

Darshan University

A Project Report on

**“Online Course Management System”**

Under the subject

**Software Engineering (2101CS503)**

B. Tech, Semester – V

Computer Science & Engineering Department

|  |  |
| --- | --- |
| Submitted By | |
| Student Name: Jethva Harsh N. | Enrollment No.:22010101072 |
| Academic Year  (2023-2024) | |
| Internal Guide  R. B. Gondaliya  Darshan University | Dean-DIET  Dr. Gopi Sanghani  Darshan University |

|  |  |
| --- | --- |
|  | **Computer Science & Engineering Department**  **Darshan University** |

**DECLARATION**

We hereby declare that the SRS, submitted along with the **Software Engineering** **(2101CS503)** for entitled **“Online Course Management System”** submitted in partial fulfilment for the Semester-5 of **Bachelor Technology (B. Tech)** in **Computer Science and Engineering (CSE)** Departmentto Darshan University, Rajkot, is a record of the work carried out at **Darshan University, Rajkot** under the supervision of **R. B. Gondaliya** and that no part of any of report has been directly copied from any students’ reports, without providing due reference.

Harsh Jethva

Student’s Signature

Date: \_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
|  | **Computer Science & Engineering Department**  **Darshan University** |

**CERTIFICATE**

This is to certify that the SRS on **“Online Course Management System” has** been satisfactorily prepared by **Harsh Jethva** (**22010101072**) under my guidance in the fulfillment of the course **Software Engineering (2101CS503)** work during the academic year 2024-2025.

|  |  |  |
| --- | --- | --- |
| Internal Guide  R. B. Gondaliya  Darshan University |  | Dean-DIET  Dr. Gopi Sanghani  Darshan University |

**Acknowledgement**

I wish to express my sincere gratitude to my project guide Prof. **R. B. Gondaliya** and all the faculty members for helping me through my project by giving me the necessary suggestions and advices along with their valuable co- ordination in completing this work.

I also thank my parents, friends and all the members of the family for their precious support and encouragement which they had provided in completion of my work. In addition to that, I would also like to mention the Darshan University personals who gave me the permission to use and experience the valuable resources required for the project from the University premises.

Thus, in conclusion to the above said, I once again thank the faculties and members of **Darshan University** for their valuable support in completion of the project.

Thanking You

**Harsh Jethva**

**ABSTRACT**

The Online Course Management System (OCMS) is a comprehensive digital platform designed to streamline the creation, administration, and delivery of educational courses over the internet. This system facilitates a seamless interaction between instructors and students, offering a user-friendly interface and robust functionalities to enhance the learning experience. Key features of OCMS include course content management, student enrollment and tracking, assignment submission, grading, and real-time communication tools. The system supports multimedia content, enabling diverse teaching methodologies and interactive learning modules. Additionally, OCMS integrates analytics tools for monitoring student progress and engagement, providing valuable insights for educators to tailor their teaching strategies. Security and scalability are integral aspects, ensuring data protection and accommodating a growing user base. By leveraging modern web technologies, OCMS addresses the evolving needs of educational institutions, promoting efficiency, accessibility, and quality in online education.

Main purpose of this system is to reduce human efforts as much as possible.

**Table of Contents**

[List of Figures I](#_Toc146706805)

[List of Tables II](#_Toc146706806)

[1 Introduction 1](#_Toc146706807)

[1.1 Product perspective 1](#_Toc146706808)

[1.2 Product features 1](#_Toc146706809)

[1.2.1 There are three different users who will be using this product: 1](#_Toc146706810)

[1.2.2 The features that are required for the Student are: 1](#_Toc146706811)

[1.2.3 The features that are required for the Faculty are: 1](#_Toc146706811)

[1.2.4 The features that are required for the Admin are: 1](#_Toc146706811)

[1.2.5 The features that are required for the Parents are: 1](#_Toc146706811)

[1.2.6 The features that are required for the Technical support staff are: 2](#_Toc146706812)

[1.3 Functional Requirement 2](#_Toc146706813)

[1.3.1 Student 2](#_Toc146706814)

[1.3.2 Faculty 2](#_Toc146706814)

[1.3.3 Admin 3](#_Toc146706814)

[1.3.4 Parent 3](#_Toc146706814)

[1.3.5 Technical support staff 3](#_Toc146706815)

[1.4 Non-Functional Requirement 4](#_Toc146706816)

[1.4.1 Usability: 4](#_Toc146706817)

[1.4.2 Accuracy: 4](#_Toc146706818)

[1.4.3 Backup and recovery: 4](#_Toc146706819)

[1.4.4 Maintainability: 4](#_Toc146706819)

[1.4.5 Performance: 4](#_Toc146706819)

[1.4.6 Security: 4](#_Toc146706820)

[2 Design and Implementation Constraints 5](#_Toc146706821)

[2.1 Use case diagram 5](#_Toc146706822)

[2.2 Activity diagram and Swimlane diagram 6](#_Toc146706823)

[2.3 Sequence diagram 8](#_Toc146706824)

[2.4 State diagram 9](#_Toc146706825)

[2.5 Class diagram 10](#_Toc146706826)

[2.6 Data flow diagram 11](#_Toc146706827)

[2.6.1 Context diagram (level-0) 11](#_Toc146706828)

[2.6.2 DFD Level-1 11](#_Toc146706829)

[2.6.3 DFD Level-2 12](#_Toc146706830)

[3 External interface requirement (Screens) 13](#_Toc146706831)

[3.1 Screen-1: Student Dashboard 13](#_Toc146706832)

[3.2 Screen-2: Course Browse Page 14](#_Toc146706833)

[3.3 Screen-3: Chapter View Page 14](#_Toc146706833)

[3.4 Screen-4: Teacher’s Course List Page 14](#_Toc146706833)

[3.5 Screen-5: Teacher’s Course Analytics Page 14](#_Toc146706833)

[3.6 Screen-6: Course create page 14](#_Toc146706833)

[4 Database design 20](#_Toc146706835)

[4.1 List of Tables 20](#_Toc146706836)

[5 Stories and Scenario 22](#_Toc146706837)

[5.1 Story-1: User registration in OCMS 22](#_Toc146706838)

[5.1.1 Scenario# S1.1 22](#_Toc146706839)

[5.1.2 Scenario# S1.2 22](#_Toc146706840)

[5.1.3 Scenario# S1.3 22](#_Toc146706840)

[5.1.4 Scenario# S1.4 23](#_Toc146706840)

[5.1.5 Scenario# S1.5 23](#_Toc146706841)

[6 Test cases 24](#_Toc146706846)

[7 References 25](#_Toc146706847)

# List of Figures

[Figure 2.1‑1 Use case diagram for online course management system 5](#_Toc146706848)

[Figure 2.2‑1 Activity diagram for course registration 6](#_Toc146706849)

[Figure 2.2‑2 Activity diagram for course payment 6](#_Toc146706849)

[Figure 2.2‑3 Swimlane diagram of course payment 7](#_Toc146706850)

[Figure 2.3‑1 Sequence diagram for course enrollment and get assignment details 8](#_Toc146706851)

[Figure 2.3‑2 Sequence diagram for assignment adding and submission 8](#_Toc146706851)

[Figure 2.4‑1 State Diagram of Student 9](#_Toc146706852)

[Figure 2.5‑1 Class diagram for online course management system 10](#_Toc146706854)

[Figure 2.6‑1 Context diagram for Online course management system 11](#_Toc146706855)

[Figure 2.6‑2 DFD level-1 for student 11](#_Toc146706856)

[Figure 2.6‑3 DFD level-2 for course enrollment 12](#_Toc146706857)

[Figure 3.1‑1 Screen-1: Student dashboard 13](#_Toc146706858)

[Figure 3.2‑1 Screen-2: Course browse Page 14](#_Toc146706859)

[Figure 3.3‑1 Screen-3: Chapter view Page 15](#_Toc146706859)

[Figure 3.4-1 Screen-4: Teacher course list Page 16](#_Toc146706859)

[Figure 3.5‑1 Screen-5: Teacher course analytics Page 17](#_Toc146706859)

[Figure 3.6‑1 Screen-6: Course create page 18](#_Toc146706860)

# List of Tables

[Table 3.1‑1 Screen element of Student dashboard 13](#_Toc146706861)

[Table 3.2‑1 Screen element of Course browse page 14](#_Toc146706862)

[Table 3.3‑1 Screen element of Chapter view page 14](#_Toc146706862)

[Table 3.4‑1 Screen element of Teacher course list page 14](#_Toc146706862)

[Table 3.5‑1 Screen element of Teacher course analytics page 14](#_Toc146706862)

[Table 3.6‑1 Screen element of Course create page 15](#_Toc146706863)

[Table 4.1‑1 Table: User 20](#_Toc146706864)

[Table 4.1‑2 Table: Course 20](#_Toc146706865)

[Table 4.1‑3 Table: Chapter 21](#_Toc146706866)

[Table 4.1‑4 Table: Purchase 21](#_Toc146706866)

[Table 4.1‑5 Table: Attachment 21](#_Toc146706866)

# Introduction

## Product perspective

The purpose of this Software Requirements Specification (SRS) document is to detail the requirements for the development of an Online Course Management System (OCMS). This system is designed to facilitate the management and delivery of educational courses over the internet, aiming to enhance the teaching and learning experience for both instructors and students. The OCMS will provide a centralized platform for course creation, student enrolment, content delivery, assessment, and communication.

