

## Functional Requirements:

### Authentication

#### Student:

- Students can log in and log out securely.
- Students can update their password.
- Students can reset their password using their registered email.
- Students can enable two-factor authentication for added security.
- Students receive email notifications for login activities.

#### Instructor:

- Instructors can log in and log out securely.
- Instructors can update their password.
- Instructors can reset their password using their registered email.
- Instructors can enable two-factor authentication for enhanced security.
- Instructors receive login activity notifications.

#### Admin:

- Admins can log in and log out securely.
- Admins can update their password.
- Admins can reset their password using their registered email.
- Admins can enforce security policies for all users.
- Admins receive notifications of any suspicious login activities.

## Profile Management:

#### Student:

- Students can manage and update their profile.
- Students can update certain fields.
- Students can change their profile image.
- Students can set preferences for notifications.
- Students can manage their privacy settings.

#### Instructor:

- Instructors can manage and update their profile.
- Instructors can update certain fields.
- Instructors can change their profile image.
- Instructors can set teaching preferences.
- Instructors can link their profiles to professional networks.

#### Admin:

- Admins can manage and update their profile.

- Admins can update certain fields.
- Admins can change their profile image.
- Admins can update system-wide notifications.
- Admins can modify security settings.

## **Course Management:**

### **Student:**

- Students can enroll in offered courses.
- Students can view their class schedule.
- Students can view course materials.
- Students can access quizzes and assignments.
- Students can see their grades.
- Students can participate in course discussions.
- Students can download lecture materials for offline study.
- Students receive notifications on assignment deadlines and course updates.

### **Instructor:**

- Instructors can create and manage courses.
- Instructors can upload course materials (videos, PDFs, slides).
- Instructors can create quizzes and assignments.
- Instructors can grade students.
- Instructors can manage course discussions.
- Instructors can set grading policies.
- Instructors can track student progress.
- Instructors can generate performance reports for students.

### **Admin:**

- Admins can manage multiple processes:
  - Course approval and management.
  - Instructor verification.
  - Category management.
  - Reports and analytics.
  - Monitoring course engagement.
  - Ensuring compliance with platform policies.

## **User Management:**

### **Admin:**

- Admins can manage multiple accounts.
- Admins can block/unblock users.
- Admins can change user passwords.
- Admins can monitor user activity logs.
- Admins can send system-wide announcements.

## **Payment Management:**

### **Student:**

- Students can make one-time payments or subscribe to courses.
- Students can view their payment history and download invoices.
- Students can request refunds or dispute transactions under defined conditions.

#### **Instructor:**

- Instructors can view earnings from course enrollments.
- Instructors can access payment reports and analytics related to their courses.

#### **Admin:**

- Admins can view and manage all payment transactions across the platform.
- Admins can process refund requests and handle payment disputes.
- Admins can configure and manage payment gateway settings and system-wide financial reports.

## **Data Model**

#### **User:**

- `_id`
- `id` (generated)
- `password`
- `needsPasswordChange`
- `role` (student, instructor, admin)
- `status`
- `isDeleted`
- `createdAt`
- `updatedAt`
- `securityQuestions`
- `lastLogin`
- `emailNotificationsEnabled`

#### **Student:**

- `_id`
- `id` (generated)
- `name`
- `gender`
- `dateOfBirth`
- `email`
- `contactNo`
- `profileImage`
- `enrolledCourses` (Array of Course IDs)
- `isDeleted`
- `createdAt`
- `updatedAt`
- `completedCourses` (Array of Course IDs)
- `discussionPosts` (Array of Post IDs)
- `preferences`

**Instructor:**

- \_id
- id (generated)
- name
- gender
- dateOfBirth
- email
- contactNo
- profileImage
- expertise
- assignedCourses (Array of Course IDs)
- isDeleted
- createdAt
- updatedAt
- reviews (Array of Review IDs)
- certifications

**Admin:**

- \_id
- id (generated)
- name
- gender
- dateOfBirth
- email
- contactNo
- profileImage
- isDeleted
- createdAt
- updatedAt
- systemAccessLogs
- managedReports

**Course:**

- \_id
- title
- description
- category
- instructor (Instructor ID)
- students (Array of Student IDs)
- materials (Array of File URLs)
- quizzes (Array of Quiz IDs)
- assignments (Array of Assignment IDs)
- createdAt
- updatedAt
- courseReviews (Array of Review IDs)
- announcements (Array of Announcement IDs)

## Quiz:

- `_id`
- `title`
- `course` (Course ID)
- `questions` (Array of Question Objects)
- `createdAt`
- `updatedAt`
- `passingCriteria`
- `timeLimit`

## Assignment:

- `_id`
- `title`
- `description`
- `course` (Course ID)
- `submissionDeadline`
- `createdAt`
- `updatedAt`
- `gradingRubric`
- `maxScore`

## Payment:

- `_id`
- `paymentId` (generated unique identifier)
- `studentId` (reference to Student)
- `courseId` or `subscriptionId` (depending on payment type)
- `amount`
- `currency`
- `status` (paid, pending, refunded, failed)
- `paymentMethod` (credit card, PayPal, etc.)
- `transactionDate`
- `createdAt`
- `updatedAt`

## API Endpoints

### User:

- `users/create-student` (POST)
- `users/create-instructor` (POST)
- `users/create-admin` (POST)
- `users/:id/reset-password` (POST)

### Student:

- `students` (GET)
- `students/:id` (GET)
- `students/:id` (PATCH)
- `students/:id` (DELETE)
- `students/my-profile` (GET)
- `students/:id/courses` (GET)

### Instructor:

- `instructors` (GET)
- `instructors/:id` (GET)
- `instructors/:id` (PATCH)
- `instructors/:id` (DELETE)
- `instructors/my-profile` (GET)
- `instructors/:id/courses` (GET)

### Admin:

- `admins` (GET)
- `admins/:id` (GET)
- `admins/:id` (PATCH)
- `admins/:id` (DELETE)
- `admins/my-profile` (GET)
- `admins/system-reports` (GET)

### Course:

- `courses` (GET)
- `courses/:id` (GET)
- `courses` (POST)
- `courses/:id` (PATCH)
- `courses/:id` (DELETE)
- `courses/:id/enroll` (POST)
- `courses/:id/materials` (GET)

### Payment:

- `payments` (GET) – Retrieve a list of all payment transactions.
- `payments/:id` (GET) – Retrieve details of a specific payment.
- `payments` (POST) – Create a new payment transaction.
- `payments/:id` (PATCH) – Update payment details (e.g., mark as refunded).
- `payments/:id` (DELETE) – Cancel or remove a payment if applicable.
- `payments/refund` (POST) – Process a refund request.

## Auth:

- `auth/login` (POST)
- `auth/refresh-token` (POST)
- `auth/change-password` (POST)
- `auth/forgot-password` (POST)
- `auth/reset-password` (POST)
- `auth/enable-2fa` (POST)