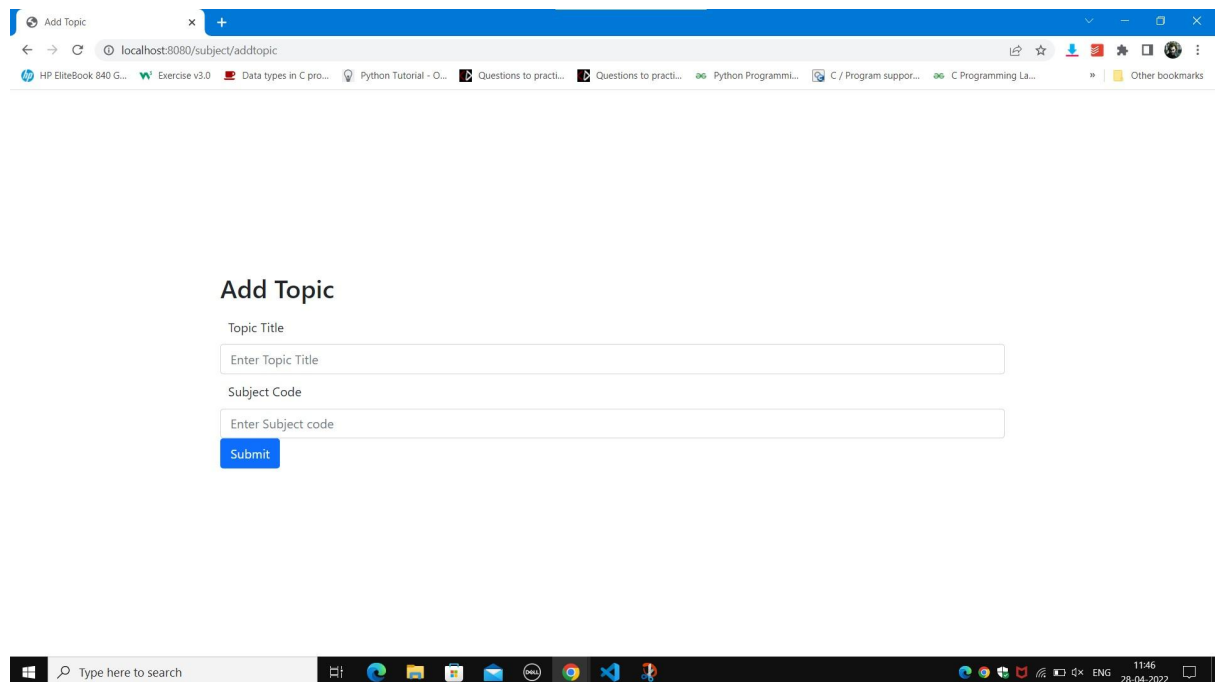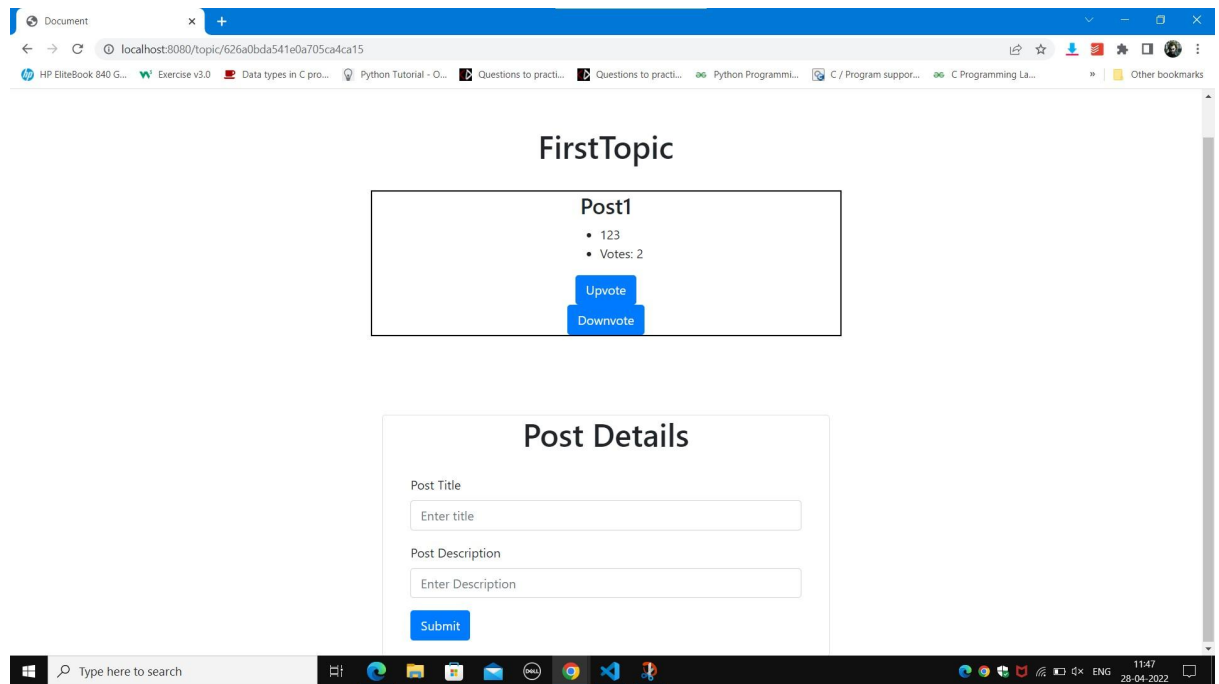
