**Problem Statement**

In today's digital age, learning is taking place online more than ever before. But there's a challenge: existing online education systems are often complex and not user-friendly. Our website aims to solve this problem. We want to create an easy-to-use online learning platform for students, teachers, and administrators. Our goal is to make learning online simple, engaging, and accessible to everyone. We believe that education should be interactive and enjoyable. With our project, we hope to provide a better way for people to access knowledge and connect with others in a global community of learners. Our mission is to make online learning a fun and enriching experience for all, bringing education into the digital age.

**Requirement specification**

**Functional Requirement**

Functional requirements (FRs) provide an explanation of the tasks or functions that a system or software application should perform. In essence they outline the characteristics, behaviors and connections that the system needs to demonstrate in order to fulfill the requirements and meet the desires of its users or stakeholders.

Functional requirements are mandatory for the any system. For this system functional requirements are mentioned below.

|  |  |
| --- | --- |
| FR-01 | Registration |
| Description | Users must have the ability to register for the system by providing their name, email, and password. |
| Stakeholders | All users (Learners, Instructors, admins) |
| Priority | High |

|  |  |
| --- | --- |
| FR-02 | Login |
| Description | Registered users should be able to log in securely using their username and password. |
| Stakeholders | All users (Learners, Instructors, admins) |
| Priority | High |

|  |  |
| --- | --- |
| FR-03 | Course Management |
| Description | Only an admin able to create new courses by providing a course title, description, and other relevant information. And also can edit existing courses, and delete courses from the system. |
| Stakeholders | Admin |
| Priority | High |

|  |  |
| --- | --- |
| FR-04 | Course Enrollment |
| Description | Students are able to browse available courses and enroll in them. |
| Stakeholders | Learners |
| Priority | High |

|  |  |
| --- | --- |
| FR-05 | Payment Processing |
| Description | The system supports payment processing for course fees. |
| Stakeholders | Learners |
| Priority | High |

|  |  |
| --- | --- |
| FR-06 | Blog Post Creation and Mange |
| Description | Teachers should have full control over their blog posts, including the ability to create new blogs, edit the content and title of their existing blogs, and delete any of their posted blogs. |
| Stakeholders | Instructors |
| Priority | High |

|  |  |
| --- | --- |
| FR-07 | Blog Post Reading |
| Description | All users should be able to read blog posts created by teachers. |
| Stakeholders | All users (students, teachers, admins) |
| Priority | High |

|  |  |
| --- | --- |
| FR-08 | Provide Course Feedback |
| Description | Students should have the capability to provide feedback on courses in which they are enrolled. |
| Stakeholders | Learners |
| Priority | Medium |

|  |  |
| --- | --- |
| FR-09 | Course Feedback Viewing |
| Description | all users should be able to view and access feedback provided by learners for a particular course. |
| Stakeholders | All users (Learners, Instructors, admins) |
| Priority | Medium |

|  |  |
| --- | --- |
| FR-10 | Course Material Management |
| Description | Instructor should have the capability to add course materials to their courses. Additionally, teachers should be able to edit and delete the course materials they have created. |
| Stakeholders | Instructors |
| Priority | High |

|  |  |
| --- | --- |
| FR-11 | Course Material Viewing |
| Description | Learners should be able to view the course materials for courses in which they are enrolled. |
| Stakeholders | Learners |
| Priority | High |

|  |  |
| --- | --- |
| FR-12 | User Role Management |
| Description | When a new user completes registration for the system, their default role should be set as "Learner." Admins should have the authority to update user roles, changing them from "Learner" to "Instructor" as needed. |
| Stakeholders | Admin |
| Priority | Medium |

|  |  |
| --- | --- |
| FR-13 | User Removal |
| Description | Admins should have the authority to remove any user from the system as needed. This action permanently deletes the user's account and associated data from the system. |
| Stakeholders | Admin |
| Priority | High |

|  |  |
| --- | --- |
| FR-14 | User Profile Editing |
| Description | Users should be able to edit their profiles and update personal information, including email and password. |
| Stakeholders | All users (Learners, Instructors, admins) |
| Priority | Medium |

|  |  |
| --- | --- |
| FR-15 | User Logout |
| Description | Users, including students, teachers, and admins, should have the ability to log out securely from the system. |
| Stakeholders | All users (Learners, Instructors, admins) |
| Priority | Medium |

