**Learning Management System**

**A Project Report**

**Submitted by:**

**Rakshit R Soni**

|  |  |
| --- | --- |
|  |  |

**Table Of Contents**

|  |  |  |
| --- | --- | --- |
| **Sr. No** | **Contents** | **Page No.** |
| **1.** | **Acknowledgement** | **2** |
| **2.** | **Introduction** | **5** |
| **3.** | **Existing system** | **6** |
| **4.** | **Need for the new system** | **7** |
| **5.** | **Scope for the new system** | **8** |
| **6.** | **Modules** | **9** |
| **7.** | **Tools & Technology** | **10** |
| **8.** | **System Flow Diagram** | **11** |
| **9.** | **UML Diagrams : - Class Diagram** | **12** |
| **10.** | **Activity Diagram** | **13** |
| **11.** | **Use Case Diagram** | **15** |
| **12.** | **Sequence Diagram** | **17** |
| **13.** | **Data Dictionary** | **19** |
| **14.** | **Student Interface** | **25** |
| **15.** | **Instructor Interface** | **31** |
| **16** | **Test Plan and Test Cases** | **35** |
| **17** | **Future Enhancement** | **37** |
| **18** | **Conclusion** | **38** |
| **19** | **References / Bibliography** | **39** |

**Introduction**

A learning management system (LMS) is a software application or web-basedtechnology used to plan, implement and assess a specific learning process. In its most common form, an LMS consists of a server that performs the base functionality anda user interface .

LMS’es have a range of use cases in education institutions and companies. They're primarily used for [knowledge management](https://www.techtarget.com/searchcontentmanagement/definition/knowledge-management-KM): the gathering, organizing, sharing and analysis of an organization's knowledge in terms of resources, documents and people skills. The role of the LMS varies according to the organization's training strategy and goals.

An LMS is a large repository where users store and track information in one place. Any user with a login and password can access the system and its online learning resources. If the system is self-hosted, the user must either install the software on their computer or access it through their company's server.

**Existing System**

An existing Learning Management System (LMS) is a software platform used by schools, universities, or organizations to manage, deliver, and track educational content and training. It allows instructors to create courses, upload materials, administer assessments, and communicate with students. Learners can access course content, submit assignments, track their progress, and interact with peers and instructors, all in one platform.

**Some key features of an LMS include:**

1. **Course creation and management**
2. **Tracking learner progress and performance**
3. **Discussion boards and communication tools**
4. **Review of courses**
5. **User management (instructors, students)**

**Need of the system**

The need for an improved LMS includes:

* **Better User Experience** – Easier navigation.
* **Customization** – Tailored to specific needs.
* **Integration** – Seamless connection with other tools.
* **Mobile Access** – Learning anytime, anywhere.
* **Scalability** – Handling large user numbers.
* **Advanced Analytics** – Better tracking and reporting.
* **Collaboration** – Real-time interaction and group work

**Scope of system**

**The scope of an LMS system includes:**

* **Course Video Access –** Allow students to watch course videos.
* **User Management –** Manage students and instructors**.**
* **Tracking Progress** – Monitor students’ course completion status.
* **Basic Analytics –** Provide insights into video engagement and completion rates.
* **Review System** – Enable students to submit, edit, and delete course reviews.
* **Mobile Access-** Ensure a responsive, mobile-friendly interface.
* **Security and Privacy –** Protect user data and maintain compliance with security standards.

**Modules**

**Instructor** :

* **Login**
* **Add Course**
* **Remove Course**
* **View Student**
* **View Payment**
* **Reply Reviews**
* **Logout**

**Student:**

* **Register/Login**
* **View Course**
* **Add To Cart**
* **View Course** **Progress**
* **View Payment**
* **Review**
* **Logout**

**Tools and technology**

Development Tool - Visual studio code

**Web Server** - Localhost

**Front End** - React (18.x),HTML,CSS (Tailwind CSS),JavaScript, Tailwind

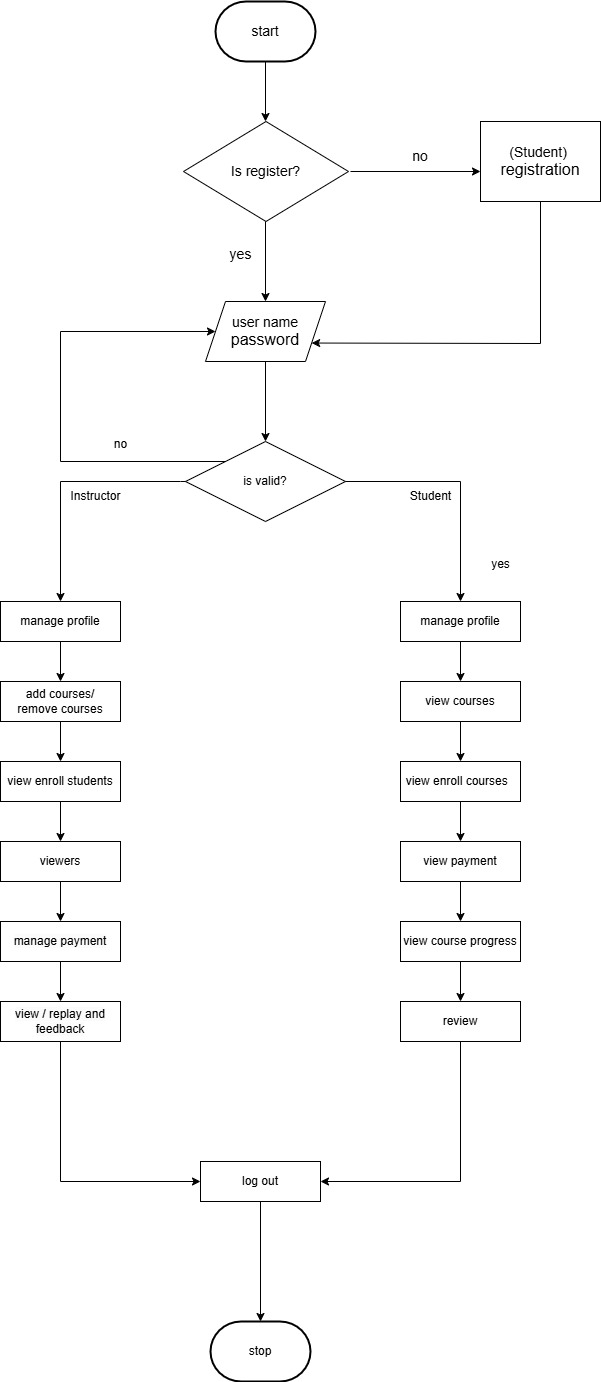
**Back End** - Node.js (18.x or 20.x),Express.js (4.x), MongoDB (6.x)

