**STUDENT BOARD APPLICATION**

**THOKALA LALITHA**

**700740741**

**Under the guidance of**

**Prof. Dr. Narasimha Paravastu**

**Computer Information Systems & Information technology**

**University of Central Missouri**

**Introduction:**

This application gives Students the information they want, allows students to view course content of their department and enroll in courses. Each student needs to sign up before they can login into blackboard. Student can enroll into courses, access or download books from library which are added by the admin or professor and upload submissions for assignments that are uploaded by Admin or Professor. Admin is responsible to add the courses and upload books related to the course. Admin will grade the assignments uploaded by students and upload marks of the assignments submitted by students. Professor is also responsible to add the courses and upload books related to the course. Professor can grade the assignments uploaded by students and upload marks of the assignments submitted by students. For Front End Programming I used HTML, CSS, Redux, Redux Toolkit. For Back End Programming I used Node JS and used MongoDB for database and used amazon S3 to upload images into the bucket. I have also used JsonWebToken mechanism for authentication and authorization. Used NodeMailer to send email to students.

**Roles and Responsibilities:**

**Admin Role:**

Admin role plays a crucial role in the project, because this role is for department head who maintains the information required for the application like Courses, Books, Assignments. Admin will not have a sign up process. He will directly be added into the database. He can login into the application by using his credentials that were added in database. Once he is logged in the application, he can add courses, books and assignments related to the courses. He will even upload marks for the assignments uploaded by the students. Then an email will be sent to the student’s mail ID giving them information of the grades that they received.

After admin logged in to the website there is a button to logout. When he clicks on the logout button the site will redirect to the home page where registered students or admins can login.

Functionalities:

* Admin can add Courses.
* Admin can add books related to a course that can be downloaded.
* Admin can add assignments related to a course that can be downloaded.
* Admin will upload marks to the assignments submitted by the students.

Login Credentials:

Email: admin1@gmail.com

Password: admin@123456

**Student Role:**

Students need to register by filling the form details, before they can login into the application. Once student is registered, he will then login into the application using his email id and password. He can then see three pages after login. The firs page displays the Courses that are available and gives him an opportunity to enroll into his desired course. In the second page he can see the list of books available for the courses enrolled and can download books if needed. In the third page he can see the list of assignments for the courses that he has enrolled. He will need to upload the submissions for the assignments and once they are graded he will receive an email about the marks and he can also see this information in the application.

Functionalities:

* Student can Sign up and login into the application.
* Student can view list of courses and enroll into them.
* Student can view list of books and download them.
* Student can view list of assignments and upload submissions to them.

Login Credentials:

Email: lxt07410@ucmo.edu

Password: lalitha@1234

**Professor Role:**

Professor maintains the information required for the application like Courses, Books, Assignments. Professor will not have a sign up process. He will directly be added into the database. He can login into the application by using his credentials that were added in database. Once he is logged in the application, he can add courses, books and assignments related to the courses. He will even upload marks for the assignments uploaded by the students. Then an email will be sent to the student’s mail ID giving them information of the grades that they received.

After Professor logged in to the website there is a button to logout. When he clicks on the logout button the site will redirect to the home page where registered students or admins can login.

Functionalities:

* Professor can add Courses.
* Professor can add books related to a course that can be downloaded.
* Professor can add assignments related to a course that can be downloaded.
* Professor will upload marks to the assignments submitted by the students.

Login Credentials:

Email: paravastu@gmail.com

Password: paravastu@123456

**Use Cases:**

|  |  |
| --- | --- |
| **Use Case:** | 1 |
| **Use Case Name:** | **Admin Login** |
| **Actors:** | Admin. |
| **Description:** | The admin logs into the application using their credentials. |
| **Trigger:** | The admin initiates the login process by clicking the login button. |
| **Preconditions:** | Admin credentials are stored in the database. |
| **Postconditions:** | The admin gains access to their admin dashboard. |

Registration

|  |  |
| --- | --- |
| **Use Case:** | 2 |
| **Use Case Name:** | **Student Sign Up** |
| **Actors:** | Student |
| **Description:** | A student creates a new account in the application. |
| **Trigger:** | A student initiates the sign-up process. |
| **Preconditions:** | The student's email is not already registered in the system. |
| **Postconditions:** | The student's account is created, and they receive a confirmation email. |

|  |  |
| --- | --- |
| **Use Case:** | 3 |
| **Use Case Name:** | **Student Login** |
| **Actors:** | Student |
| **Description:** | A student logs into the application using their credentials. |
| **Trigger:** | The student initiates the login process. |
| **Preconditions:** | The student has a registered account in the system. |
| **Postconditions:** | The student gains access to their user account. |

View Profil

|  |  |
| --- | --- |
| **Use Case:** | **4** |
| **Use Case Name:** | **View Course Content** |
| **Actors:** | Student |
| **Description:** | A student accesses and views course content within their department. |
| **Trigger:** | The student selects a course to view. |
| **Preconditions:** | The student is logged into the application, and the course content is available. |
| **Postconditions:** | The student can see the course content. |

Edit Profile

|  |  |
| --- | --- |
| **Use Case:** | 5 |
| **Use Case Name:** | **Enroll in Courses** |
| **Actors:** | Student |
| **Description:** | A student enrolls in one or more courses. |
| **Trigger:** | The student selects courses to enroll in. |
| **Preconditions:** | The student is logged into the application, and the courses are available for enrollment. |
| **Postconditions:** | The student is registered for the selected courses. |

|  |  |
| --- | --- |
| **Use Case:** | 6 |
| **Use Case Name:** | **Access/Download Books** |
| **Actors:** | Student |
| **Description:** | A student accesses or downloads books from the library added by the admin. |
| **Trigger:** | The student selects a book to access or download. |
| **Preconditions:** | The student is logged in, and the books are available in the library. |
| **Postconditions:** | The student can access or download the selected book. |

|  |  |
| --- | --- |
| **Use Case:** | 7 |
| **Use Case Name:** | **Upload Assignment Submission** |
| **Actors:** | Student |
| **Trigger:** | The student selects an assignment to submit and uploads the file. |
| **Preconditions:** | The student is logged in, and the assignment is open for submissions. |
| **Postconditions:** | The student's assignment is submitted for grading. |
| **Description:** | A student uploads their assignment submission. |

|  |  |
| --- | --- |
| **Use Case:** | 8 |
| **Use Case Name:** | **Grade Assignments** |
| **Actors:** | Admin |
| **Description:** | The admin grades assignments submitted by students. |
| **Trigger:** | The admin selects an assignment for grading. |
| **Preconditions:** | The admin is logged in and has the necessary permissions. |
| **Postconditions:** | The assignment is graded, and the grade is recorded in the system. |

|  |  |
| --- | --- |
| **Use Case:** | 9 |
| **Use Case Name:** | **Upload Books and Assignments** |
| **Actors:** | Admin |
| **Description:** | The admin uploads books and Assignments related to courses. |
| **Trigger:** | The admin selects a course and uploads the associated books and Assignments . |
| **Preconditions:** | The admin is logged in and has the necessary permissions. |
| **Postconditions:** | The books and Assignments are added and associated with the selected course. |

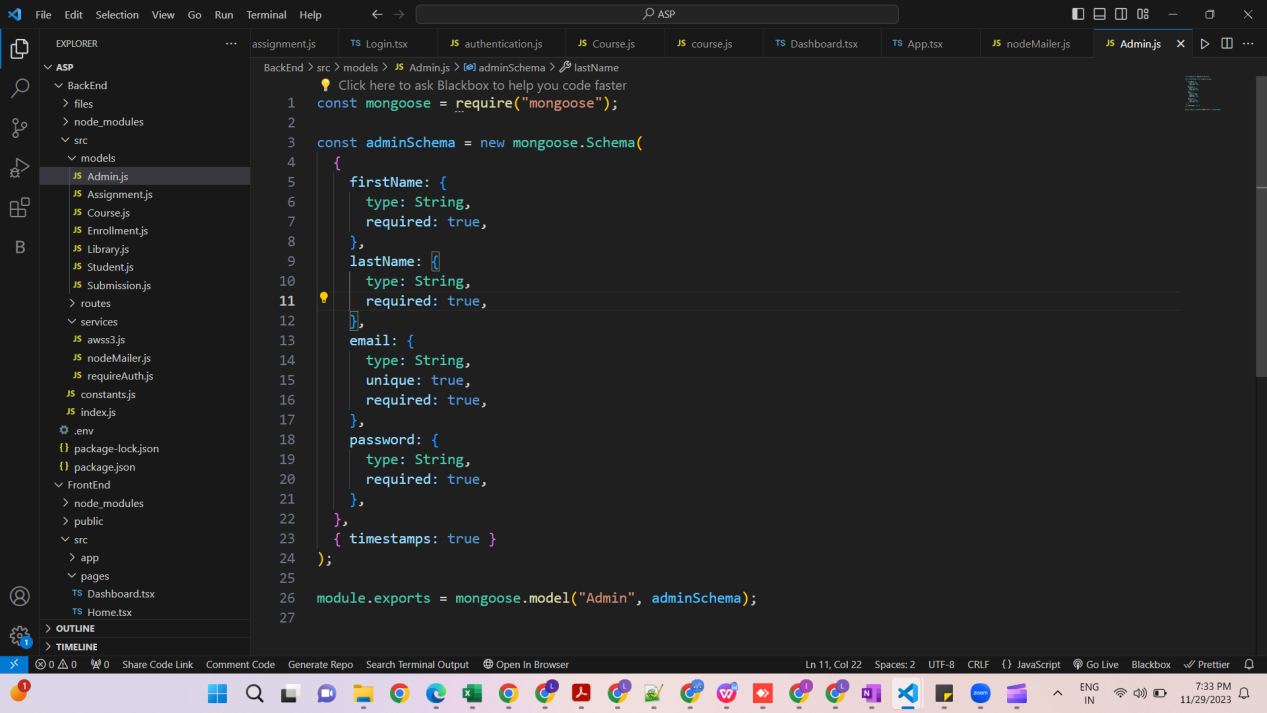
|  |  |
| --- | --- |
| **Use Case:** | 10 |
| **Use Case Name:** | **Professor Login** |
| **Actors:** | Professor. |
| **Description:** | The professor logs into the application using their credentials. |
| **Trigger:** | The professor initiates the login process by clicking the login button. |
| **Preconditions:** | Professor credentials are stored in the database. |
| **Postconditions:** | The professor gains access to their professor dashboard. |

|  |  |
| --- | --- |
| **Use Case:** | 11 |
| **Use Case Name:** | **Grade Assignments** |
| **Actors:** | Professor |
| **Description:** | The Professor grades assignments submitted by students. |
| **Trigger:** | The Professor selects an assignment for grading. |
| **Preconditions:** | The Professor is logged in and has the necessary permissions. |
| **Postconditions:** | The assignment is graded, and the grade is recorded in the system. |

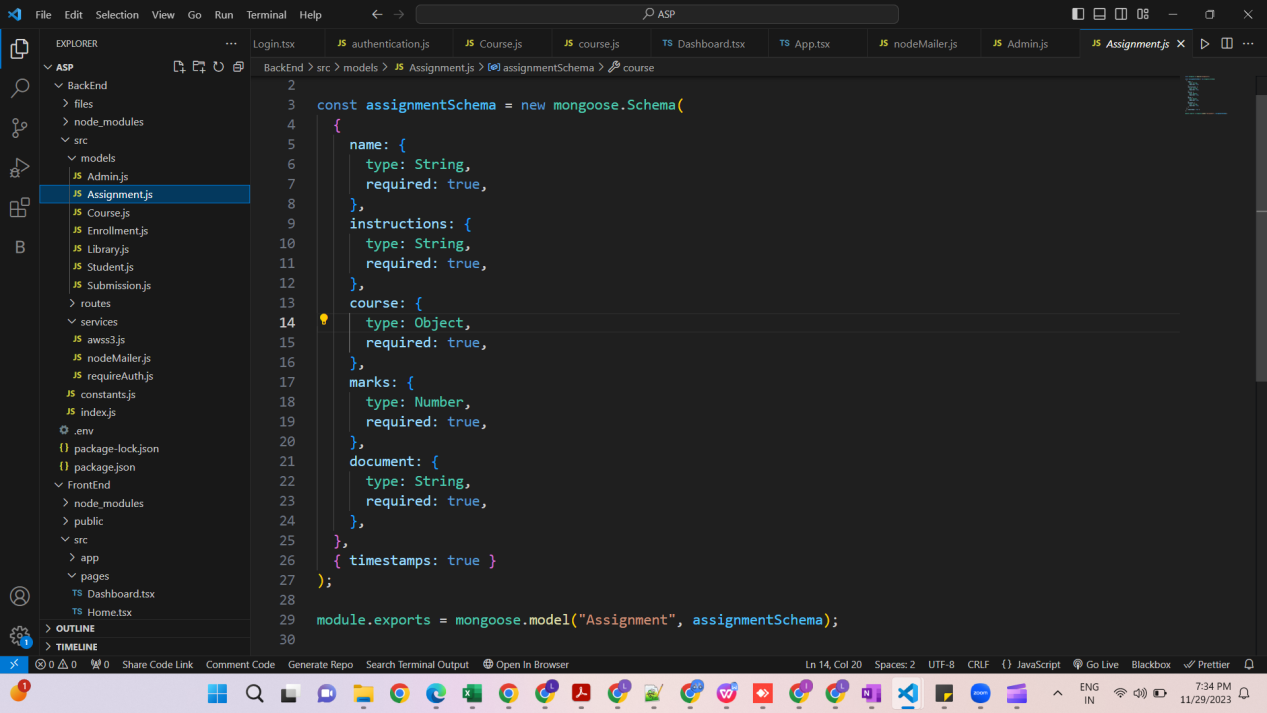
|  |  |
| --- | --- |
| **Use Case:** | 12 |
| **Use Case Name:** | **Upload Books and Assignments** |
| **Actors:** | Professor |
| **Description:** | The Professor uploads books and Assignments related to courses. |
| **Trigger:** | The Professor selects a course and uploads the associated books and Assignments . |
| **Preconditions:** | The Professor is logged in and has the necessary permissions. |
| **Postconditions:** | The books and Assignments are added and associated with the selected course. |