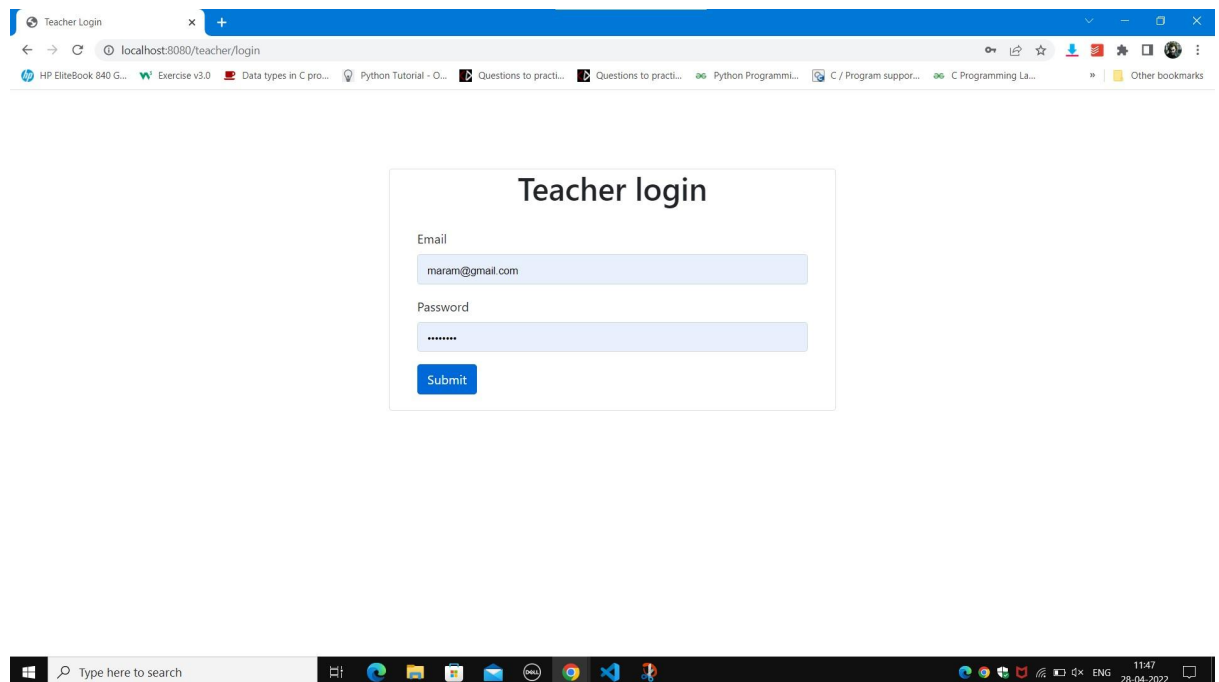### Home page