**Components** – ui.shadcn

**Documentation Tools** - MS Word, Google Docs

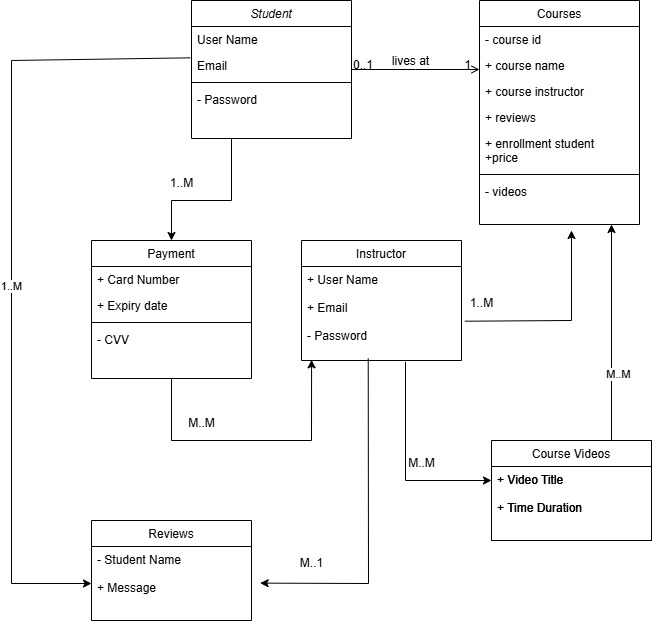
**Drawing Tools** - Draw.io

**System Flow Diagram**

****

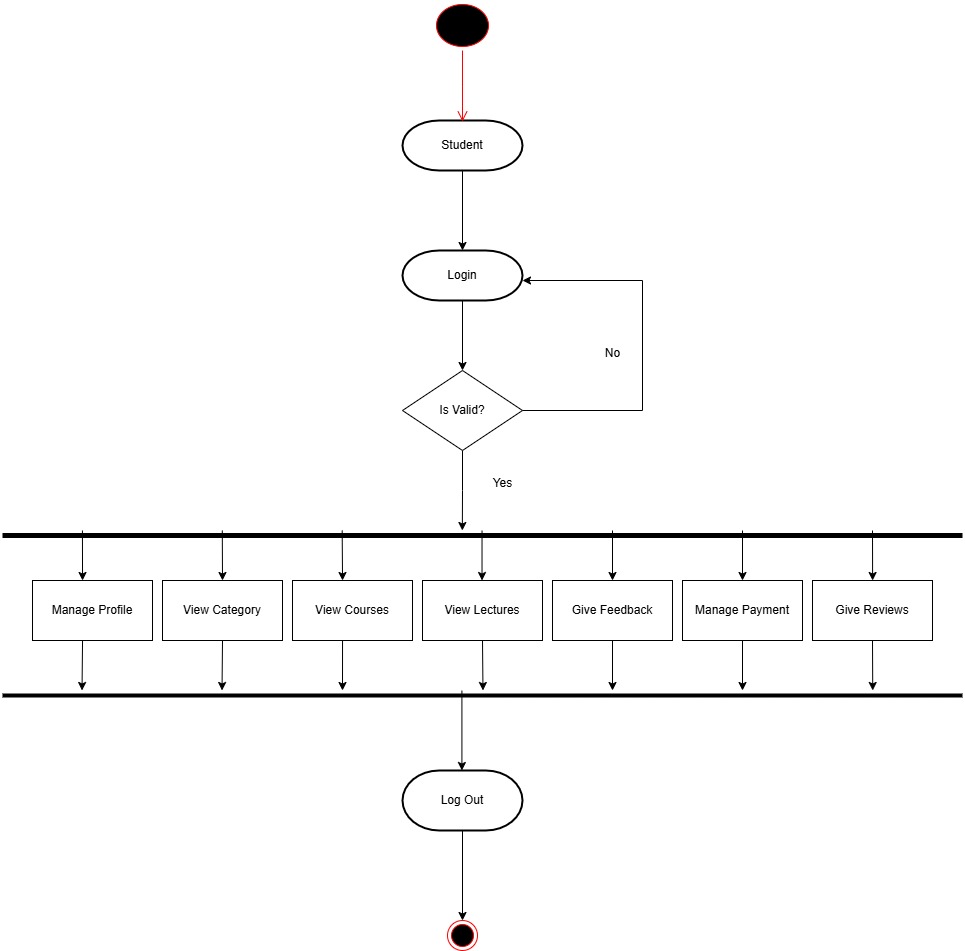
**UML Diagram**

**Class Diagram:-**

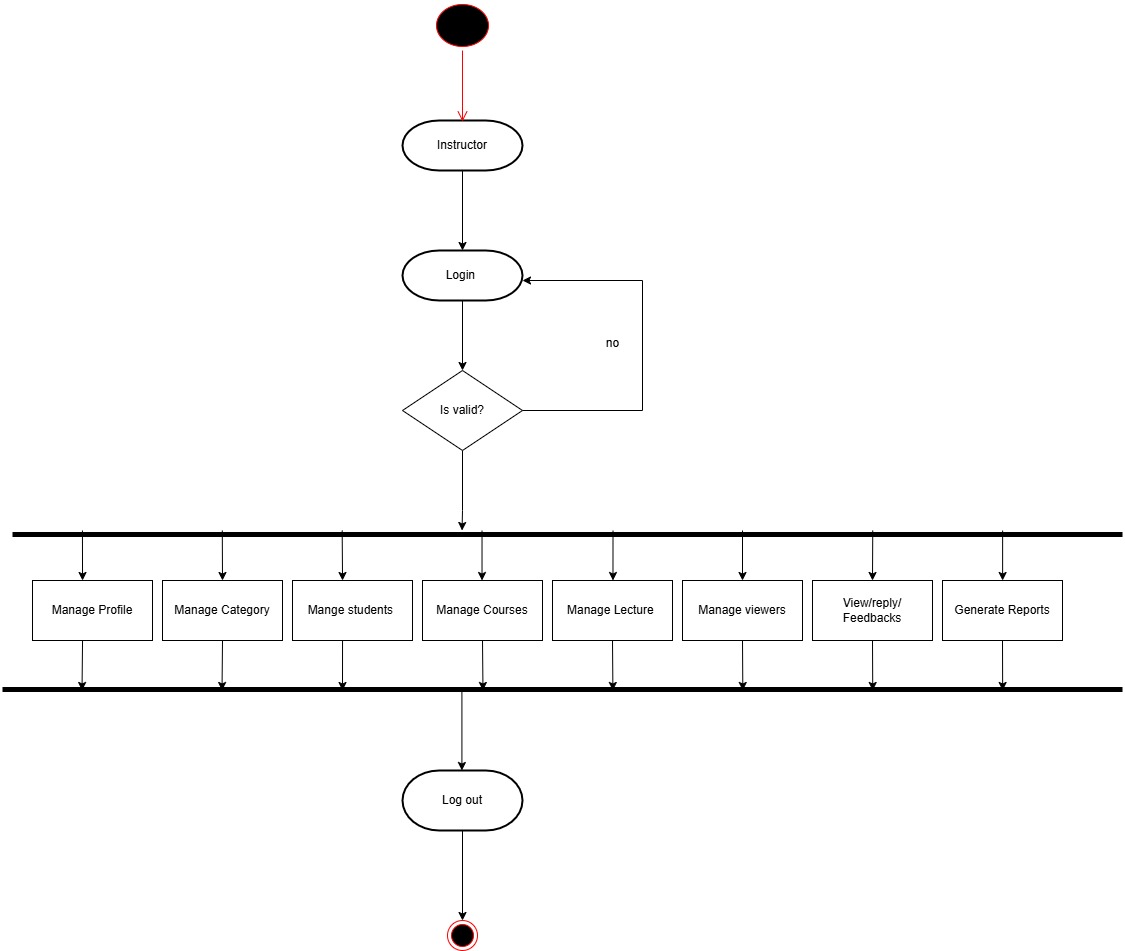
****

**Activity Diagram**

**1.Student Activity Diagram:**

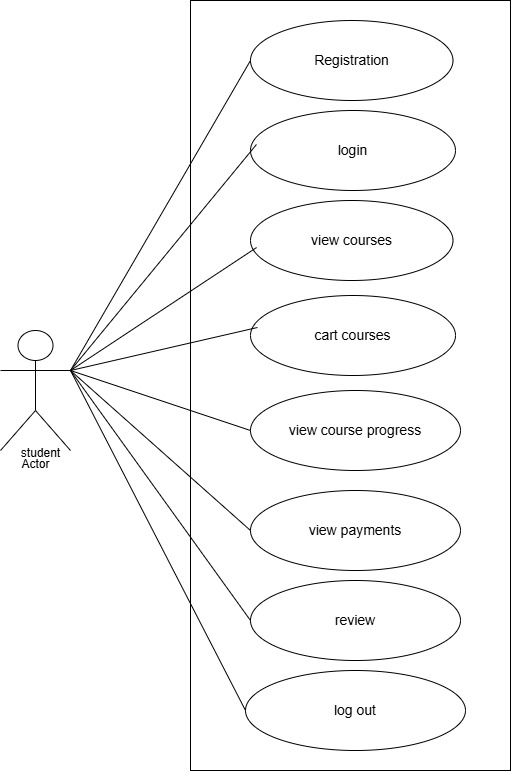
****

**2.Instructor Activity Diagram:**

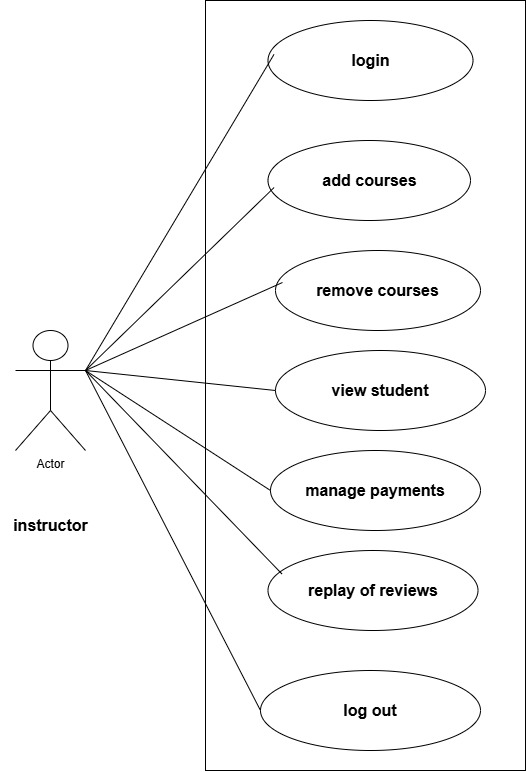
****

Use Case Diagram

1.student:

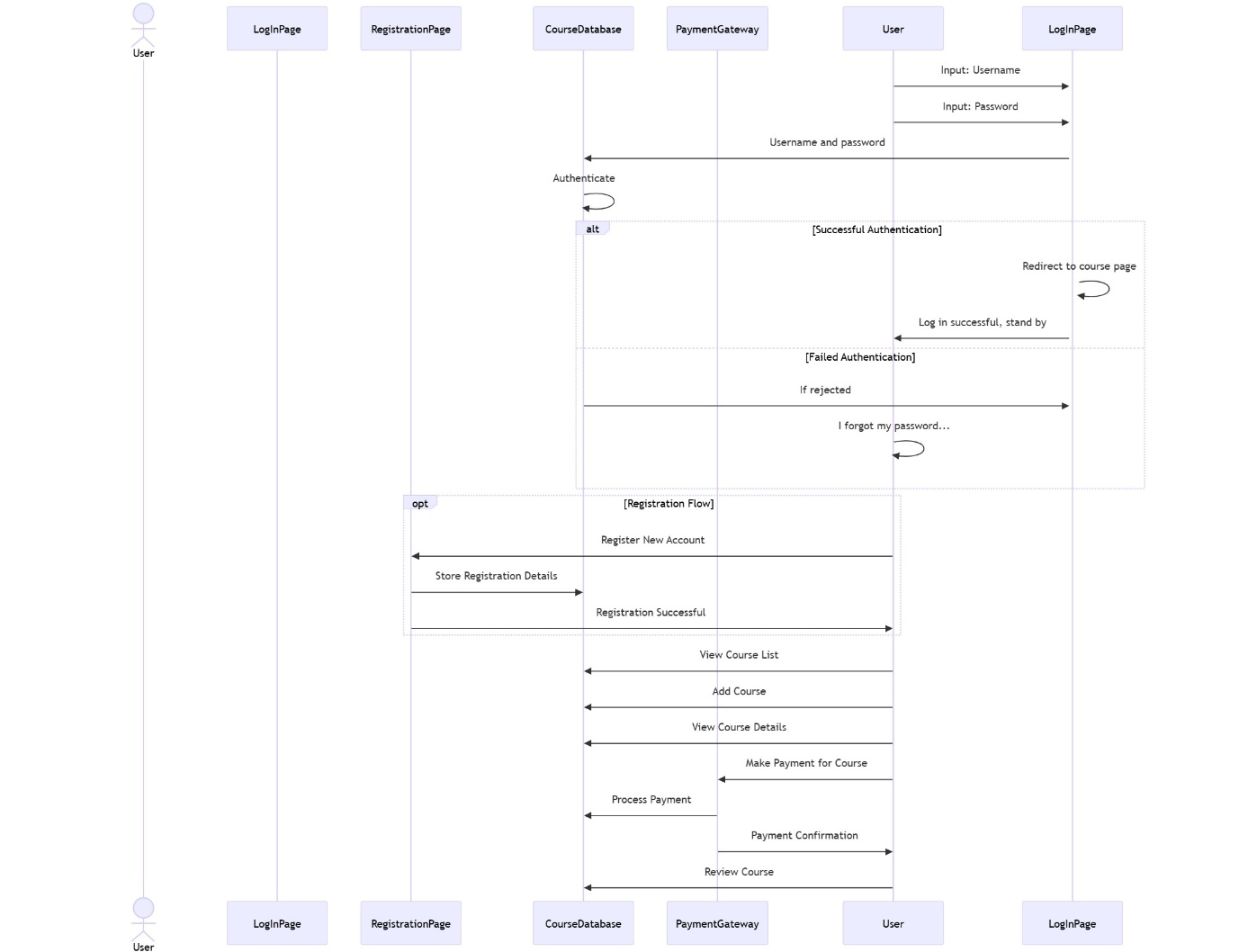


2.instructor

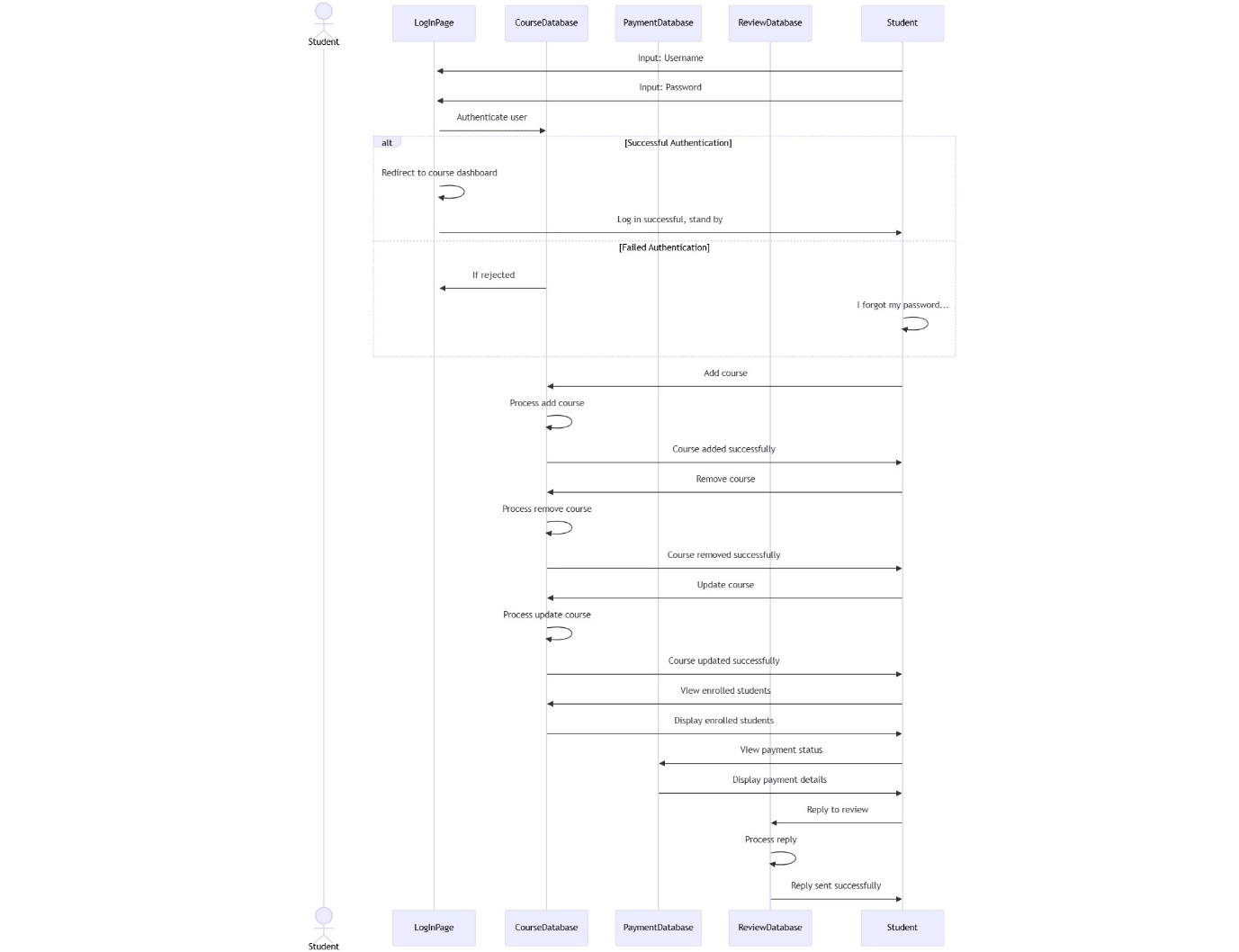


**Sequence Diagram**

**1 Student**



**2 instructor:**



**Data Dictionary**

1 Student/ instructor:

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Description** |
| Name | String | Full name of the user |
| Email | String (Email) | User's email address |
| Password | String (Hashed) | Encrypted password stored securely (hashed using bcrypt) |
| Role | String | Role of the user (student, instructor, or admin) |
| enrolledCourses | Array of ObjectIds | List of courses the user is enrolled in |
| photoUrl | String (URL) | URL of the user's profile picture |
| createdAt | Date | Date the user account was created |
| updatedAt | Date | Date the user data was last updated |
| \_\_v | Number | Version key (used internally by Mongoose) |

