



UE21CS352B - Object Oriented Analysis & Design using Java

Mini Project Report

Assignment Submission and Evaluation Portal

Submitted by:

Name 1: Akhil M	PES1UG21CS328
Name 2: M Sandeep Kumar Reddy	PES1UG21CS348
Name 3: N Kapil Naidu	PES2UG21CS320
Name 4: M Sai Vikas Reddy	PES1UG21CS344

6th Semester 'F' Section

Prof. Bhargavi Mokashi

January - May 2024

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING
PES UNIVERSITY**

(Established under Karnataka Act No. 16 of 2013)
100ft Ring Road, Bengaluru – 560 085, Karnataka, India

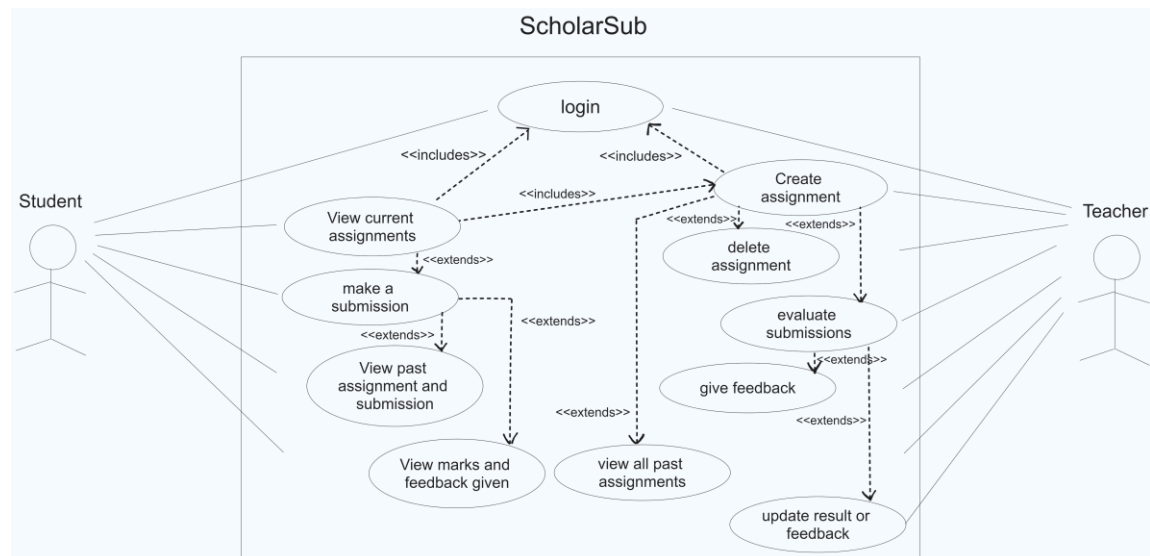
Problem Statement:

This project aims to build a web application in which teachers and students can interact academically. A teacher would create and upload an assignment, for students to view. A student can view the assigned assignments by the teachers, and correspondingly generate submissions to the related assignment(s). A teacher would then be able to view the submissions given by the students for each assignment, and accordingly generate a list of results, for assignments, based on their submissions.

