FPT ACADEMY INTERNATIONAL

FPT – APTECH COMPUTER EDUCATION

E-Learning Management System

Supervisor: *LÊ THANH NHÂN*

Semester: *II*

Batch No: *T2.2410.E0*

Group No: *03*

Order:	Full name	Roll No.
1.	*Lê Thị Hạnh Mai*	*Student1614875*
2.	*Cống Hiển Khoa*	*Student1614878*
3.	*Dương Minh Mẫn*	*Student1614886*

***Month**:08 **Year**:2025*

This is to certify that

Ms. **Lê Thị Hạnh Mai**

Mr. **Cống Hiển Khoa**

Mr. **Dương Minh Mẫn**

Have successfully Designed & Developed

E-Learning Management System

Submitted by:

Date Of Issue:

Authorized Signature:

Content

Content

Acknowledge

Synopsis

Review1 (điền số trang theo thực tế)

Review2 (điền số trang theo thực tế)

Review3 (điền số trang theo thực tế)

Testting Document (điền số trang theo thực tế)

Final Checklist (điền số trang theo thực tế)

Tasksheet (điền số trang theo thực tế)

Acknowledge

As we understand, the eProject is a structured step-by-step learning experience that simulates both classroom and lab-based training into real-world application. It provides an electronic, hands-on platform where learners can practice systematically, apply concepts gradually, and develop practical skills that lead to complete and functional applications. Through this process, students are encouraged to build larger, more robust systems, make use of essential utilities, and integrate unified code effectively. The phased implementation enhances knowledge, adds value, and strengthens confidence by engaging learners in realistic scenarios.

Furthermore, the eProject emphasizes the importance of real-life projects, allowing students to gain practical exposure and prepare for professional challenges. It also fosters creativity and problem-solving by guiding learners to design applications that are more useful and adaptable to actual needs. Continuous mentoring, including support through email, ensures that students are not only guided but also motivated throughout the journey.

We would like to sincerely thank our professor and fellow classmates for their kind guidance and support during the project. Despite some mistakes made along the way, these experiences have become valuable lessons that help us improve and broaden our knowledge in web development. This project has given us greater confidence, practical skills, and motivation to keep learning.

With the lessons and experience gained, we are determined to continue our journey of growth and strive for future success.

Yours sincerely,
Team Group 03

Synopsis

The objective of our semester project is to provide students with practical experience by working on a real-life web application using PHP, Laravel, and MySQL. Rather than focusing only on theoretical knowledge, this project emphasizes applying what we have learned in class to build a functional system. By doing so, we not only strengthen our understanding of programming concepts but also develop the ability to design and implement complete solutions.

This project is not intended to re-teach the basics of PHP or Laravel, but to give us a real scenario where these tools can be applied to create dynamic and interactive web applications. It encourages us to connect various concepts, such as database management, backend logic, and user interface design, into one unified application.

The project is designed for students who have already completed the PHP and MySQL modules, and it should ideally be carried out during lab sessions with faculty guidance when necessary. Through this process, we gain exposure to essential skills such as CRUD operations, authentication, routing, and database connectivity.

Ultimately, this project helps us develop a deeper understanding of how modern web applications are built and prepares us for future challenges in professional development.

Review 01

Problem Definition

Our project focuses on developing an E-Learning Management System (LMS) designed to enhance the teaching and learning experience in a professional and accessible way. The system aims to provide students, instructors, and administrators with a centralized platform to manage courses, resources, and communication effectively.

The main objective of the LMS is to create a responsive, user-friendly, and scalable solution that supports both individual learners and institutional needs. This platform allows users to easily navigate through course categories, access detailed learning materials, view grades, submit assignments, and participate in discussions. Instructors, on the other hand, can design courses, upload content, evaluate students, and track progress seamlessly.

To ensure usability, the system is developed with the following criteria:

1. User-friendly design – intuitive navigation and responsive layout.
2. Well-structured menus – clear organization of courses, lessons, and assessments.
3. Consistent design scheme – professional interface with modern visuals.
4. Smooth interactions – reliable performance and seamless communication tools.

Customer Requirements Specifications

I. CRS/Requirements Specifications

Client/Project Undertaken:		
1.	List of inputs to the system	
		User login credentials (username, password)
		Student registration details (name, email, course selection, etc.)
		Instructor/teacher information (profile, subjects taught, schedules)
		Course content (documents, videos, quizzes, assignments)
		Assessment data (quiz answers, assignment submissions)
		Feedback/ratings from students
		Admin configurations (course creation, user management, reports)
2.	List of outputs expected from the system	
		User authentication confirmation (login success/failure)
		Course catalog and details available to students
		Learning materials accessible by enrolled students
		Progress tracking (course completion percentage, grades, scores)
		Reports for instructors (student performance, attendance, submissions)
		Notifications/announcements (deadlines, exam schedules, updates)
		Certificates of completion for students
		Admin dashboard with statistics and analytics
3.	Overview of processes involved in the system	

	Prepared by (Student)	**Approved by (Faculty)**
	Project Group No: 03	LE THANH NHAN
Signature		
Date		

		User Management: Register, login, update profile, manage roles (student, teacher, admin)
		Course Management: Create, upload, update, and delete courses/content
		Learning Process: Students enroll in courses, access materials, watch lectures, complete assignments
		Assessment Process: Conduct quizzes, evaluate assignments, calculate grades automatically
		Progress Tracking: Monitor student performance and generate progress reports
		Communication: Forum, messaging, announcements between teachers and students
		Administration: Manage users, courses, permissions, and generate system reports

Admin/Project Undertaken:		
1.	List of inputs to the system	
		User login credentials (username, password)
		Student registration details (name, email, course selection)
		Instructor details (profile, subject expertise, timetable)
		Course materials (PDFs, videos, presentations, assignments)
		Quiz/assignment submissions by students
		Feedback, comments, and ratings from users
		Admin settings (role assignment, course management, reporting options)
2.	List of outputs expected from the system	

	Prepared by (Student)	**Approved by (Faculty)**
	Project Group No: 03	LE THANH NHAN
Signature		
Date		

		Authentication results (login success/failure message)
		List of available courses and modules
		Learning resources accessible to enrolled students
		Quiz/assignment results and grades
		Student progress reports and completion certificates
		Notifications and announcements (exam dates, deadlines, updates)
		Analytics dashboard for admin (active users, performance stats, course usage)
3.	Overview of processes involved in the system	
		User Management: Registration, login, role assignment (student, teacher, admin)
		Course Management: Create, upload, update, and delete courses/materials
		Learning Process: Students access content, complete lessons, submit assignments
		Assessment: Online quizzes, automated grading, manual evaluation of assignments
		Progress Tracking: Monitoring student progress and generating reports
		Communication: Forums, messaging, and announcements between users
		Administration: Manage system settings, generate analytics, oversee overall system performance

Hardware and Software required for implementing the project

Hardware Requirements

- Client (Student/Teacher):