2 Courses:

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Description** |
| courseTitle | String | Title of the course |
| Category | String | Course category |
| courseThumbnail | String (URL) | Thumbnail image URL |
| enrolledStudents | Array of ObjectIds | List of students enrolled |
| Lectures | Array of ObjectIds | List of lecture IDs |
| Creator | ObjectId | Creator (instructor) ID |
| isPublished | Boolean | Whether course is published or not |
| createdAt | Date | Date when course was created |
| updatedAt | Date | Date of last course update |
| \_\_v | Number | Version key (used internally by Mongoose) |
| courseLevel | String | Difficulty level (e.g., Beginner) |
| coursePrice | Number | Course price in currency units |
| Description | String (HTML) | Course description |
| subtitle | String | Short subtitle or summary |

**3 Lectures:**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Description** |
| lectureTitle | String | Title of the lecture |
| createdAt | Date | Timestamp when the lecture was created |
| updatedAt | Date | Timestamp of the last update to the lecture |
| \_\_v | Number | Version key (used internally by Mongoose) |
| isPreviewFree | Boolean | Indicates if the lecture is available as a free preview |
| publicId | String | Cloudinary public ID of the uploaded video |
| videoUrl | String (URL) | URL of the video hosted on Cloudinary |

**4**)**course purchases:**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Description** |
| courseId | ObjectId | Reference to the course being purchased |
| userId | ObjectId | Reference to the user who made the purchase |
| amount | Number | Amount paid for the course |
| status | String | Payment status (e.g., pending, completed) |
| paymentId | String | Unique payment transaction ID |
| createdAt | Date | Timestamp when the payment record was created |
| updatedAt | Date | Timestamp of the last update to the payment record |
| \_\_v | Number | Version key (used internally by Mongoose) |

**5) course progress:-**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Description** |
| userId | ObjectId (String) | ID of the student/user tracking their progress |
| coursed | ObjectId (String) | ID of the course being tracked |
| Completed | Boolean | Indicates whether the student has completed the course |
| lectureProgress | Array of Objects | Tracks progress for individual lectures |
| lectureProgress[].lectureId | ObjectId | ID of the lecture |
| lectureProgress[].viewed | Boolean | Indicates whether the lecture was viewed |
| \_\_v | Number | Version key (used internally by Mongoose) |

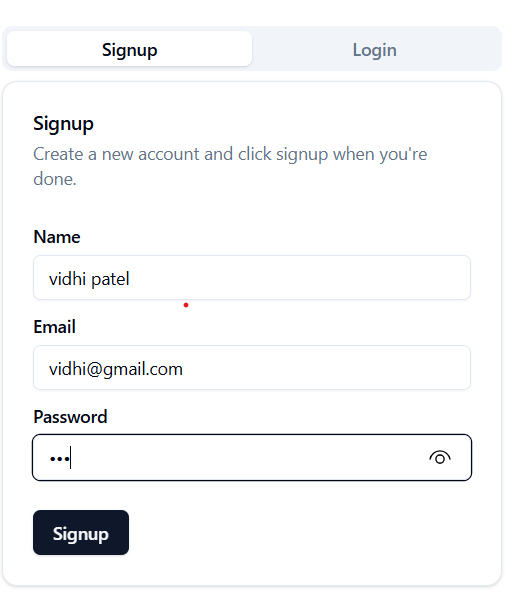
**6)Reviews:**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Description** |
| course | ObjectId | ID of the reviewed course |
| instructor | ObjectId | ID of the instructor of the course |
| student | ObjectId | ID of the student who submitted the review |
| rating | Number (1–5) | Rating given to the course |
| comment | String | Textual feedback from the student |
| reply | String | Reply from the instructor to the review (optional) |
| createdAt | Date | Date and time when the review was submitted |
| updatedAt | Date | Last updated timestamp for the review or reply |
| \_\_v | Number | Version key (used internally by Mongoose) |

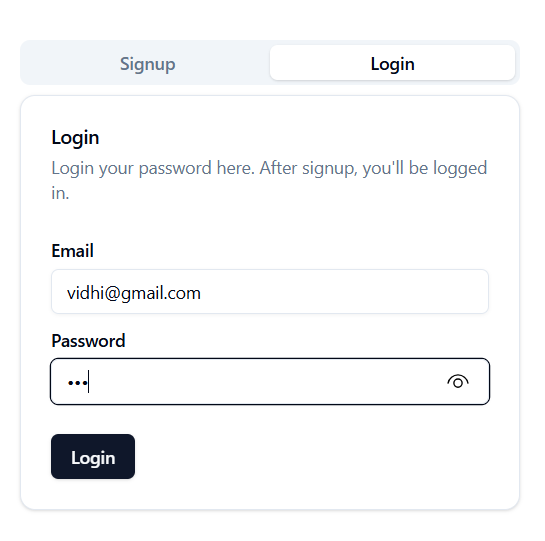
**Screenshots**

**1 Student**

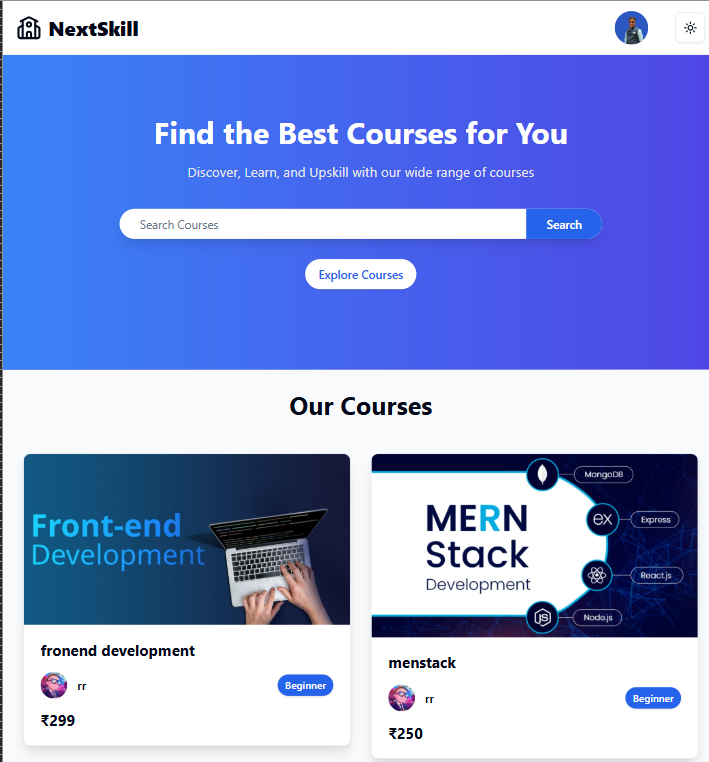
Student registration:



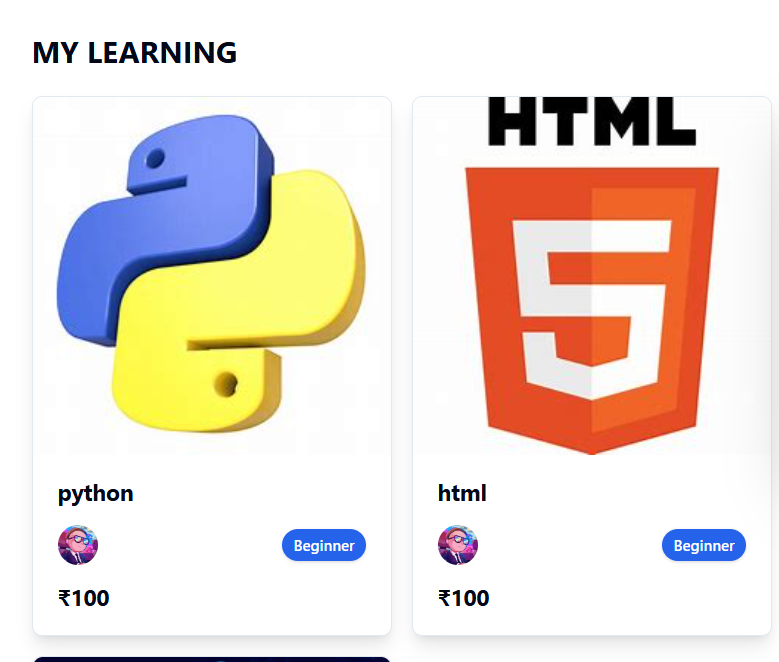
Student login:



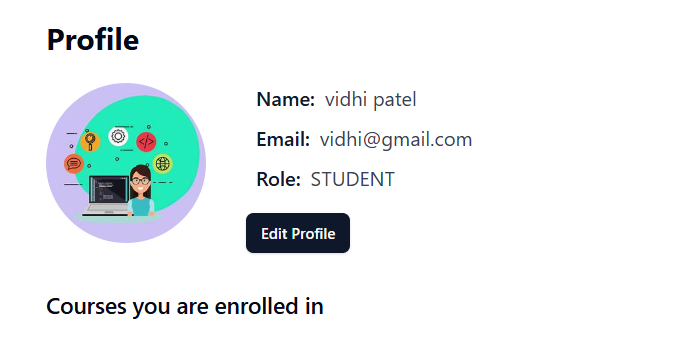
Homepage:



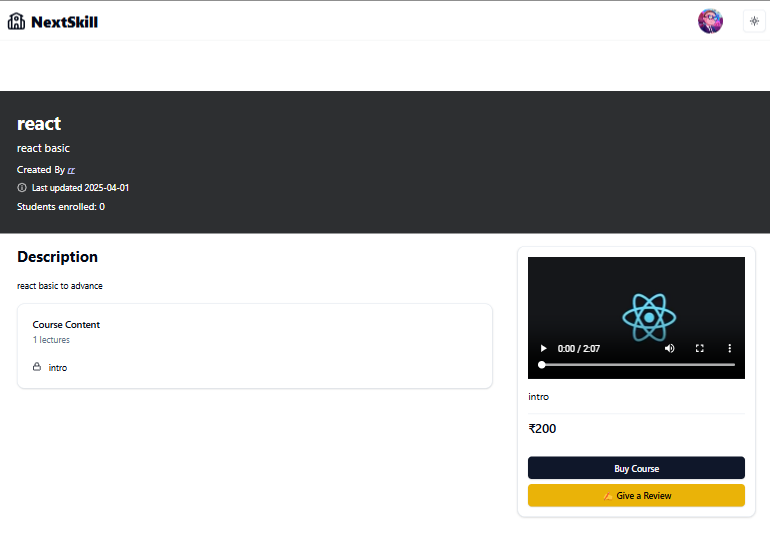
In account section also shows the enrollcourses

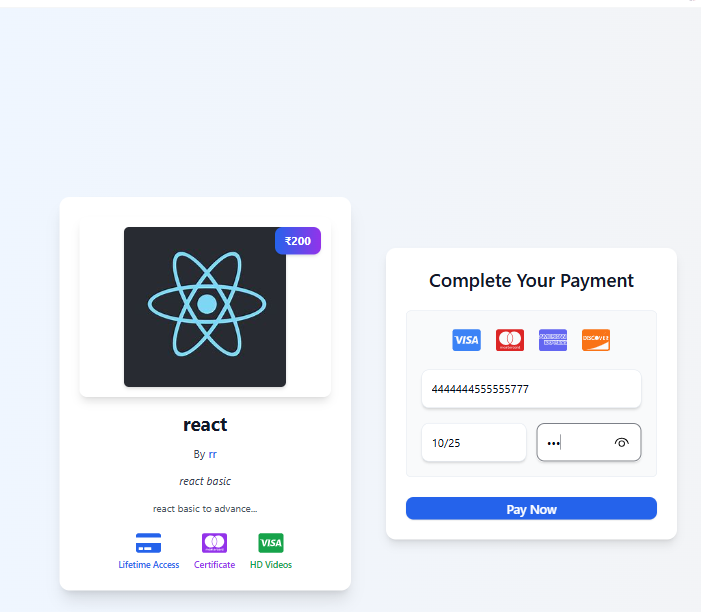


Profile

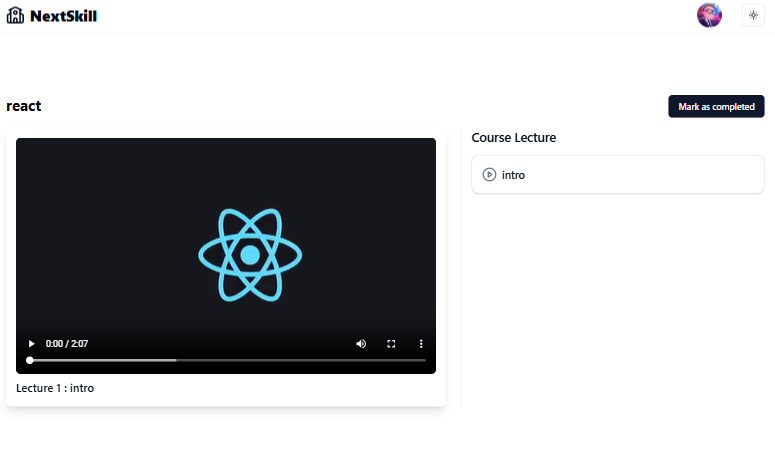


Course details page:-

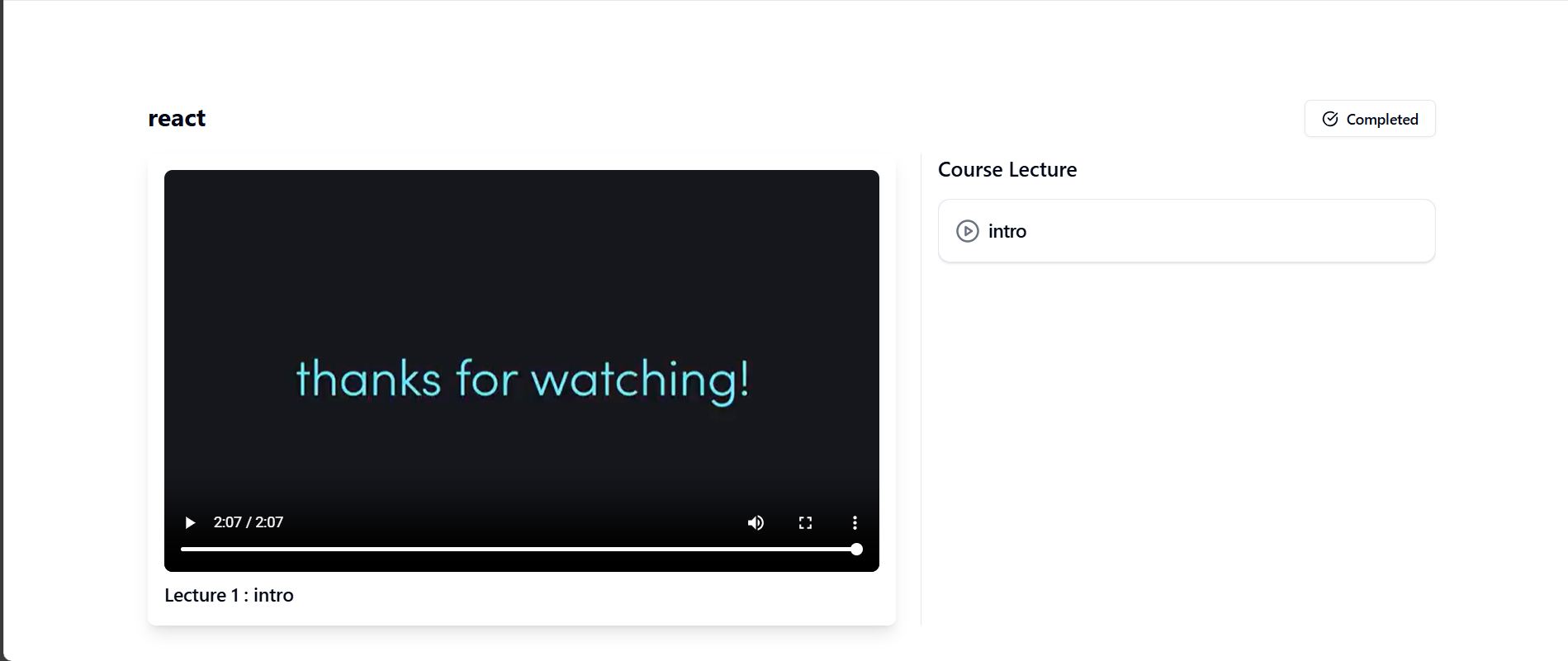


Course payment : 

