Java Auto Grading Document

1. **Project Description**

1. Login system.

2. Professor and Student can log in.

3. Professor can create assignments, test cases (expected output), and due dates. For example, hello world is created as an assignment problem. we can have only one test case with the expected output.

For example. shape.java program is there. and we are expecting 4 shapes (circle, triangle, square, and rectangle). So, the number of test cases will be 4 with the expected output.

4. Students can submit the assignment

5. Submitted assignment will start compiling and if the observed output is compared with expected output via regular expression are equal then give marks.

6. Marks will be given on how many test cases have passed.

7. We save the marks, student, and lab\_id when the professor releases the marks. The data of students and their marks are sent to another web page in JSON format via RESTful API.

9. Basically auto-grading and university portal is different think. results need to be declared on the university portal. So, data is sent in a JSON format from the auto grading site to the university portal via REST API.

10. Frontend (jsp/ thyleaf) + middleware (spring boot) + backend (MySQL) all will be done by you. Frontend should not be fancy. Please keep it simple. Only desktop site support needs to be there.

Timeline

Description automatically generated

Diagram

Description automatically generated

I will need the below things: -

1. Personal contact number for two people.

2. Daily code integration and execution on my laptop.

3. And report progress to me daily (EOD).

3. Deadline 24th July 2022.

1. **Database Tables**

Graphical user interface, application

Description automatically generated

* + - 1. First insert the dummy data into tables.
      2. No need to touch admin table. Except that add Created\_By (Save Admin’s or professor’s or student’s primary key), Created\_Date, Deleted\_By (Save Admin’s or professor’s or student’s primary key) and Deleted\_Date columns in every table (all above).
      3. Maintain schema and data script backup.
      4. Change username-> email Id in frontend and database too.
      5. Admin can create and delete data from course, module, role, professor and student’s table.
      6. Professor can create and delete data from Assignmment and Assignmnet\_Test\_cases table.
      7. Student can create data from Student\_Assignment table. And Admin can delete from

1. **Basic login syste**Text

   Description automatically generated with medium confidence
   * + 1. Use the spring login module using role-based login.

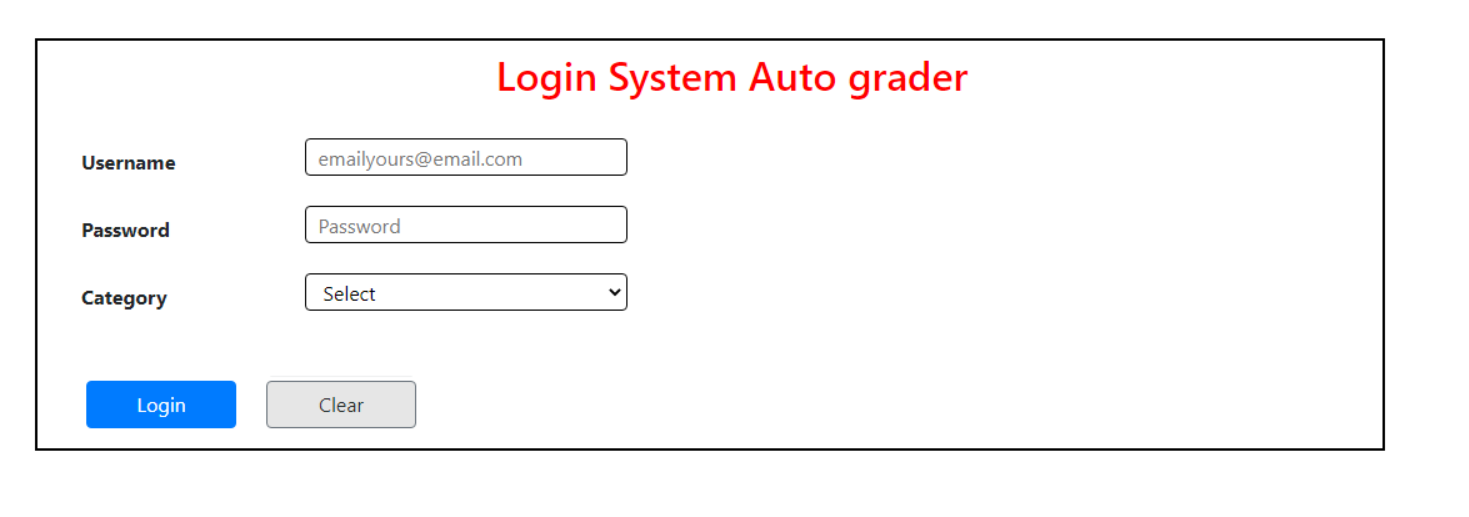
**Admin Login ->** Welcome Page (Menus- Create Role, Review Role, Create Course, Review Course, Create Module, Review Module)

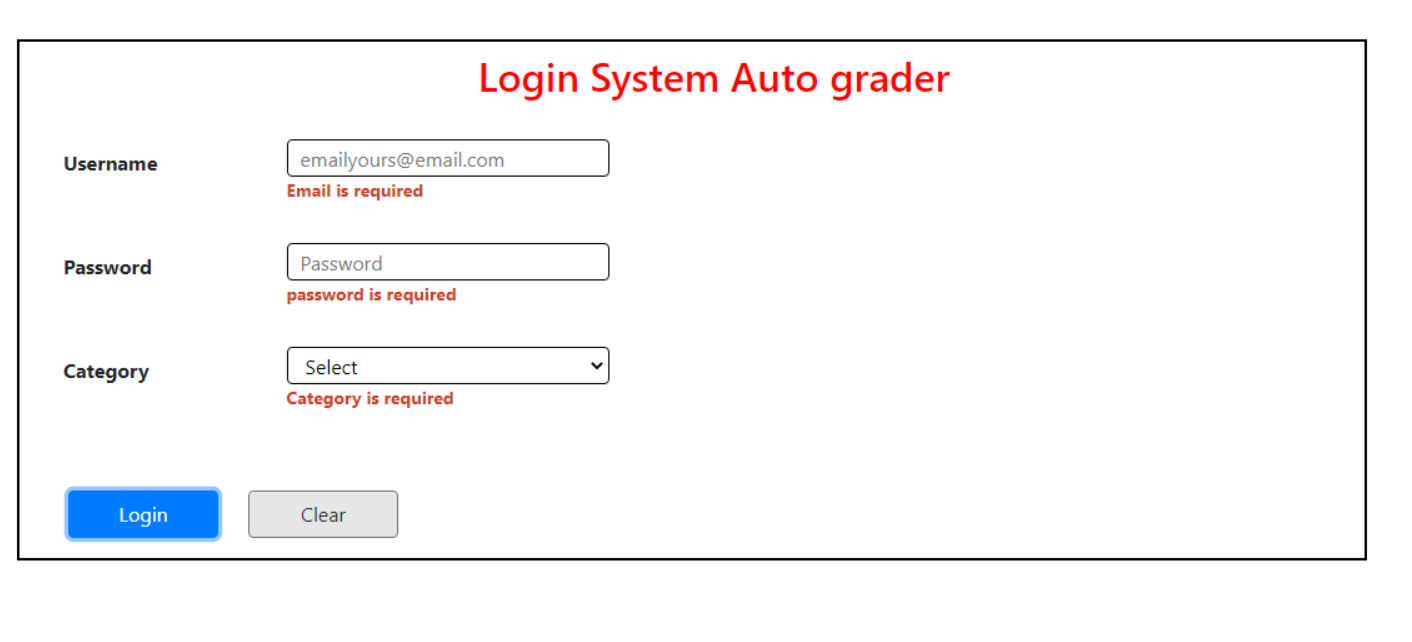
**Professor Login ->** WelcomePage (Menus- Assignment Creation, Review student submission)