  - Computer/Laptop/Smartphone with internet connection
  - CPU: Dual Core or higher
  - RAM: Minimum 4GB
  - Hard Disk: Minimum 10GB free space
  - Modern Web Browser: Chrome, Firefox, Edge
- Server (Deployment):
  - CPU: Quad Core or higher
  - RAM: Minimum 16GB
  - SSD Storage: ≥ 256GB
  - High-speed Internet connection with 99.9% uptime
  - Operating System: Linux/Windows Server

Software Requirements

- Frontend: ReactJS, HTML5, CSS3, JavaScript, Bootstrap/Tailwind
- Base: PHP (Laravel Framework)
- Database: MySQL
- IDE/Editor: Visual Studio Code / NetBeans
- Web Server: Apache/Nginx
- Version Control: Git/GitHub

- Testing Tools: Postman, PHPUnit/Jest
- Deployment (Optional): Docker, AWS/Azure/Heroku

II. CRS/Customer Acceptance Criteria

S.No	Customer's Acceptance Criteria
1	Given a registered student, when they log in, then they should be able to access their enrolled courses
2	Given a teacher uploads study materials, when students access the course, then they should be able to view/download them.
3	Given an administrator creates an exam, when students log in at the scheduled time, then they should be able to attempt the online test.
4	Given a student submits an exam, when results are processed, then the student should be able to view their score immediately.
5	Given a teacher posts announcements, when students log in, then they should see the notifications.
6	Given multiple users access the system simultaneously, when up to 100 users log in, then the system should remain stable and responsive.
7	Given incorrect login credentials, when a user attempts login, then the system should display an appropriate error message.
8	Given a student loses internet connection, when they reconnect, then their learning progress should be saved and resumed.

III. Project Plan

1. Project Details	
+	Name of the Client: FPT Aptech
+	Date of Project Plan: 20 August 2025
+	Project Vision/Objectives: To build an online E-Learning Management System that supports teaching, learning, course management, online assessments, and result tracking, providing a modern and accessible learning environment.
+	Scope: ● User management (Admin, Teacher, Student) ● Course, lecture, and study material management ● Online examination with automatic grading ● Notification and feedback system for communication between teachers and students
+	Our understanding of the client organization: The client requires a scalable system that can handle a large number of students, provide cross-platform accessibility, and ensure a user-friendly interface for smooth adoption.
+	Project Organization with Responsibilities and Authorities: ● Administrator: Manage users, courses, and overall system configuration ● Teacher: Create courses, upload materials, create/manage exams, evaluate students

	• Student: Enroll in courses, access materials, complete assignments, take exams online
2. Project Initiation/Requirement Documents	
	Project Charter: Document defining E-Learning Management System objectives, scope, and stakeholder roles for FPT Aptech
	Business Requirements Document (BRD): Specifications for online teaching, learning, course management, and assessment functionalities
	Functional Requirements Specification (FRS): Detailed user stories and acceptance criteria for Admin, Teacher, and Student roles
	Technical Architecture Document: System design using ReactJS frontend, PHP Laravel backend, and MySQL database
	User Interface/User Experience (UI/UX) Design: Wireframes and mockups for responsive web application across devices
	Security Requirements: Authentication, authorization, data encryption, and secure exam environment specifications
	Performance Requirements: System capacity for 100+ concurrent users with 99.9% uptime guarantee
3. Deliverables	
	Web-based E-Learning Platform: Complete responsive application accessible via modern browsers
	Admin Dashboard: User management, course oversight, system configuration, and reporting modules
	Teacher Portal: Course creation, material upload, exam management, and student evaluation tools

	Student Interface: Course enrollment, content access, online examination, and progress tracking features
	Database Schema: Normalized MySQL database with user, course, exam, and result management tables
	API Documentation: RESTful API endpoints documentation for frontend-backend communication
	User Manuals: Comprehensive guides for administrators, teachers, and students
	Deployment Package: Docker containers and cloud deployment scripts for production environment
4. Project Dependencies	
	External Dependencies: ● Internet connectivity and hosting infrastructure availability ● Third-party libraries and frameworks (ReactJS, Laravel) compatibility updates ● SSL certificate procurement for secure HTTPS communication ● Cloud service provider (AWS/Azure/Heroku) service availability
	Internal Dependencies: ● FPT Aptech content and curriculum materials for course population ● User data and existing student/teacher information for system migration ● IT infrastructure setup and server configuration by client technical team ● Stakeholder availability for requirement validation and testing phases
5. Major Project Milestones	
	Milestone 1 (Week 1): Requirements finalization, technical architecture approval, and project setup

	Milestone 2 (Week 2): Database design completion, UI/UX prototypes approval, and development environment setup
	Milestone 3 (Week 3): Backend API development and authentication system implementation
	Milestone 4 (Week 4): Frontend development and API integration for core functionalities
	Milestone 5 (Week 5): Online examination module, grading system implementation, and system integration testing
	Milestone 6 (Week 6): User acceptance testing, deployment preparation, and project handover completion
6. Quality Plan	
+	Review Activities:
	Code Reviews: Peer review process for all developed modules using Git pull request workflow
	Architecture Reviews: Weekly technical architecture assessment to ensure scalability and maintainability
	Requirements Traceability: Regular verification that all customer acceptance criteria are being addressed
	UI/UX Reviews: Usability testing sessions with sample users from each role (Admin, Teacher, Student)
	Security Reviews: Penetration testing and vulnerability assessment for authentication and data protection
	Performance Reviews: Load testing with simulated 100+ concurrent users to validate system capacity