Student login

Student homepage
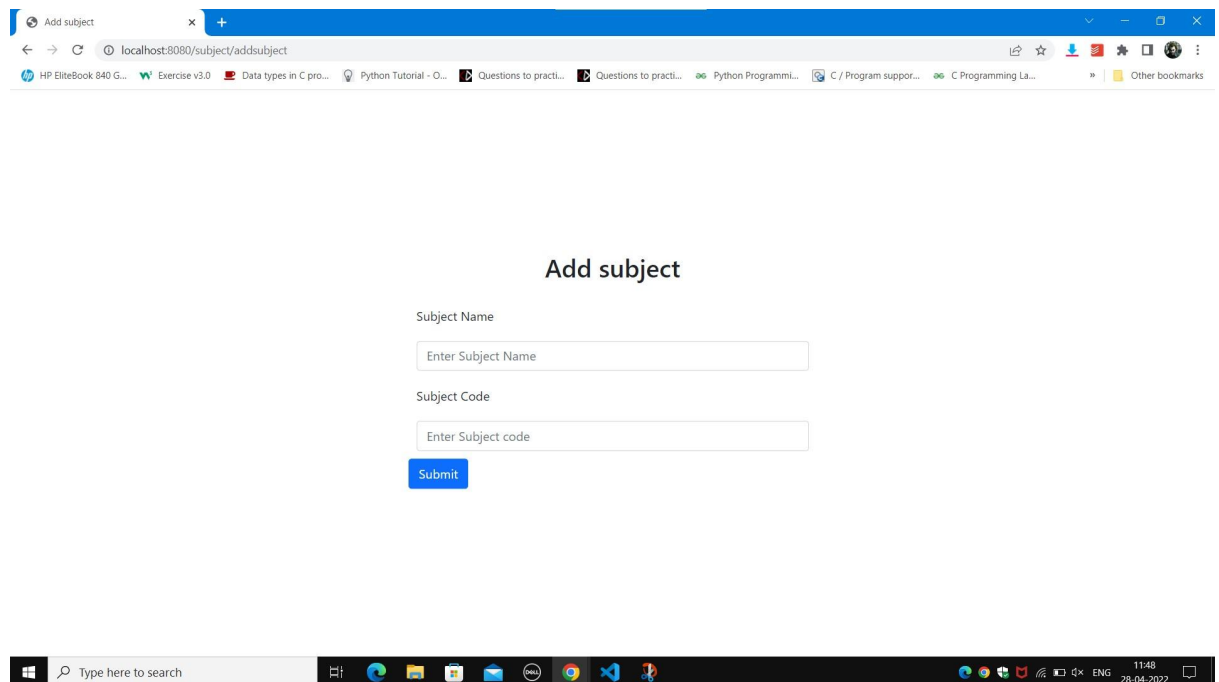
Subject homepage

Add topic

Topic home and add post

Teacher Login

Teacher homepage
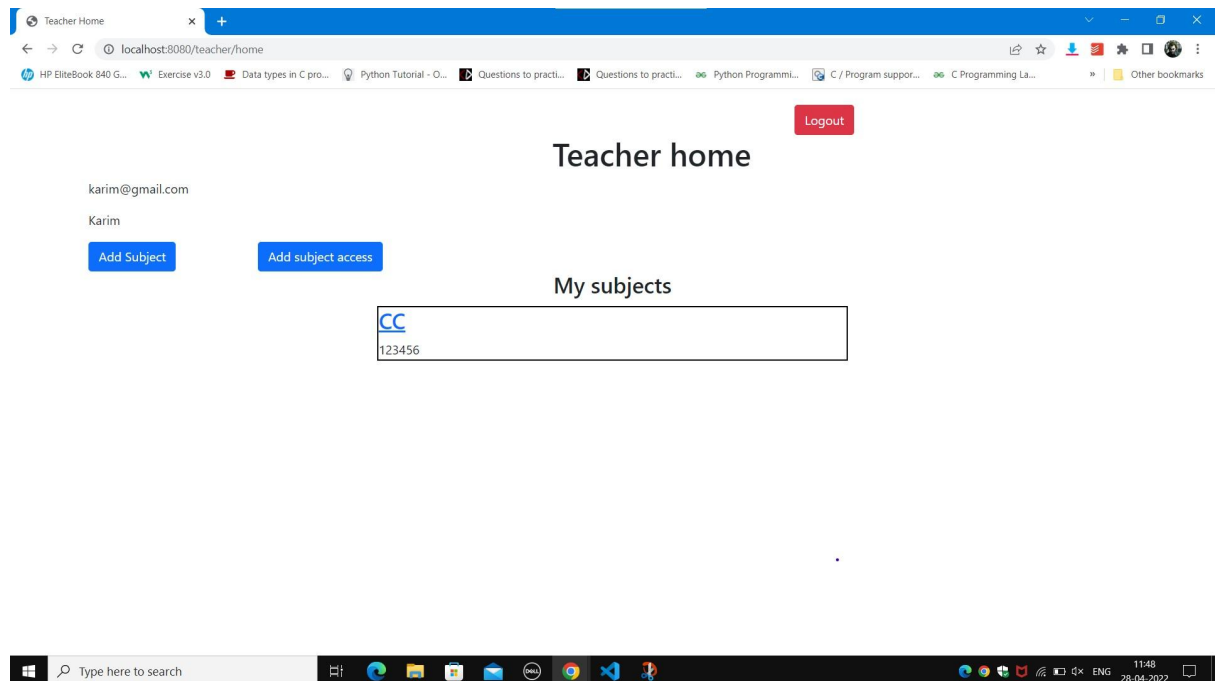
Add subject by teacher

Add subject access

6. Team member contributions

Team members	contribution
Gokul Karthik	Student/Teacher roles and session authentication of the users present.
Gurukiran S	Implementation of subjects and topics within the subjects and UI .
Gaurav J	Voting of posts within the topics and authentication of users.