SCHEMA IN THIS APPLICATION  
------------------------------------------------------------------------------------------------

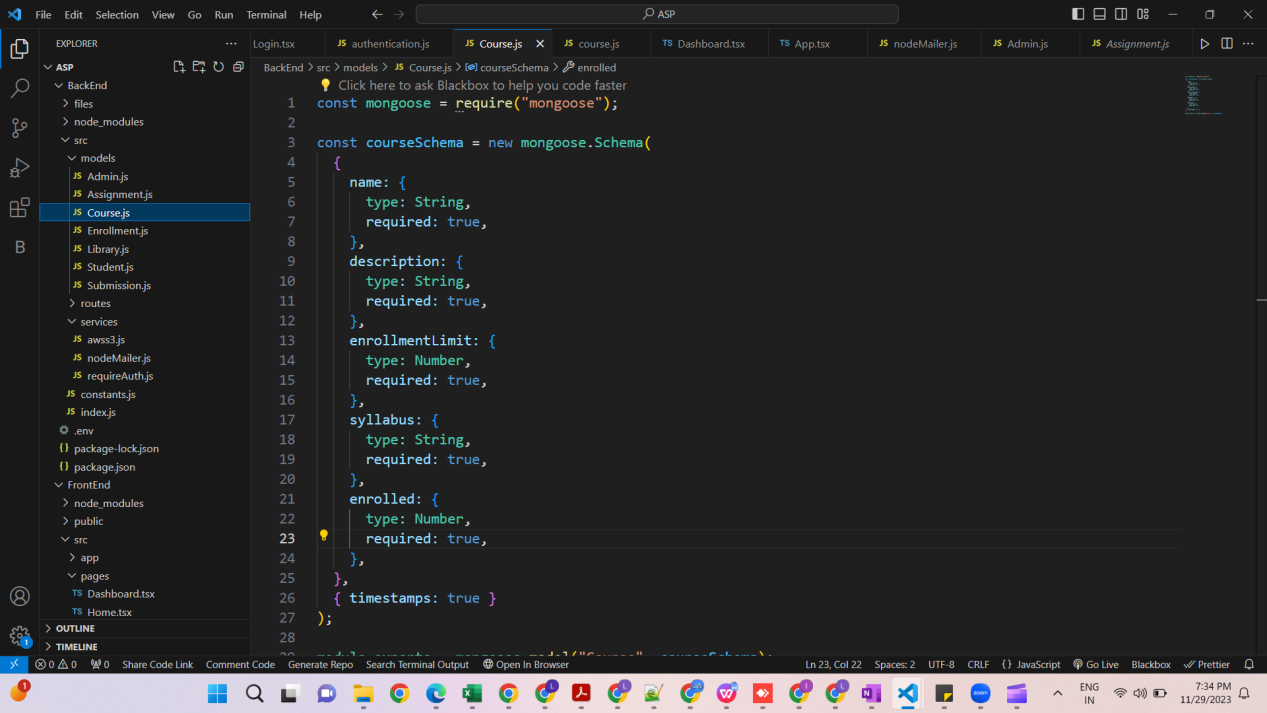
1. Admin Schema



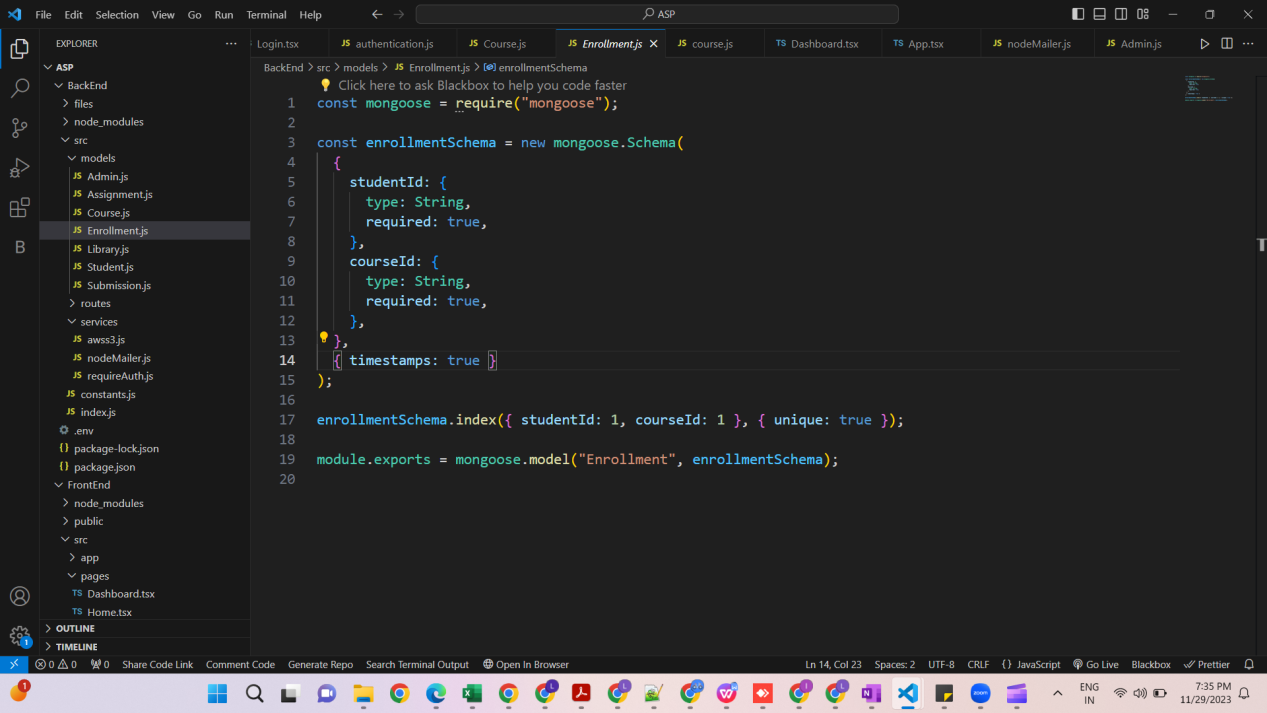
1. Assignments Schema



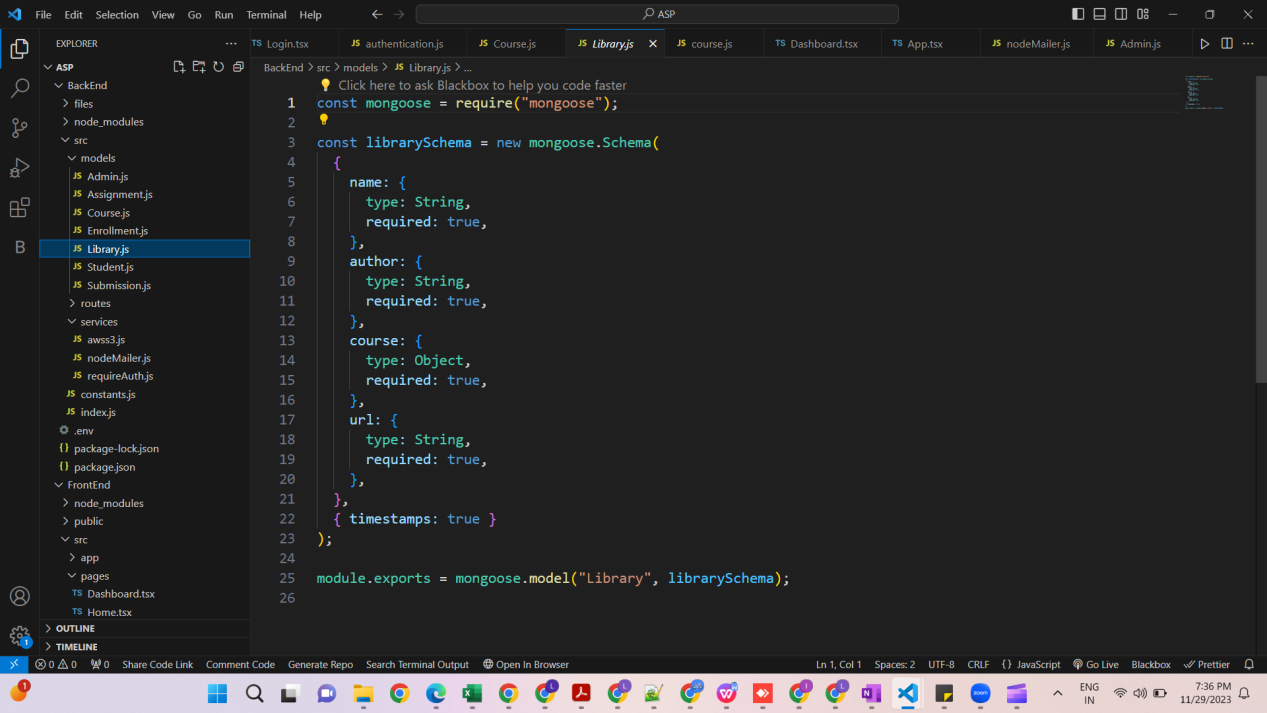
1. Courses Schema



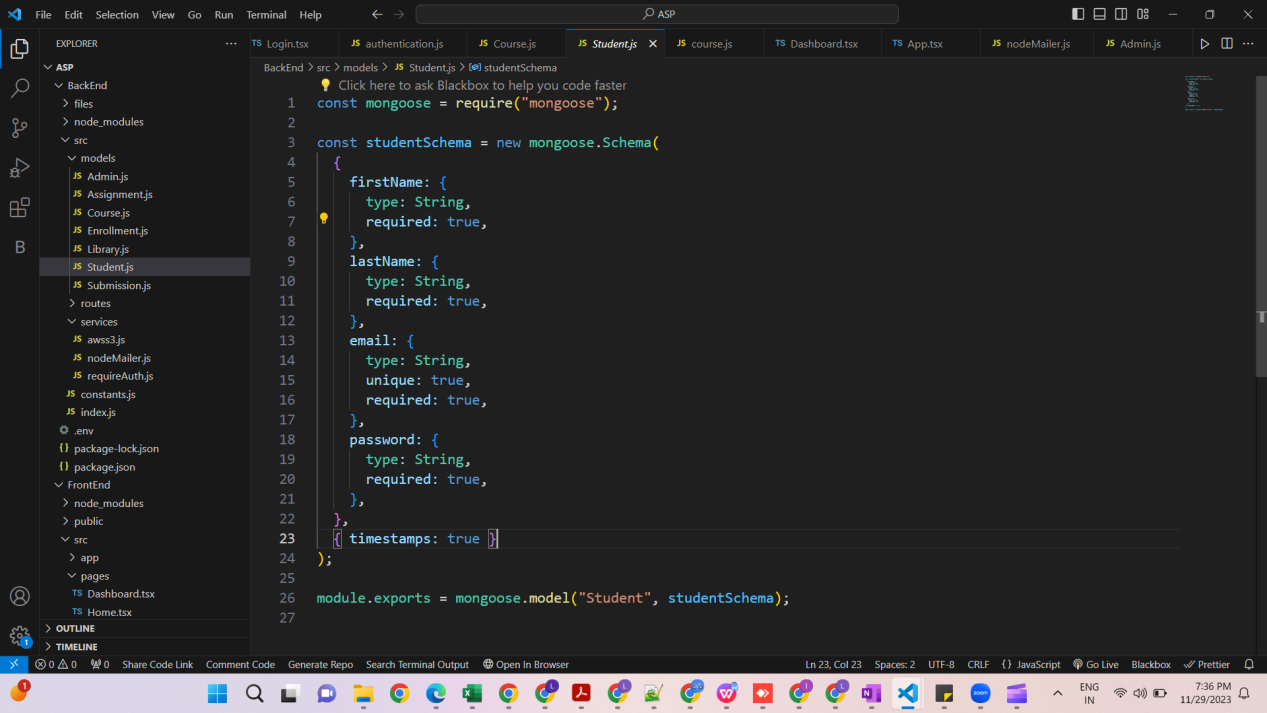
1. Enrollment Schema



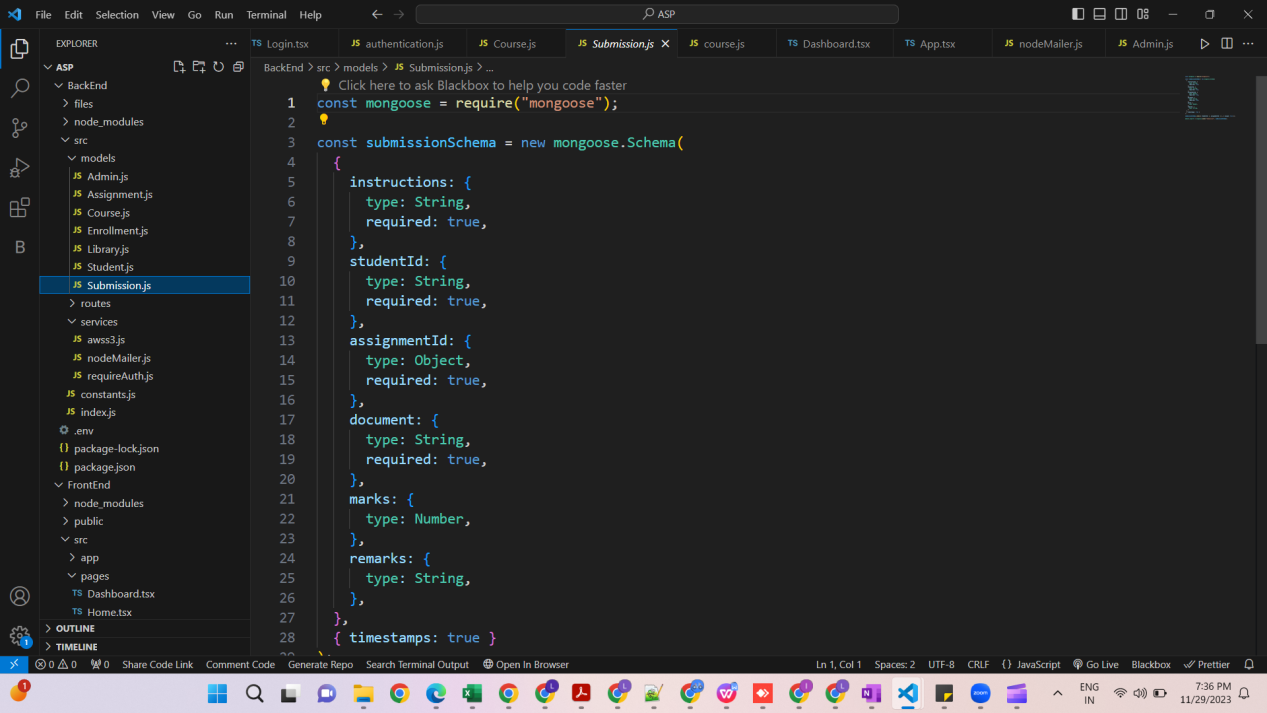
1. Library Schema



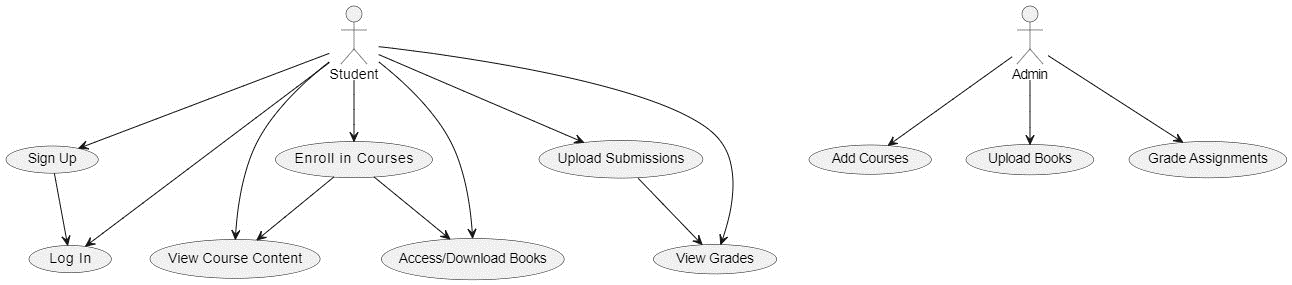
1. Student Schema

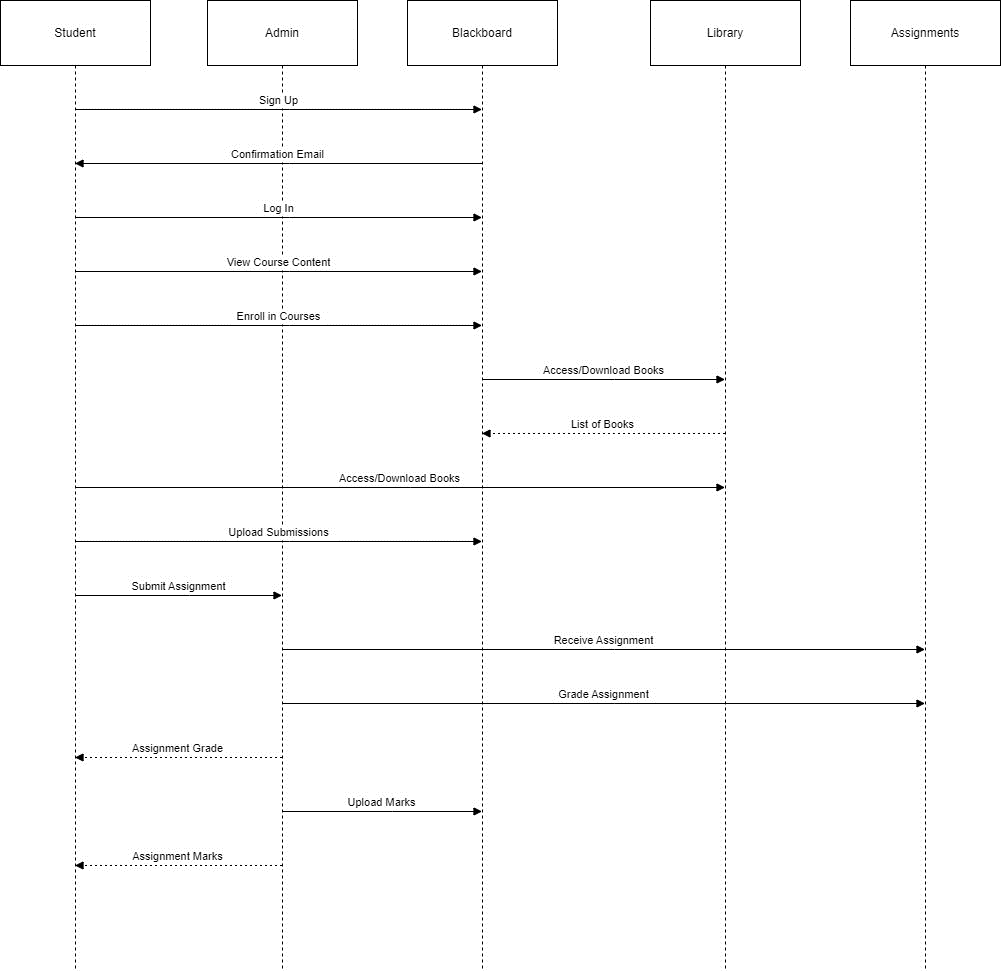


1. Submission Schema

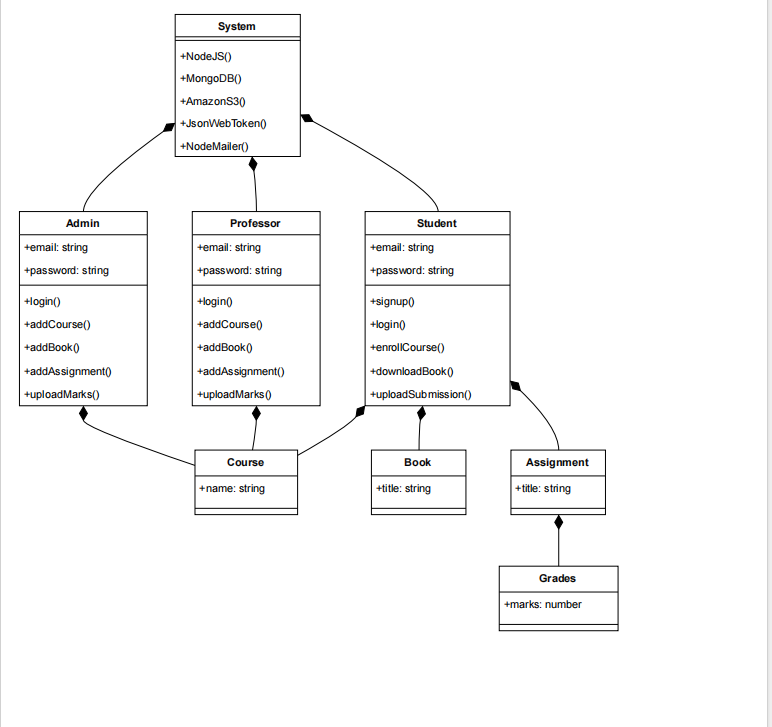


**USE CASE DIAGRAMS:**





Class Diagram



The technologies used in this application are HTML, CSS, Redux, Redux Toolkit, Node JS, Mongo DB, AWS S3

**HTML**

HTML is the standard markup language used for creating web pages and web applications.

It provides a structured way to define the content and layout of a web page using elements like headings, paragraphs, lists, links, and more.

HTML is essential for creating the structure of a web page and is complemented by other technologies like CSS for styling and JavaScript for interactivity.

It uses a tag-based syntax, where elements are enclosed in opening and closing tags, allowing developers to specify the type and properties of content.

HTML5, the latest version of HTML, introduced new elements and features that enhance multimedia support, offline capabilities, and improved semantics.

**CSS (Cascading Style Sheets):**

CSS is a stylesheet language used for controlling the presentation and styling of HTML and XML documents.

It allows developers to define the layout, colors, fonts, and other visual aspects of a web page, making it visually appealing and user-friendly.

CSS follows a rule-based system, where selectors target HTML elements, and properties define their styling.

It supports responsive design, enabling web pages to adapt to various screen sizes and devices.

CSS preprocessors like Sass and LESS enhance CSS development by adding features like variables and nesting.

CSS frameworks and libraries, such as Bootstrap and Materialize, provide pre-designed styles and components for faster web development.