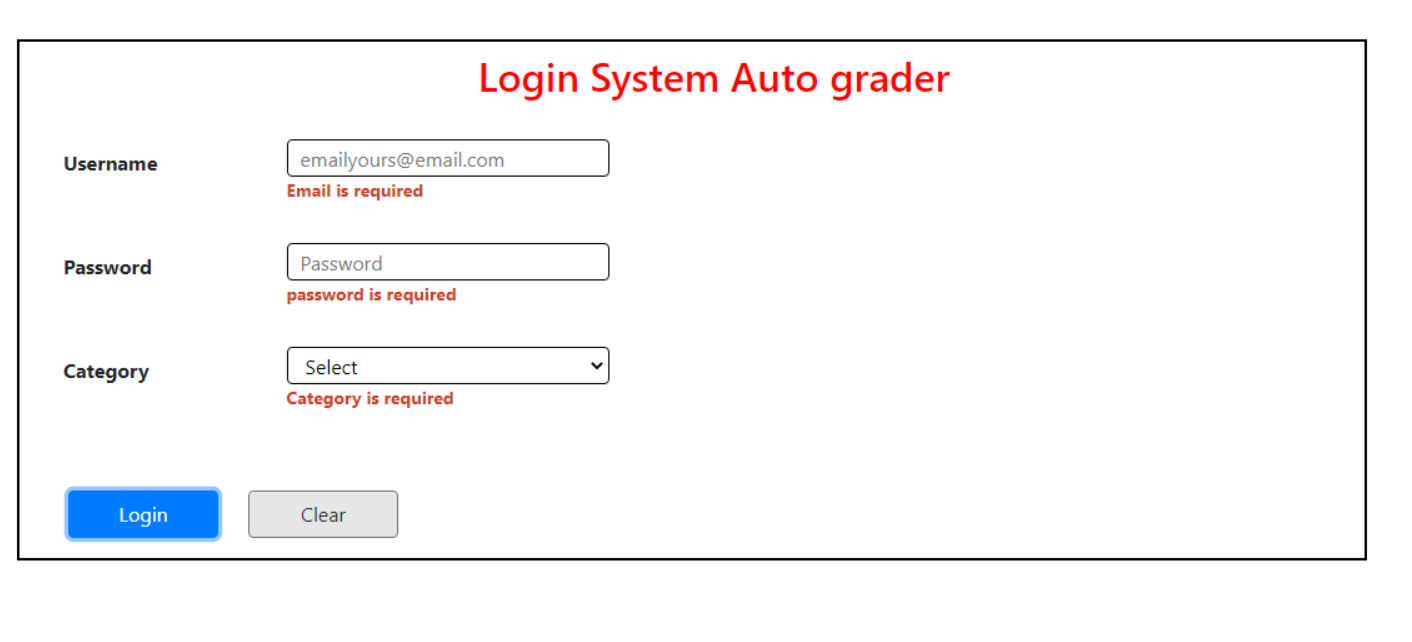
**Student Login ->** Welcome Page (Menus- Assignment Submission, Review Submission)

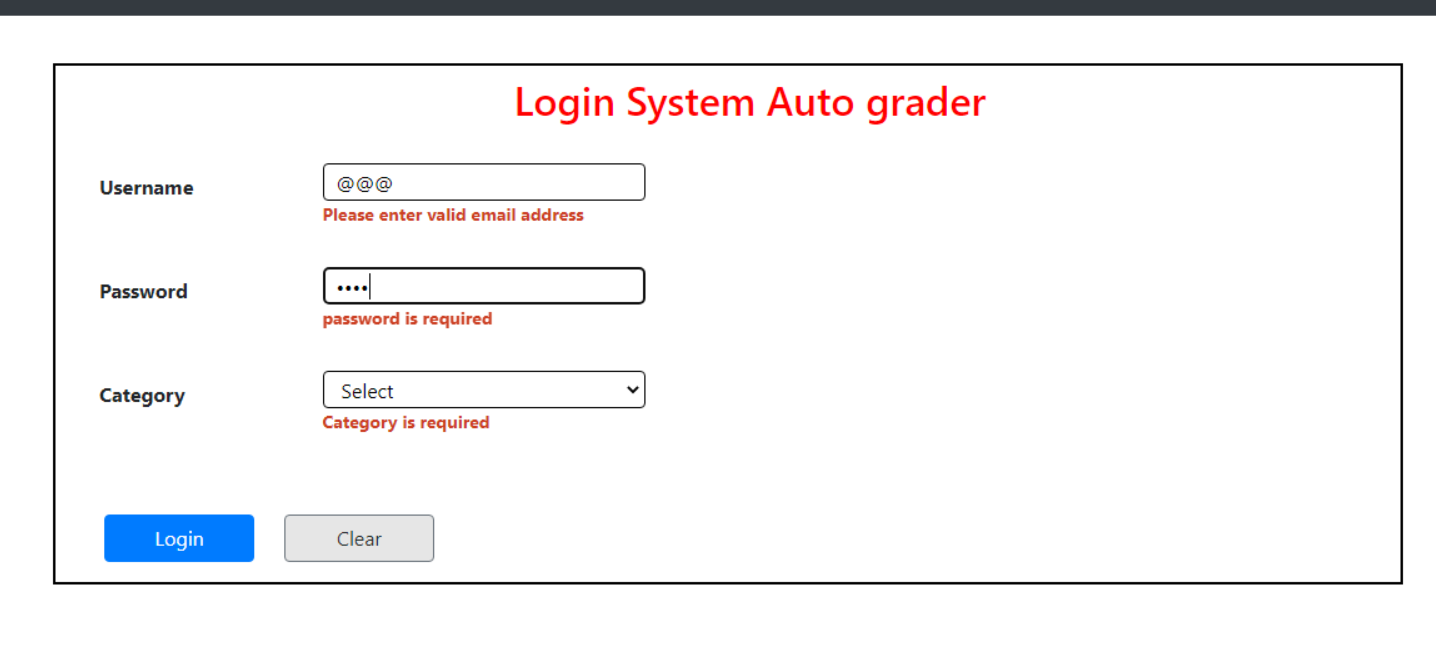
* + - 1. Will have username as email id and validate it. Select the category. And press login button. After login will check the respective table as student, professor or admin based on category selection. And verify the email id and password.
      2. After successful login send the user to welcome page with menus mentioned in 1st point.