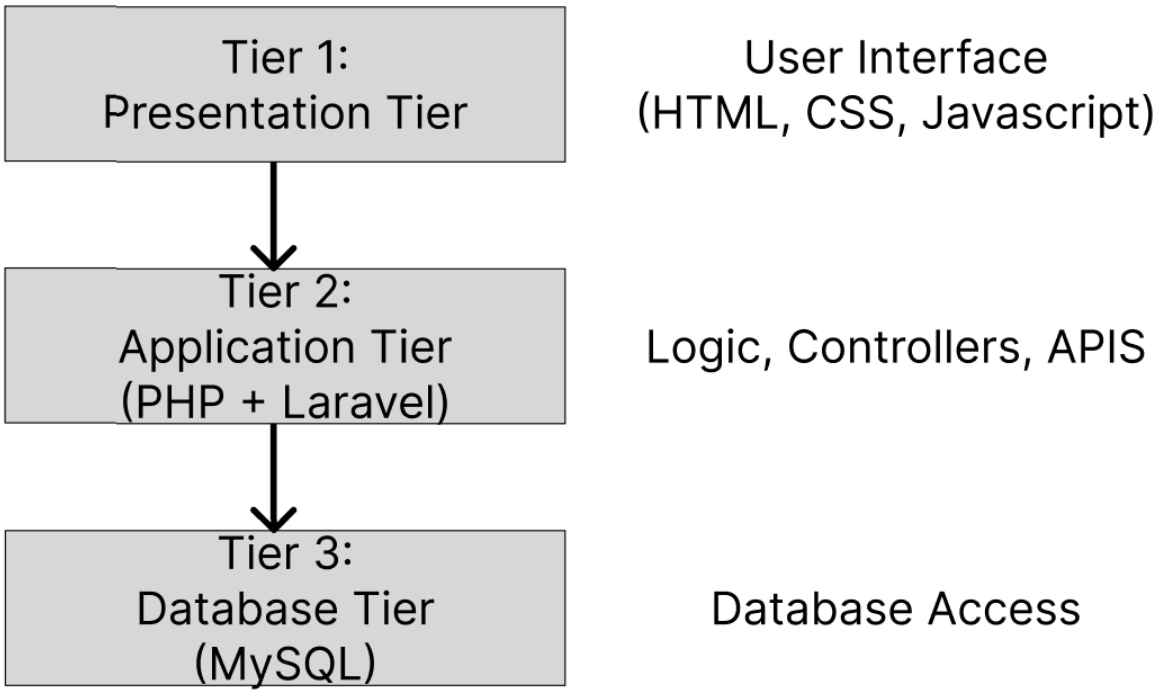
	Database Reviews: Query optimization and data integrity validation for MySQL operations
+	Testing Activities:
	Unit Testing: Individual component testing using PHPUnit for backend and Jest for frontend modules
	Integration Testing: API endpoint testing using Postman to ensure seamless frontend-backend communication
	System Testing: End-to-end testing of complete user workflows for all three user roles
	User Acceptance Testing: Client validation of all acceptance criteria with real-world scenarios
	Performance Testing: Load and stress testing to verify system handles 100+ concurrent users
	Security Testing: Authentication bypass attempts, SQL injection, and cross-site scripting vulnerability tests
	Compatibility Testing: Cross-browser testing on Chrome, Firefox, and Edge browsers
	Mobile Responsiveness Testing: Verification of functionality across smartphones and tablets
+	Backup and recovery strategies:
	Database Backup: Automated daily MySQL database backups with 30-day retention policy stored in separate cloud storage
	Application Code Backup: Version control through Git repository with multiple remote backup locations

	System Configuration Backup: Weekly snapshots of server configurations and deployment scripts
	Disaster Recovery Plan: Complete system restoration procedure with Recovery Time Objective (RTO) of 4 hours
	Data Recovery Testing: Monthly backup restoration drills to ensure data integrity and recovery processes
	Incremental Backups: Hourly incremental backups during peak usage hours to minimize data loss
	Cloud Storage Redundancy: Multi-region backup storage to protect against regional outages or disasters