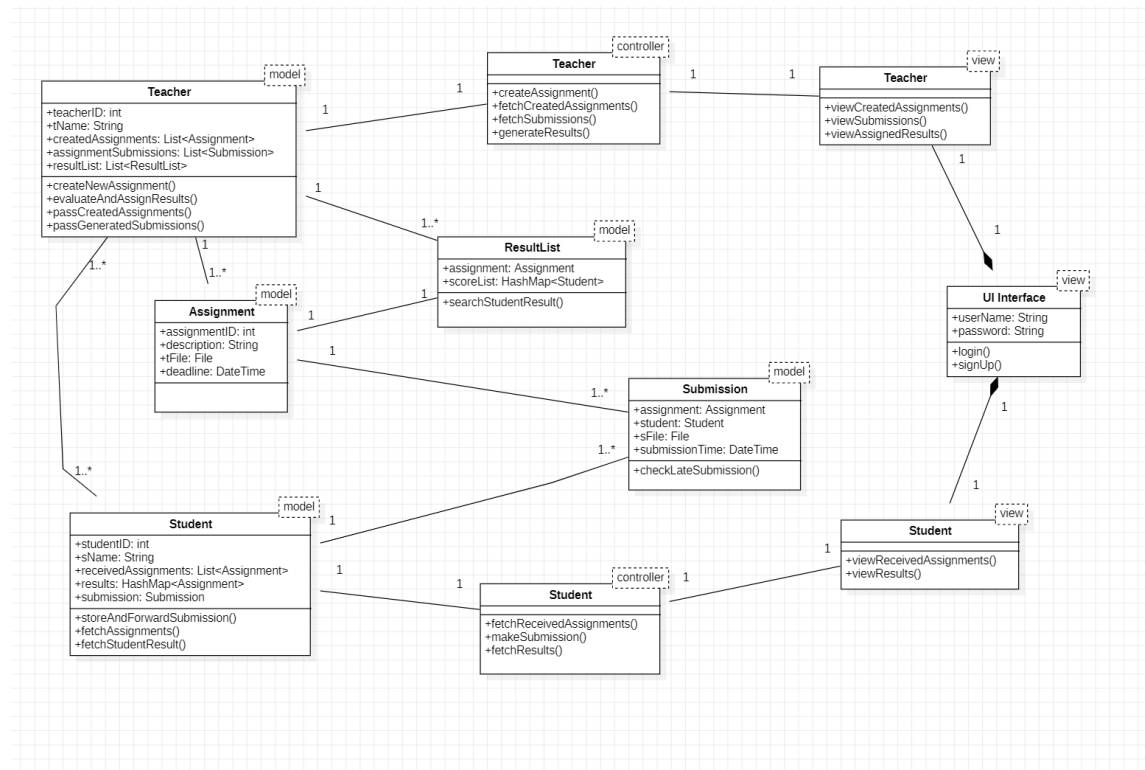
Apart from teachers and students, there would also be an admin, who would be responsible for performing CRUD operations on teacher/student accounts in the application, which would be reflected in the database.

Models:

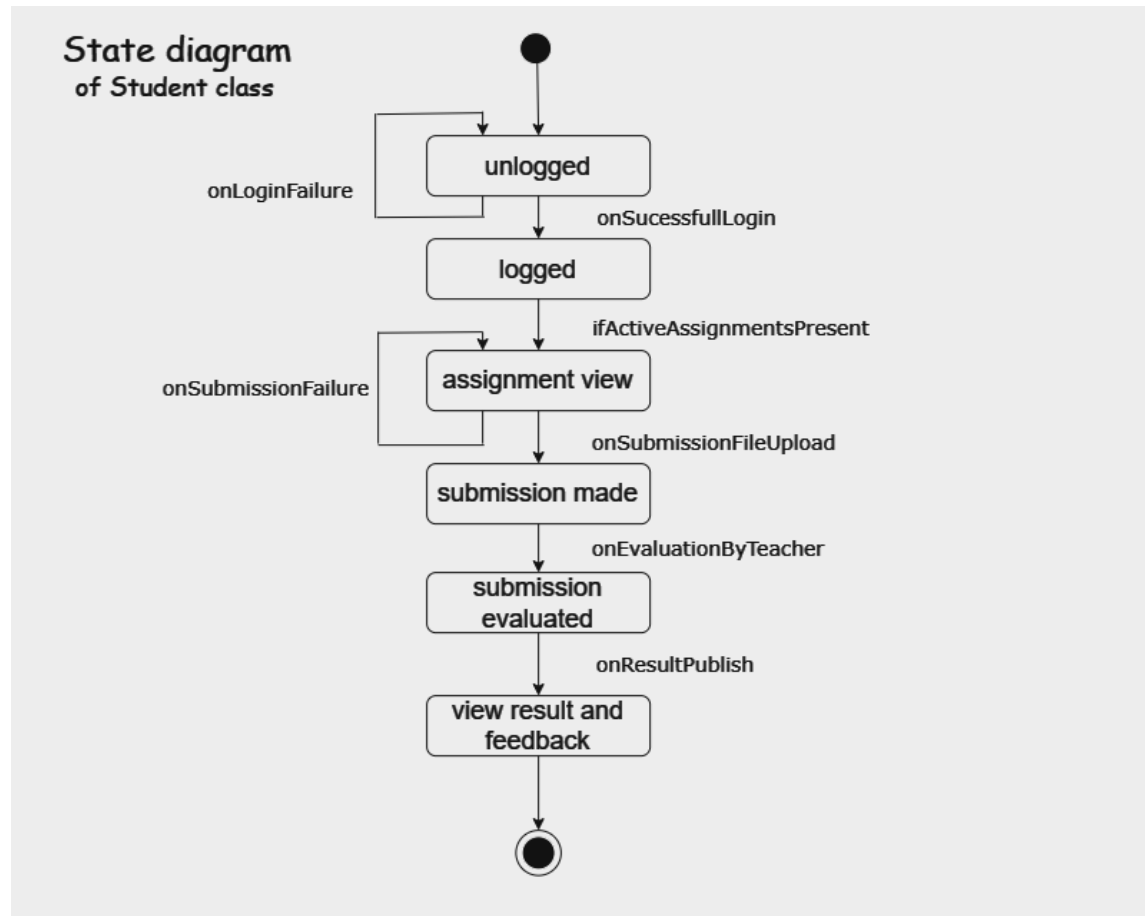
Use Case Diagram:



Class Diagram:

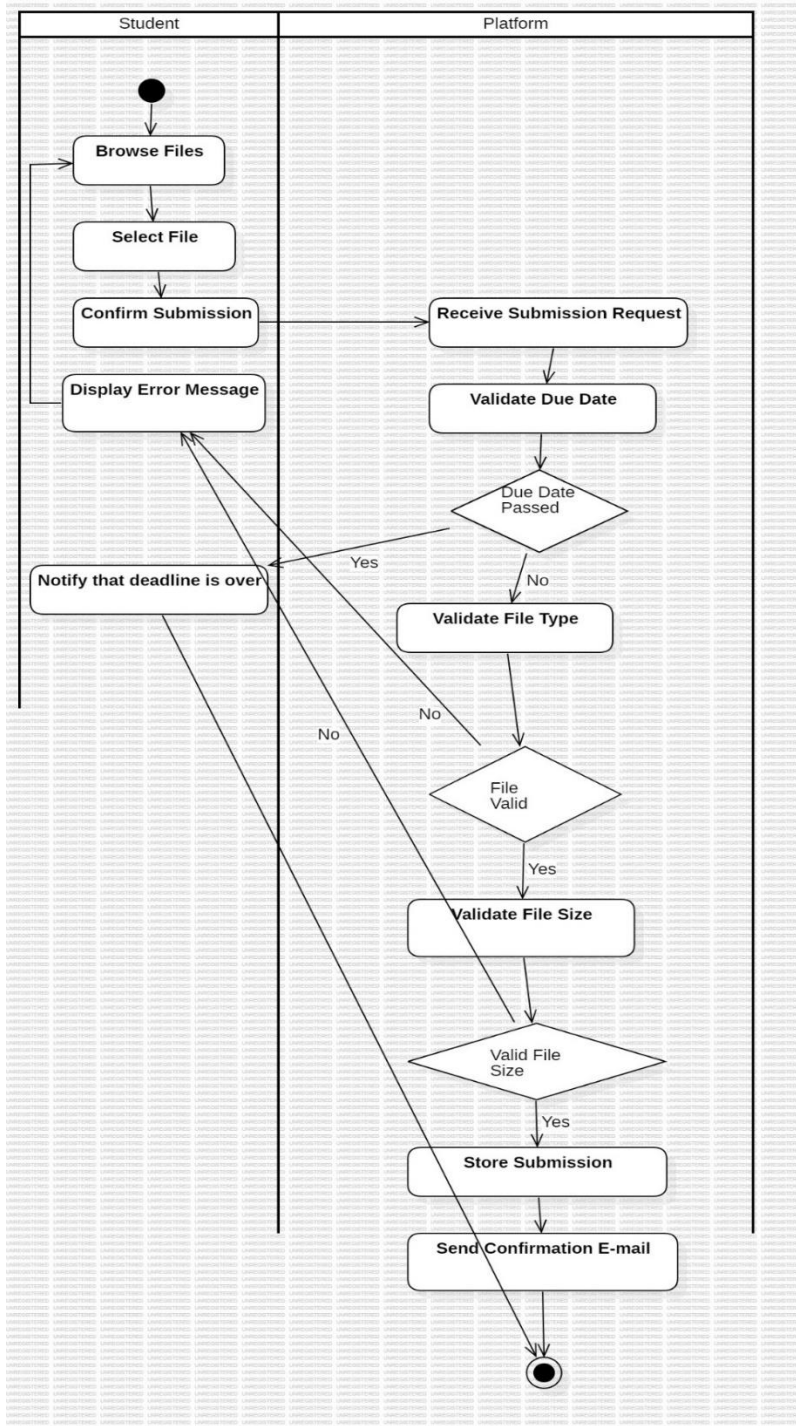


State Diagram:

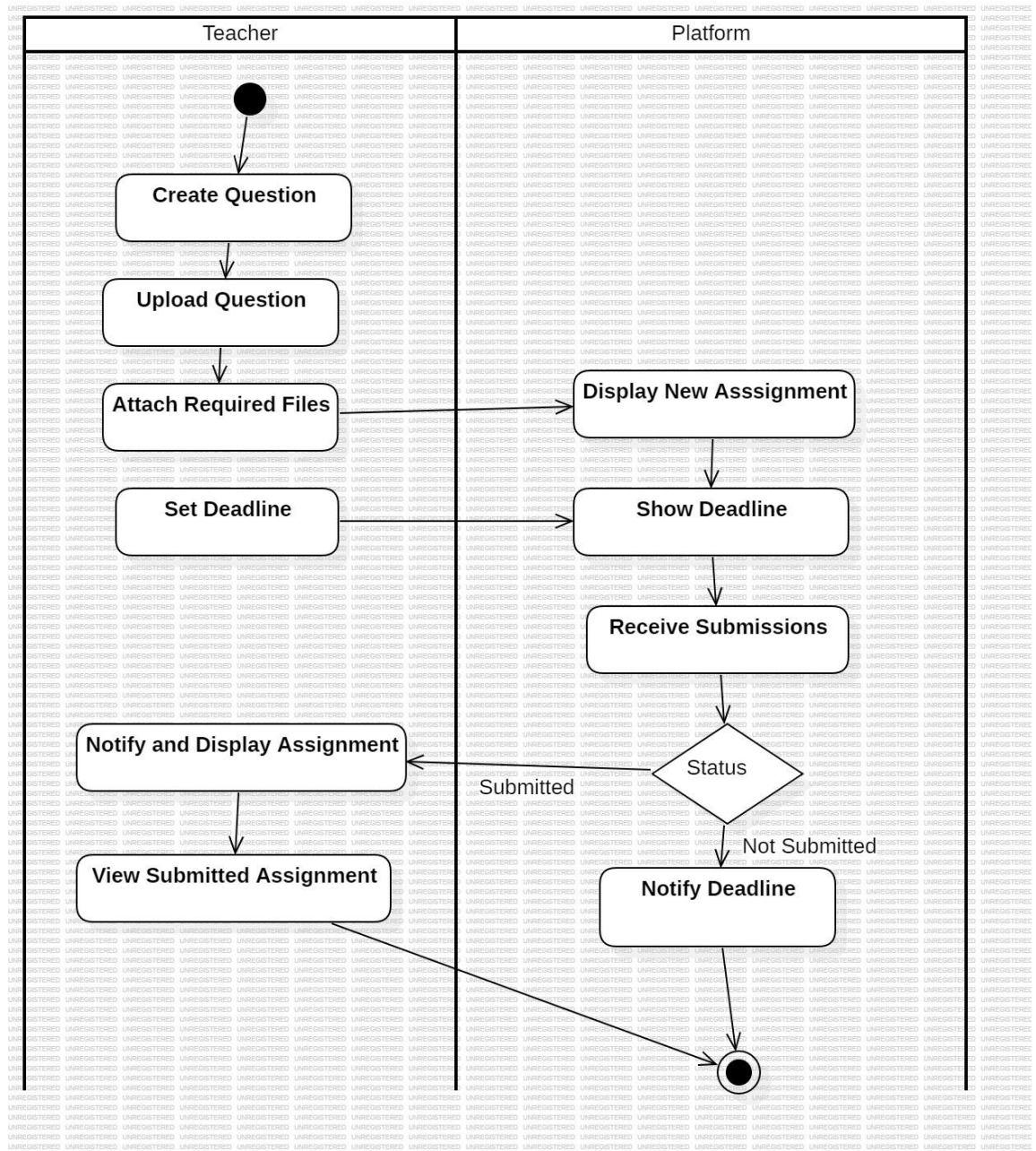


Activity Diagram(s):

1) Student:



2) Teacher:



Design Patterns/Principles:

For this project, we have used four design patterns: Builder + Factory, Singleton, and Iterator.