**Redux:**

Redux is a predictable state container for JavaScript applications, commonly used with React.

It helps manage the state of an application in a consistent and predictable manner, making it easier to debug and maintain.

Redux stores application data in a single, immutable state object that can only be modified by dispatching actions.

Actions are dispatched to update the state, and reducers specify how the state should change in response to actions.

Middleware can be used to add additional functionality, such as logging or asynchronous data fetching.

Redux is popular for complex and data-intensive applications where centralized state management is crucial.

**Redux Toolkit:**

Redux Toolkit is an official package that simplifies the usage of Redux, reducing boilerplate code.

It includes utilities like createSlice for defining reducers, configureStore for setting up the Redux store, and createAsyncThunk for handling asynchronous actions.

Redux Toolkit promotes best practices and reduces the amount of manual configuration required when setting up Redux.

It encourages the use of the "slice" pattern to manage individual parts of the state.

Redux Toolkit is widely adopted in the Redux community for its developer-friendly features and improved ergonomics.

**Node.js:**

Node.js is a server-side runtime environment that allows developers to build scalable and efficient network applications using JavaScript.

It is built on the V8 JavaScript engine and uses an event-driven, non-blocking I/O model, making it suitable for real-time applications.

Node.js has a package manager called npm (Node Package Manager) that hosts a vast ecosystem of open-source libraries and frameworks.

It's commonly used for building web servers, RESTful APIs, microservices, and serverless applications.

Node.js enables full-stack JavaScript development by running JavaScript on both the client and server sides.

**MongoDB:**

MongoDB is a NoSQL database system known for its flexibility and scalability.

It stores data in a document format, allowing developers to work with data in a more natural way.