FileName:- loginPage.html







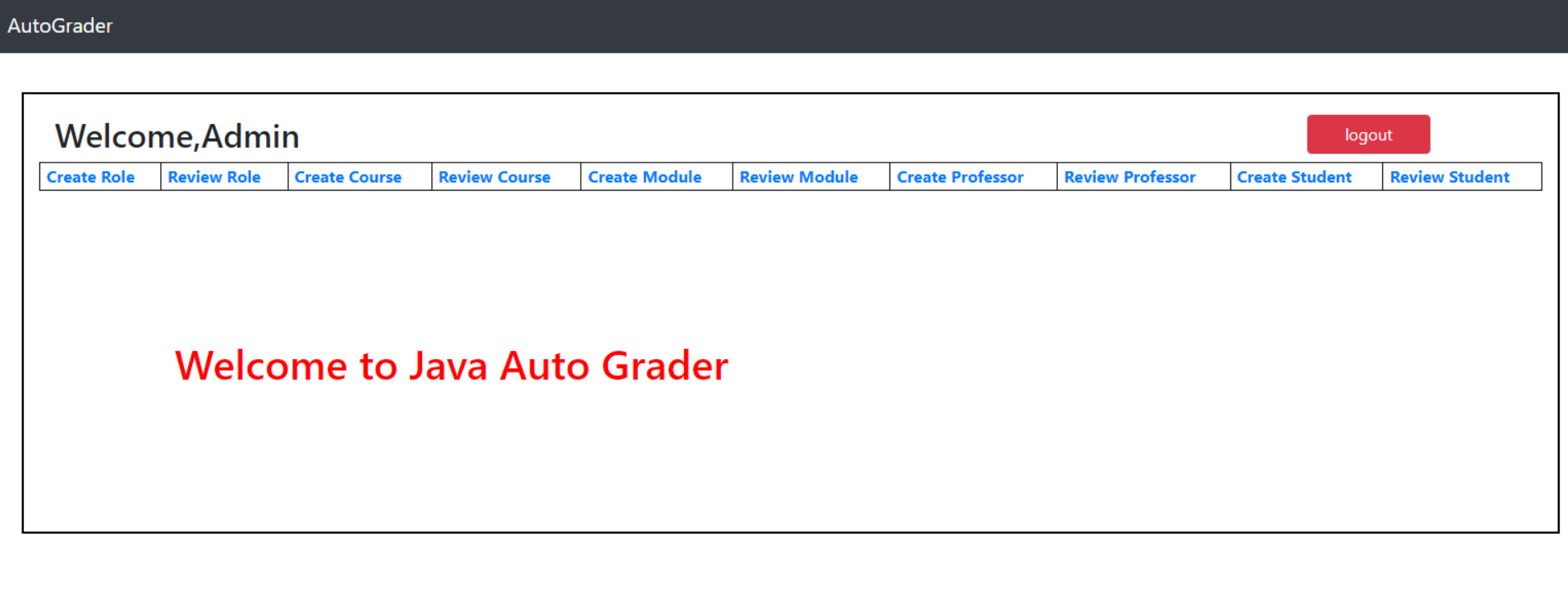


1. **After Admin get logged in, it will be redirected to below welcome Page. And able to see below menus. Professor and student won’t be able to see this menu after they get logged in.**