- 1) For the assignment abstract class, there are three types of subclasses: AssignmentWithOnlyFile, AssignmentWithOnlyDescription, AssignmentWithFileAndDescription. For instantiating each of these different subclasses, we have used a separate class, which would correspond to the Factory design pattern. And also, since in the subclasses, all the attributes are not necessarily being used (with only file/description would not require description/file attributes and with file and description would require both), instantiating the assignment with the necessary attributes only would correspond to the Builder design pattern.
- 2) For the LateSubmissionChecker class, we have used a Singleton design pattern (eager instantiation, in specific). This class checks submission times to their corresponding assignment deadlines to determine late/in-time submission.
- 3) While fetching the results of the students from ResultList class, we are iterating over the HashMap of student id and scores (both integer), to fetch the results of that particular student id. Iterator design pattern can be claimed here, although we haven't specifically made a class for iteration purpose, which is the textbook method.

Coming to the principles, our project follows all five SOLID principles.

- 1) Single Responsibility Principle – Since we have given each class only one specific responsibility (only one reason to change).
- 2) Open Closed Principle – The assignment abstract class has some defined attributes, which cannot be modified/changed, but can be extended by the subclasses to make use of them differently.
- 3) Liskov Substitution Principle – The assignment abstract class variable (parent class) can be replaced by any one of the child class (WithOnlyFile, WithOnlyDescription, WithFileAndDescription), and the properties of the program would not change, and neither do these classes change the behaviour of their parent class.
- 4) Interface Segregation Principle – In our project, we have specifically dedicated one entire folder in the backend for interfaces. Each of those interfaces offer specific functions related to the backend functionality (repositories).
- 5) Dependency Inversion Principle – In the submission class, the variable assignment is of the abstract class. This means that the submission class (high level module) does not depend on any of the concrete assignment classes (low level modules), but instead depends on their abstraction, which is the assignment abstract class variable.

Github Link:

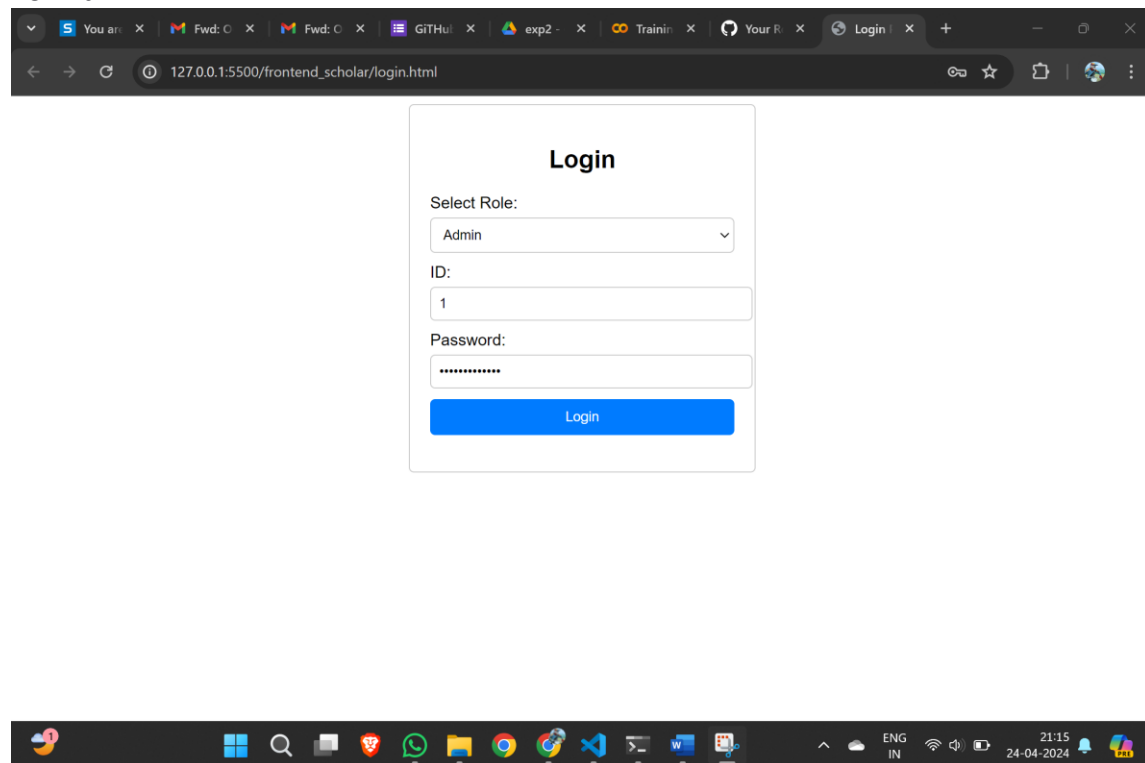
<https://github.com/akhil-manne/scholar-app>