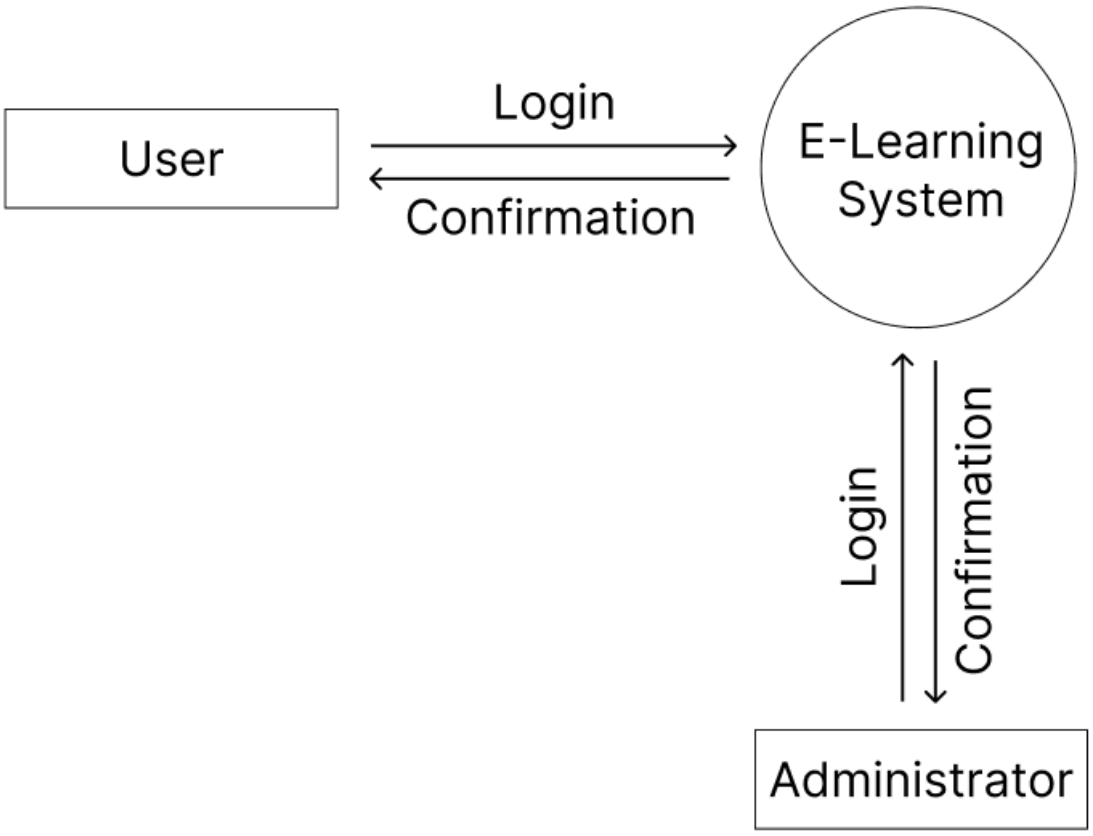
Review 02

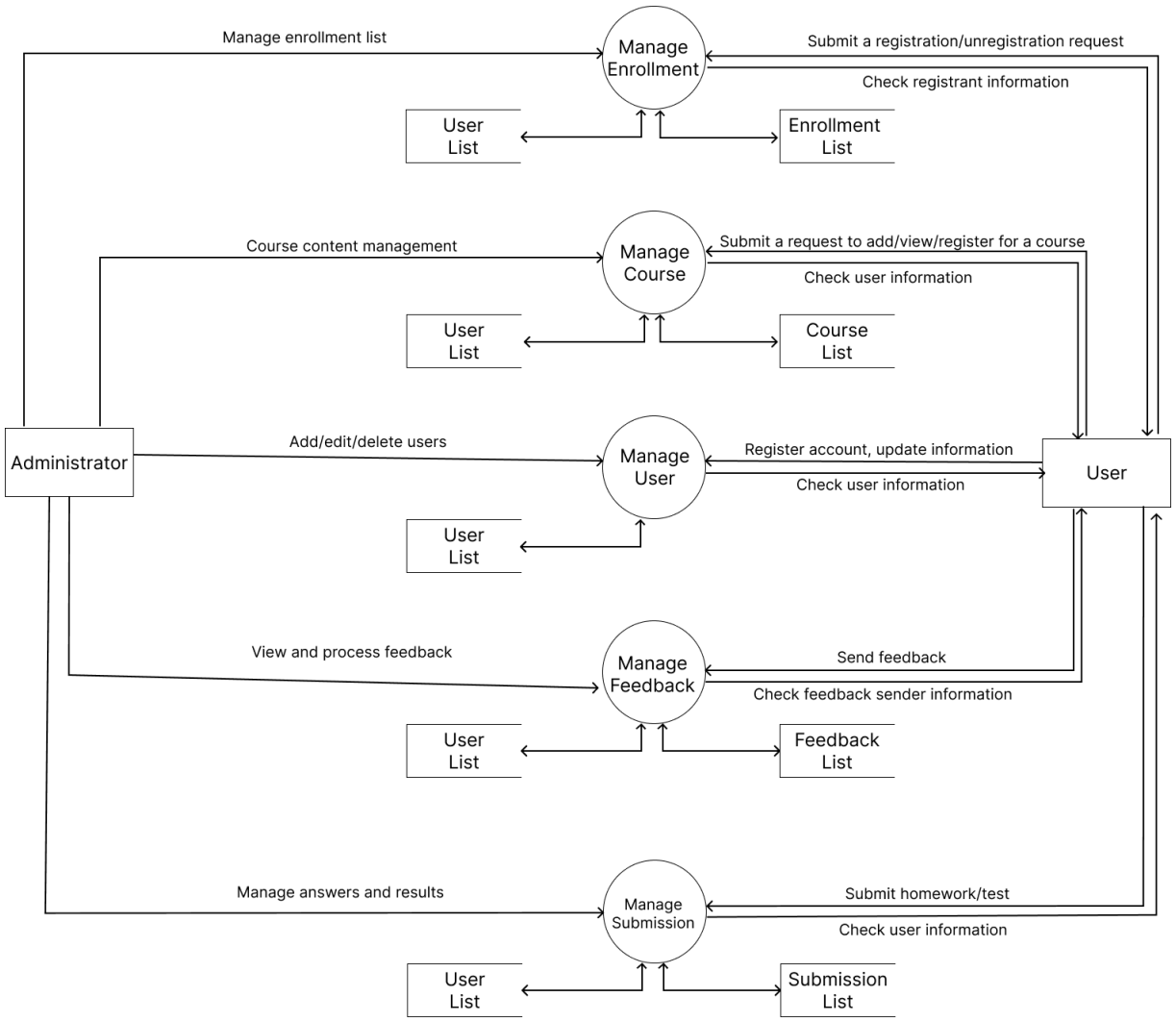
Architecture and design of the Application

I. Application Architecture



II. Data Flow Diagram