MongoDB is schema-less, which means documents in a collection can have different structures.

It is well-suited for handling large volumes of unstructured or semi-structured data.

MongoDB uses a JSON-like query language for data retrieval and manipulation.

It is often used in modern web applications and big data projects.

**AWS S3 (Amazon Simple Storage Service):**

AWS S3 is an object storage service provided by Amazon Web Services (AWS) for storing and retrieving data, including images, videos, and documents.

It offers high durability and availability, making it a reliable solution for data storage.

S3 uses a flat namespace, allowing users to organize data in buckets and folders.

It provides fine-grained access control using policies and access control lists (ACLs).

AWS S3 integrates with other AWS services, making it a key component in cloud-based architectures.

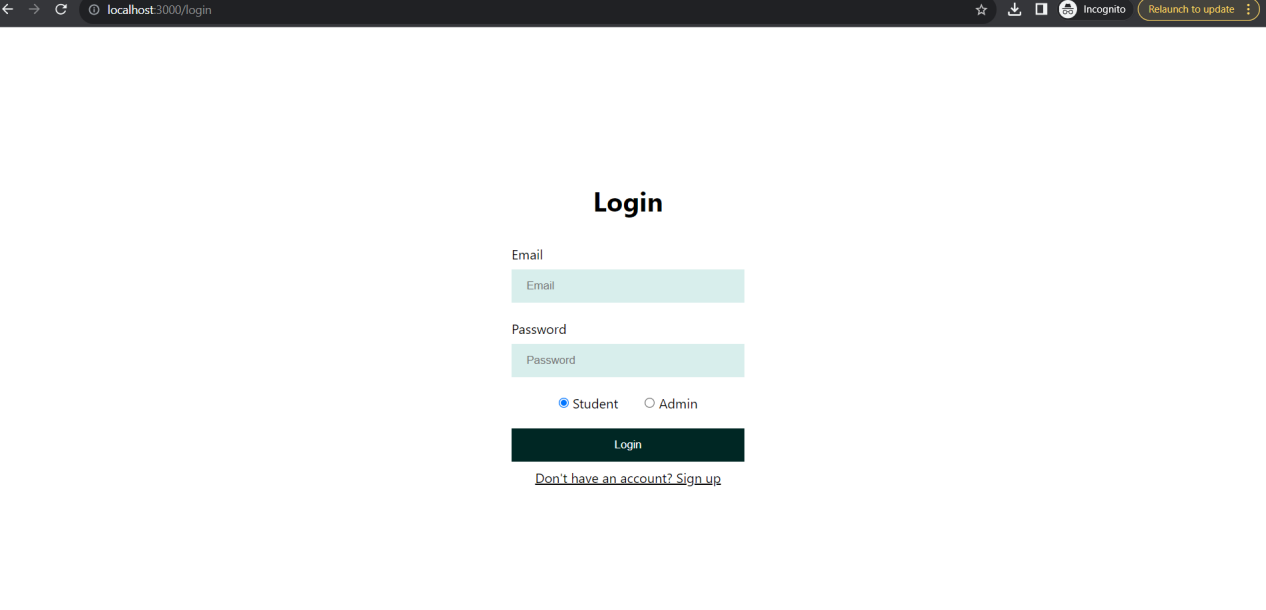
S3 supports versioning, data lifecycle management, and server-side encryption for enhanced data security.

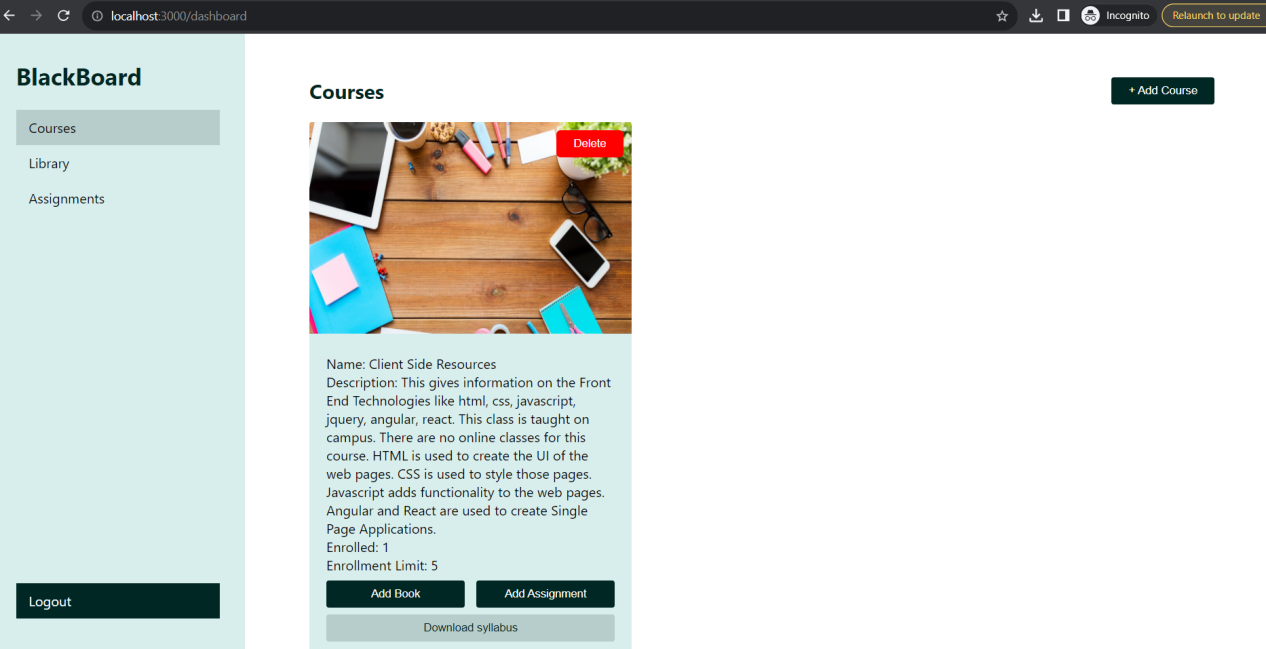
**Application Flow:**

In this project there will be two types of roles. One is admin and other is student.

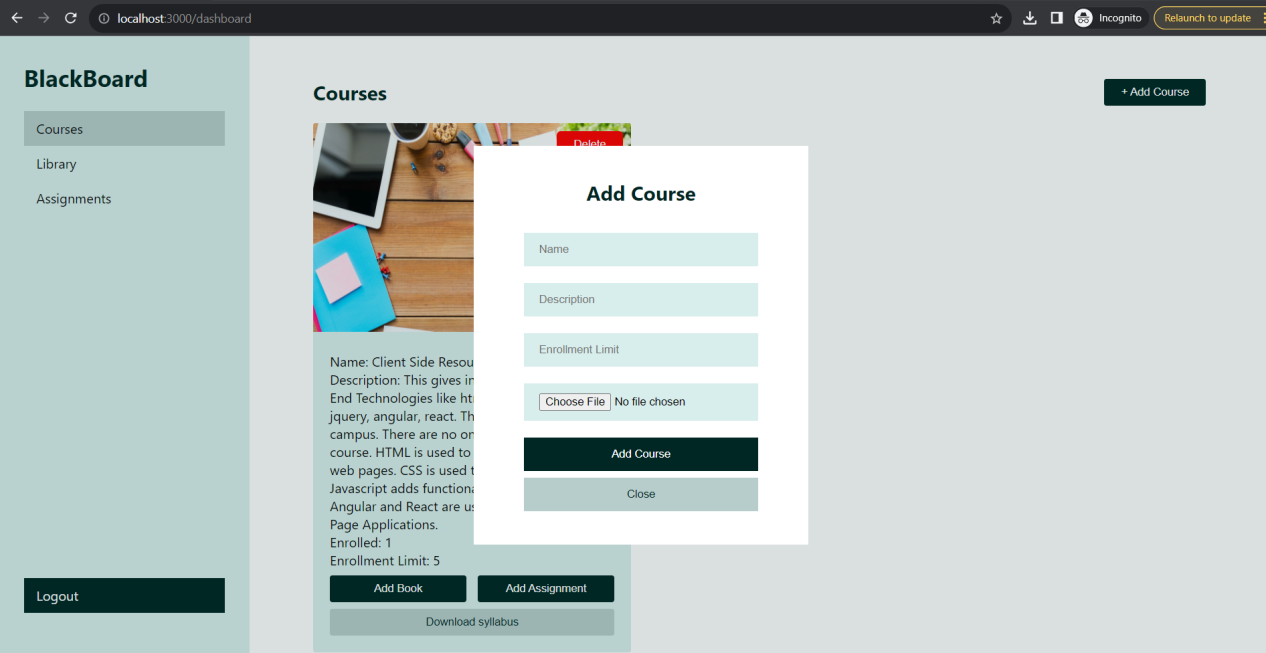
**Admin Flow**

The admin login page looks like below.

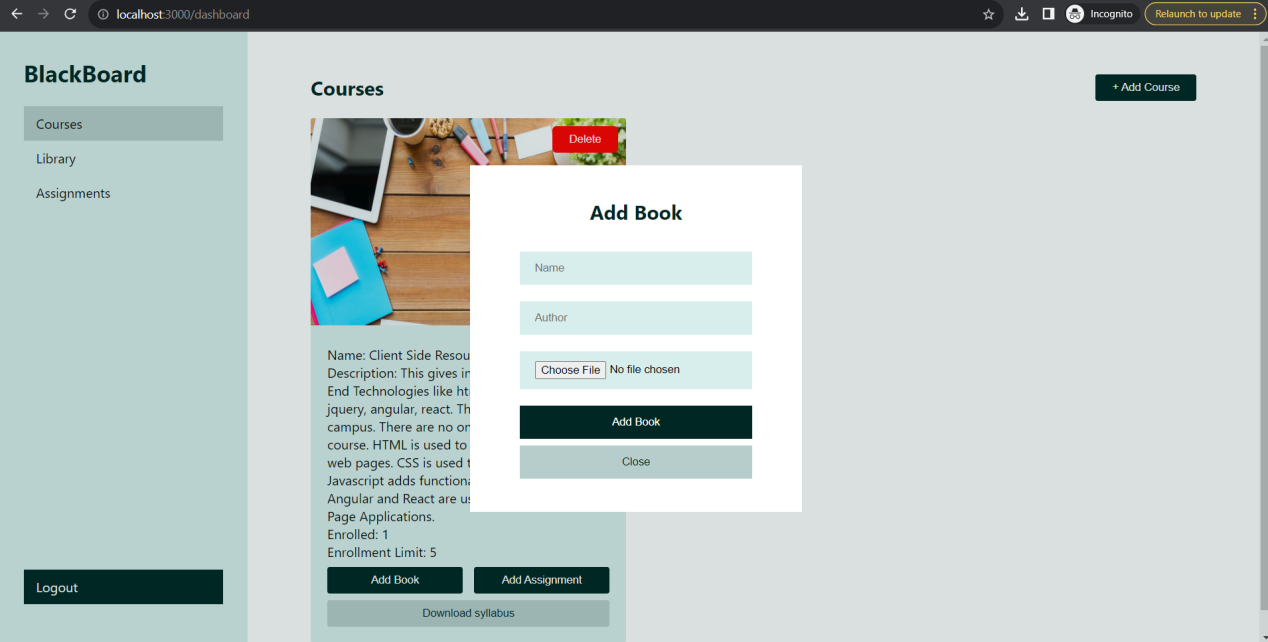
The below page is displayed when we login with Admin Credentials.