Contributions:

Myself (Akhil) and Sandeep focused on the combined implementation of the Builder and Factory pattern (for Assignment class). Kapil worked on the said iterator pattern (for iterating through student's results), and Vikas worked on the Singleton pattern (for LateSubmissionChecker class).

Screenshots:

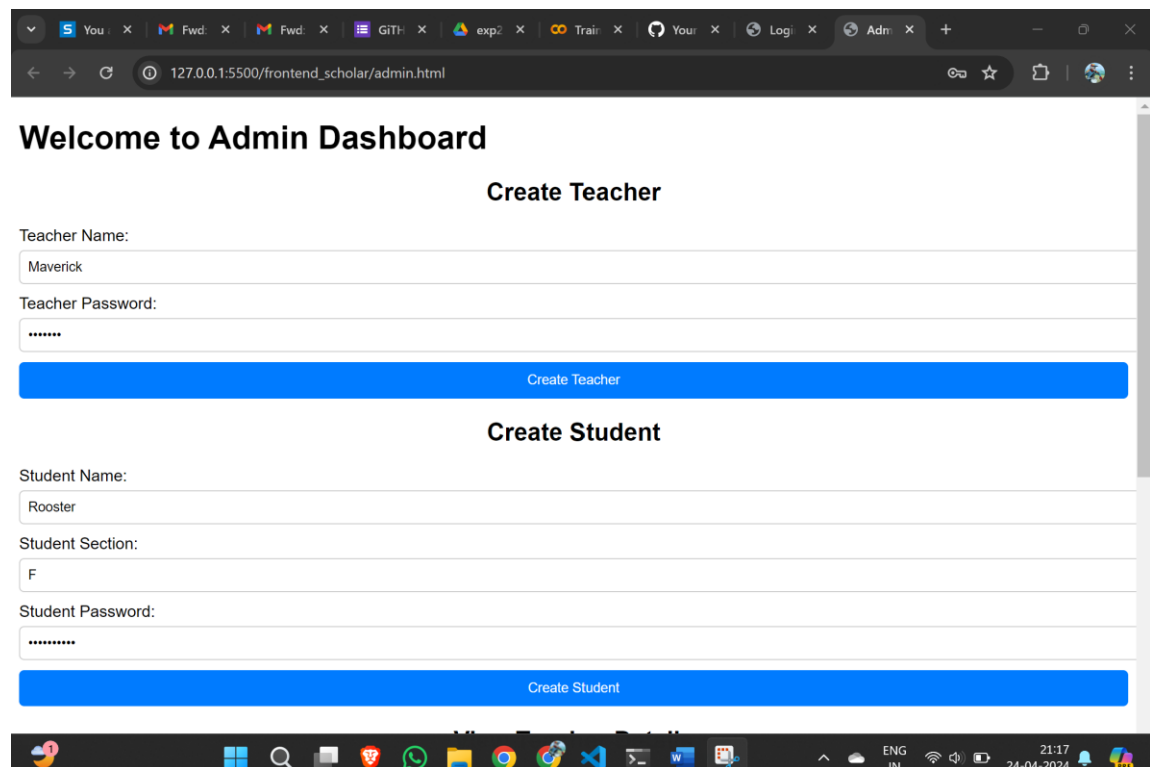
For Admin:

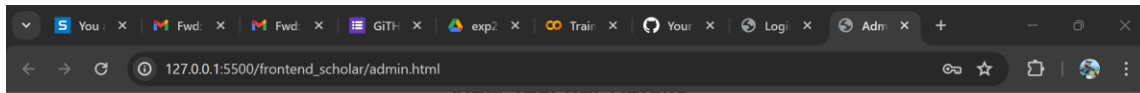


The screenshot shows a web browser window with the address bar displaying `127.0.0.1:5500/frontend_scholar/login.html`. The browser has several tabs open, including 'You are', 'Fwd: C', 'Fwd: C', 'GitHub', 'exp2', 'Trainin', 'Your R', and 'Login'. The main content area displays a 'Login' form with the following fields:

- Select Role:** A dropdown menu with 'Admin' selected.
- ID:** A text input field containing the value '1'.
- Password:** A text input field with masked characters (dots).
- Login:** A blue button labeled 'Login'.