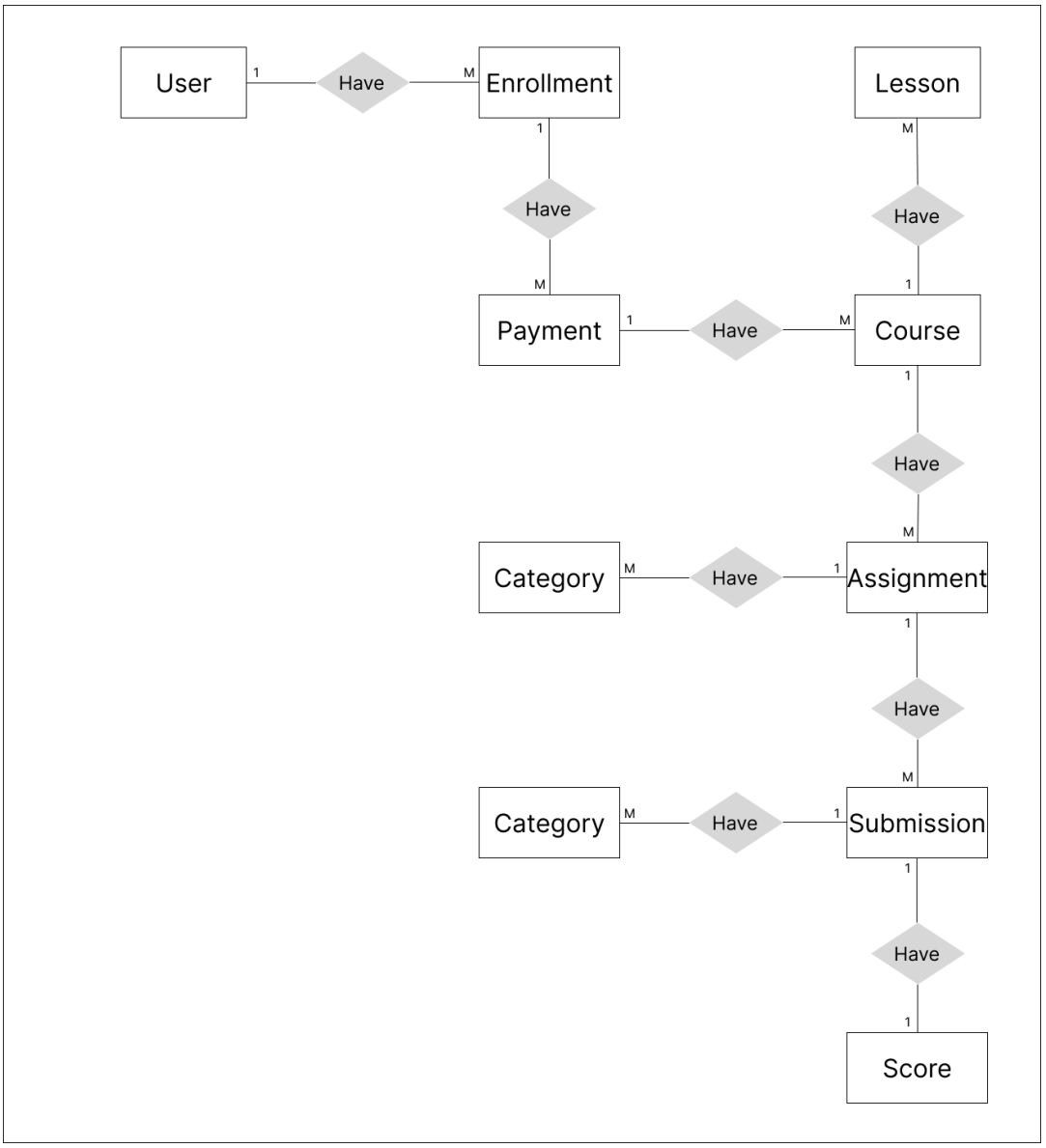




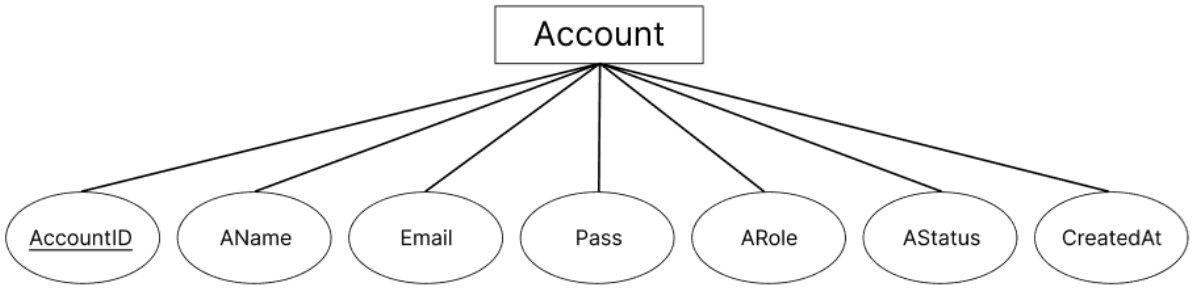
	Prepared by (Student)	**Approved by (Faculty)**
	Project Group No: 03	LE THANH NHAN
Signature		
Date		