## Product features

### There are four different users who will be using this product:

* Student who will be accessing the courses.
* Faculty who will create courses.
* Admin who will be acting as the administrator.
* Parent who will oversee the progress of student.
* Technical support staff who will create and modify software and manage hardware devices.

### The features that are required for the Student are:

* Course Enrolment: Ability to browse, enrol in, and drop courses.
* Assignment Submission: Upload assignments, receive feedback, and track grades.
* Course Materials Access: View and download lecture notes, slides, and other resources.
* Discussion Forums: Participate in class discussions, ask questions, and collaborate with peers.

### The features that are required for the Faculty are:

* Course Creation: Design and publish course content, including syllabi, assignments, and exams.
* Grade Management: Enter, edit, and publish student grades.
* Communication Tools: Send announcements and communicate with students via email or messaging systems.
* Attendance Tracking: Record and monitor student attendance.

### The features that are required for the Admin are:

* User Management: Create, update, and delete user accounts and manage roles and permissions.
* Course Scheduling: Oversee the scheduling of courses, classrooms, and resources.
* Reporting and Analytics: Generate reports on student performance, course popularity, and system usage.
* System Configuration: Manage system settings, integrations, and customizations.

### The features that are required for the Parent are:

* Progress Tracking: View child's grades, assignments, and overall progress in courses.
* Communication Portal: Contact faculty and staff regarding the child's performance or issues.
* Event Notifications: Receive updates about school events, meetings, and important dates.
* Financial Management: Access and manage billing information, payment history, and financial aid details.

### The features that are required for the Technical Staff are:

* System Maintenance: Perform regular system updates, backups, and maintenance tasks.
* Technical Support: Provide helpdesk support for system issues and user inquiries.
* Security Management: Implement and monitor security measures to protect data integrity and privacy.
* Integration Management: Manage integrations with other systems such as LMS, CRM, and third-party tools.

## Functional Requirement

### Student

* Login/Logout: Login to the platform by using userid and password with role based access and can edit profile. User can logout.
* Courses and curriculum: They can view and purchase course. Additionally, access content inside courses as chapters.
* Apply for financial aid: Students can apply for financial aid
* Track Progress: Student can track progress their own progress.
* Provide Feedback: They can provide feedback for the course they purchased.
* Give tests: Give tests after getting learnings.
* Submit assignment: Submit assignment asked accordingly.
* Download attachments: Students can download access-based attachments.
* Resource Bookmarking: Bookmark important resources and materials for easy access.
* Peer reviewer: They can check and can give grades to the assignment of another student.
* View certificates and grades: After completing the course they will get certificate of completion.
* Course History: Access history of completed courses and achievements.
* Study Groups: Create and join study groups with other students.
* Language Support: Access the system in multiple languages to support diverse student populations.
* Goal Setting and Tracking: Set academic goals and track progress towards achieving them.

### Faculty

* Create course with chapters: Create course by adding all type of content into modules or units as chapters. Additionally, can attach attachments and arrange test.
* Assignment: Create, distribute, manage and grade assignments.
* Monitor student progress: Monitor and track student progress and performance.
* Monitor course analysis: Access detailed analytics on course engagement, content effectiveness, and student outcomes.
* Attendance tracking: Record and track student attendance for in-person or virtual classes.
* Conduct live classes: Conduct live classes to the students for better understanding
* View transactions: Teachers can view transactions history.
* View notifications: To see the notifications from admin and other activities.
* Schedule management: Manage online course scheduling hours.
* Surveys and Polls: Conduct surveys and polls to gather student feedback and opinions.
* Learning Path Customization: Create customized learning paths for different student needs and levels.
* Join discussion: To join discussion with students and teachers.
* Certification Issuance: Issue certificates of completion or participation to students.
* External Resource Linking: Link to external resources and references for supplemental learning.

### Admin

* Access all user accounts: They will allow users to create, edit and delete their id on the platform.
* Set up courses: Set up courses by reviewing and ensure course content integrity and compliance.
* Role-based permission: Configure and manage role-based access to different parts of the CMS. Ensure that users can only access features and data relevant to their role.
* Calendar and scheduling: Manage course schedules, deadlines, and events.
* Course requirements meetup: Ensure courses meet requirements for professional certification and licensure. Provide an external perspective on course quality and effectiveness.
* Financial reports: Provide financial reports for the organization.
* Financial aid: Assist students in understanding and meeting these requirements
* Promote online courses: Ensure courses meet requirements for professional certification and licensure
* Analyse enrolment trends: Assist students in understanding and meeting these requirements
* Integrate with 3rd party tools: Provide 3rd party tools and providing platform to build real-time projects for students to get better understanding.
* Allow offline Access: Allow students to download course materials or any other services for offline access. Manage synchronization of offline and online content.
* Standard criteria: These criteria ensure that faculty members maintain high educational standards, contribute to institutional goals, and must meet, including qualifications, teaching effectiveness.
* Customization and Personalization: Customize system features and interfaces to meet institutional needs.

### Parent

* Student Progress Monitoring: Access detailed reports on their child’s academic performance. View grades, attendance records, and progress in individual courses.
* Behaviour and Conduct Reports: Access reports on their child's behaviour and conduct at school.
* Parental Controls: Set restrictions and controls on student's account and activities.
* Parent-Teacher Meetings: Schedule and participate in parent-teacher meetings.
* Health and Wellness: Access information on student's health and wellness activities.

### Technical Support Staff

* Troubleshooting issues: Provide technical support to users for system-related issues.
* API: Technical members can manage API access and integration with external systems.
* Setup system and Configuration: Install, configure, and maintain the CMS software and its dependencies.
* System performance: Maintain system performance and updates
* Generate reports: Generate reports of the platform steadiness and performance and all the data.
* Data backup and recovery: Implement data backup and recovery procedures.
* Data security: Ensure data security
* Payment resolving: Ensure payment be consistent
* Server and Database manage: Manage servers and databases to ensure optimal performance.
* Log and Audit Management: Maintain system logs and audit trails for monitoring and compliance.
* Scalability planning: Plan for and implement system scalability to handle increased load.
* Software updates and patching: Apply software updates, patches, and upgrades to the CMS.

## Non-Functional Requirement

### Usability:

* The UI should be simple enough for everyone to understand and get the relevant information without any special training. Different languages can be provided based on the requirements. Preparing documentation for software.

### Accuracy:

* The data stored about the courses and the payments calculated should be correct, consistent, and reliable.

### Backup and recovery:

* The System should be available for the duration when the website operates and must be recovered within an hour or less if it fails. The system should respond to the requests within two seconds or less.

### Maintainability:

* The software should be easily maintainable and adding new features and making changes to the software must be as simple as possible. In addition to this, the software must also be portable. Code quality should be good.

### Performance:

* The software should be scalable with good response time.

### Security:

* The software should include authentication and authorization, data encryption and compliance.