**File:- welcomeAdmin**

Graphical user interface, application

Description automatically generated



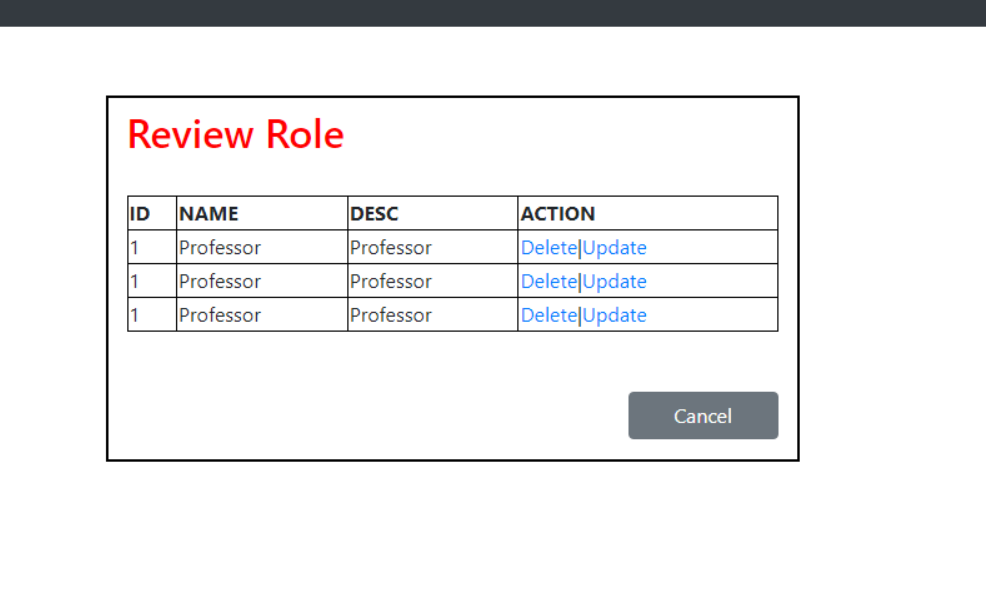
1. **Role**

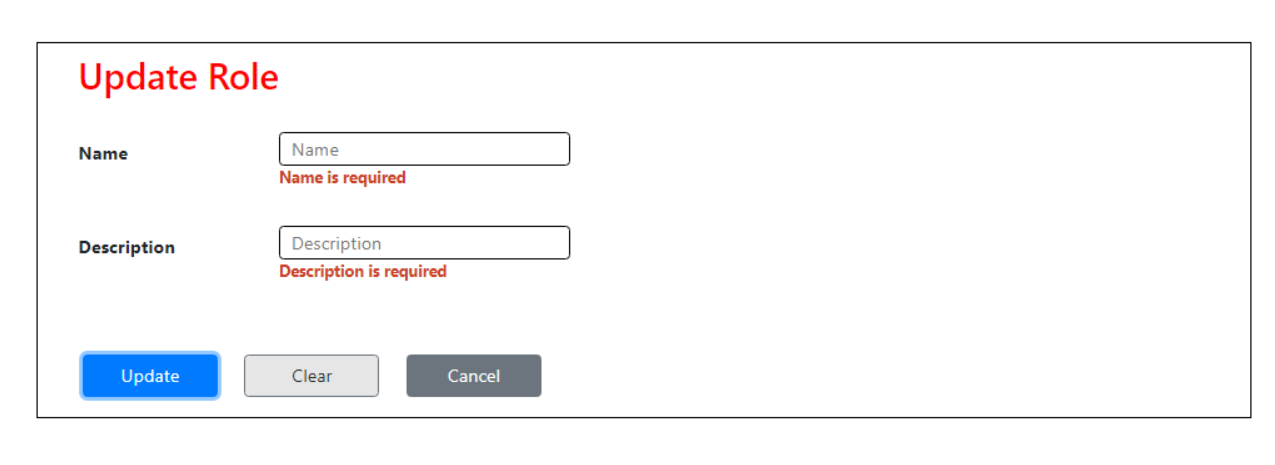
Graphical user interface

Description automatically generated**\***

**When on “Role Creation” data is entered, will be saved to “Role” table.**

**Review Role :-reviewRole.html**

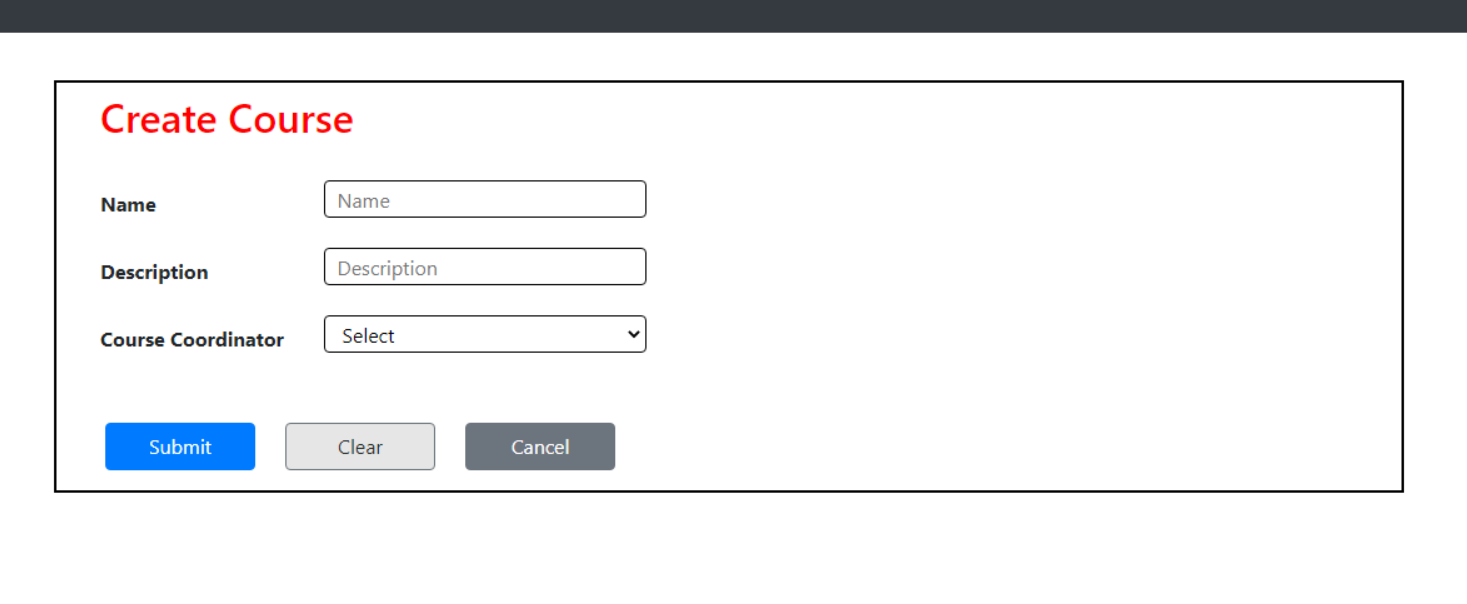
****

****

1. **Create Course file:-createCourse**

Graphical user interface, text

Description automatically generated

****

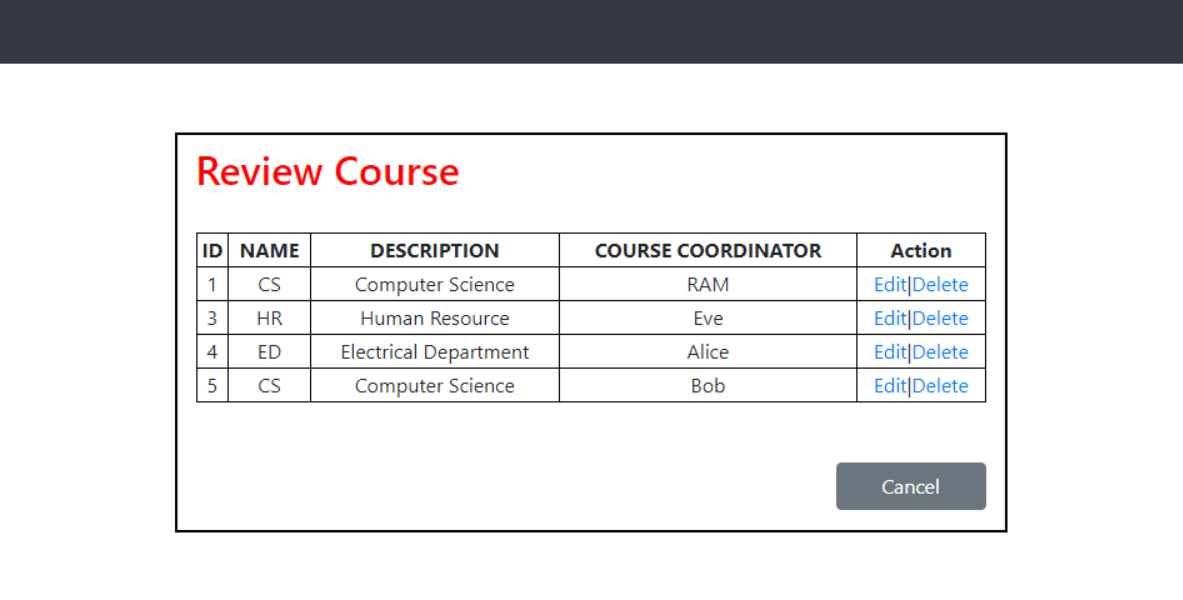
1. **Review Course- File:-reviewCourse.html**

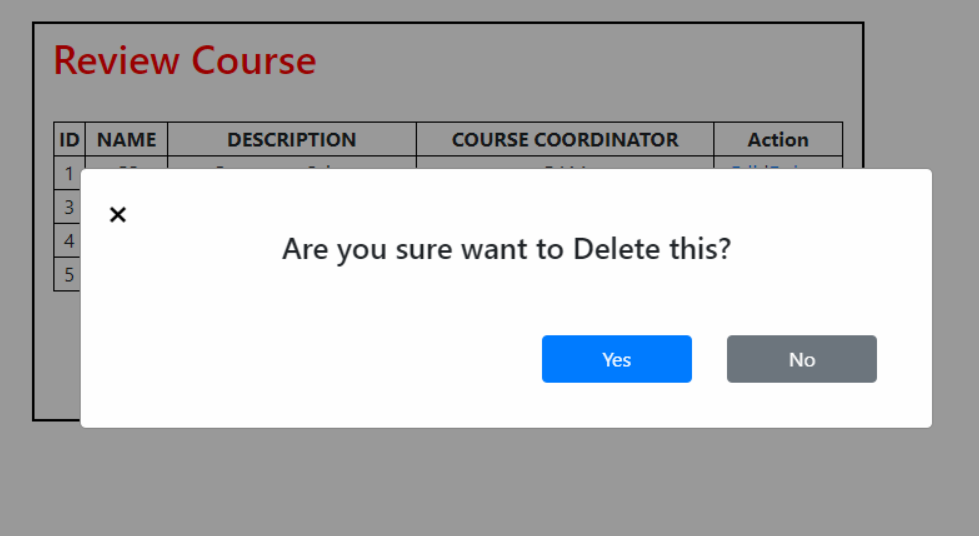
Table

Description automatically generated

**If we click on “Edit” button, it should redirect to below “Edit Course” screen. Where we can update the course related information.**