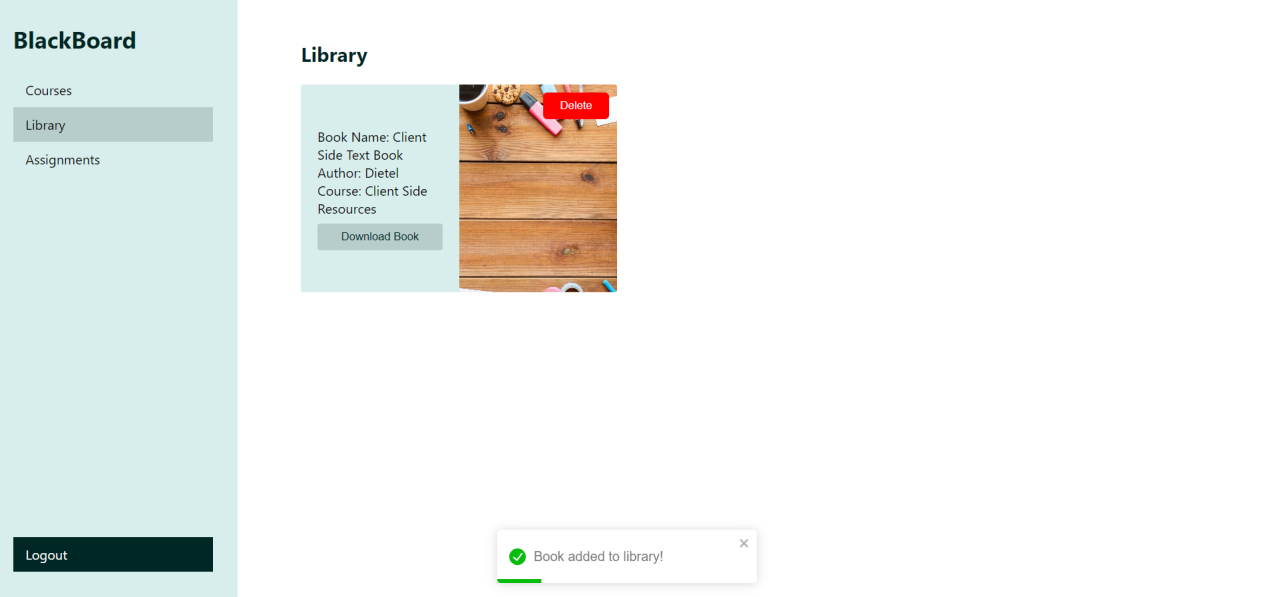


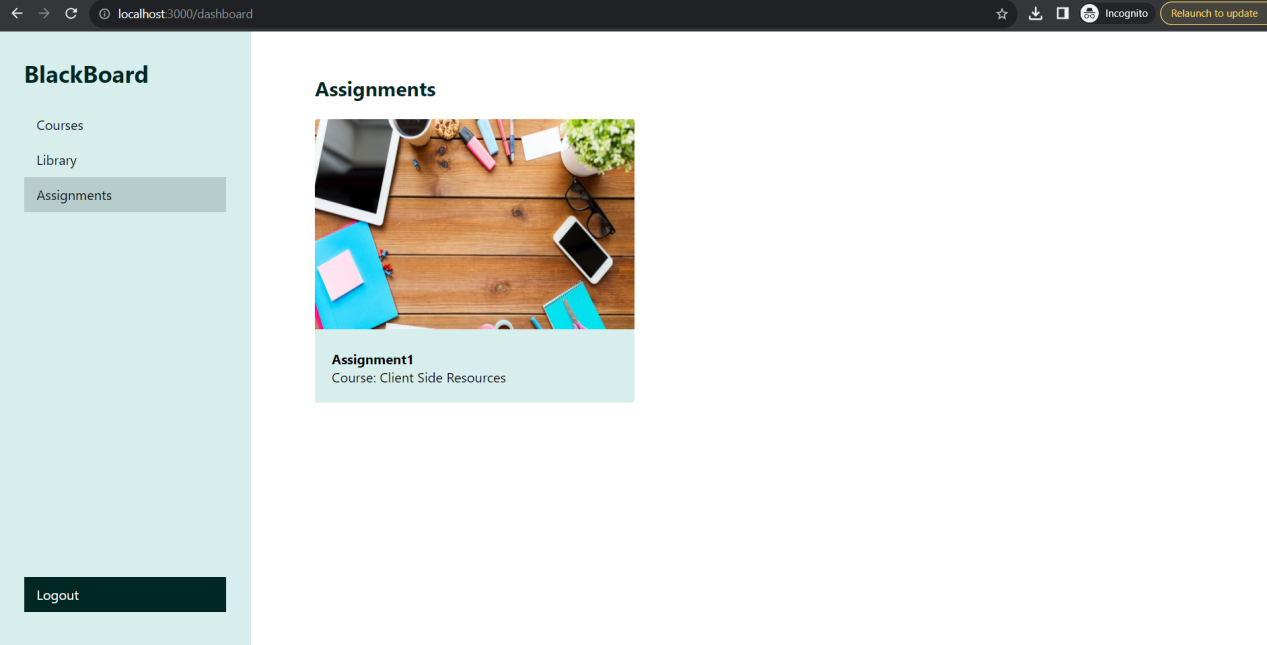
The admin can create Courses in which the student can enroll.



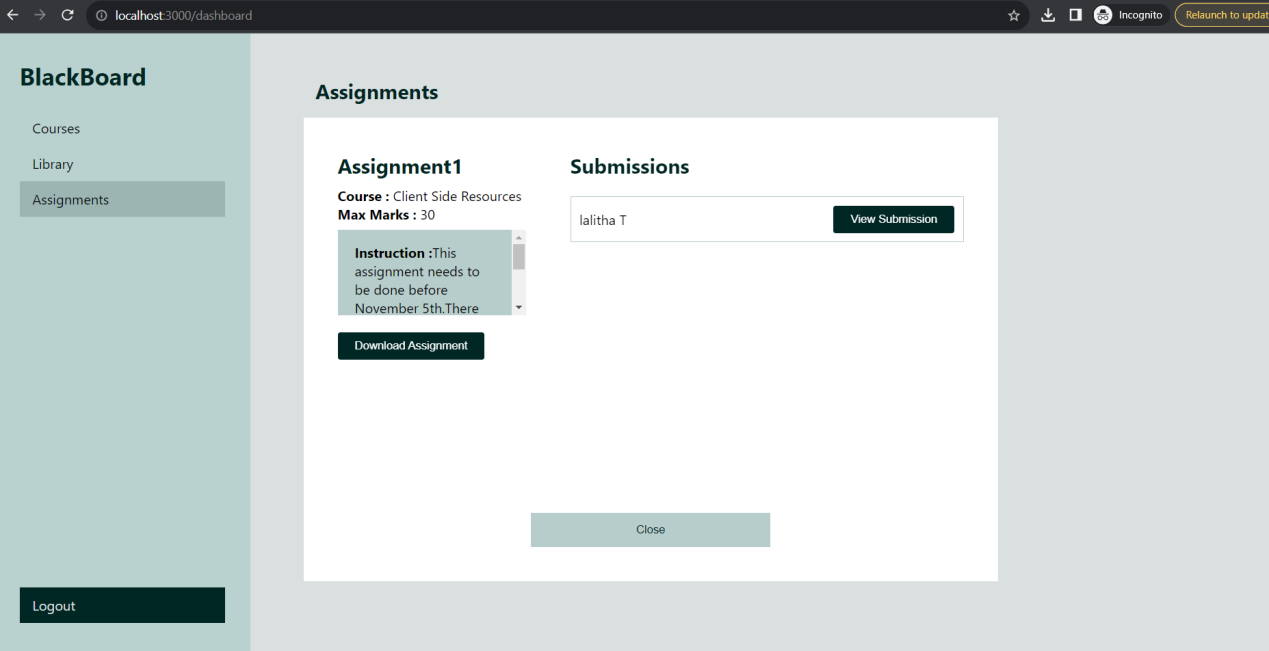
Once he creates the courses, he can then add books, assignments related to those courses .



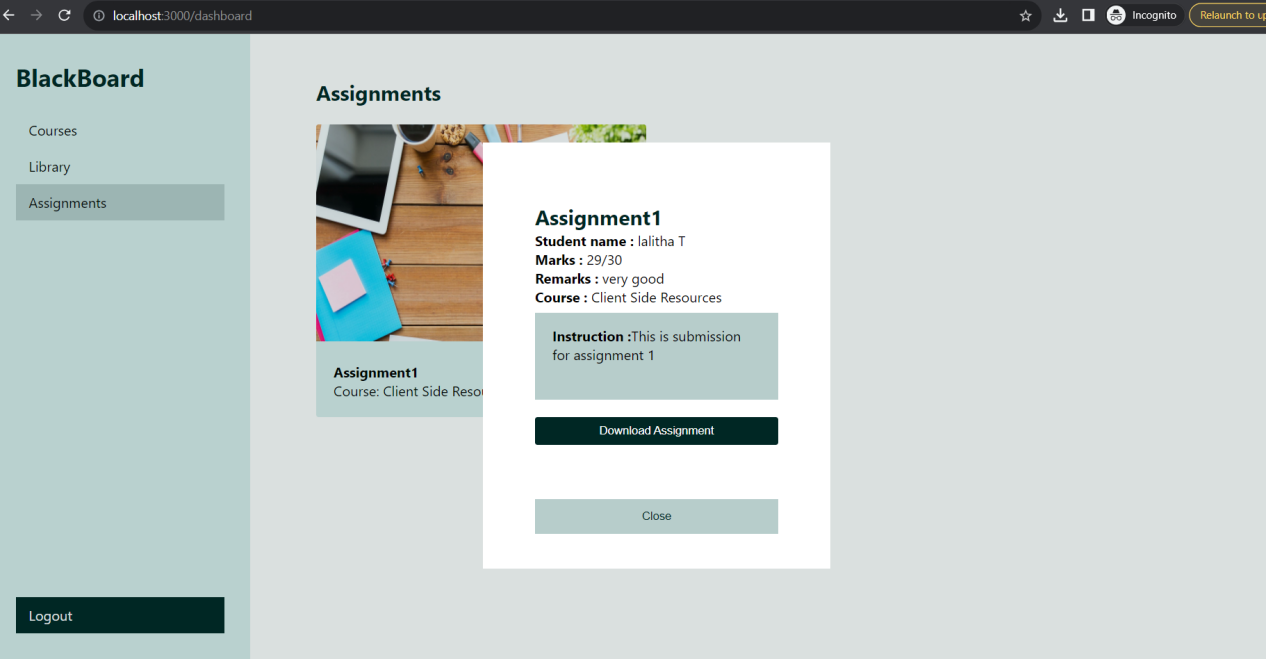




When he clicks on the particular assignment, he can see the information about the assignment and see how many students have submitted the assignment.

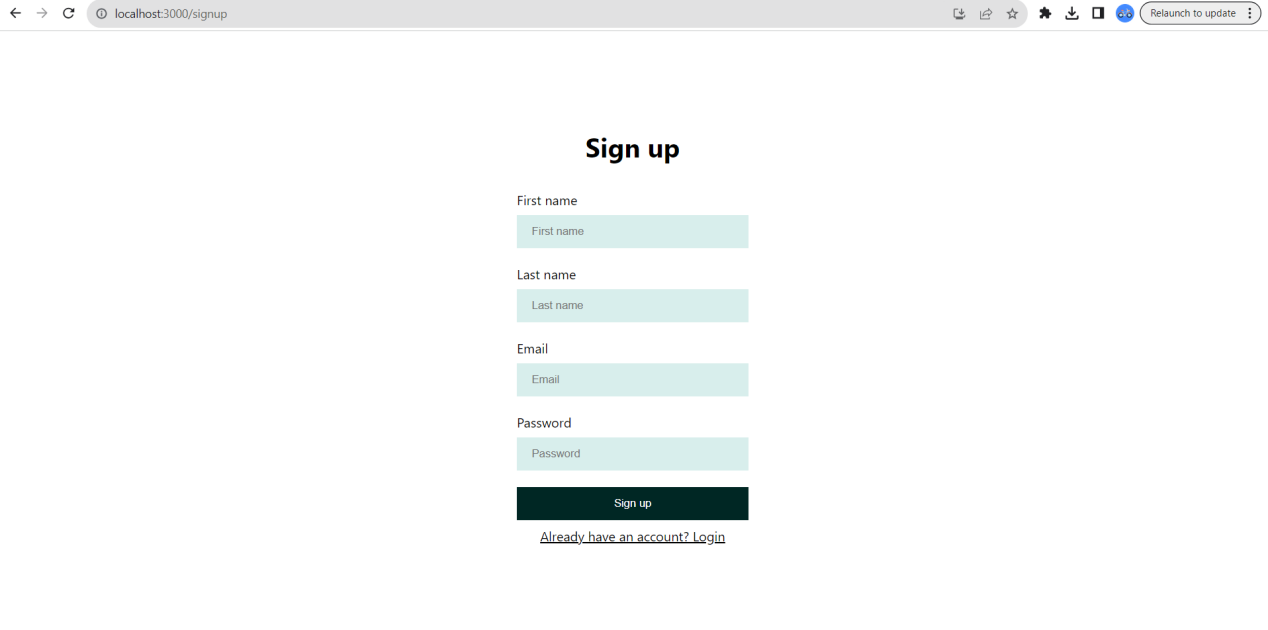


The admin then clicks on View Submission button to see the submissions made by the students. In this page he can even download the assignment and grade it and upload the marks in the portal.