# Design and Implementation Constraints

## Use case diagram

 Figure 2.1-1 Use case diagram for online course management system

## Activity and Swimlane diagram

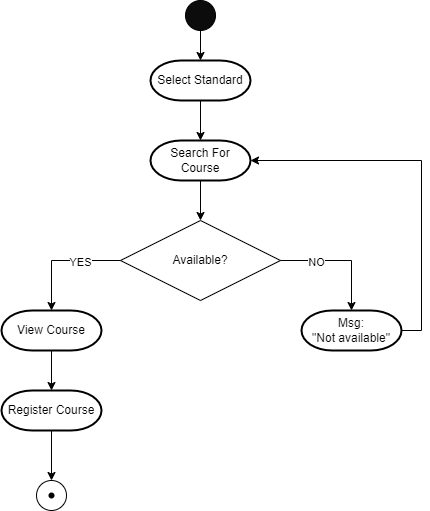


Figure 2.2‑1 Activity diagram for course registration

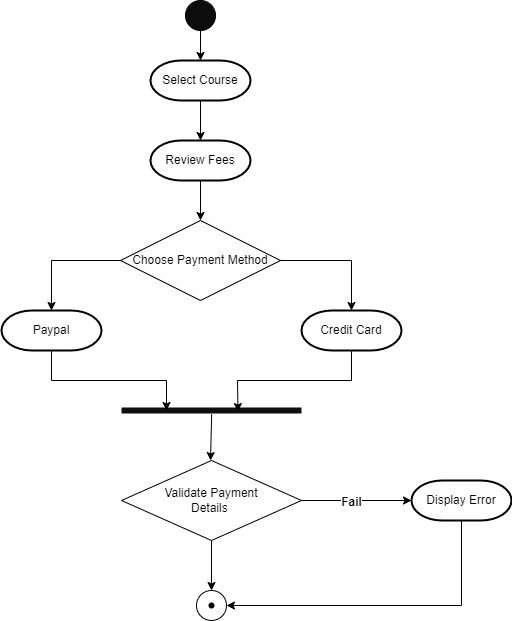


Figure 2.2‑2 Activity Diagram of course payment

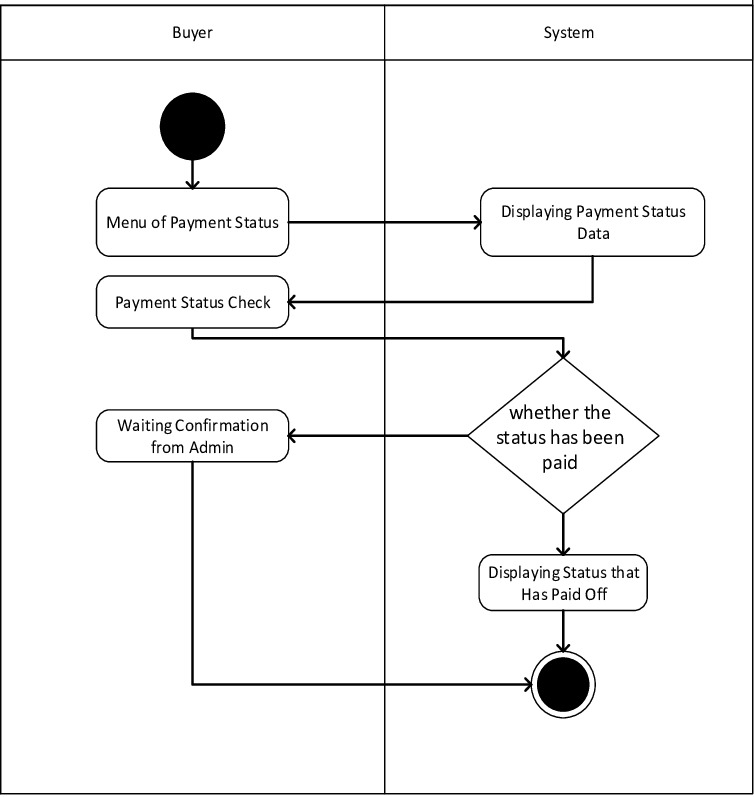


Figure 2.2‑3 Swimlane diagram of course payment

## Sequence diagram

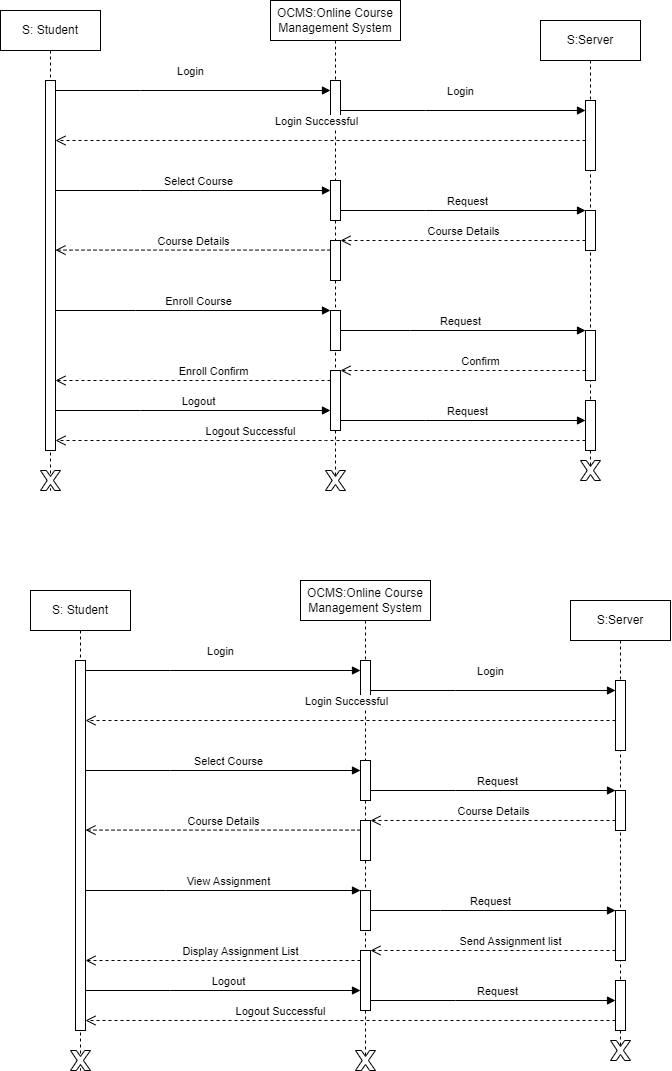


Figure 2.3‑1 Sequence diagram for course enrollment and get assignment details

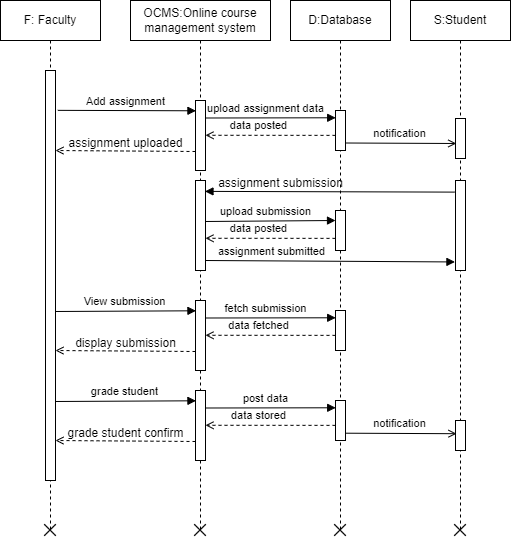


Figure 2.3‑2 Sequence diagram for assignment adding and submission

## State diagram



2.4-1 State Diagram of Student

## Class diagram

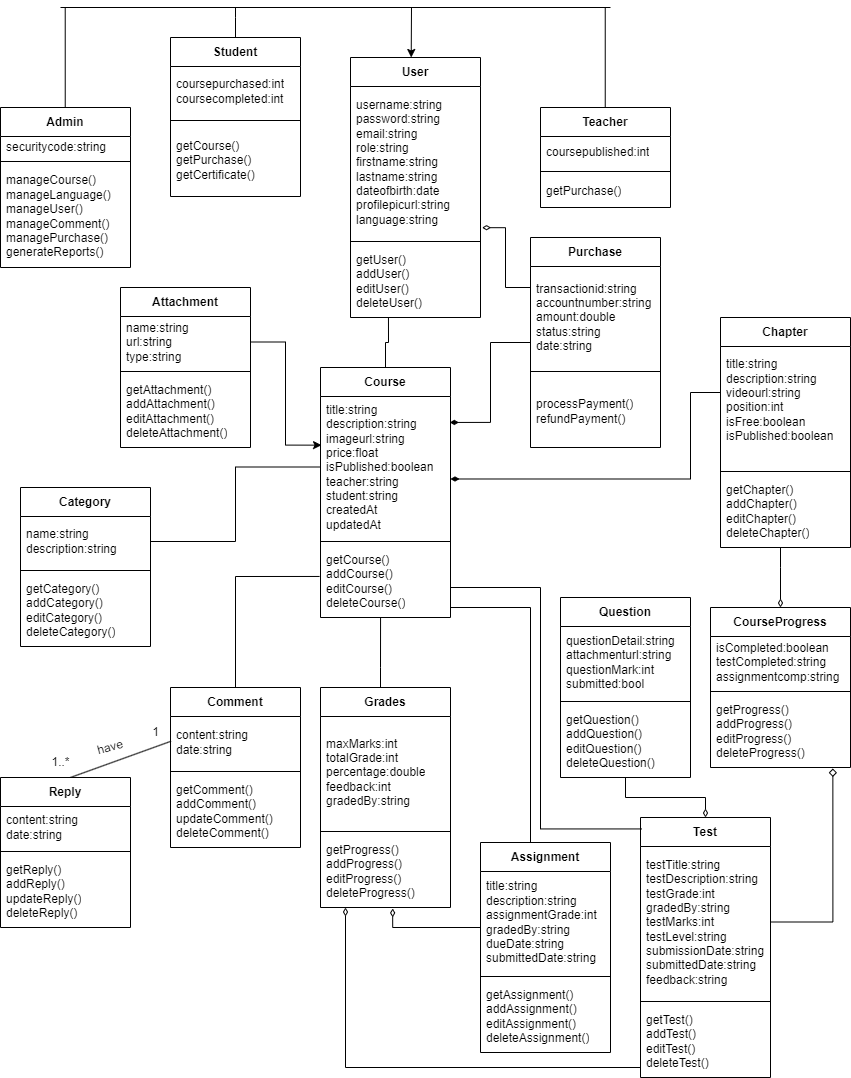