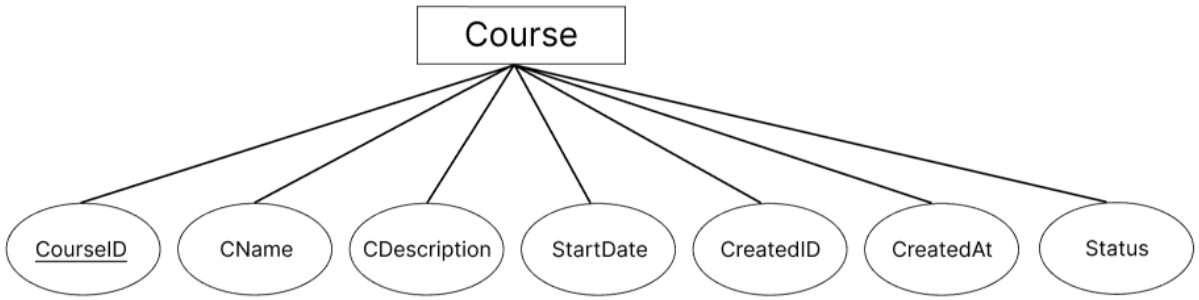
Entity Relationship Diagram

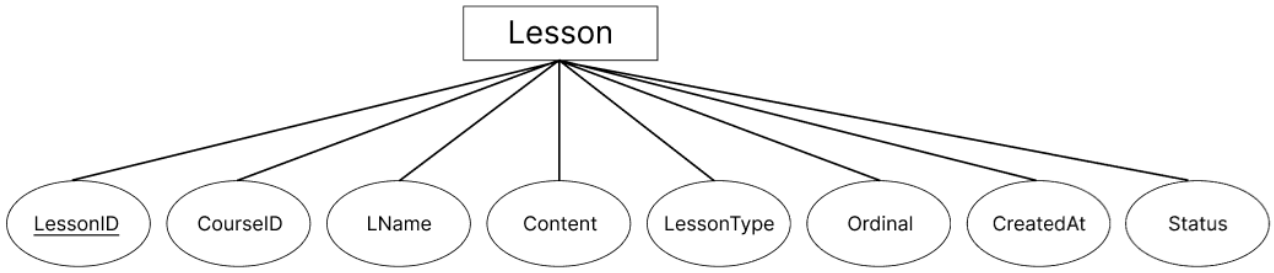
I. ERD

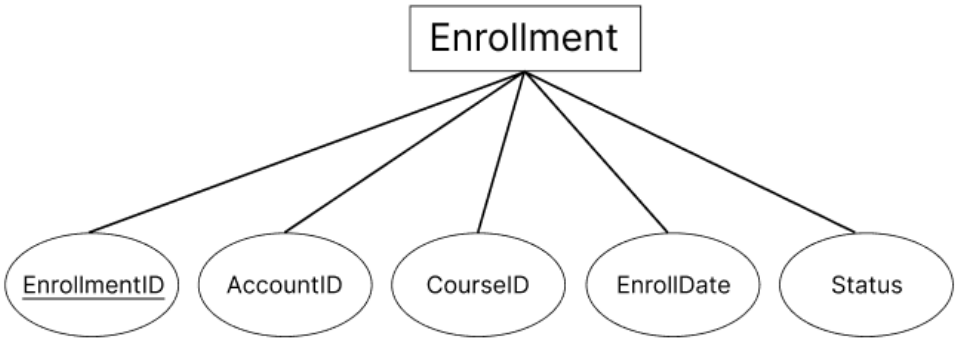


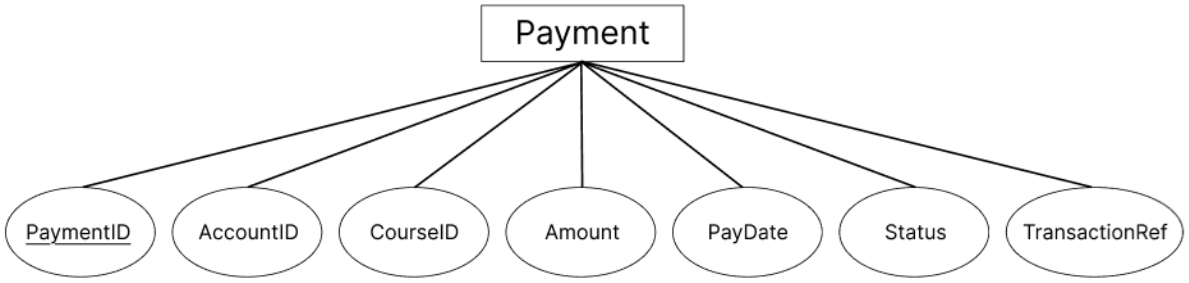
II. ERD Properties

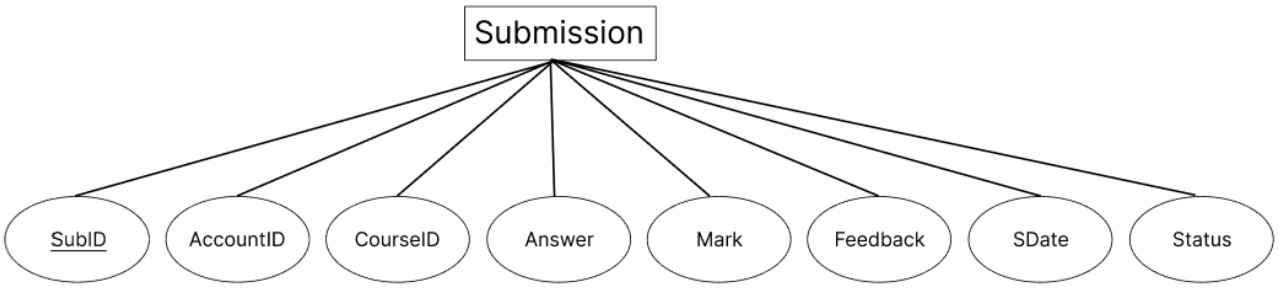





Lesson
LessonID
CourseID
LName
Content
LessonType
Ordinal
CreatedAt
Status


Enrollment
EnrollmentID
AccountID
CourseID
EnrollDate
Status


Payment
PaymentID
AccountID
CourseID
Amount
PayDate
Status
TransactionRef


Submission
SubID
AccountID
CourseID
Answer
Mark
Feedback
SDate
Status

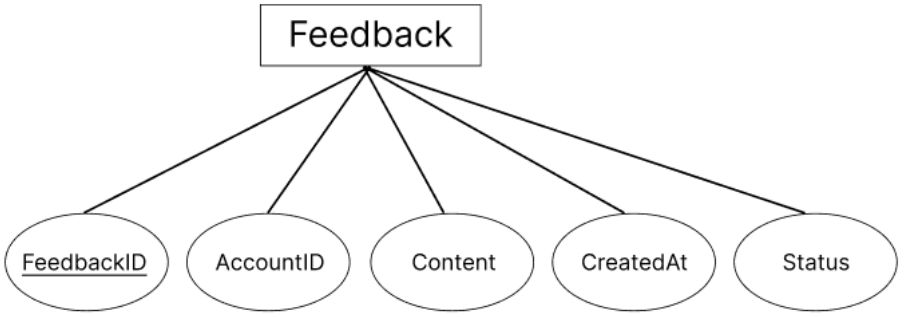

Feedback
FeedbackID
AccountID
Content
CreatedAt
Status

Table Design

I. Table Design

Accounts					
Serial	**Fields**	**Keys**	**Data Type**	**Constraint**	**Description**
1	AccountID	Primary key	int	Identity	Unique identifier for each user (auto-increment).
2	AName		nvarchar(100)	Not null	Full name or display name of the user.
3	Email		nvarchar(100)	Unique, not null	Email address of the user (used for login and notifications).
4	Pass		nvarchar(100)	Not null	Encrypted password for account authentication.
5	ARole		nvarchar(50)	Not null	Role of the user (e.g., Admin, Instructor, Student).
6	AStatus		bit	Default 1	Account status (1 = active, 0 = inactive).
7	CreatedAt		datetime	Default getdate()	Date and time when the account was created.

Courses

Serial	Fields	Keys	Data Type	Constraint	Description
1	CourseID	Primary key	int	Identity	Unique identifier for each course (auto-increment).
2	CName		nvarchar(200)	Not null	Name of the course.
3	CDescription		nvarchar(max)	Null	Detailed description of the course content.
4	StartDate		datetime	Not null	Course start date
5	CreatorID	Foreign key	int	Not null	References Account(AccountID), identifies the creator or instructor of course.
6	CreatedAt		datetime	Default getdate()	Date and time when the course was created.
7	Status		bit	Default 1	Account status (1 = active, 0 = inactive).

Enrollments

Serial	Fields	Keys	Data Type	Constraint	Description
1	EnrollmentID	Primary key	int	Identity	Unique identifier for each enrollment record (auto-increment).
2	AccountID	Foreign key	int	Not null	References User(UserID), identifies the student who enrolled.
3	CourseID	Foreign key	int	Not null	References Course(CourseID), identifies the course being enrolled.
4	EnrollDate		datetime	Default getdate()	Date and time when the enrollment was created.
5	EStatus		nvarchar(50)	Null	Enrollment status (e.g., Active, Completed, Dropped).

Lessons

Serial	Fields	Keys	Data Type	Constraint	Description
1	LessonID	Primary key	int	Identity	Unique identifier for each lesson (auto-increment).
2	CourseID	Foreign key	int	Not null	References Course(CourseID), identifies which course the lesson belongs to.
3	LName		nvarchar(200)	Not null	Title or name of the lesson.
4	Content		nvarchar(max)	Null	Lesson content (text, material, or description).
5	LessonType		nvarchar(50)	Null	Type of lesson (e.g., Video, Quiz, Document).
6	Ordinal		int	Null	Order of the lesson in the course sequence.
7	CreatedAt		datetime	Default getdate()	Date and time when the lesson was created.
8	LStatus		bit	Default 1	Account status (1 = active, 0 = inactive).

Submissions

Serial	Fields	Keys	Data Type	Constraint	Description
1	SubID	Primary key	int	Identity	Unique identifier for each submission (auto-increment).
2	AccountID	Foreign key	int	Not null	References User(UserID), identifies the student who made the submission.
3	CourseID	Foreign key	int	Not null	References Lesson(LessonID), identifies the lesson being submitted.
4	Answer		nvarchar(max)	Null	Content of the student's answer or uploaded solution.
5	Mark		float	Null	Score/grade given for the submission.
6	Feedback		nvarchar(max)	Null	Teacher's comments or feedback on the submission.
7	SDate		datetime	Default getdate()	Date and time when the submission was created.
8	LStatus		bit	Default 1	Account status (1 = active, 0 = inactive).