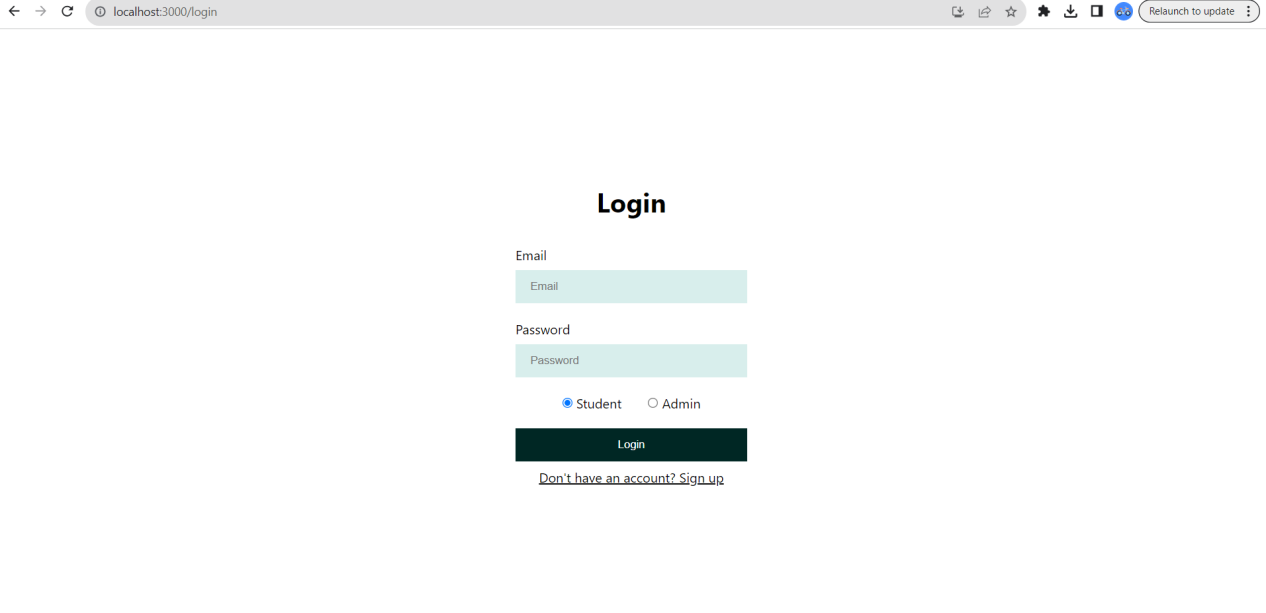


Student Flow

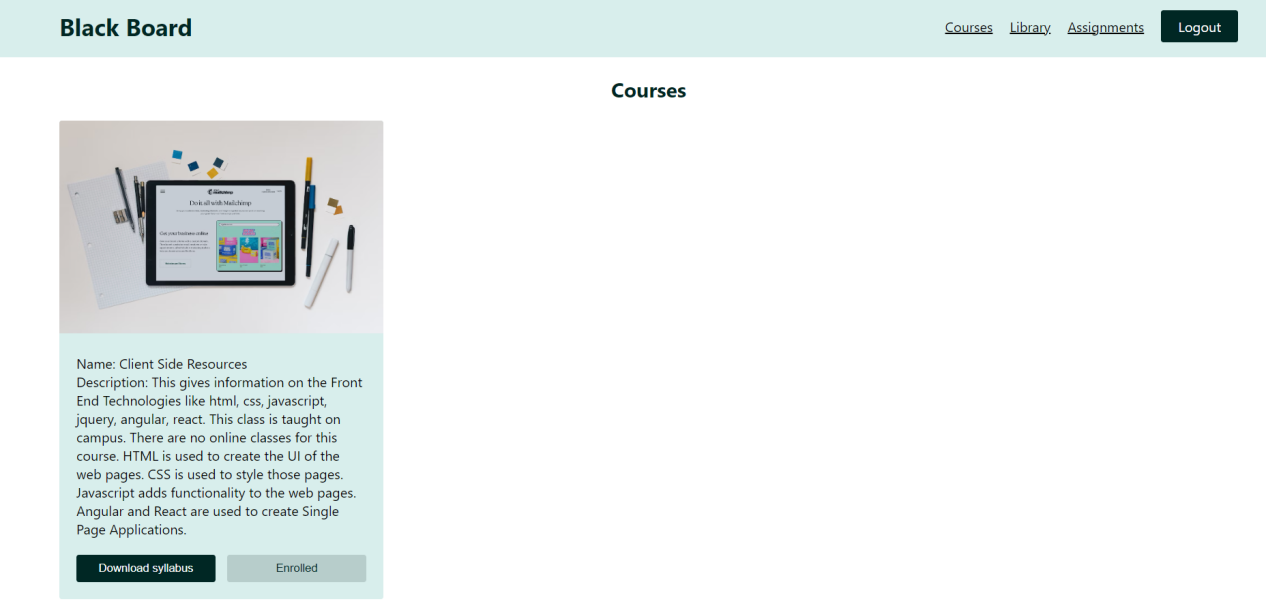
Students need to sign up before they can login. The below is the UI for Students to sign up.



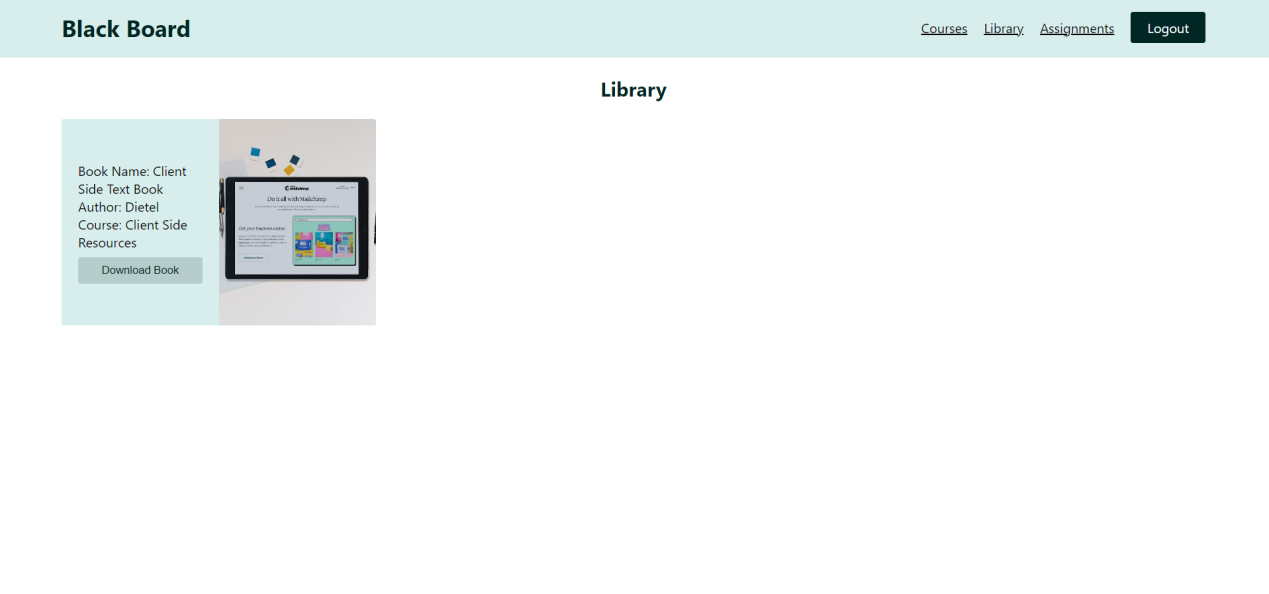
The below is the screen for student to login.



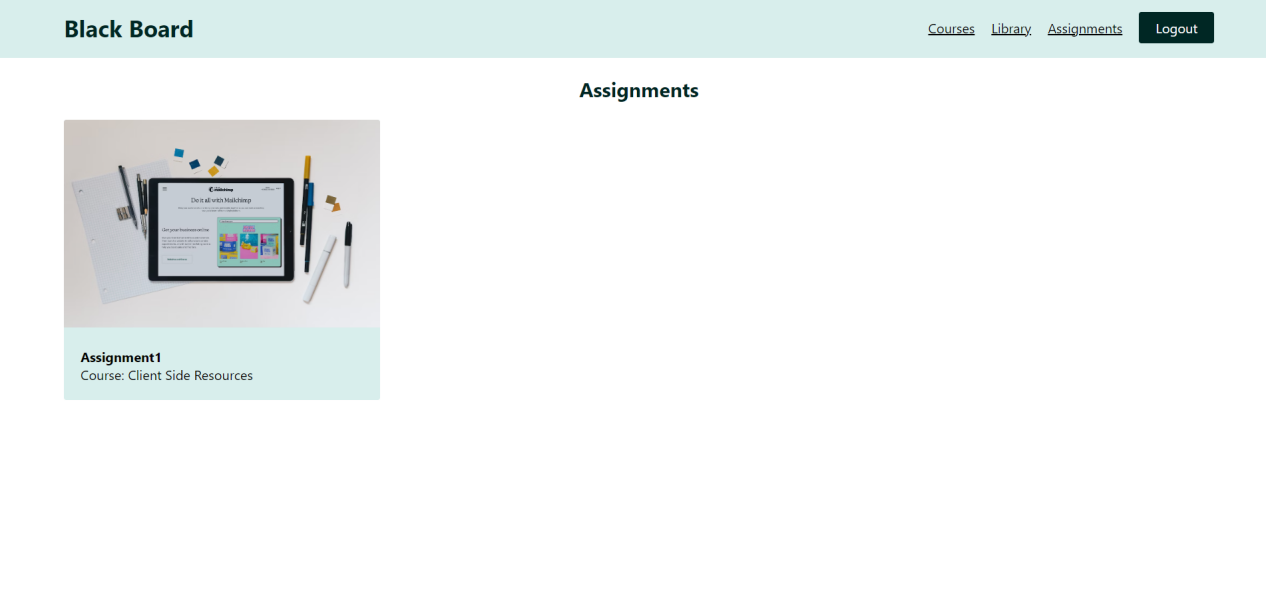
After the Student logs in, he will be able to see the list of courses.



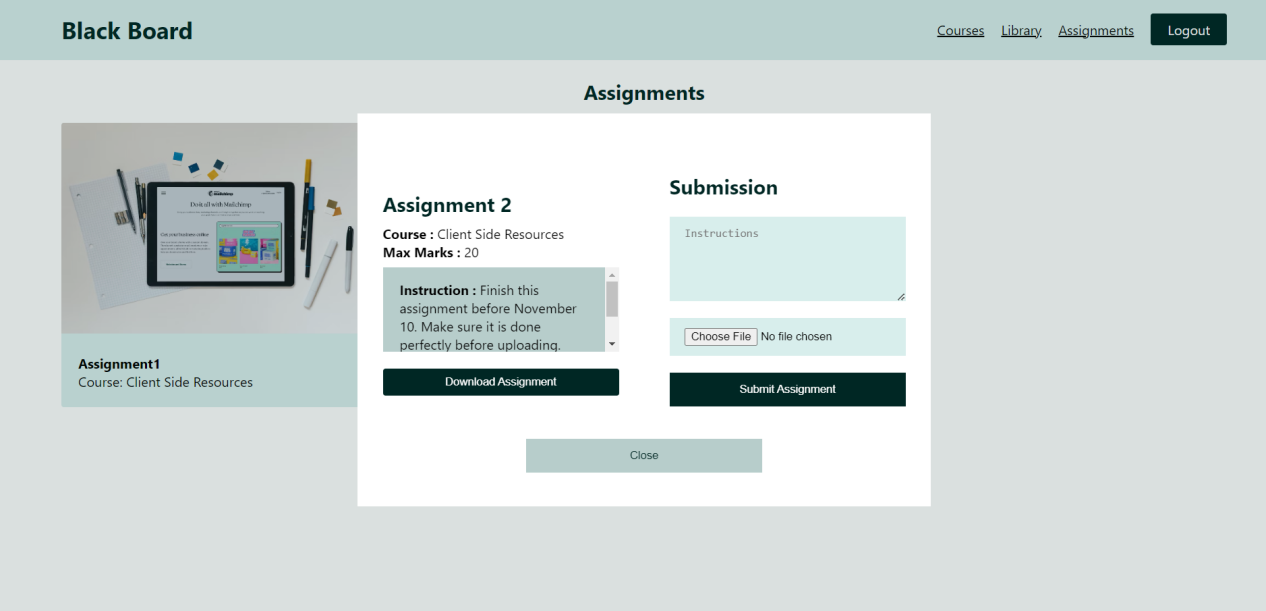
Below is the screen, that displays the list of books related to courses that the student has enrolled.