Figure 2.5‑1 Class diagram for online course management system

## Data flow diagram

### Context diagram (level-0)

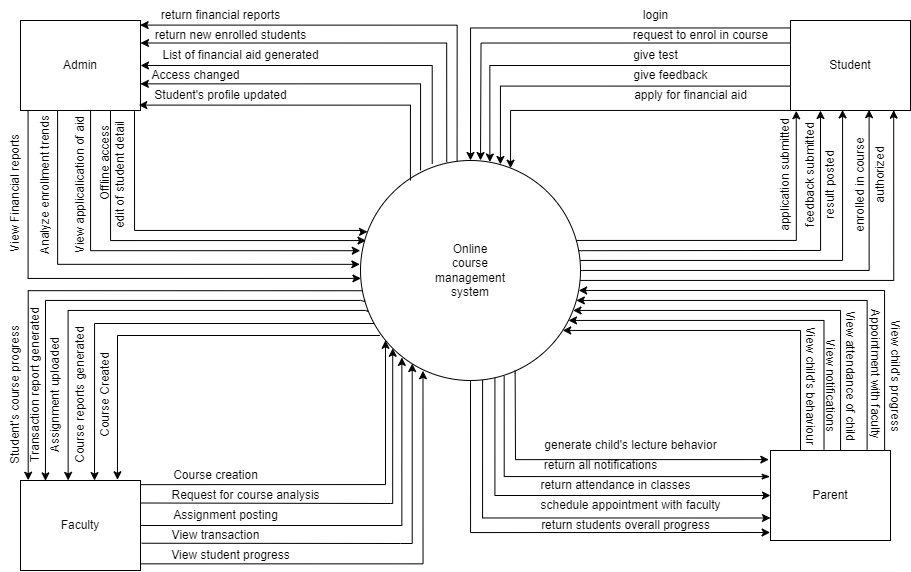


Figure 2.6‑1 Context diagram for online course management system

### DFD Level-1

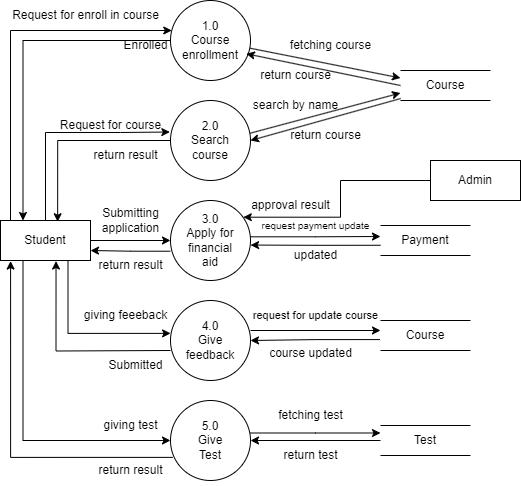


Figure 2.6‑2 DFD level-1 for student

### DFD Level-2

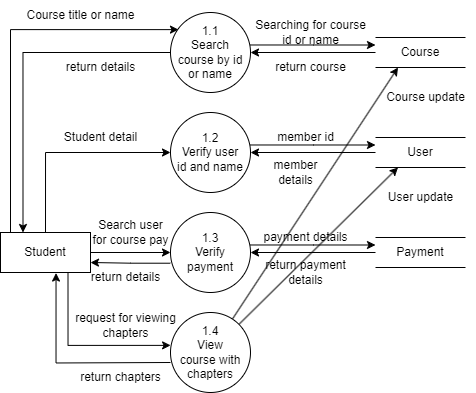


Figure 2.6‑3 DFD level-2 for course enrollment

# External interface requirement (Screens)

## Screen-1: Student Dashboard

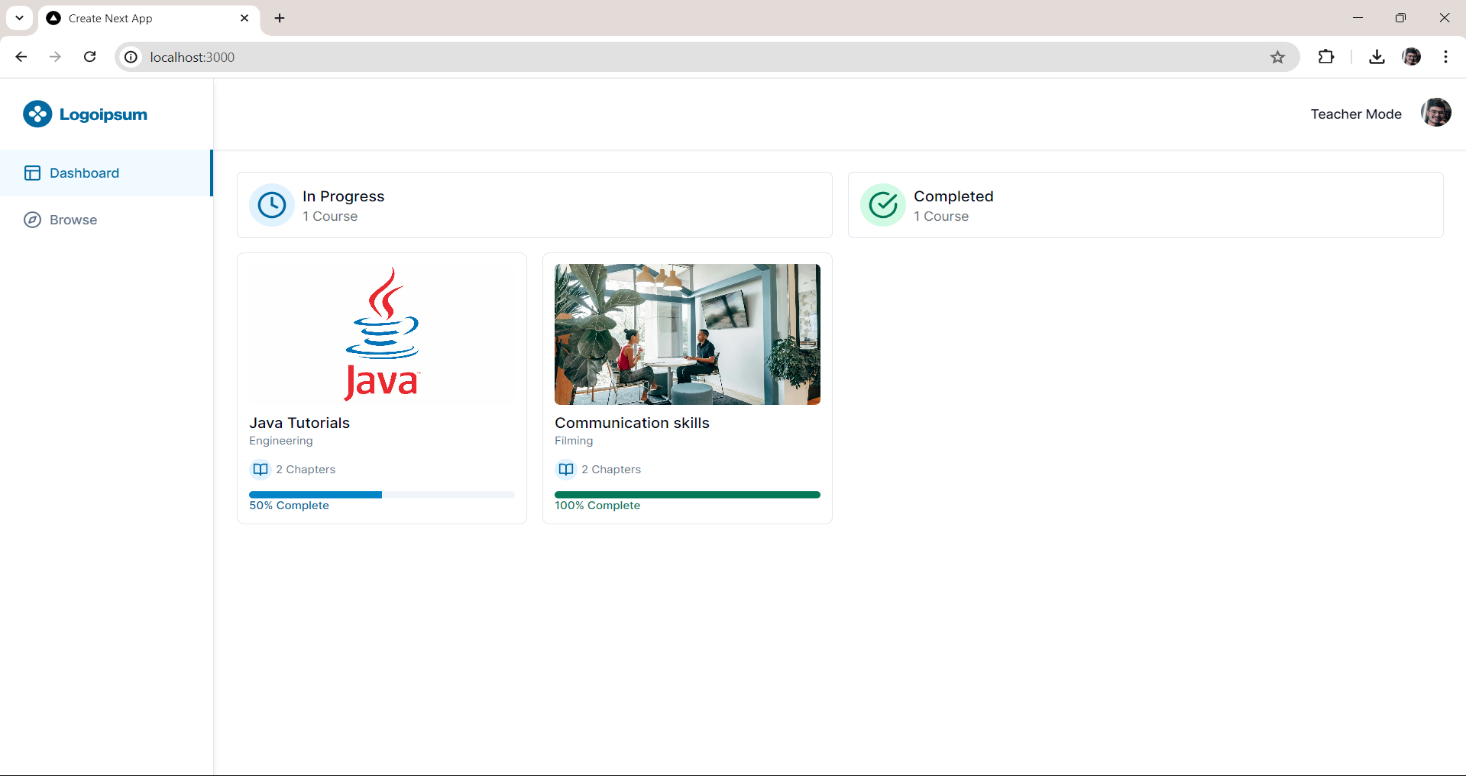


Figure 3.1‑1 Screen-1: Student Dashboard

**Purpose:** This screen will be used by the students to access the purchased course of the system. The users will see their completed courses and in-progress courses. On that basis student can choose courses.