**And if click “Delete” button, “Are you sure want to delete this?” pop up should be shown.**

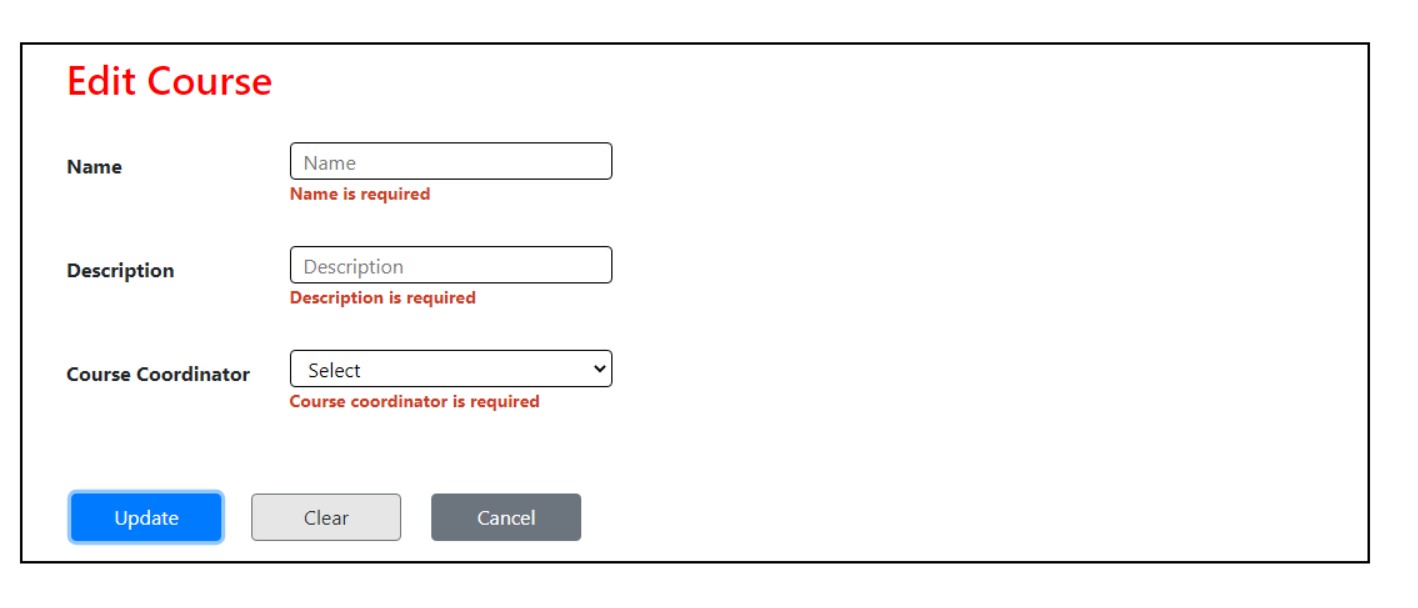
****

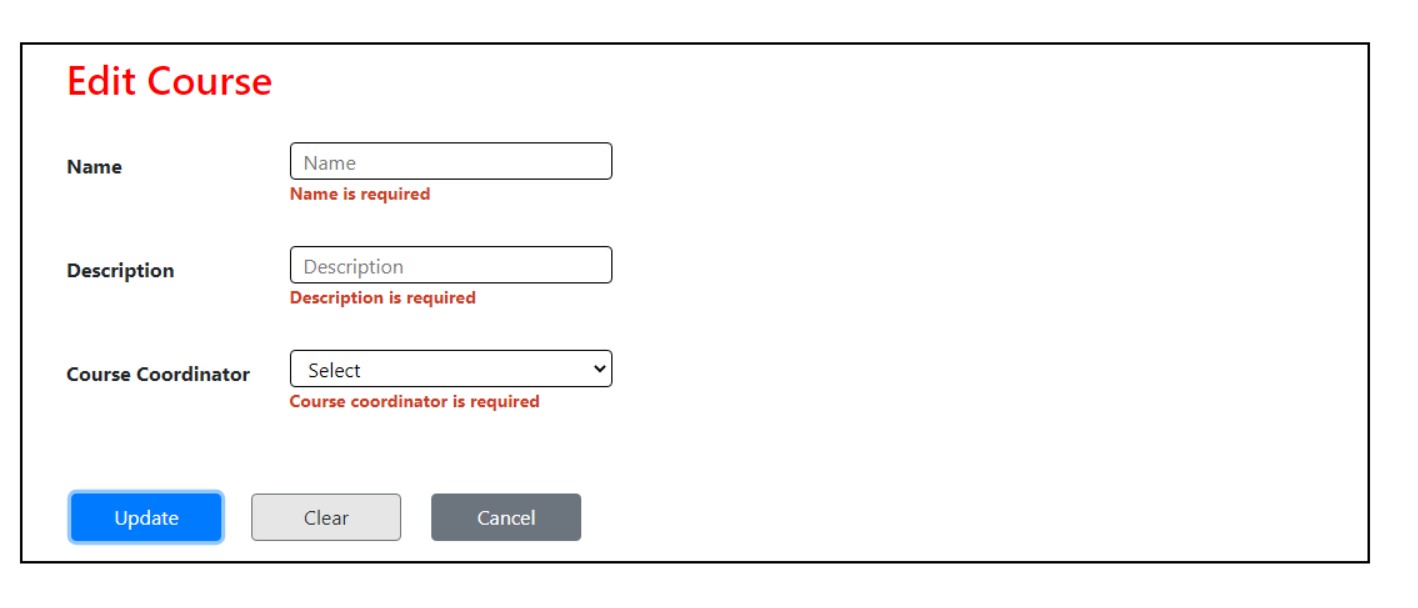
****

1. **Edit Course file:-editCourse**

Graphical user interface

Description automatically generated

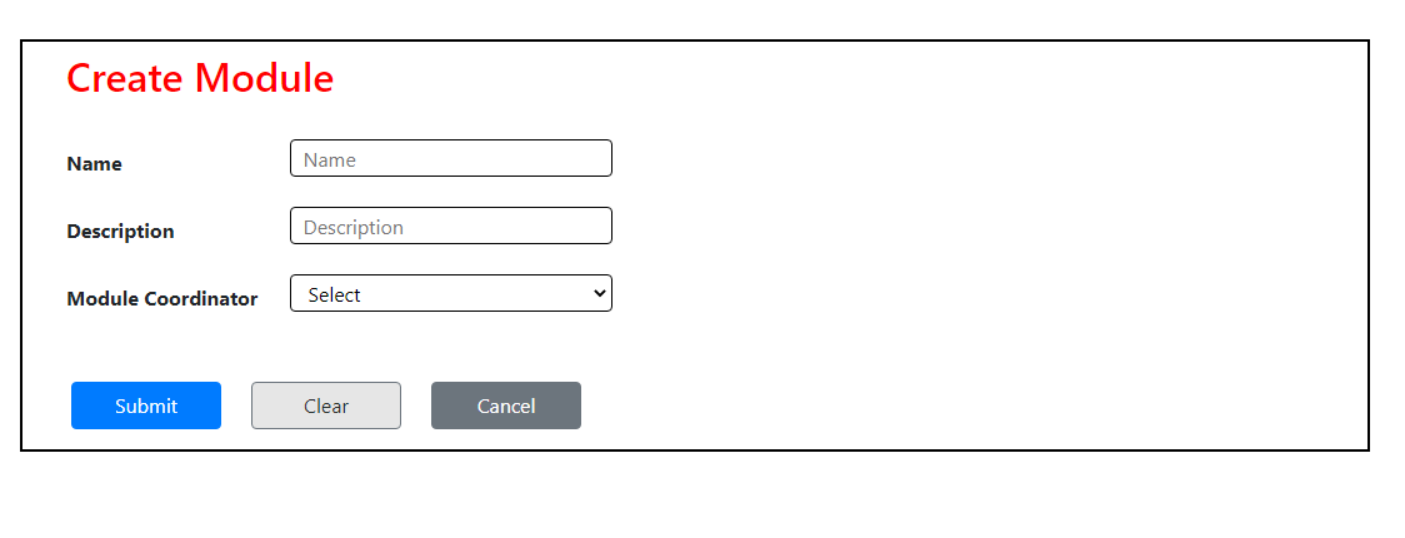
****

****

1. **Create Module , createModule**

Table

Description automatically generated with low confidence