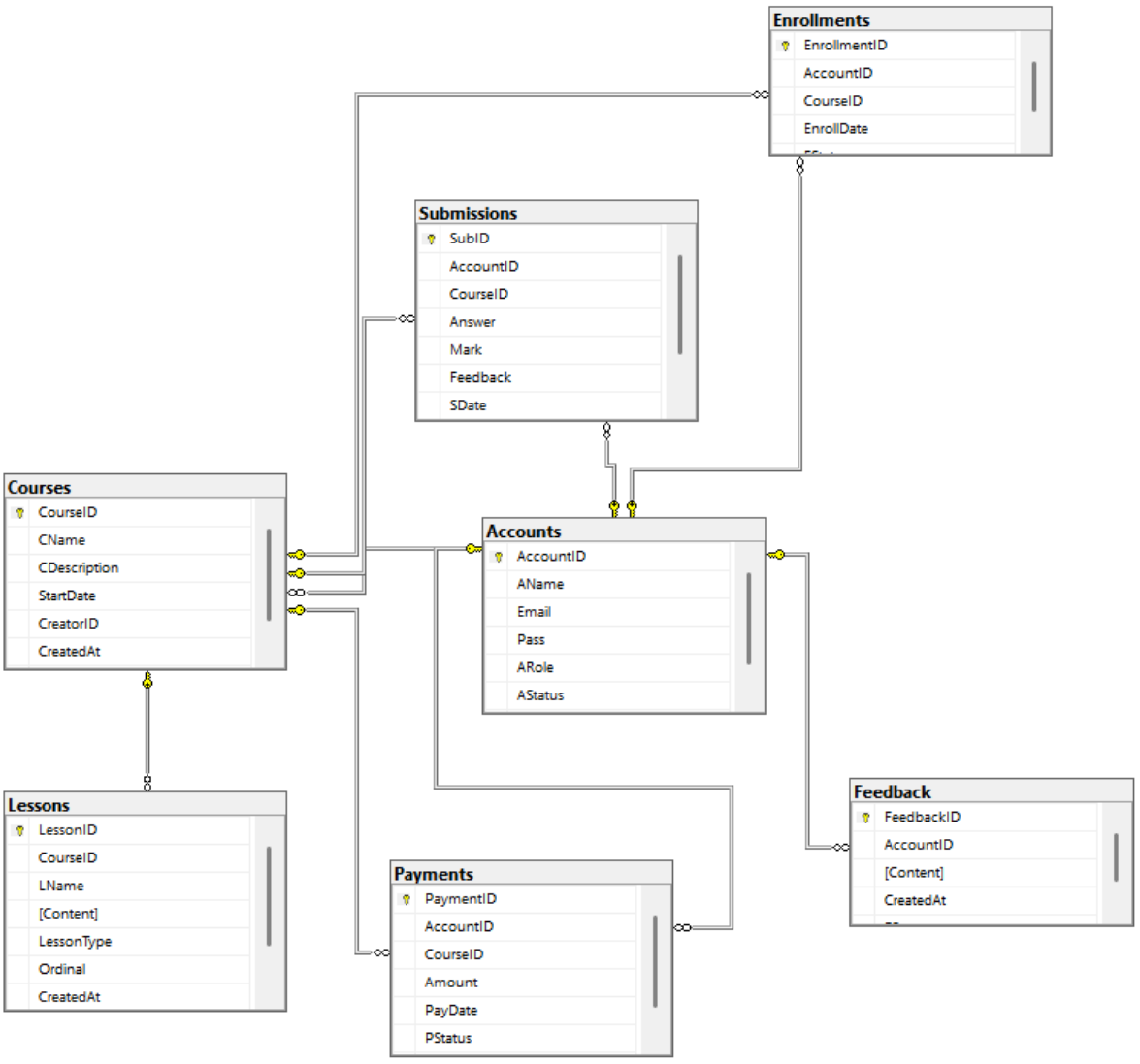
Payments

Serial	Fields	Keys	Data Type	Constraint	Description
1	PaymentID	Primary key	int	Identity	Unique identifier for each payment transaction (auto-increment).
2	AccountID	Foreign key	int	Not null	References User(UserID), identifies the user who made the payment.
3	CourseID	Foreign key	int	Not null	References Course(CourseID), identifies the course being purchased.
4	Amount		decimal(10,2)	Not null	Payment amount for the course.
5	PayDate		datetime	Default getdate()	Date and time when the payment was made.
6	PStatus		nvarchar(50)	Null	Current status of payment (e.g., Pending, Completed, Failed, Refunded).
7	TransactionRef		nvarchar(100)	Unique	Unique transaction reference or identifier from payment gateway.

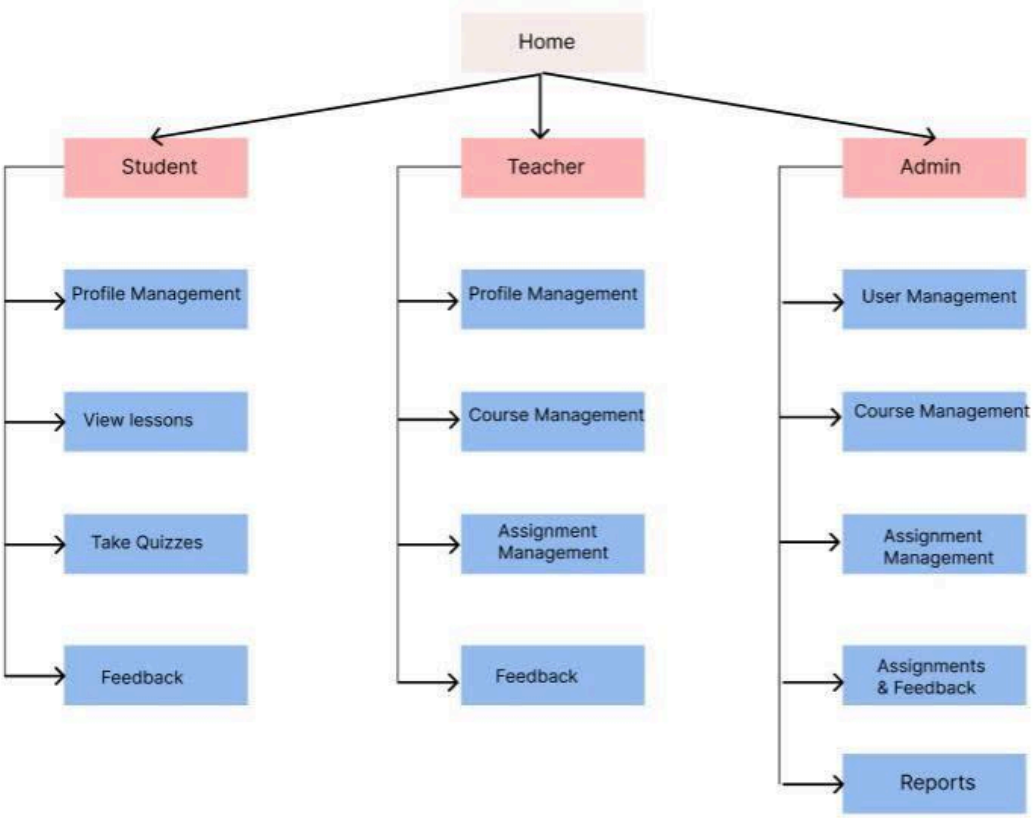
Feedback

Serial	Fields	Keys	Data Type	Constraint	Description
1	FeedbackID	Primary key	int	Identity	Unique identifier for each feedback entry (auto-increment).
2	AccountID	Foreign key	int	Not null	References User(UserID), identifies the user who gave the feedback.
4	Content		nvarchar(max)	Null	The text content of the feedback message.
5	CreatedAt		datetime	Default getdate()	Date and time when the feedback was created.
6	FStatus		bit	Default 1	Feedback status (1 = active/visible, 0 = hidden/inactive).

II. Relationship Diagram



Site Map


Home
Student
Teacher
Admin
Profile Management
View lessons
Take Quizzes
Feedback
Profile Management
Course Management
Assignment Management
Feedback
User Management
Course Management
Assignment Management
Assignments & Feedback
Reports