Table 3.1‑1 Screen element of Student Dashboard

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Logo-image | Button | M | 1 | Button link to dashboard page. |
| 2 | Dashboard | Button | M | 1 | Button link to the user purchased course list pages and portfolio. |
| 3 | Browse | Button | M | 1 | Button link to the all-course list page. |
| 4 | Teacher mode | Button | M | 1 | Button to redirect to the teacher-role pages. |
| 5 | User-image | Button | M | 1 | Image button to logout and view details of the session. |
| 6 | In Progress | Text | M | 1 | Text displaying the in-progress or pending courses that student might have to complete. |
| 7 | Completed | Text | M | 1 | Text displaying the completed courses that student completed. |
| 8 | Card | Card | M | N | Details of the user-purchased respective courses. |

## Screen-2: Course Browse Page

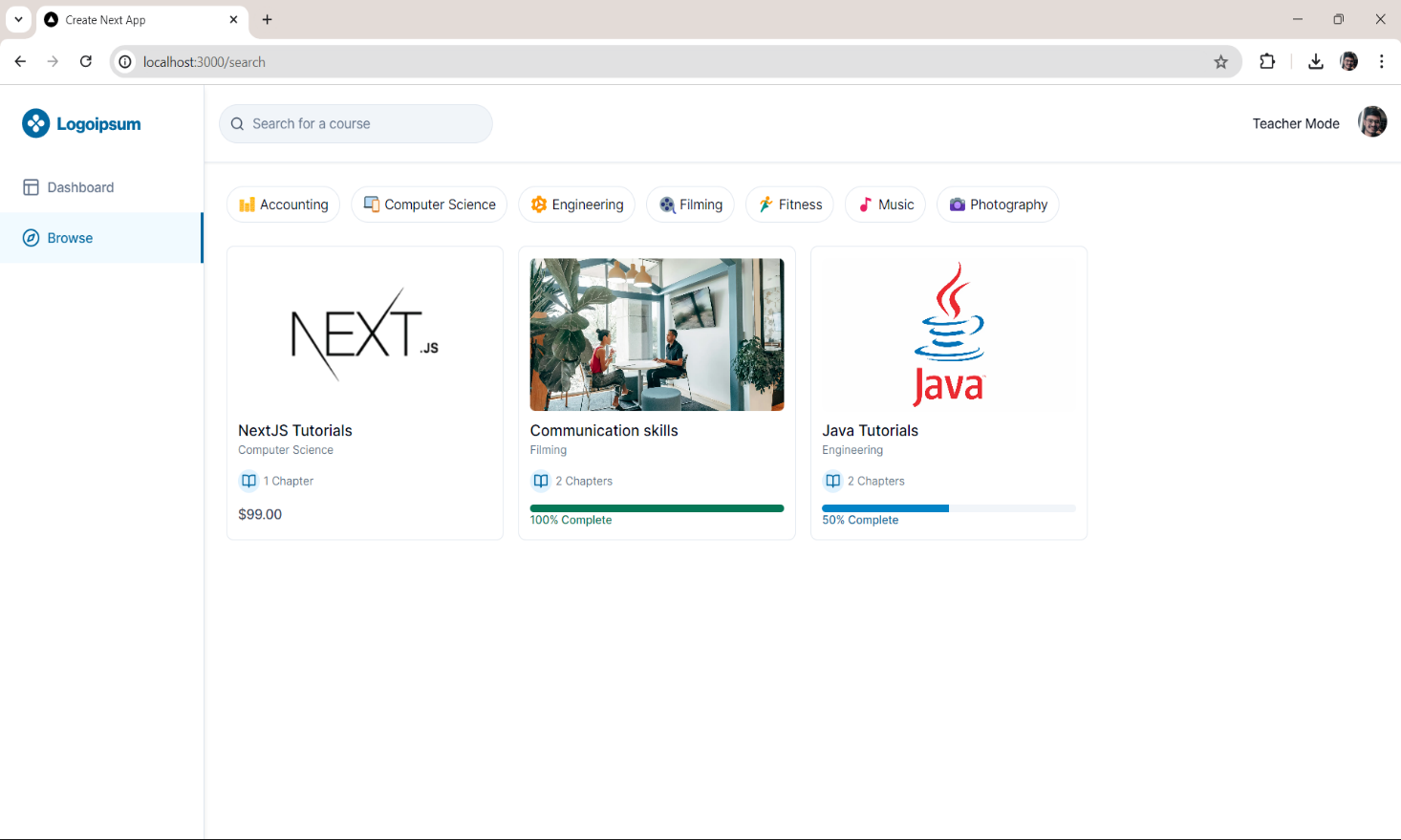


Figure 3.2‑1 Screen-2: Course browse page

**Purpose:** This form will be used by the system’s users to access records and features of the system. The users will input the correct combination of their username and password to be able to login to the system.

Table 3.2‑1 Screen element of course browse page

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Logo-image | Button | M | 1 | Button link to dashboard page. |
| 2 | Dashboard | Button | M | 1 | Button link to the user purchased course list pages and portfolio. |
| 3 | Browse | Button | M | 1 | Button link to the all-course list page. |
| 4 | Search | Textbox | M | 1 | Course search be filled for the search related course-list. |
| 5 | Teacher mode | Button | M | 1 | Button to redirect to the teacher-role pages. |
| 6 | User-image | Button | M | 1 | Image button to logout and view details of the session. |
| 7 | Carousel | Button | M | N | Categories list for the course category related searches. |
| 8 | Card | Button | M | N | List of all the types of courses. |

## Screen-3: Chapter View Page

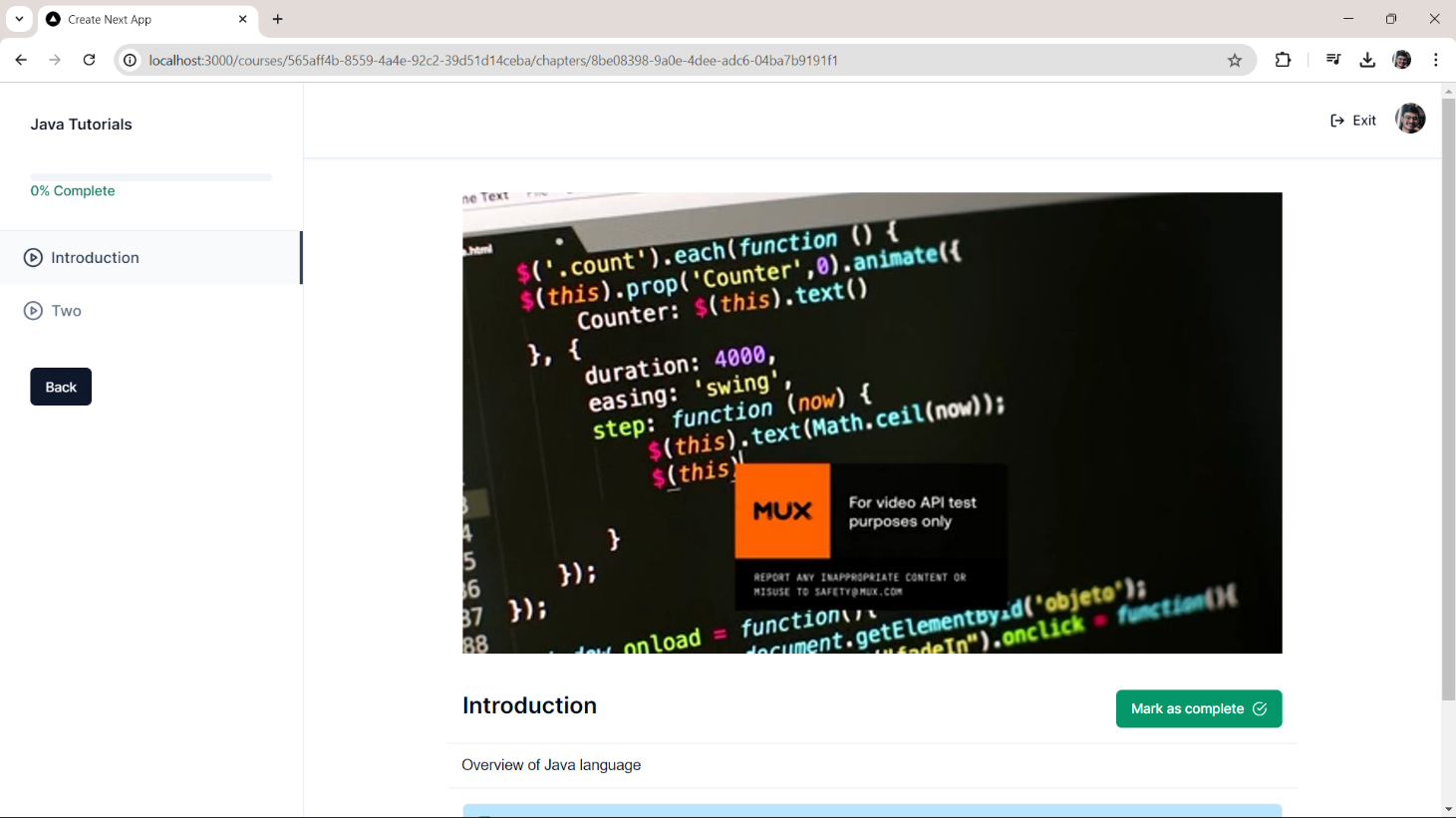


Figure 3.3‑1 Screen-3: Chapter View Page

**Purpose:** This module will allow the system administrator to add, edit, update or delete borrowers of book. The admin can add borrower information and manage it.