****

1. **Review Module**

Table

Description automatically generated

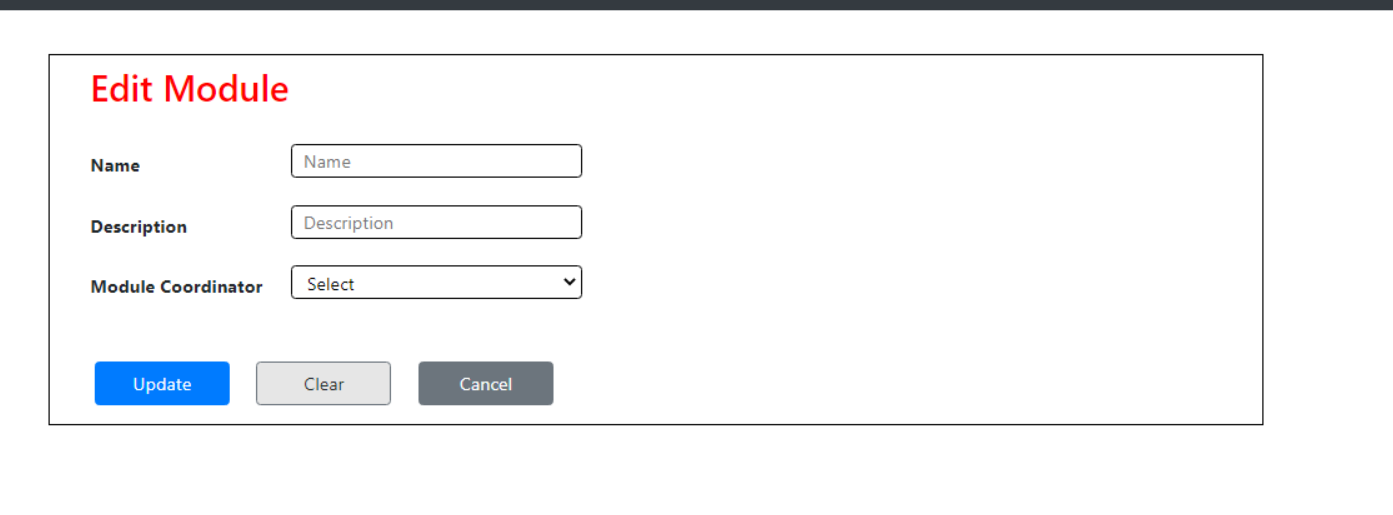
1. **Edit Module**

Table

Description automatically generated with medium confidence

1. Add page for Review Courses and Review Module menus. Updating and deleting functionality will be same as review role.

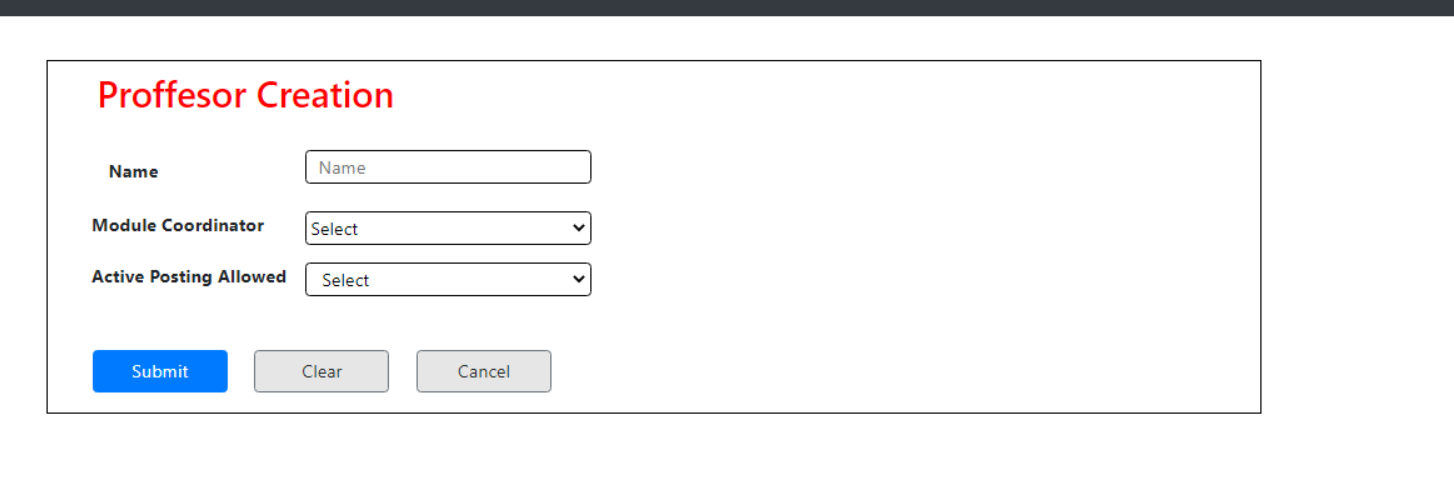
2. After clicking on “Cancel” button redirect page to welcome page.