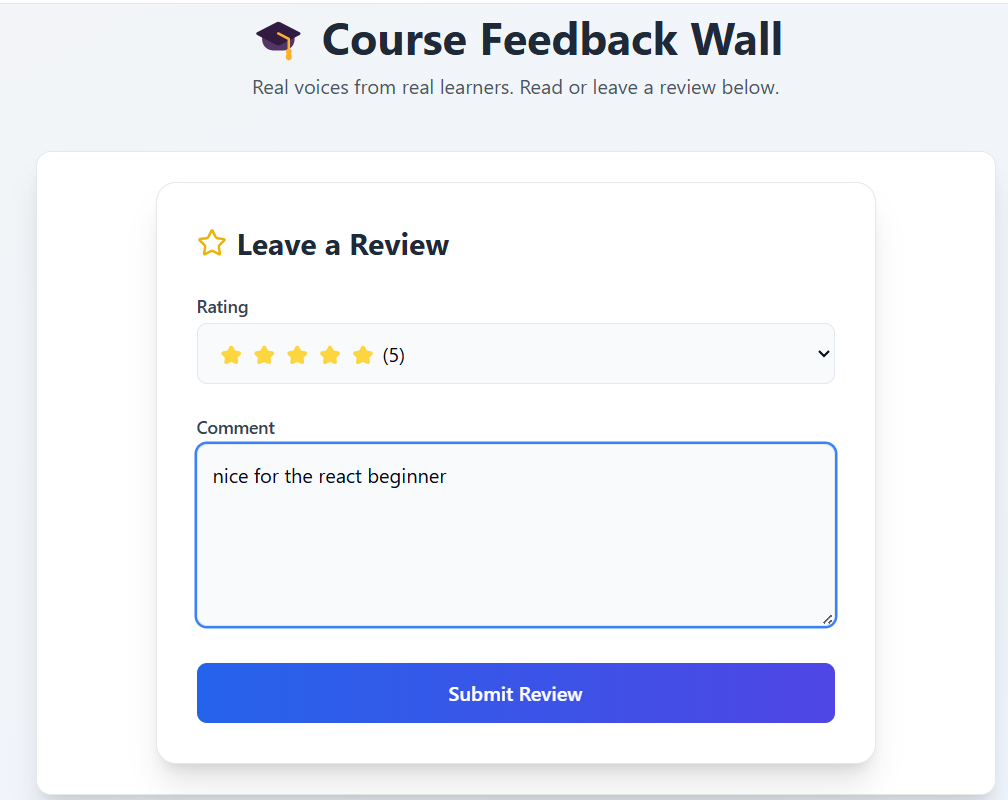
complete the payment course payment:



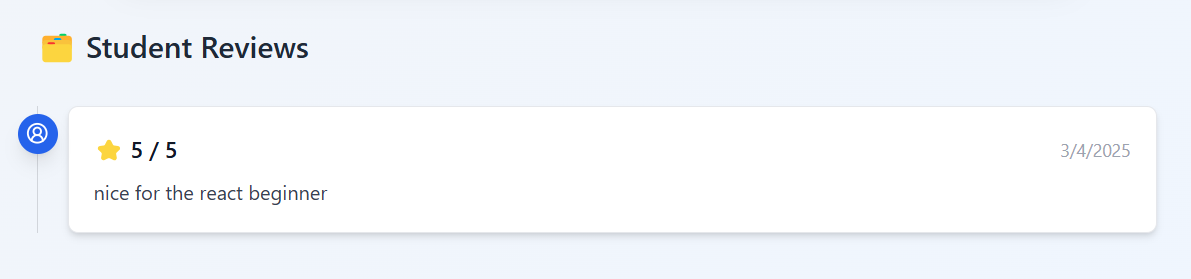
Complete the payment



Review of course

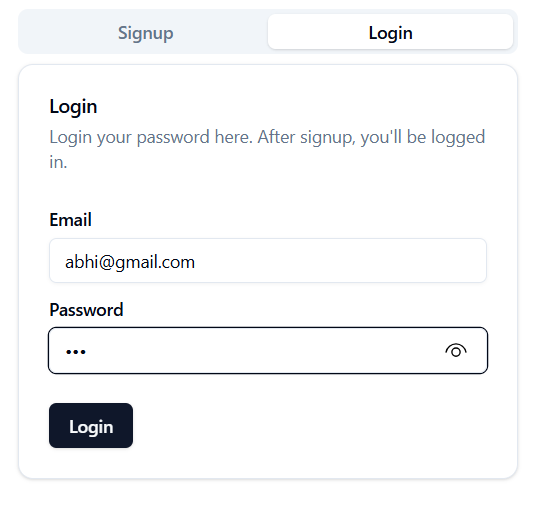


Submit after the also show their:



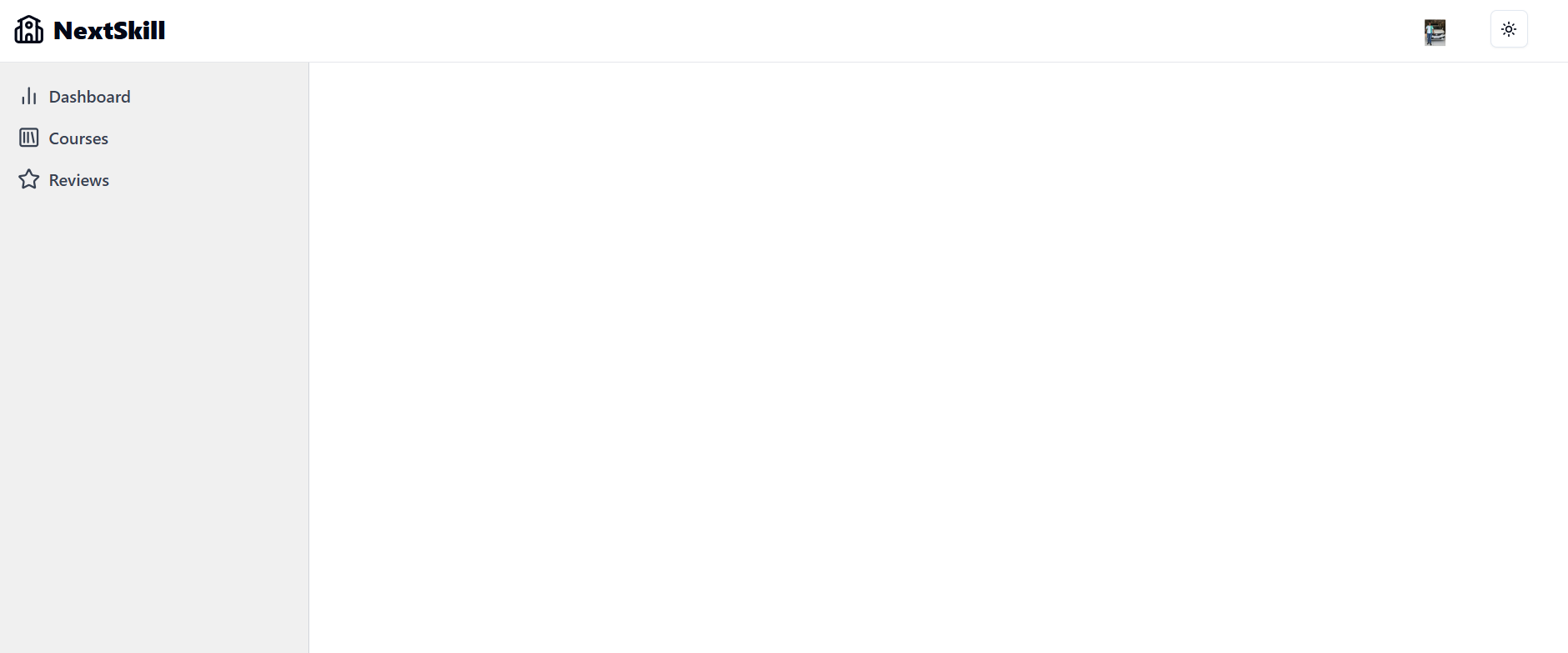
**2 Instructor:**

**Instructor login**

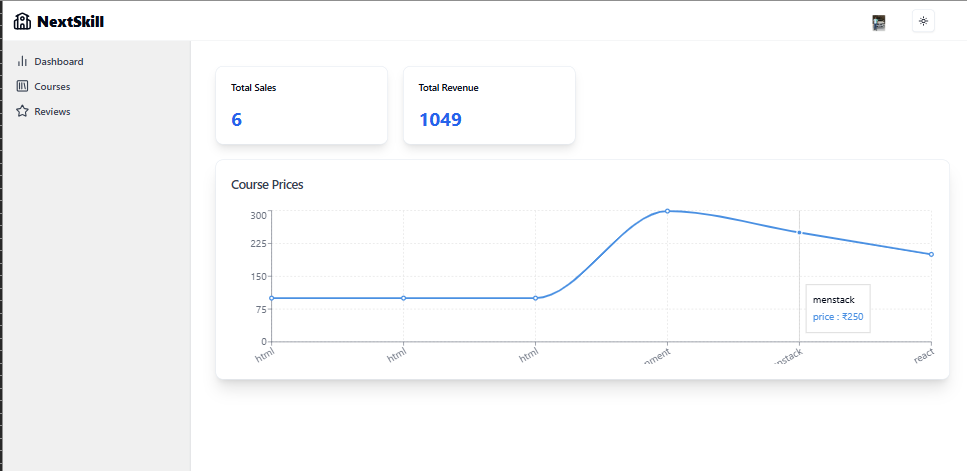


In profile :