**Non-functional Requirement**

|  |  |
| --- | --- |
| NFR-01 | Performance |
| Description | The system should provide a responsive and fast user experience, with page load times not exceeding [specified time limit] even with a large number of concurrent users. |
| Stakeholders | All Users (students, teachers, admins) |
| Priority | High |

|  |  |
| --- | --- |
| NFR-02 | Security |
| Description | User data, including personal information and payment details, should be securely stored and transmitted using encryption and other appropriate security measures. Authentication and authorization mechanisms should prevent unauthorized access. |
| Stakeholders | All Users |
| Priority | High |

|  |  |
| --- | --- |
| NFR-03 | Usability |
| Description | The user interface should be intuitive and user-friendly, providing a positive user experience. The system should be accessible to users with disabilities, adhering to relevant accessibility standards. |
| Stakeholders | All Users |
| Priority | Medium |

|  |  |
| --- | --- |
| NFR-04 | Reliability |
| Description | The system should be available 24/7, with minimal downtime for maintenance. Backup and recovery mechanisms should be in place to prevent data loss and ensure system reliability. |
| Stakeholders | All Users |
| Priority | High |

|  |  |
| --- | --- |
| NFR-05 | Documentation |
| Description | Comprehensive documentation should be available for users, administrators, and developers, providing clear instructions on system usage, configuration, and maintenance. |
| Stakeholders | All Users |
| Priority | Medium |

**Use Case Description**

|  |  |
| --- | --- |
| Use Case ID | UC-01 |
| Use Case Name | Registration/Login |
| Actor | All users (Learners, Instructors, admins) |
| Description | Users (learners, instructors, admins) can register for the system and log in to access their accounts. |
| Precondition | User is not logged in. |
| Trigger | User accesses the system. |
| Flow of Events | * User selects the "Register" option. * User provides registration details (name, email, username, password). * User submits the registration form. * User receives a confirmation email. * User selects the "Login" option. * User enters their username and password. * User clicks "Login." |
| Postcondition | User is logged in and can access their account. |

|  |  |
| --- | --- |
| Use Case ID | UC-02 |
| Use Case Name | Manage Course |
| Actor | Admin |
| Description | Admins can create, edit, and delete courses in the system. |
| Precondition | Admin is logged in. |
| Trigger | Admin selects the course management option. |
| Flow of Events | * Admin selects "Create Course" to add a new course. * Admin enters course details (title, description, instructor). * Admin clicks "Create." * Admin selects an existing course to edit. * Admin modifies course details. * Admin clicks "Save Changes." * Admin selects a course to delete. * Admin confirms deletion. |
| Postcondition | Course management is updated according to admin actions. |

|  |  |
| --- | --- |
| Use Case ID | UC-03 |
| Use Case Name | View Course Detail |
| Actor | All Users |
| Description | All users can view details of a course, including its title, description, and instructor. |
| Precondition | User is logged in. |
| Trigger | User selects a course to view. |
| Flow of Events | * User navigates to the course details page. * User selects a course to view. |
| Postcondition | User can see the details of the selected course. |

|  |  |
| --- | --- |
| Use Case ID | UC-04 |
| Use Case Name | Manage Course Material |
| Actor | Instructor |
| Description | Instructors can add, edit, and delete course materials for the courses they teach. |
| Precondition | Instructor is logged in. |
| Trigger | Instructor selects the course material management option. |
| Flow of Events | * Instructor selects "Add Material" for a course. * Instructor uploads course material (documents, videos, etc.). * Instructor clicks "Add." * Instructor selects an existing material to edit. * Instructor modifies material details. * Instructor clicks "Save Changes." * Instructor selects a material to delete. * Instructor confirms deletion. |
| Postcondition | Course materials are updated according to instructor actions. |