The Windows taskbar is visible at the bottom of the screen, showing various application icons and the system clock indicating 21:15 on 24-04-2024.





View Teacher Details

View Teachers

ID: 5, Name: testteacher, Password: secret

ID: 7, Name: bot, Password: professor

ID: 11, Name: Maverick, Password: Captain

View Student Details

View Students

ID: 6, Name: Test, Section: A, Password: hidden

ID: 9, Name: Rooster, Section: F, Password: Lieutenant

Delete Teacher

Teacher ID:

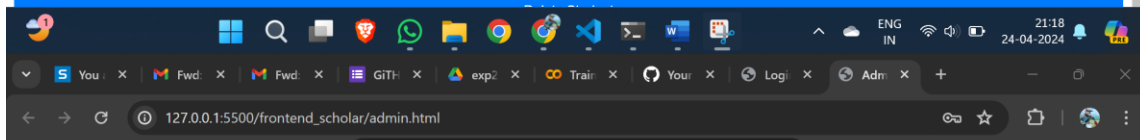
Enter teacher ID

Delete Teacher

Delete Student

Student ID:

Enter student ID



127.0.0.1:5500 says

Teacher deleted successfully

OK

ID: 5, Name: testteacher, Password: secret

ID: 7, Name: bot, Password: professor

ID: 11, Name: Maverick, Password: Captain

View Student Details

View Students

ID: 6, Name: Test, Section: A, Password: hidden

ID: 9, Name: Rooster, Section: F, Password: Lieutenant

Delete Teacher

Teacher ID:

5

Delete Teacher

Delete Student

Student ID:

Enter student ID



For Teacher:

Welcome, Teacher!

ID: 11

Name: Maverick

Create Assignment

Description:

Test Assignment 1.

File Link:

Submission Link:

don't need

Section:

F

Deadline:

22-04-2024

Create Assignment

Assignments

Create Assignment

Assignments

View Assignments

ID: 46

Description: null

Deadline: 2024-04-22

File Link: [dummylink.com](#)

Submission Link: [notrequired](#)

Section: B

ID: 47

Description: TestAssignmentDescription

Deadline: 2024-04-22

File Link: [dummylink.com](#)

Submission Link: [notrequired](#)

Section: C

```

Description: Test Assignment 1.
File Link: null
Section: F
Deadline: 2024-04-22
Submission Link: don't need
Form Data:
description=Test+Assignment+1.&section=F&deadline=2024-04-22&submissionLink=don%27t+need
Response status: 200
Assignment created successfully:
{type: 'AssignmentWithOnlyDescription', assignmentID: 52, description: 'Test Assignment 1.', fileLink: null, submissionLink: 'don't need', ...}
assignmentID: 52
deadline: "2024-04-22"
description: "Test Assignment 1."
fileLink: null
section: "F"
submissionLink: "don't need"
type: "AssignmentWithOnlyDescription"
[[Prototype]]: Object

```

21:20

24-04-2024

```

Description: Test Assignment 1.
File Link: null
Section: F
Deadline: 2024-04-22
Submission Link: don't need
Form Data:
description=Test+Assignment+1.&section=F&deadline=2024-04-22&submissionLink=don%27t+need
Response status: 200
Assignment created successfully:
{type: 'AssignmentWithOnlyDescription', assignmentID: 52, description: 'Test Assignment 1.', fileLink: null, submissionLink: 'don't need', ...}
assignmentID: 52
deadline: "2024-04-22"
description: "Test Assignment 1."
fileLink: null
section: "F"
submissionLink: "don't need"
type: "AssignmentWithOnlyDescription"
[[Prototype]]: Object

```

View Submissions

Assignment ID:

View Submissions

Submission ID: 2

Assignment ID: 46

Student ID: 6

Submission File Link: [dummy24.com](#)

Submission Status: In-Time

Submission Time: 2024-04-16

Create or Update Result List

Assignment ID:

Student ID:

Score:

Create/Update Result List

View Results

Console

top | Filter | Default levels | No Issues

Description: Test Assignment 1. [teacherscript.js:31](#)