Table 3.3‑1 Screen element of Chapter View Page

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Course name | Textbox | M | 1 | Name of the course. |
| 2 | Course progress | Textbox | M | 1 | Course progress based on completed chapters. |
| 3 | Chapter list | Textbox | M | N | List of Chapter lister inside Course. |
| 4 | Back | Textbox | O | 1 | Button for the last chapter visited. |
| 5 | Exit | Textbox | M | 1 | Button to exit the course. |
| 6 | User-image | Textbox | M | 1 | Image button to logout and view details of the session. |
| 7 | Chapter video | Textbox | O | 1 | Chapter video for the learnings. |
| 8 | Chapter name | Button | ----- | ----- | Chapter name for brief overview. |
| 9 | Description | Button | ----- | ----- | Text to describe the chapter outcome and other details. |
| 10 | Attachment | Button | O | N | It may be in the form of pdf, zip or jpg image for other important learnings. |
| 11 | Mark as  complete | Button | M | 1 | Button to store the user complete course in the database. |

## Screen-4: Teacher Course List Page

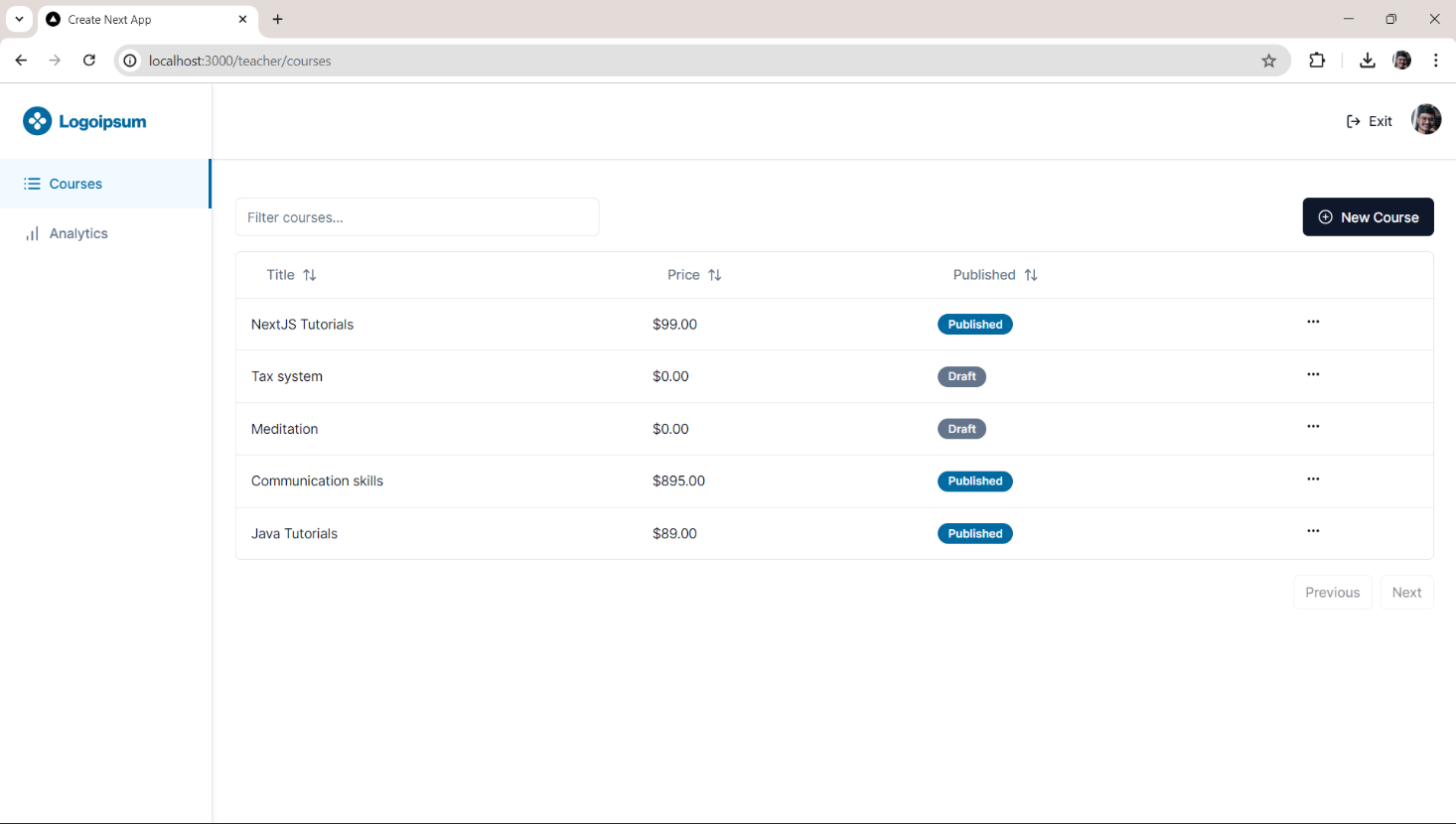


Figure 3.4‑1 Screen-4: Teacher Course List Page

**Purpose:** This module will allow the system administrator to add, edit, update or delete borrowers of book. The admin can add borrower information and manage it.

Table 3.4‑1 Screen element of Teacher Course List Page

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Logo-image | Button | M | 1 | Button link to courses page. |
| 2 | Courses | Button | M | 1 | Button link to the list of course made by teacher. |
| 3 | Analytics | Button | M | 1 | Button link to the course analytics page. |
| 5 | Exit | Button | M | 1 | Button to redirect to the student-role pages. |
| 8 | User image | Button | M | 1 | Image button to logout and view details of the session. |
| 6 | Search | Textbox | M | 1 | Get search-related result list of course. |
| 7 | New course | Textbox | O | 1 | Button to create new course. |
| 8 | Course list | Button | ----- | ----- | List of the course made by teacher. |
| 9 | Previous | Button | ----- | ----- | Link to predecessor course list made by teacher. |
| 10 | Next | Button | M | 1 | Link to successor course list made by teacher. |

## Screen-5: Teacher Course Analytics Page

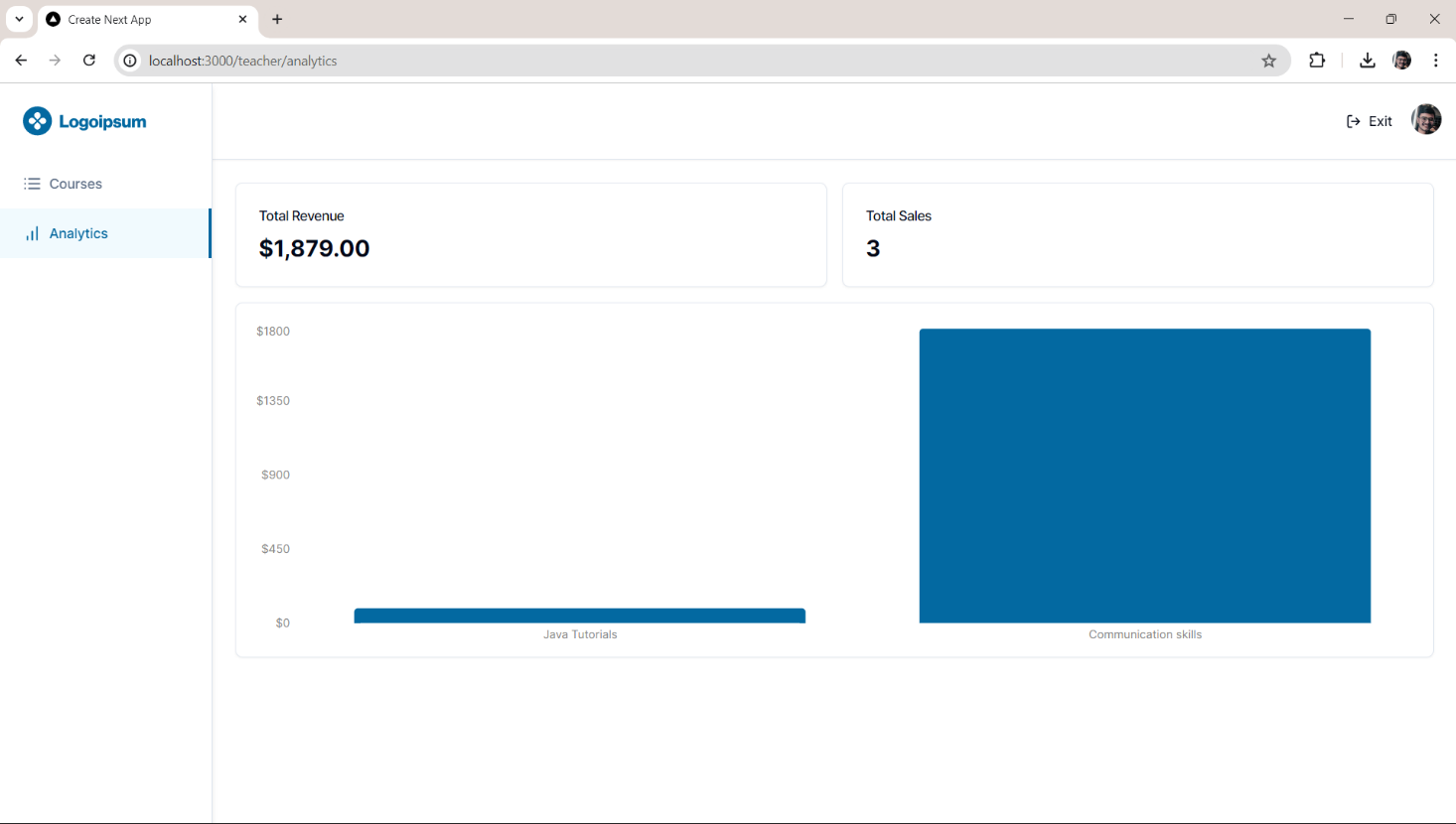


Figure 3.5‑1 Screen-5: Teacher Course Analytics Page

**Purpose:** This module will allow the system administrator to add, edit, update or delete borrowers of book. The admin can add borrower information and manage it.