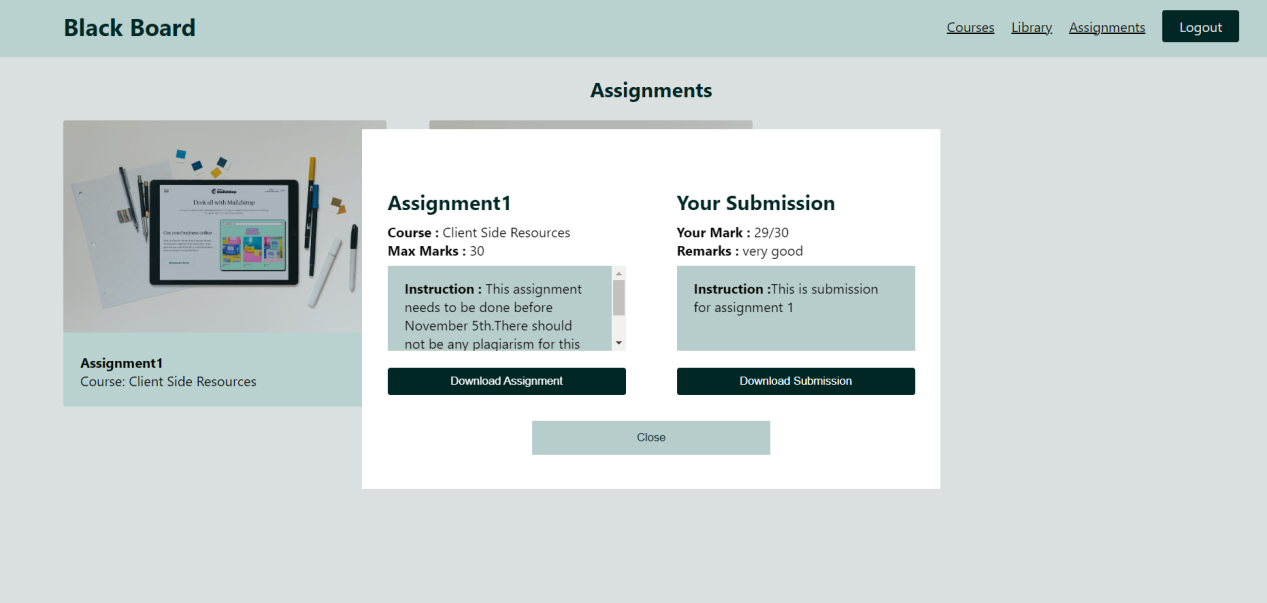
Below is the screen, that displays the list of assignments related to courses that the student has enrolled.



On click of Assignment , he will see the information about the assignment, will be able to download the assignment, and has an option to upload the assignment.

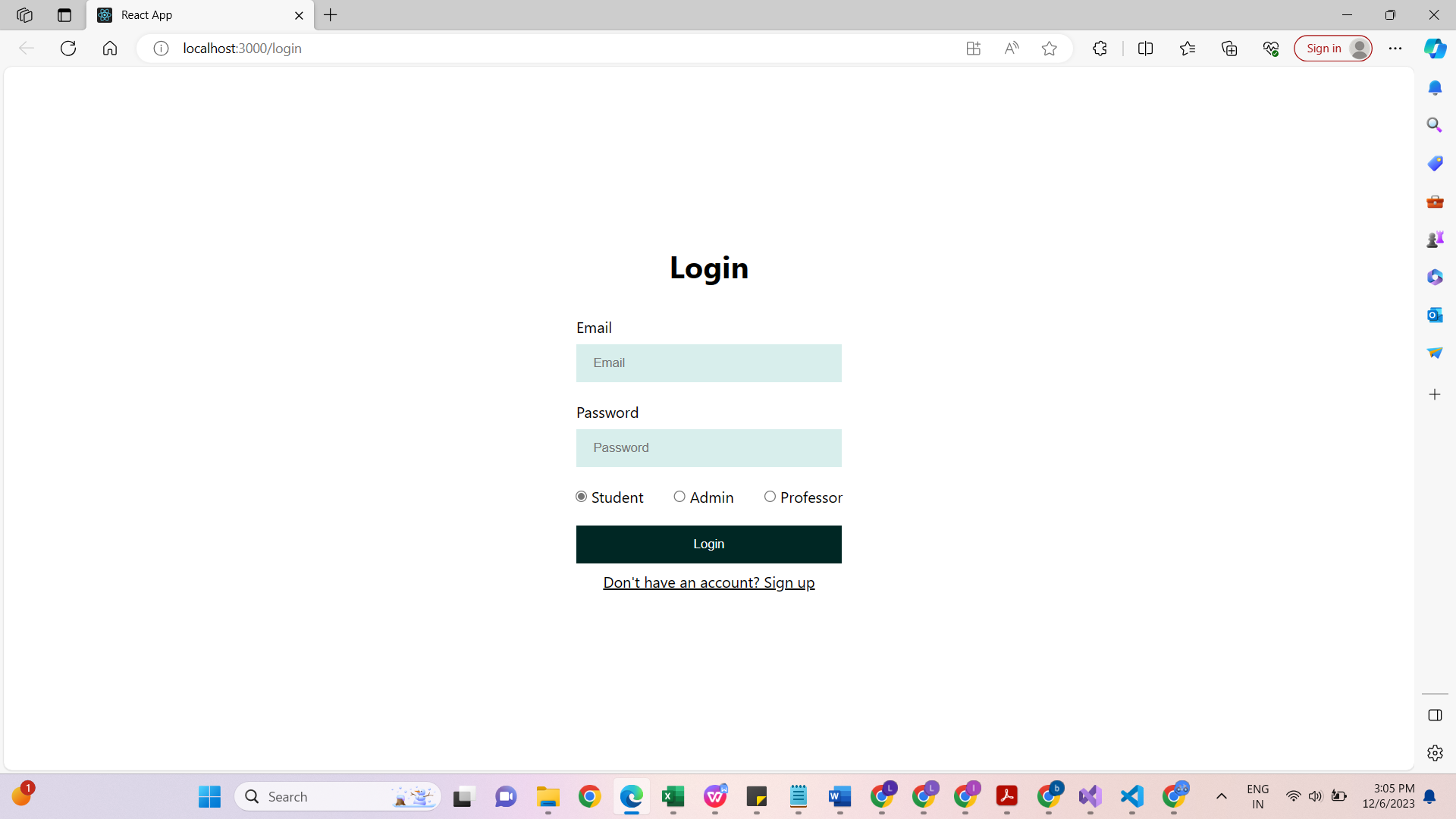


If the assignment is graded by admin, then the below UI is displayed with marks and remarks for that particular assignment.

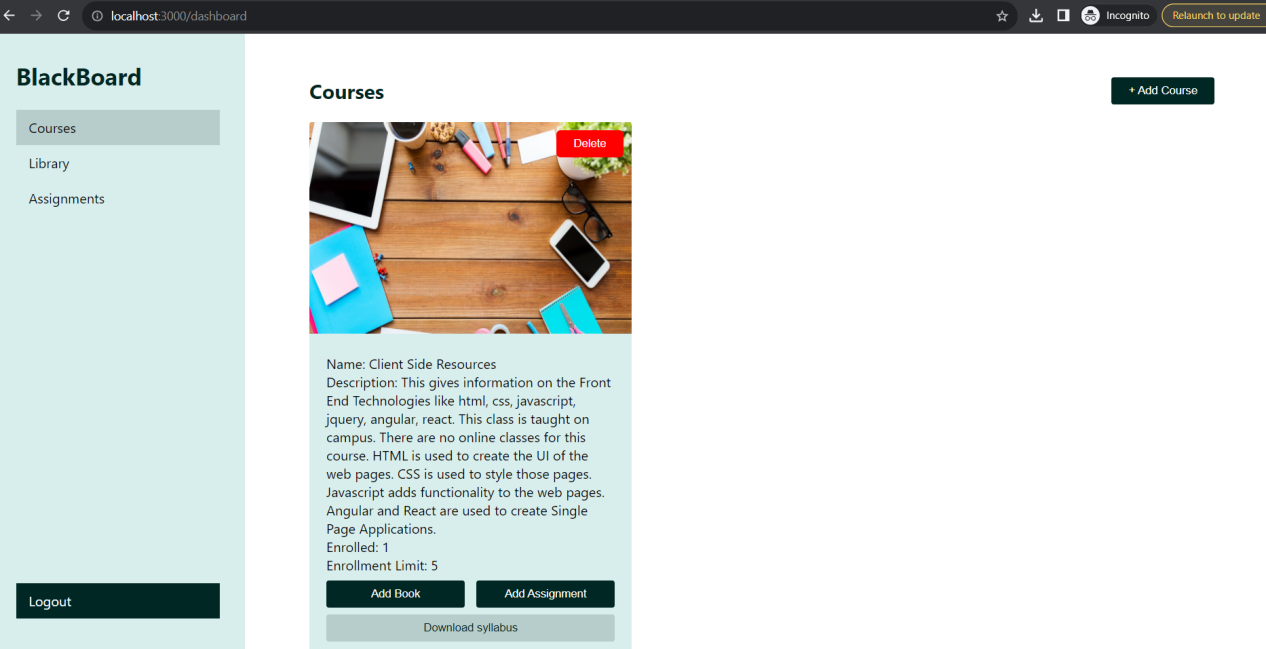


Professor Flow

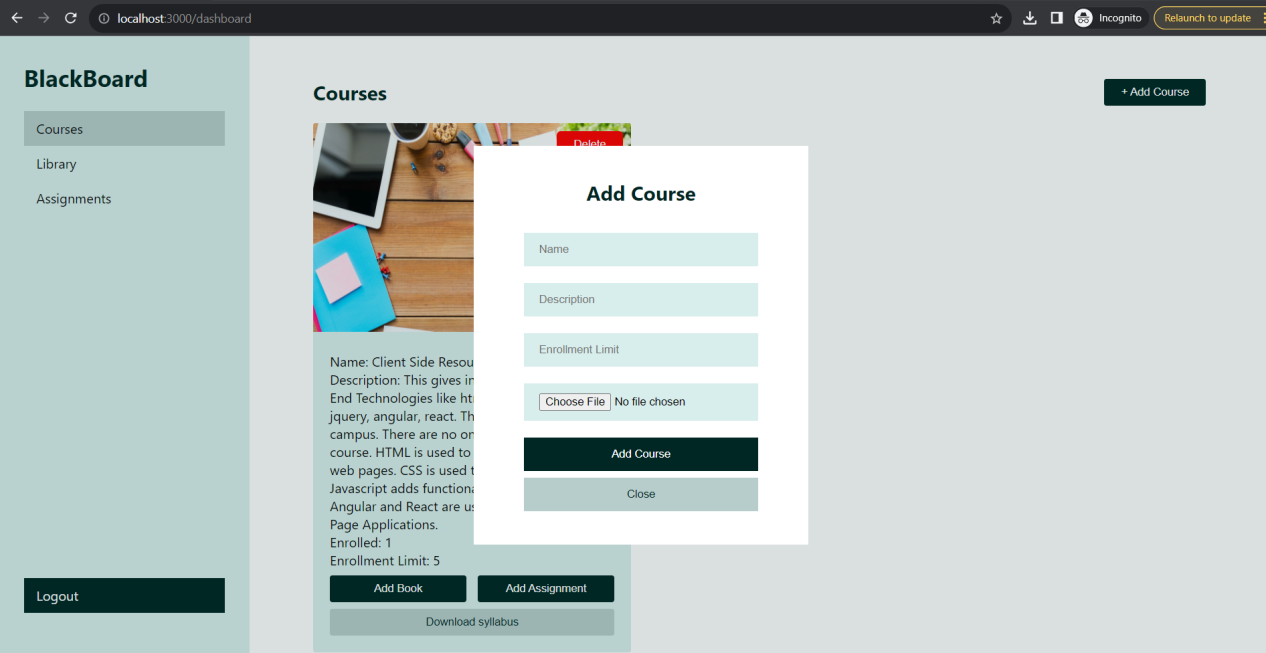
The Professor login page looks like below.