1. **Professor insertion page**

Graphical user interface, application

Description automatically generated

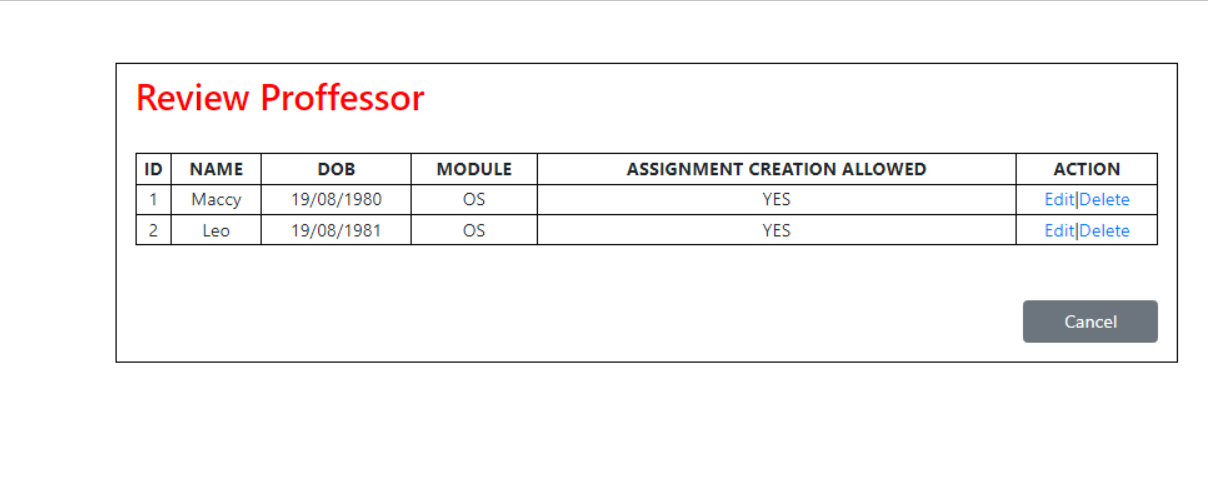


1. **Review Professor**

Table

Description automatically generated

reviewProffesor

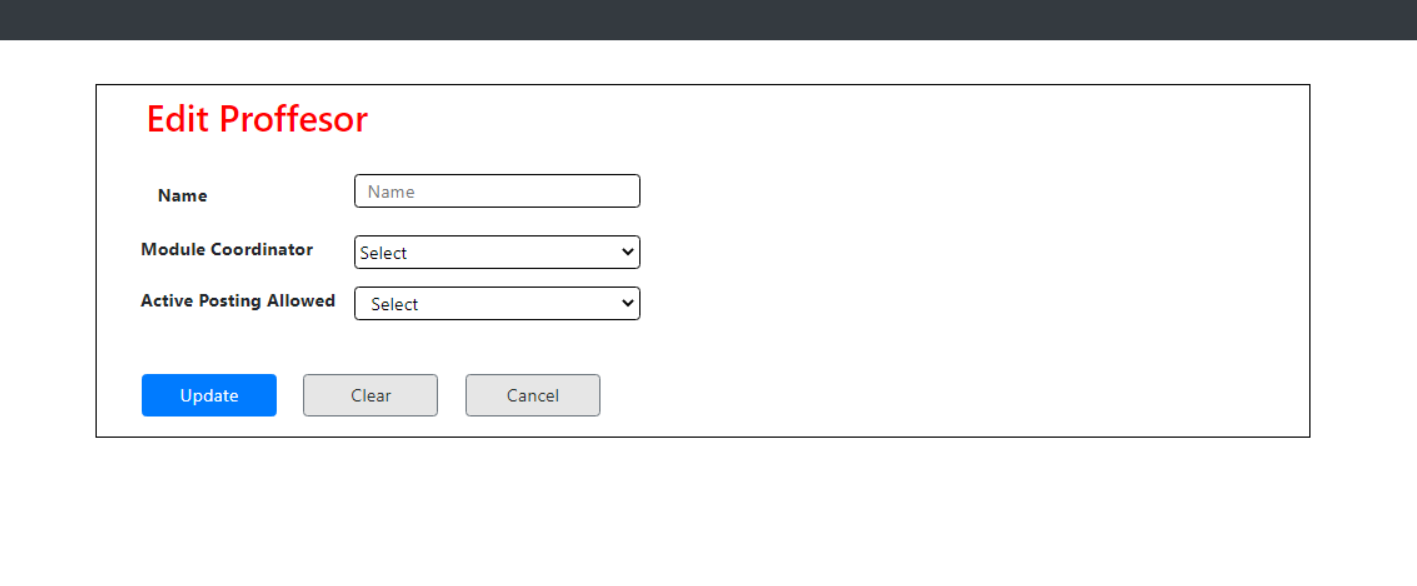


1. **Edit Professor**

Graphical user interface, text, application

Description automatically generated

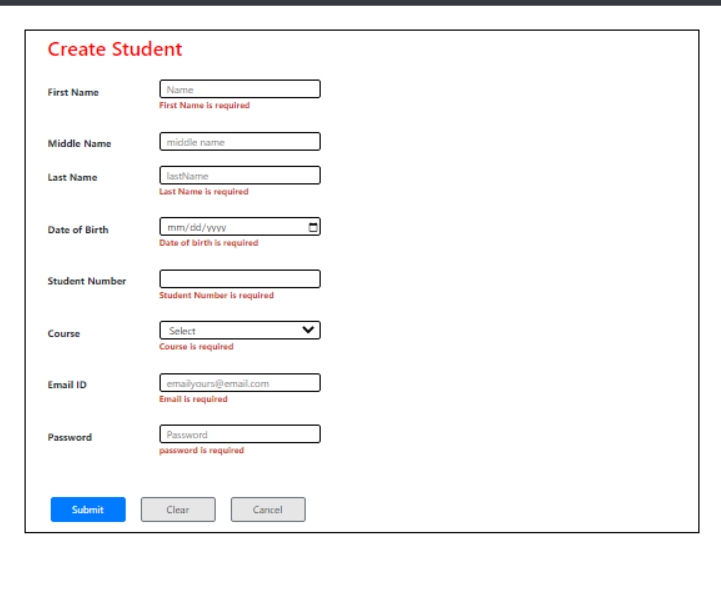
Add page for Review Professor. Updating and deleting functionality will be same as review role.