File Link: null [teacherscript.js:32](#)

Section: F [teacherscript.js:33](#)

Deadline: 2024-04-22 [teacherscript.js:34](#)

Submission Link: don't need [teacherscript.js:35](#)

Form Data: [teacherscript.js:57](#)
description=Test+Assignment+1.§ion=F&deadline=2024-04-22&submissionLink=don%27t+need

Response status: 200 [teacherscript.js:68](#)

Assignment created successfully: [teacherscript.js:83](#)
{type: 'AssignmentWithOnlyDescription', assignmentID: 52, description:
▼ 'Test Assignment 1.', fileLink: null, submissionLink: "don't need", ...}
assignmentID: 52
deadline: "2024-04-22"
description: "Test Assignment 1."
fileLink: null
section: "F"
submissionLink: "don't need"
type: "AssignmentWithOnlyDescription"
▶ [[Prototype]]: Object

46 [teacherscript.js:131](#)

Result list created or updated successfully. [teacherscript.js:187](#)

Submission Time: 2024-04-10

Create or Update Result List

Assignment ID:

Student ID:

Score:

Create/Update Result List

View Results

View Result List

Assignment ID: 46

- Student ID: 2, Score: 10
- Student ID: 6, Score: 9

Assignment ID: 48

- Student ID: 6, Score: 6

Elements

Console

Sources

Network

Filter

Default levels

No Issues

Description: Test Assignment 1. teacherscript.js:31

File Link: null teacherscript.js:32

Section: F teacherscript.js:33

Deadline: 2024-04-22 teacherscript.js:34

Submission Link: don't need teacherscript.js:35

Form Data: teacherscript.js:57

description=Test+Assignment+1.§ion=F&deadline=2024-04-22&submissionLink=don%27t+need

Response status: 200 teacherscript.js:68

Assignment created successfully: teacherscript.js:83

{type: 'AssignmentWithOnlyDescription', assignmentID: 52, description: 'Test Assignment 1.', fileLink: null, submissionLink: "don't need", ...}

assignmentID: 52

deadline: "2024-04-22"

description: "Test Assignment 1."

fileLink: null

section: "F"

submissionLink: "don't need"

type: "AssignmentWithOnlyDescription"

[[Prototype]]: Object

46 teacherscript.js:131

Result list created or updated successfully. teacherscript.js:187

For Student:

Welcome to Student Dashboard

Student Information

Student ID: 6

Student Name: Test

View Assignments

View Assignments

ID: 46

Description: null

Deadline: 2024-04-22

File Link: [dummylink.com](#)

Submission Link: [notrequired](#)

Section: B

ID: 47

Description: TestAssignmentDescription

Deadline: 2024-04-22

File Link: N/A

Submission Link: [don't need](#)

Section: F

ID: 52

Description: Test Assignment 1.

Deadline: 2024-04-22

File Link: N/A

Submission Link: [don't need](#)

Section: F

Create Submission

Assignment ID:

52

Submission File Link:

testlink.com

Submit

View Your Results

127.0.0.1:5500/frontend_scholar/student.html

Elements Console Sources Network

Submission created successfully: studentscript.js:76

{submissionID: 7, assignment: {type: 'AssignmentWithOnlyDescription', assignmentID: 52, dateOfSubmission: '2024-04-24', ...}, student: {studentID: 6, studentName: 'Test', studentSection: 'A', stuSubmissionFileLink: 'testlink.com', submissionID: 7, submissionStatus: 'Late'}, [[Prototype]]: Object

21:29 24-04-2024

21:31 24-04-2024

127.0.0.1:5500/frontend_scholar/student.html

File Link: N/A

Submission Link: [don't need](#)

Section: F

Create Submission

Assignment ID:

Submission File Link:

Submit

View Your Results

Assignment ID: 46

Score: 9

Assignment ID: 48

Score: 6

Submission created successfully: [studentscript.js:76](#)

submissionID: 7, assignment: (-), student: (-), submissionFileLink: testlink.com, dateOfSubmission: 2024-04-24, -)

assignment: {type: AssignmentWithOnlyDescription, assignmentID: 52, dateOfSubmission: 2024-04-24}

student: {studentID: 6, studentName: Test, studentSection: A, stu submissionFileLink: testlink.com}

submissionID: 7

submissionStatus: Late

[[Prototype]]: Object

ENG IN

21:31

24-04-2024