Table 3.5‑1 Screen element of Teacher Course Analytics Page

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Logo-image | Button | M | 1 | Button link to courses page. |
| 2 | Courses | Button | M | 1 | Button link to the list of course made by teacher. |
| 3 | Analytics | Button | M | 1 | Button link to the course analytics page. |
| 5 | Exit | Button | M | 1 | Button to redirect to the student-role pages. |
| 8 | User image | Button | M | 1 | Image button to logout and view details of the session. |
| 6 | Total revenue | Textbox | M | 1 | Total money collected from all courses. |
| 7 | Total sales | Textbox | O | 1 | Total courses sell to students. |
| 8 | Graph | Image | ----- | ----- | Graph of revenue for respective courses. |

## Screen-6: Course create page

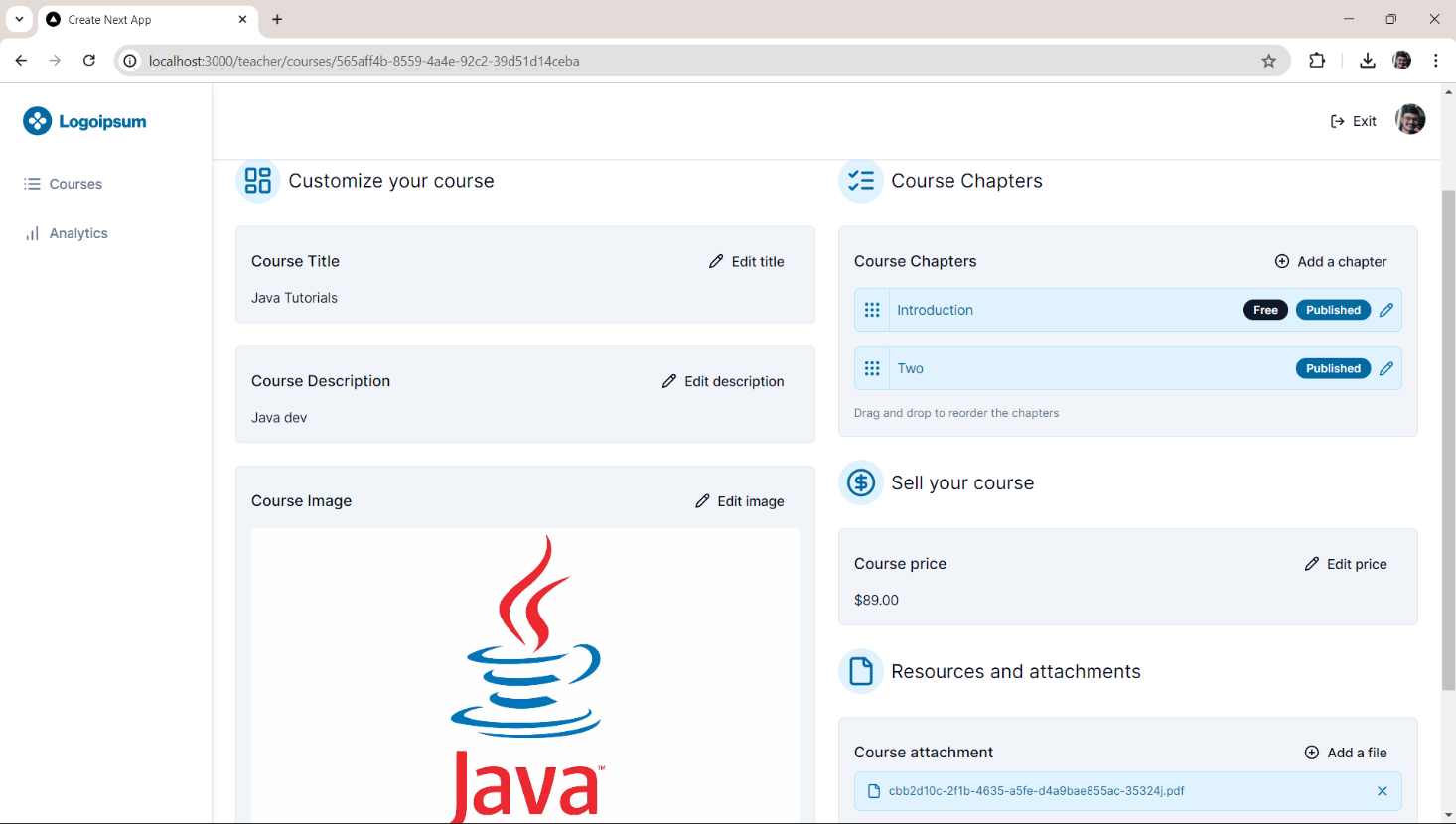


Figure 3.6‑1 Screen-6: Course create Form

**Purpose:** This module will allow the system administrator to add, edit, update or delete borrowers of book. The admin can add borrower information and manage it.

Table 3.6‑1 Screen element of Course create page

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Logo-image | Button | M | 1 | Button link to courses page. |
| 2 | Courses | Button | M | 1 | Button link to the list of course made by teacher. |
| 3 | Analytics | Button | M | 1 | Button link to the course analytics page. |
| 5 | Exit | Button | M | 1 | Button to redirect to the student-role pages. |
| 8 | User image | Button | M | 1 | Image button to logout and view details of the session. |
| 6 | Course title | Textbox | M | 1 | Title for course is displayed. |
| 7 | Course  description | Textbox | O | 1 | Description for course is displayed. |
| 8 | Course-image | Image | ----- | ----- | Image for course stored in database |
| 9 | Chapter card | Card | ----- | ----- | Cancel is a button for close add borrower dialog. |
| 10 | Add chapter | Button | M | 1 | Add chapter add a new chapter |
| 11 | Chapter name | Text | M | N | List of chapter are mentioned |
| 12 | Free | Text | O | 1 | Free chapter is displayed. |
| 13 | Published | Text | O | 1 | Published chapter is displayed |
| 14 | Chapter edit Icon | Button | M | 1 | Button for editing course. |
| 15 | Price | Textbox | M |  | Display price of course. |
| 16 | Add a file | Button | M | 1 | Add a new attachment. |
| 16 | Attachment | Text | M | 1 | Displays attachments. |
| 17 | Course Published | Button | M | 1 | It will publish chapter. |

# Database design

## List of Tables

* User
* Course
* Chapter
* Purchase
* Attachment

Table 4.1‑1 Table: User

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| UserId | int | NN | PK (Auto Increment) |  |
| CourseId | int | AN | FK |  |
| Name | varchar(30) | NN |  |  |
| Email | varchar(40) | NN |  |  |
| ImageUrl | varchar(40) | AN |  | http://localhost:3000/profile.jpg |
| Password | varchar(10) | NN |  |  |
| Role | varchar(10) | NN |  | student |
| CreatedAt | datetime | NN |  | Current date |
| UpdatedAt | datetime | NN |  | Current date |

Table 4.1‑2 Table: Course

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| CourseId | int | NN | PK (Auto Increment) |  |
| UserId | int | NN | FK |  |
| ChapterId | int | NN | FK |  |
| AttachmentId | int | AN | FK |  |
| PurchaseId | int | NN | FK |  |
| Title | varchar(50) | NN |  |  |
| Description | varchar(100) | NN |  |  |
| ImageUrl | varchar(50) | NN |  |  |
| Price | decimal(10,2) | NN |  |  |
| IsPublished | boolean | NN |  | False |
| CreatedAt | datetime | NN |  | Current date |
| UpdatedAt | datetime | NN |  | Current date |

Table 4.1‑3 Table: Chapter

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| ChapterID | int | NN | PK (Auto Increment) |  |
| CourseId | Int | NN | FK |  |
| Title | varchar(50) | NN |  |  |
| Description | varchar(100) | NN |  |  |
| VideoUrl | varchar(40) | NN |  |  |
| IsPublished | boolean | NN |  | False |
| IsFree | boolean | NN |  | False |