1. **Student creation \*\*\***

A picture containing graphical user interface

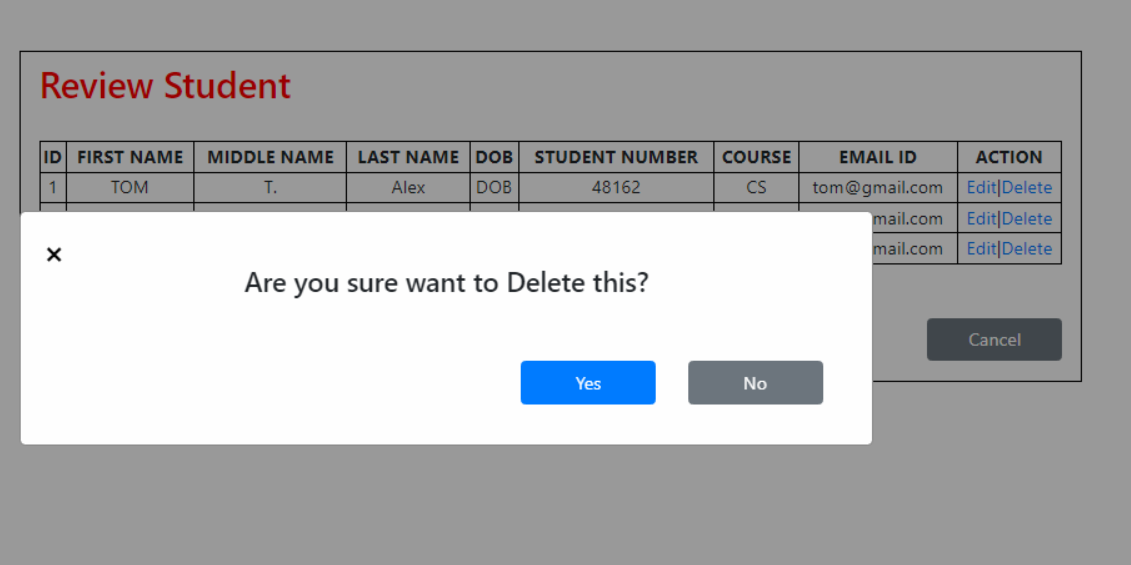
Description automatically generated

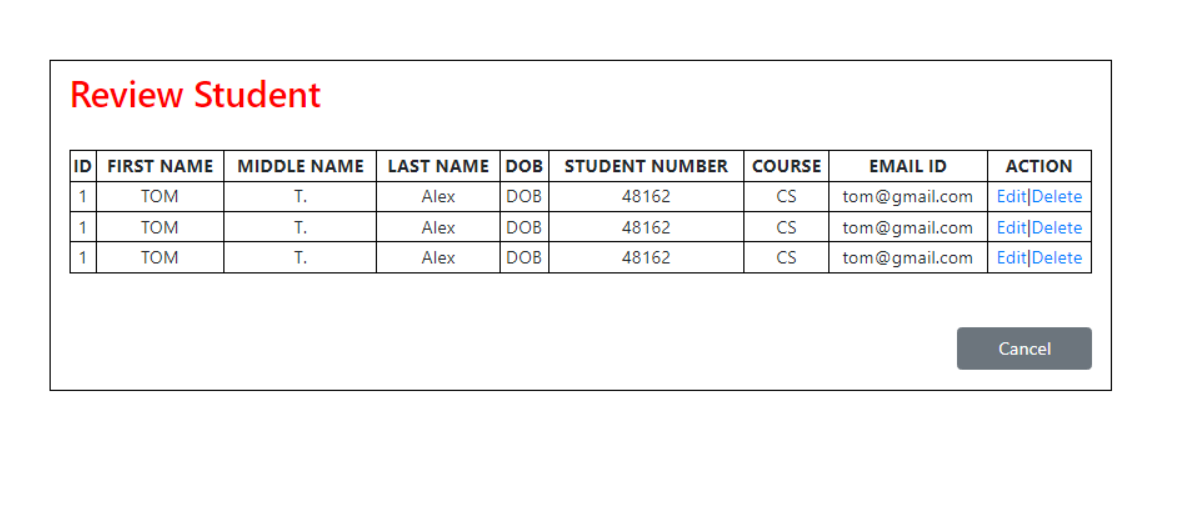
createStudent

1. **Review Student :-reviewStudent**

Table

Description automatically generated



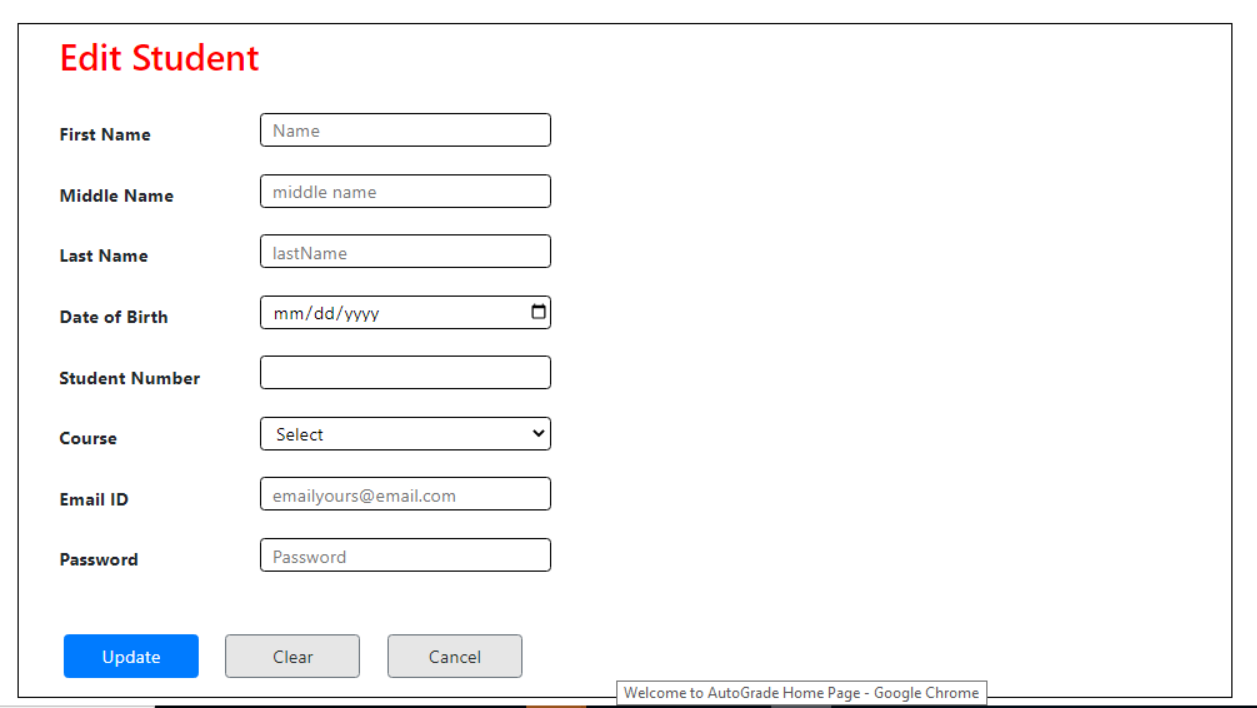


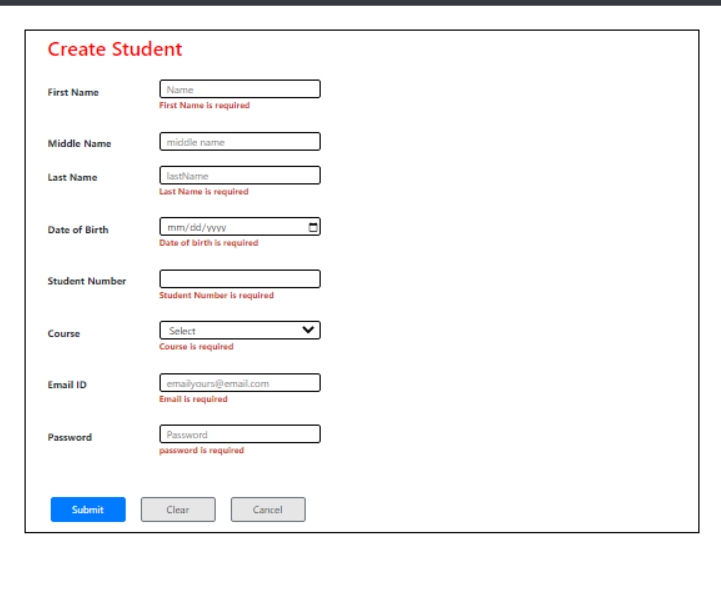
1. **Edit Student**

Graphical user interface, application

Description automatically generated with medium confidence

Add page for Review Student. Updating and deleting functionality will be same as review role.





1. **Assignment creation Page (by the professor)**

Graphical user interface

Description automatically generated

* + - 1. Add page for “Assignment Review” having update and delete functionality.
      2. Default value for “Marking Completed” column is “No”.

1. **Student Assignment Submission Page**

|  |
| --- |
| Graphical user interface, text, application  Description automatically generated |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. No. | Description | Status | ETA for Demo | Client Notes | Developer Notes |
| * + - 1. 1. | * + - 1. Course, module, assignment will be dropdown. The values will come from database. Upload file will be file controller. | Not Started |  | Course:- ACS, CS, DS, Cyber Security  Module:- OS, WAD, NS  Assignment:- CW1, CW2, CW3 |  |
| * + - 1. 2. | * + - 1. Right Side message will be shown only if submission done at least one time. And show him last done submission. After clicking on this link. It will go to “Review Student Submission”. | Not Started |  |  |  |
| 3. |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

1. **Review student submission**

A picture containing chart

Description automatically generated

* + - 1. If the user clicks on the above blue link, the below page child page will be open (point 10th). Please don’t open a new page.
      2. This “Review Student Submission” menu will be seen by both professor and student.

1. **Review Student Assignment Test cases.**

Table

Description automatically generated

Above all component (1-10) is of Auto grader site. We need to send this data to university portal like canvas in JSON format via REST API. Just want to show data in a table form. Tell me when you reach here. You have done a great job. Hat’s off to you and your team. Contact me immediately.

Diagram

Description automatically generated

1. **Compiler API**

**Reference: -** https://docs.jdoodle.com/compiler-api/compiler-api

**Jdoodle API URL:-** <https://api.jdoodle.com/v1/execute>

**Request Json: -**

{

    "clientId": "257e8932d3f5bf0701240e4c08f57e1",

    "clientSecret": "fbb4318fb5e1cbd79ecbdb4cc4eb33998aa8bf9456b4115145ca2a602354ac9e",

    "script": "public class program{public static void main(String [] args){System.out.println(5+5+7);System.out.println(9+1);}}",

    "language": "java",

    "versionIndex": "0"

}

**Response Json:-**

{

    "output": "17\n10\n",

    "statusCode": 200,

    "memory": "24084",

    "cpuTime": "0.05"

}

Course :- to complete today night

Graphical user interface, text, application, email

Description automatically generated

**Spring and Jackson support**

* + - 1. Spring will automatically handle Jackson integration
      2. As long as the Jackson project is on the classpath or POM.xml.
      3. JSON data being passed to REST controller is converted to Java POJO.
      4. Please user Jackson for all JSON data handling via REST API calls.

MOM for 08/07/2022

* Page 8 till Tuesday Morning (12th July,2022). Saving values to “Student Assignment” and “Student Assignment Test Cases”.
* CRUD application for remaining tables 18/07/2022.
* Alternative to jdoodle compliler API.

MOM for 08/07/2022