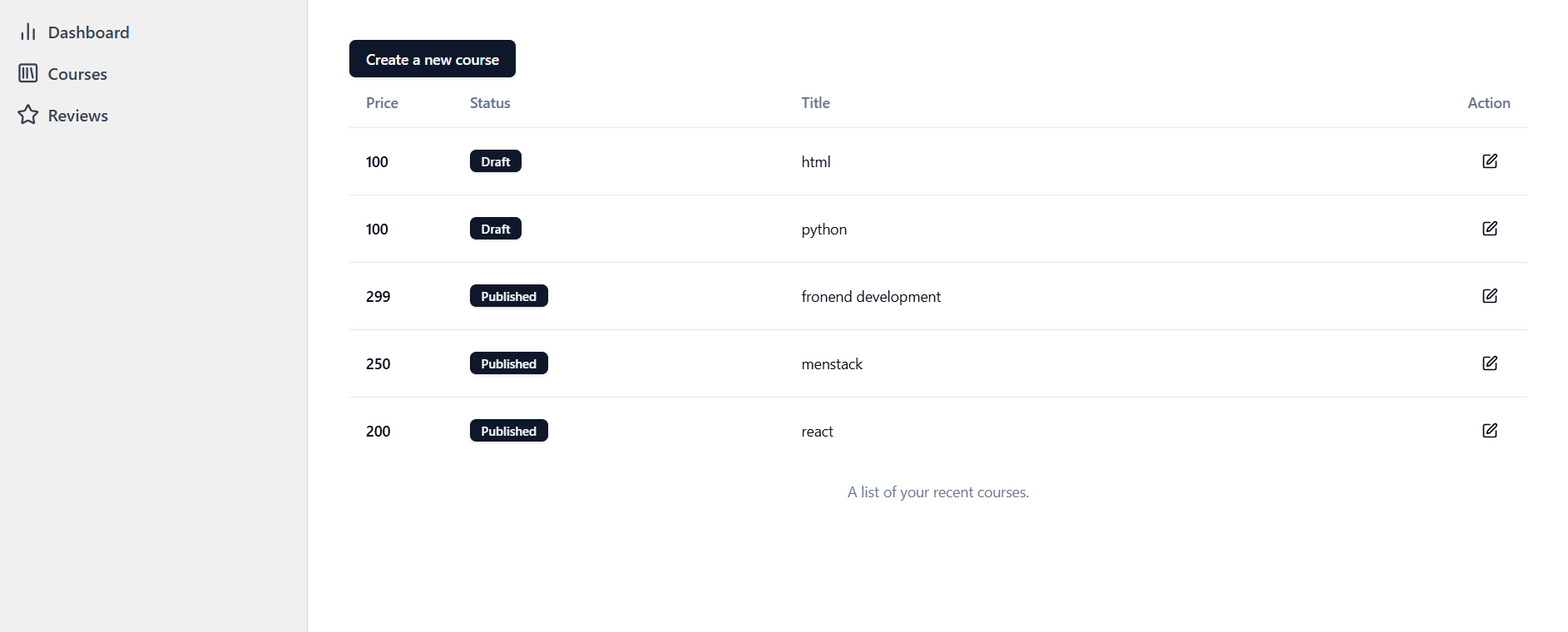


Dashboard : 

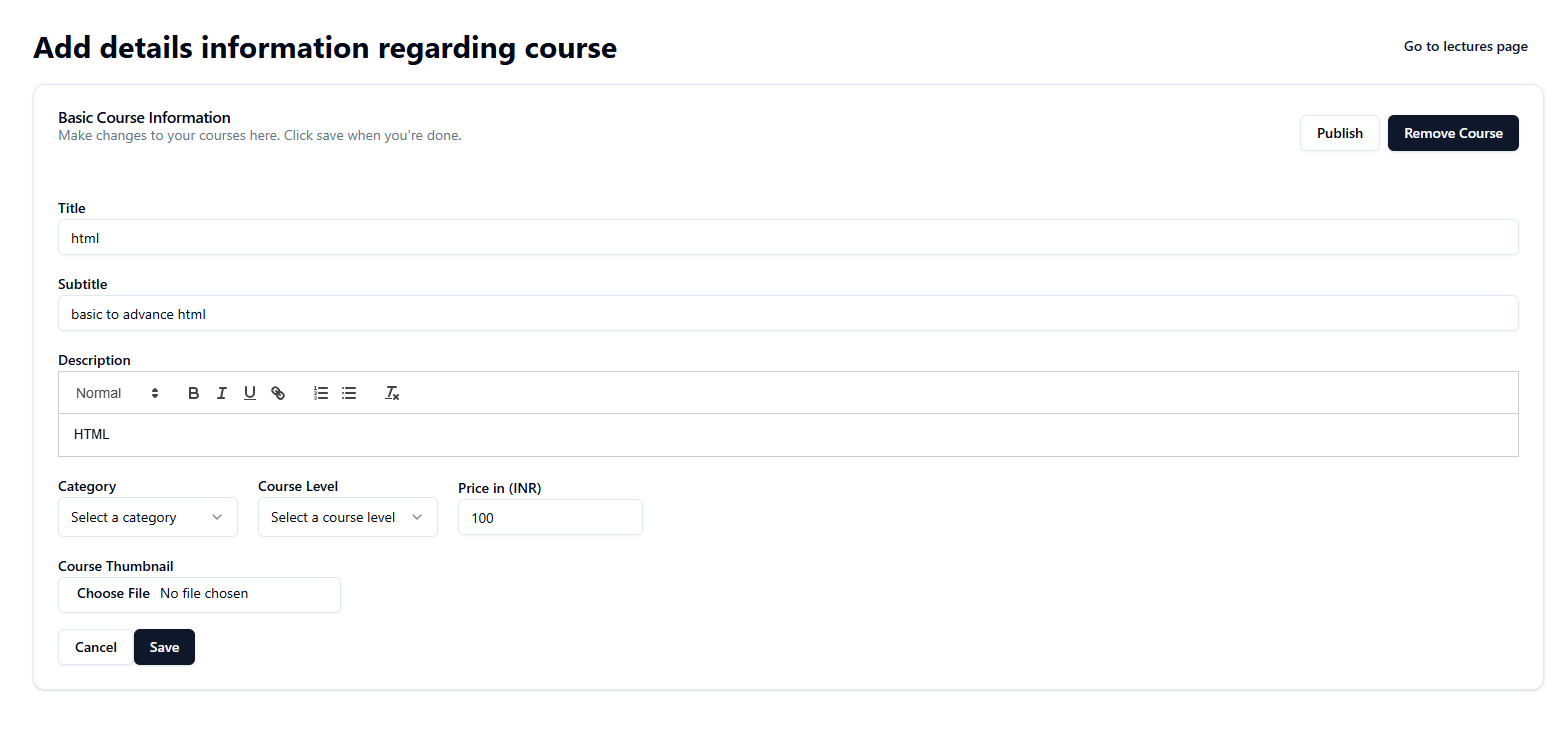
Onclick on dashboard:



Courses:



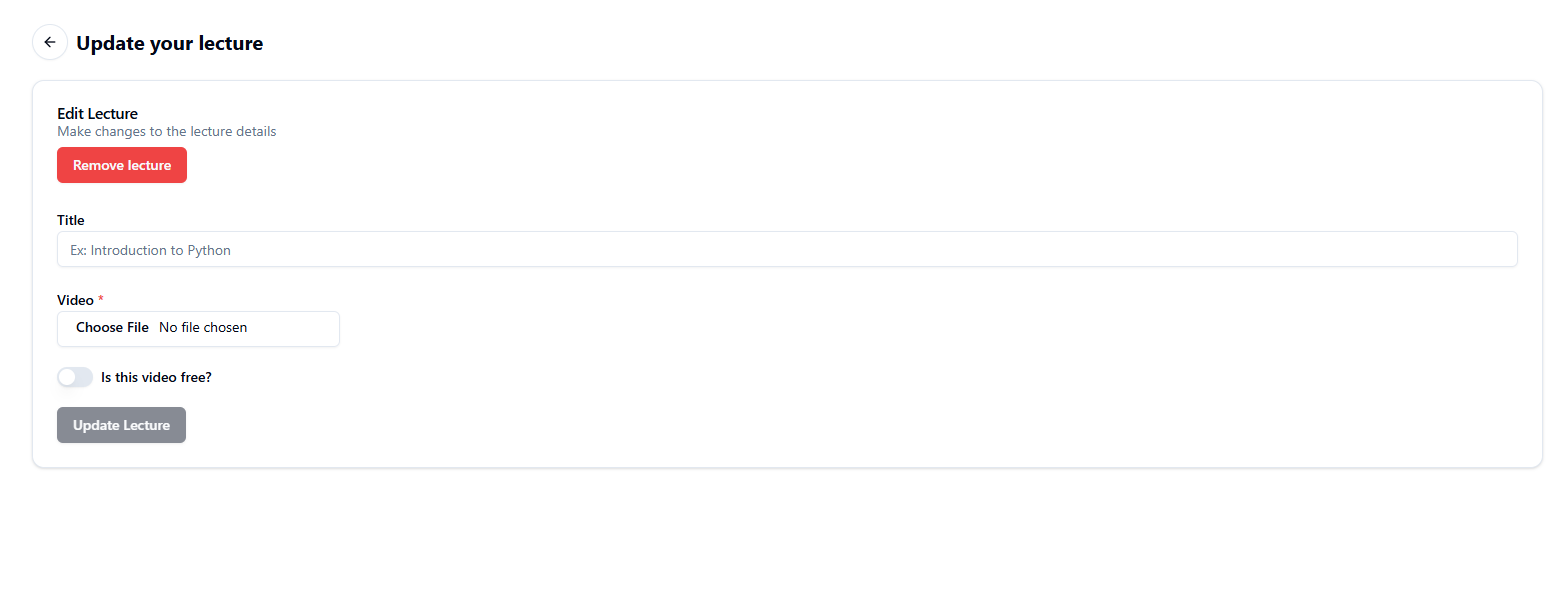
Couse information page



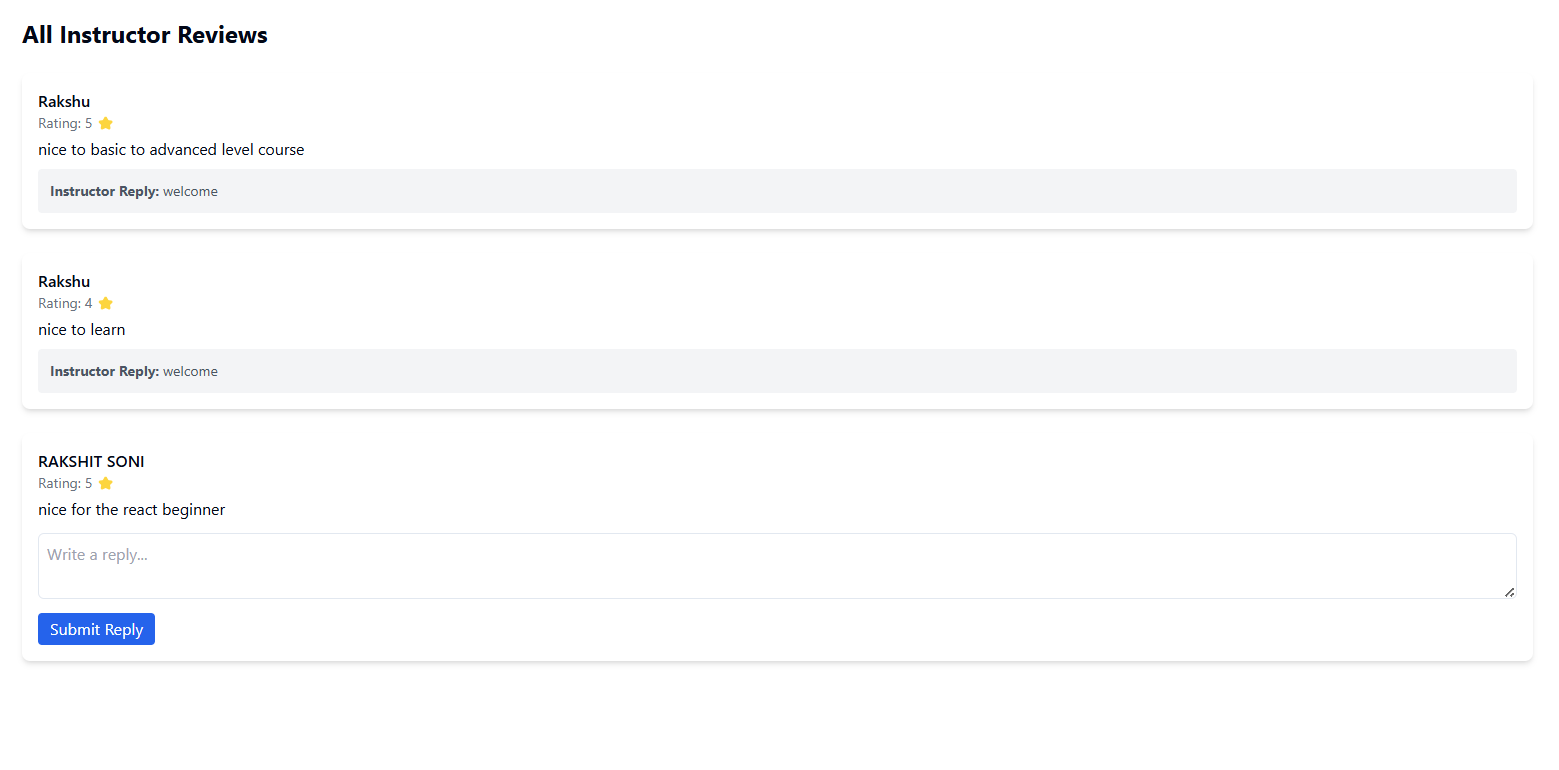
Add Lectures



Upload/ update the course



Course Reviews:



**Test Plan and Test Cases**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case ID | Module | Test Scenario | Test Steps | Expected Result |
| TC-001 | Instructor | Add a new course | 1. Log in as Instructor 2. Navigate to 'Add Course' 3. Enter course details 4. Submit | Course is successfully added |
| TC-002 | Instructor | Add a video to an existing course | 1. Log in as Instructor 2. Navigate to 'Manage Courses' 3. Select a course 4. Upload a video 5. Save changes | Video is successfully added |
| TC-003 | Instructor | Reply to a student review | 1. Log in as Instructor 2. Go to 'Reviews' 3. Select a review 4. Enter a response 5. Submit | Reply is displayed under review |
| TC-004 | Instructor | Set price for a course | 1. Log in as Instructor 2. Navigate to 'Manage Courses' 3. Select a course 4. Enter price 5. Save changes | Course price is updated |
| TC-005 | Student | View available courses | 1. Log in as Student 2. Navigate to 'Courses' 3. Browse through the list | Courses are displayed |
| TC-006 | Student | Buy a course | 1. Log in as Student 2. Select a course 3. Click 'Buy Now' 4. Complete payment | Purchase is successful, course unlocked |
| TC-007 | Student | Watch a purchased course video | 1. Log in as Student 2. Navigate to 'My Courses' 3. Select course 4. Click on video | Video plays without issues |
| TC-008 | Student | Complete a course | 1. Log in as Student 2. Watch all videos 3. Click 'Mark as Complete' | Course is marked as completed |
| TC-009 | Student | Submit a review for a completed course | 1. Log in as Student 2. Navigate to 'My Courses' 3. Select course 4. Submit a review | Review is posted successfully |
| TC-010 | Security | Ensure unauthorized users cannot access instructor panel | 1. Log in as Student 2. Try to access Instructor features | Access is denied |

Future Enhancement:

* Advanced Course Management
* Interactive Learning Features
* Improved Payment & Subscription Model
* Enhanced Review & Feedback System
* AI-Powered Recommendations
* Mobile App Version
* Security & Compliance
* Gamification Features
* Online Exam & Certification

**Conclusion**

The developed Learning Management System (LMS) provides an efficient platform for students to **watch course videos, purchase courses, and leave reviews**, while instructors can **add courses, manage content, and reply to reviews**. Future enhancements like **exams, interactive learning, and mobile support** will further improve user experience. This LMS ensures seamless learning with **security, accessibility, and scalability**.

**References/ Bibliography**

* **Coursera :** **www.coursera.org**
* **Gfg :www.geeksforgeeks.org**
* **W3school :** **www.w3schools.com**
* **Udemy :** **www.udemy.com**