Table 4.1‑4 Table: Purchase

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| PurchaseId | int | NN | PK (Auto Increment) |  |
| CourseId | int | NN | FK |  |
| TransactionId | int | NN |  |  |
| Name | varchar(40) | NN |  |  |
| Amount | decimal(10,2) | NN |  |  |
| CreatedAt | datetime | NN |  | Current date |

Table 4.1‑5 Table: Attachment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| AttachmentId | int | NN | PK (Auto Increment) |  |
| CourseId | int | NN | FK |  |
| Name | varchar(50) | NN |  |  |
| AttachmentUrl | Varchar(50) | NN |  |  |
| CreatedAt | datetime | NN |  | Current date |
| UpdatedAt | datetime | NN |  | Current date |

# Stories and Scenario

## Story: User Registration in Online Course Management System

|  |  |  |
| --- | --- | --- |
| *Story # S1* | : | As a new user,  I want to register on the Online Course Management System  so that I can access and enrol in courses. |
| Priority | **:** | High |
| Estimate | **:** | XL |
| Reason | **:** | User registration is the first interaction a new user will have with the system. Ensuring a smooth and error-free registration process is crucial for user retention and satisfaction. |

### Scenario# S1.1

|  |  |  |
| --- | --- | --- |
| *Scenario# S1.1* | : | Successful User Registration with Valid Details |
| Prerequisite | **:** | The user is on the registration page of the Online Course Management System.  All registration form fields are visible and enabled. |
| Acceptance Criteria | **:** |  **Given** the user is on the registration page,   **When** the user enters valid details (name, email, password, confirm password) and clicks "Register",   **Then** the system should create a new user account and redirect the user to the login page with a success message. |

### Scenario# S1.2

|  |  |  |
| --- | --- | --- |
| *Scenario# S1.2* | : | Registration Fails with Already Registered Email |
| Prerequisite | **:** | The email address to be used is already registered in the system. |
| Acceptance Criteria | **:** |  **Given** the user is on the registration page and the email is already registered,   **When** the user attempts to register with this email,   **Then** the system should display an error message indicating that the email is already registered and keep the user on the registration page. |

### Scenario# S1.3

|  |  |  |
| --- | --- | --- |
| *Scenario# S1.3* | : | Registration Fails with Invalid Email Format |
| Prerequisite | **:** | The user is on the registration page of the Online Course Management System. |
| Acceptance Criteria | **:** |  **Given** the user is on the registration page,   **When** the user enters an invalid email format and clicks "Register",   **Then** the system should display an error message near the email field and prevent account creation. |

|  |  |  |
| --- | --- | --- |
| *Scenario# S1.4* | : | Registration Fails with Mismatched Password and Confirm Password |
| Prerequisite | **:** | The user is on the registration page of the Online Course Management System. |
| Acceptance Criteria | **:** |  **Given** the user is on the registration page,   **When** the user enters a password and a different confirm password and clicks "Register",   **Then** the system should display an error message near the confirm password field indicating the mismatch and prevent account creation. |

|  |  |  |
| --- | --- | --- |
| *Scenario# S1.5* | : | Registration Fails When Required Fields Are Left Blank |
| Prerequisite | **:** | The user is on the registration page of the Online Course Management System. |
| Acceptance Criteria | **:** |  **Given** the user is on the registration page,   **When** the user leaves one or more required fields blank and clicks "Register",   **Then** the system should display error messages near each blank required field and prevent account creation. |

# Test cases

|  |  |  |  |
| --- | --- | --- | --- |
| Project Name: | Online Course  Management System | Test Designed by: | R. B. Gondaliya |
| Module Name: | **User Registration** | **Test Designed date:** | 01-10-2024 |
| Release Version: | **1.0** | **Test Executed by:** | **Harsh Jethva** |
|  |  | **Test Execution date:** | 15-01-2024 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pre-condition: 1) The user is on the registration page of the Online Course Management System. | | | | |
| 2) The user is not currently logged in. | | | | |
| 3) The registration form fields are visible and enabled. | | | | |
| Test Case ID | **Test Title** | **Test Type** | **Description** | **Test Case ID** |
| TC\_001 | User Registration  with Valid Details | Functional | Registering user to Online Course Management System with Valid details | TC\_001 |
| TC\_002 | User Registration with an Already Registered Email | Negative | Registering user to Online Course Management System with an already registered email | TC\_002 |
| TC\_003 | Verify Register page elements | GUI | verify that all elements are available on Register page | TC\_003 |
| TC\_004 | User Registration with Invalid Email Format | Negative | Registering user to Online Course Management System with an invalid email format | TC\_004 |

|  |  |
| --- | --- |
| **Test Case Title** | User Registration with Valid Details |
| **Test Type** | Functional |
| **Test Priority** | High |
| **Pre-condition** | The user is on the registration page of the Online Course Management System. |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Step** | **Test Case Description** | **Expected Result** | **Actual Result** | **Status** | **Comment** | **Data** | **BUG ID** |
| 1 | Navigate to the registration page. | The site launched properly | Site launched successfully | Pass |  |  |  |
| 2 | Enter valid Username in username field | Username field should be editable and accept the Username | Username input accepted | Pass |  | Username:  John Doe |  |
| 3 | Enter a valid email address that is not already registered | Email field should be editable and accept the Email | Email input accepted | pass |  | Email: johndoe@example.com |  |
| 4 | Enter a valid password | Password field should be editable and accept the Password | Password input accepted | Pass |  | Password: SecurePass123 |  |
| 5 | Click the "Register" button. | User details should be Stored into Database and navigated to login Screen | User navigated to Login screen | pass |  |  |  |

|  |  |
| --- | --- |
| **Test Case Title** | User Registration with an Already Registered Email |
| **Test Type** | Negative |
| **Test Priority** | Medium |
| **Pre-condition** | The email address to be used is already registered in the system. |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Step** | **Test Case Description** | **Expected Result** | **Actual Result** | **Status** | **Comment** | **Data** | **Bug ID** |
| 1 | Navigate to the registration page. | The site launched properly | Site launched successfully | Pass |  |  |  |
| 2 | Enter a valid name | Username field should be editable and accept the Username | Username input accepted | Pass |  | Username:  John Doe |  |
| 3 | Enter an email address that is already registered | The system should not create a new account with the same email address | Display an error Email Already exist | Pass |  | Email: johndoe@example.com (already registered) |  |
| 4 | Verify that User is not able to Register with blank Username or Password | Set required field validation message for Username and Password | Display an error of not having username and password | Fail |  |  | Bug\_002 |

|  |  |
| --- | --- |
| **Test Case Title** | Verify Register page elements |
| **Test Type** | GUI |
| **Test Priority** | Medium |
| **Pre-condition** | The registration form fields (e.g., Name, Email, Password, Confirm Password) are visible and enabled. |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Step** | **Test Case Description** | **Expected Result** | **Actual Result** | **Status** | **Comment** | **Data** | **Bug ID** |
| 1 | Launch application | The site launched properly | Site launched successfully | Pass |  |  |  |
| 2 | Verify that the register screen contains elements such as Username, Password, Sign up button, password,  Confirm Password and Store User Data. | All listed control displayed properly on the page | Register page loaded successfully | Pass |  |  |  |
| 3 | Verify that cursor is focused on “Username” text box on the page load | Cursor is focused in Username textbox | Cursor focus in Username textbox | Pass |  |  |  |
| 4 | Verify that tab functionality is working . | When tab pressed cursor move in next control | Cursor moving in next control | Pass |  |  |  |
| 5 | Verify that all the fields such as Username, Password has a valid placeholder | All text fields have proper placeholder | All text fields have proper placeholder | Pass |  |  |  |
| 6 | Verify that the labels float upward when the text field is in focus or filled (In case of floating label) | When field is focused or filled, label display on top of the filled | When field is focus or filled, label display on top of the filled | Pass | step required when fields with floating label |  |  |

|  |  |
| --- | --- |
| **Test Case Title** | User Registration with Invalid Email Format |
| **Test Type** | Negative |
| **Test Priority** | Medium |
| **Pre-condition** | The user is on the registration page of the Online Course Management System. |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Step** | **Test Case Description** | **Expected Result** | **Actual Result** | **Status** | **Comment** | **Data** | **Bug ID** |
| 1 | Navigate to the registration page. | The site launched properly | Site launched successfully | Pass |  |  |  |
| 2 | Enter a valid name | Username field should be editable and accept the Username | Username input accepted | Pass |  | Username:  John Doe |  |
| 3 | Enter an email address that is invalid format | The system should validate the email format and not accept the invalid email. | An error message should be displayed near the email field "Please enter a valid email address” | Pass |  | johndoexample.com (invalid format) |  |
| 4 | Click the "Register" button. | Not able to register | Not registering User | Pass |  |  |  |

1. **References**

* http://www.w3schools.com/html/html\_intro.asp
* https://www.w3schools.com/php/default.asp
* https://www.javatpoint.com/uml