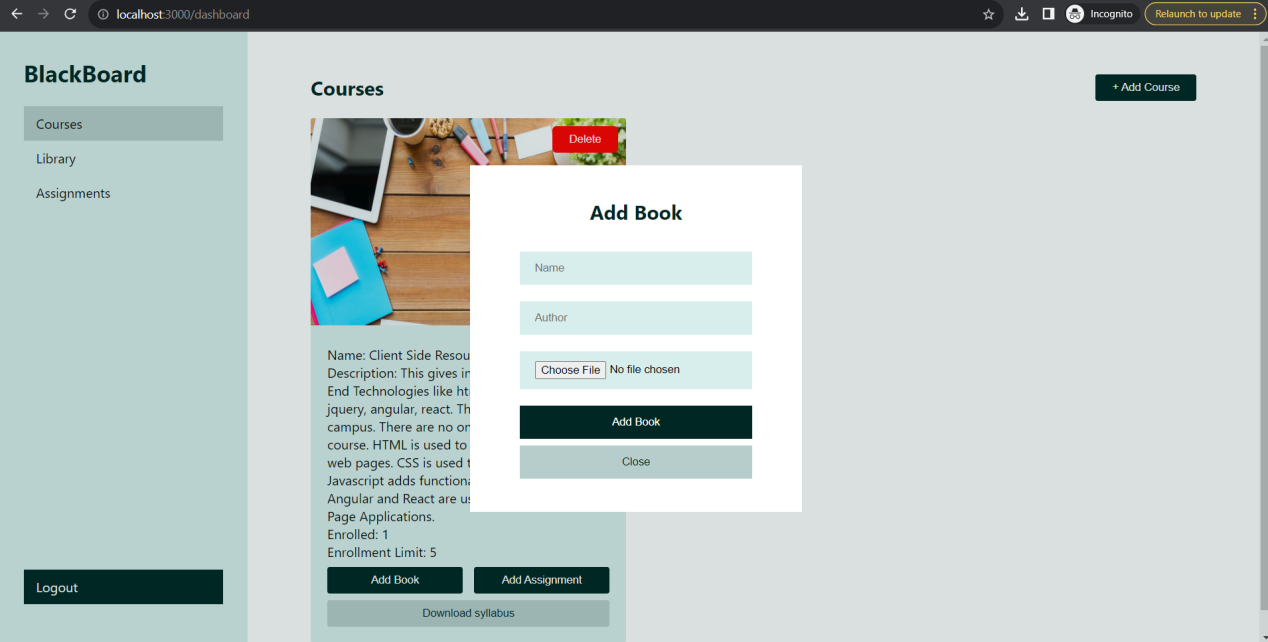
The below page is displayed when we login with Professor Credentials.

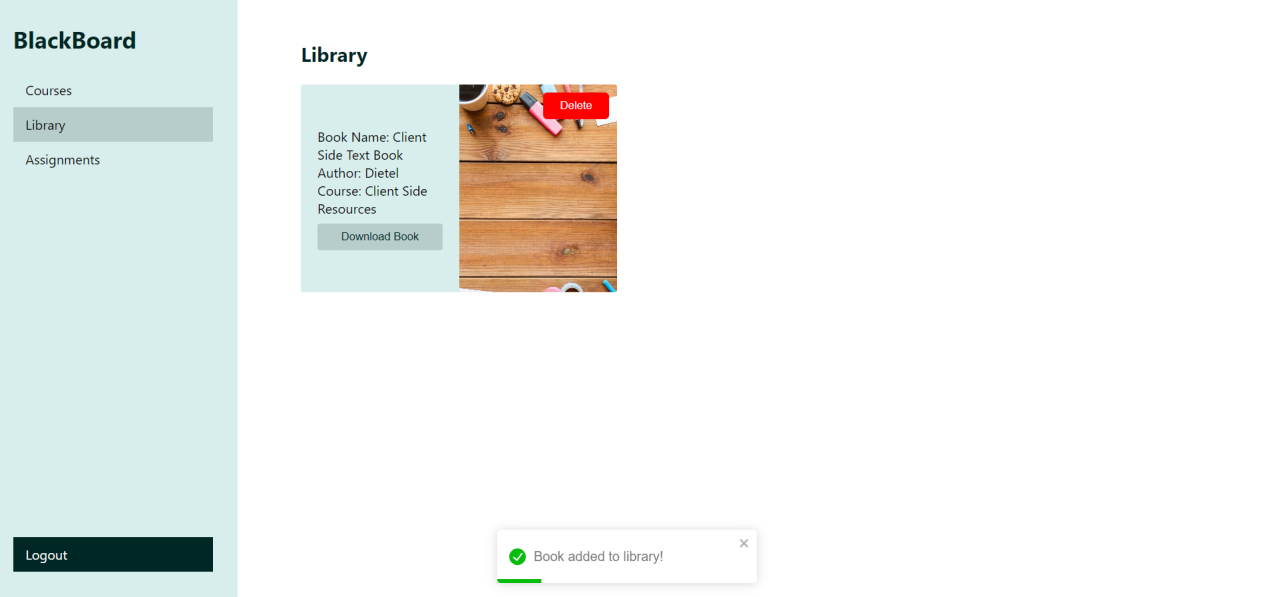


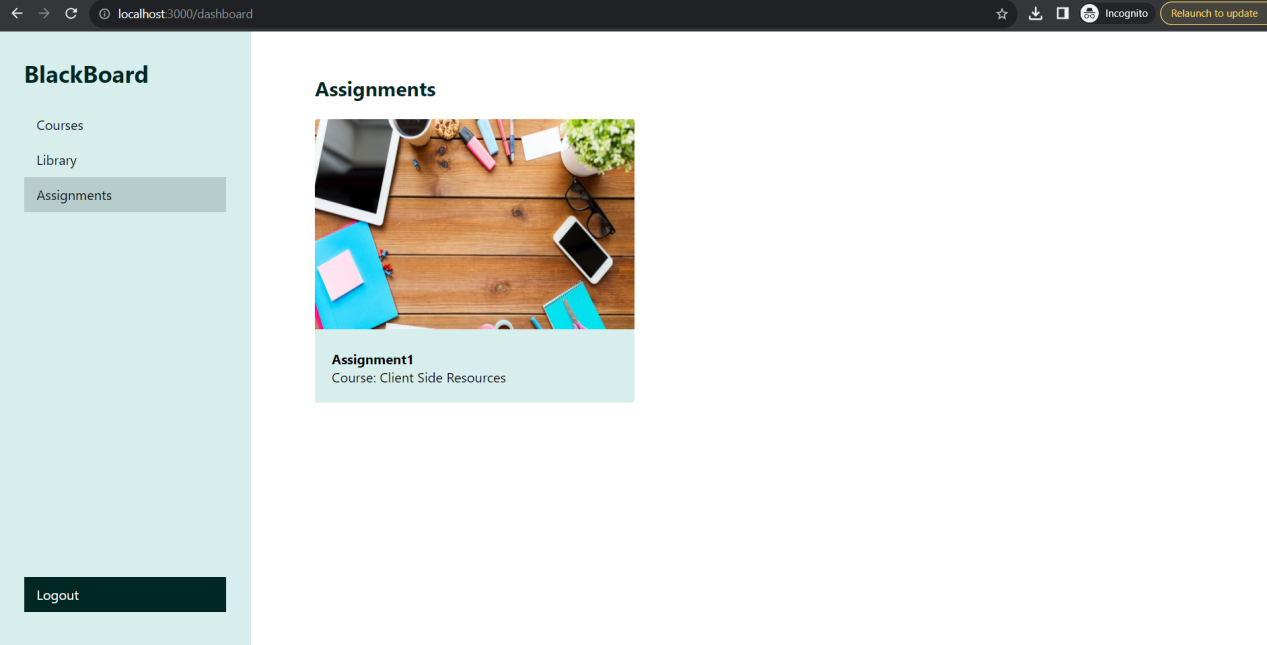
The professor can create Courses in which the student can enroll.



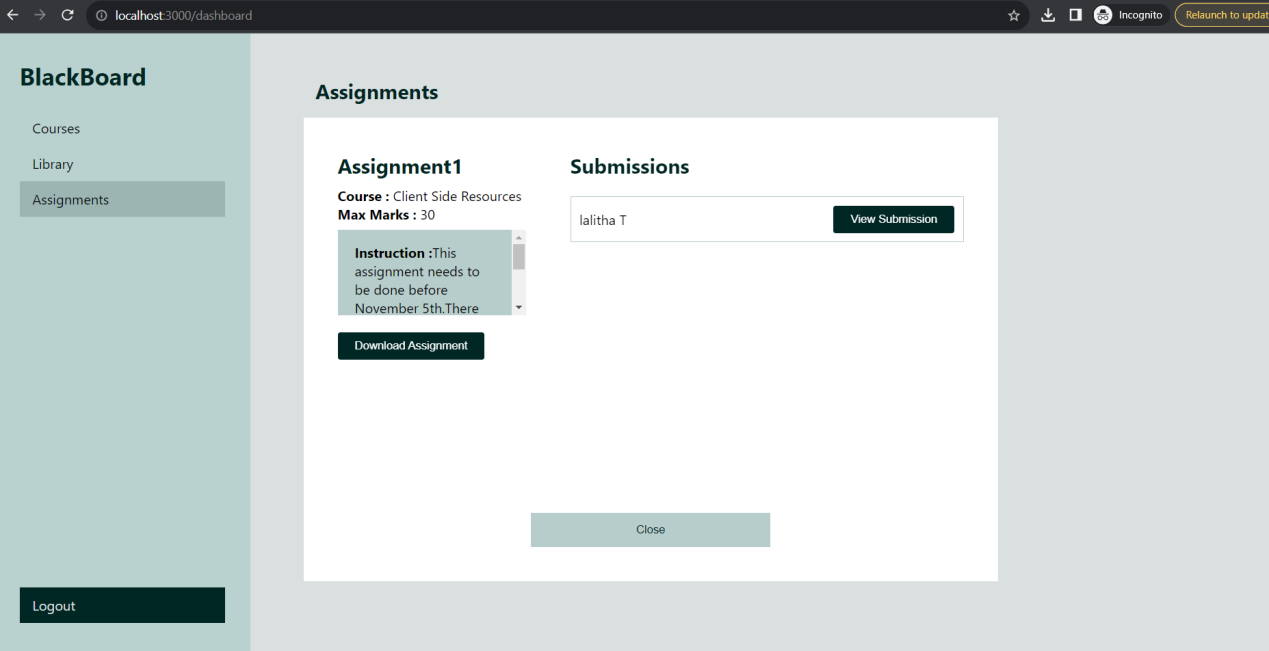
Once he creates the courses, he can then add books, assignments related to those courses .



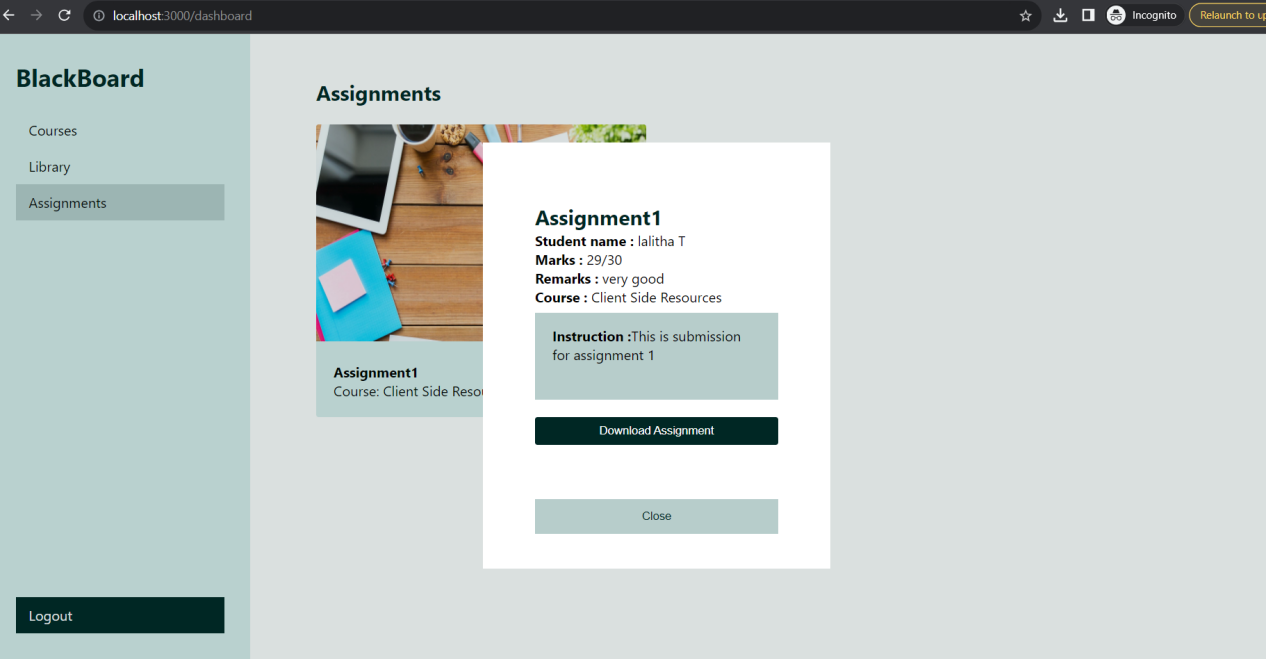




When he clicks on the particular assignment, he can see the information about the assignment and see how many students have submitted the assignment.



The professor then clicks on View Submission button to see the submissions made by the students. In this page he can even download the assignment and grade it and upload the marks in the portal.



**SOFTWARE REQUIREMENTS:**

Operating system: Windows, 7, 8, 10, 11, MAC IOS.

Cloud environment: Amazon Web Services.

Back End Languages & Frameworks: Node JS, Express JS

Front End Languages & Frameworks: HTML, CSS, React, Redux Toolkit,

Database: MongoDB