|  |  |
| --- | --- |
| Use Case ID | UC-05 |
| Use Case Name | Enroll Course with Payment |
| Actor | Learner |
| Description | Learners can enroll in courses by making payments for the chosen courses. |
| Precondition | Learner is logged in. |
| Trigger | Learner selects a course to enroll in. |
| Flow of Events | * Learner selects a course to enroll in. * Learner proceeds to payment. * Learner provides payment details and confirms payment. |
| Postcondition | Learner is enrolled in the selected course upon successful payment. |

|  |  |
| --- | --- |
| Use Case ID | UC-06 |
| Use Case Name | Write Blog |
| Actor | Instructor |
| Description | Instructors can create and publish blog posts. |
| Precondition | Instructor is logged in. |
| Trigger | Instructor selects the "Write Blog" option. |
| Flow of Events | * Instructor enters blog post content (title, content). * Instructor clicks "Publish." |
| Postcondition | The blog post is published and visible to all users. |

|  |  |
| --- | --- |
| Use Case ID | UC-07 |
| Use Case Name | View Blog |
| Actor | All Users |
| Description | All users can view blog posts created by instructors. |
| Precondition | User is logged in. |
| Trigger | User selects a blog post to view. |
| Flow of Events | * User navigates to the blog post. * User selects a blog post to view. |
| Postcondition | User can see the content of the selected blog post. |

|  |  |
| --- | --- |
| Use Case ID | UC-08 |
| Use Case Name | Manage Blog |
| Actor | Admin |
| Description | Admins can edit and delete blog posts created by instructors. |
| Precondition | Admin is logged in. |
| Trigger | Admin selects the blog management option. |
| Flow of Events | * Admin selects a blog post to edit. * Admin modifies the blog post content. * Admin clicks "Save Changes." * Admin selects a blog post to delete. * Admin confirms deletion. |
| Postcondition | Blog posts are updated or removed according to admin actions. |

|  |  |
| --- | --- |
| Use Case ID | UC-09 |
| Use Case Name | Give Feedback |
| Actor | Learner |
| Description | Learners can provide feedback on courses in which they are enrolled. |
| Precondition | Learner is logged in and enrolled in at least one course. |
| Trigger | Learner selects a course to provide feedback. |
| Flow of Events | * Learner selects a course to provide feedback. * Learner enters feedback comments and ratings (if applicable). * Learner submits the feedback. |
| Postcondition | Learner's feedback is recorded for the selected course. |

|  |  |
| --- | --- |
| Use Case ID | UC-10 |
| Use Case Name | View Feedback |
| Actor | All Users |
| Description | All users can view feedback provided by learners for specific courses. |
| Precondition | User is logged in. |
| Trigger | User selects a course to view feedback. |
| Flow of Events | * User navigates to the course feedback section. * User selects a course to view feedback. |
| Postcondition | User can see feedback comments and ratings for the selected course. |

|  |  |
| --- | --- |
| Use Case ID | UC-11 |
| Use Case Name | Manage User |
| Actor | Admin |
| Description | Admins can manage user accounts, including creating, editing, and deleting user profiles. |
| Precondition | Admin is logged in. |
| Trigger | Admin selects the user management option. |
| Flow of Events | * Admin selects "Create User" to add a new user. * Admin enters user details (name, email, role). * Admin clicks "Create." * Admin selects an existing user to edit. * Admin modifies user details. * Admin clicks "Save Changes." * Admin selects a user to delete. * Admin confirms deletion. |
| Postcondition | User management is updated according to admin actions. |

**Future Plan**

**Discussion Forums:** Implement discussion forums where learners can engage in discussions, ask questions, and collaborate with peers and instructors. Forums can be organized by course or topic.

**Interactive Assessments:** Incorporate interactive quizzes, assignments, and simulations to assess learners' understanding and skills.

**Mobile App:** Develop a mobile app for Android and iOS devices to make learning accessible on smartphones and tablets.

Programming Language: JavaScript (intermediate)  
Database: MongoDB  
CSS Framework: Tailwind CSS  
Libraries: ReactJS  
Web Framework: ExpressJS