Algorithm

I. Administrator

1. User Management Algorithm

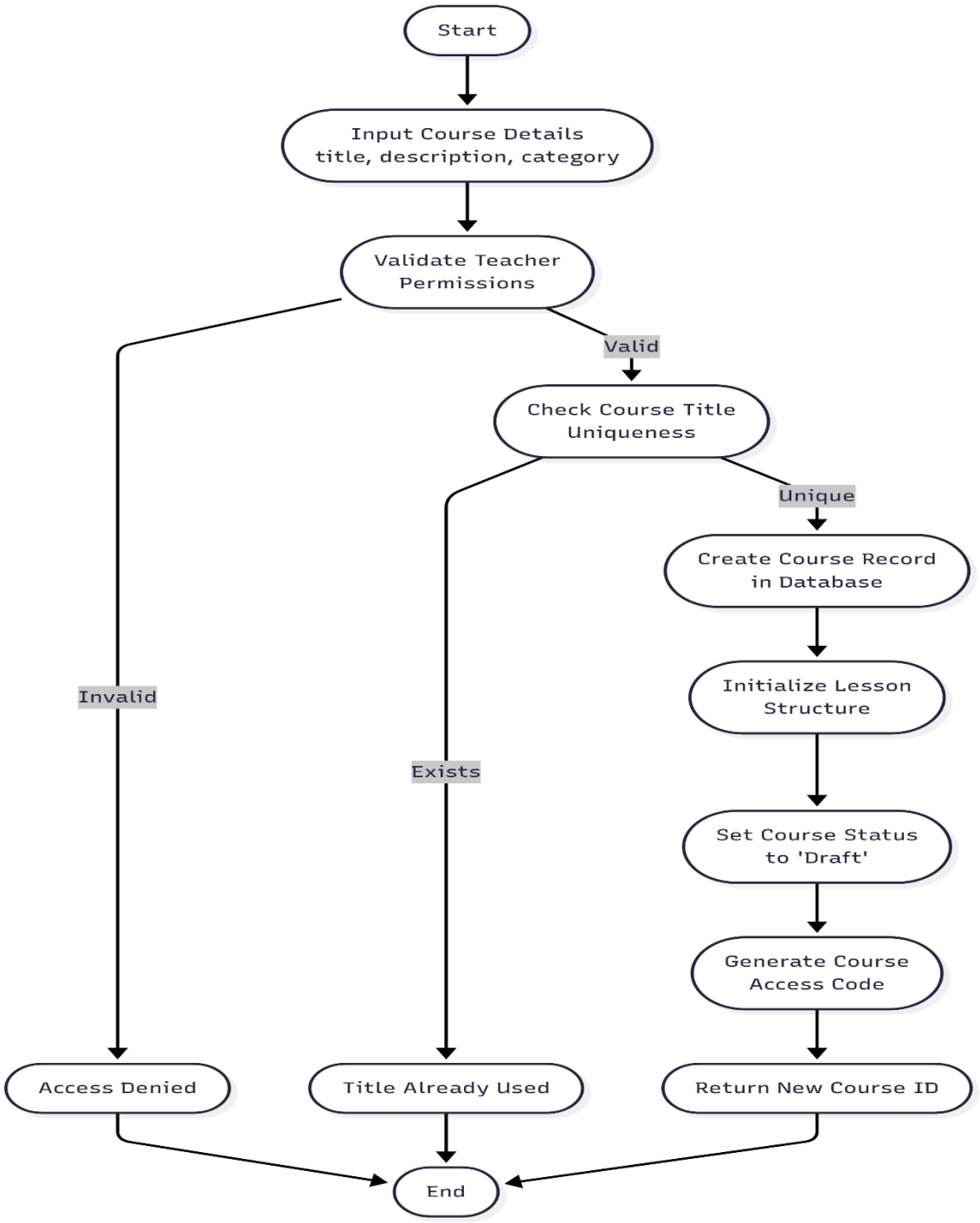
Input: User data (name, email, role, password)

Process:

1. Validate input data format
2. Check email uniqueness in database
3. Encrypt password using bcrypt
4. Insert new user record
5. Assign role-based permissions
6. Send confirmation email

Output: User creation status message

Flowchart:



	Prepared by (Student)	Approved by (Faculty)
	Project Group No: 03	LE THANH NHAN
Signature		
Date		

2. Course Approval Algorithm

Input: Course ID, approval status

Process:

1. Retrieve course details from database
2. Validate course content completeness
3. Check instructor qualifications
4. Update course status to "Active/Rejected"
5. Send notification to instructor
6. Log approval action

Output: Course approval confirmation

3. System Report Generation Algorithm

Input: Report type, date range

Process:

1. Query relevant data from multiple tables
2. Calculate statistics (enrollment, completion rates)
3. Format data for presentation
4. Generate PDF/Excel report
5. Save report to file system

Output: Comprehensive system report

II. Teacher/Staff

1. Course Creation Algorithm

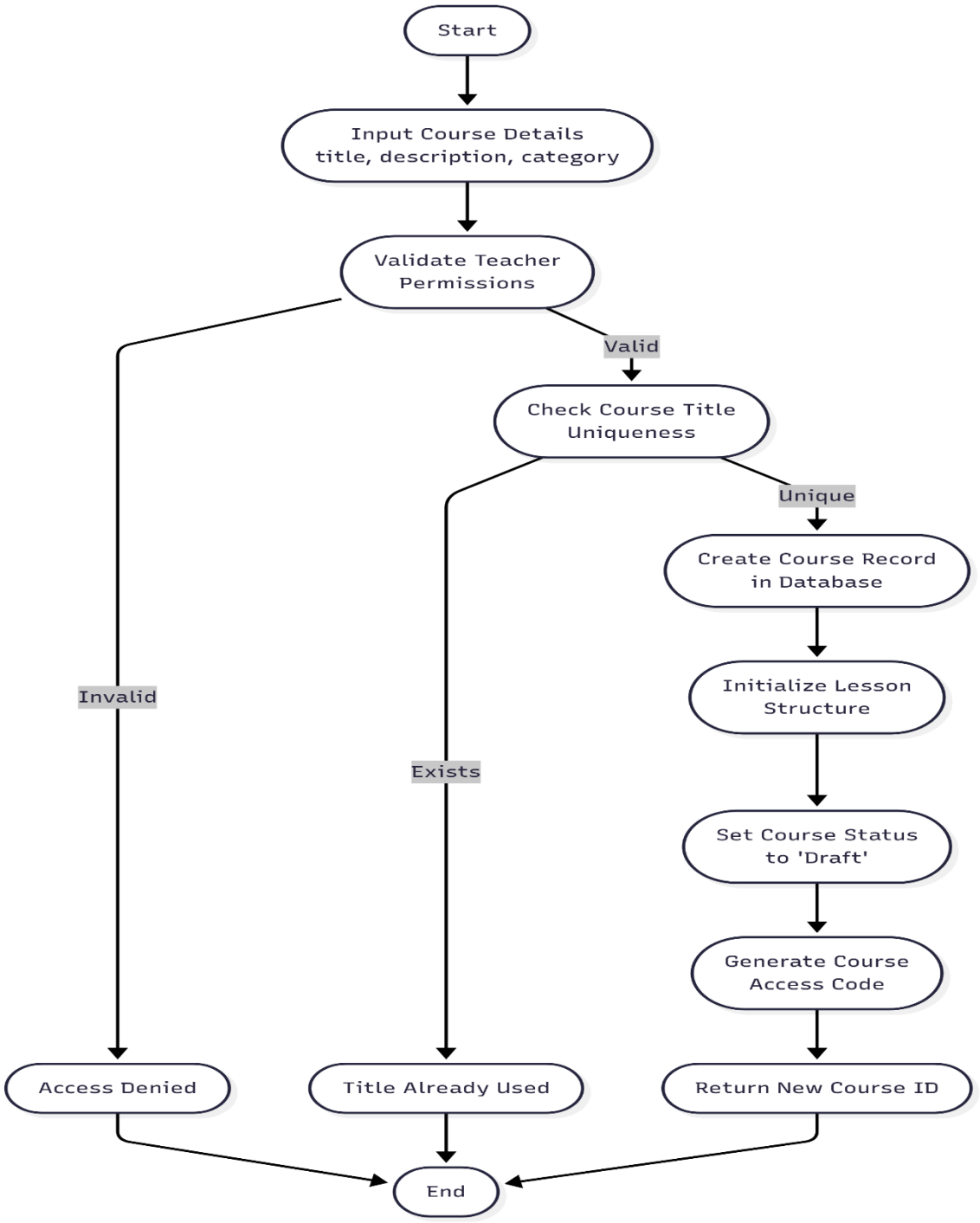
Input: Course details (title, description, category)

Process:

1. Validate teacher permissions
2. Check course title uniqueness
3. Create course record in database
4. Initialize lesson structure
5. Set course status to "Draft"
6. Generate course access code

Output: New course ID

Flowchart:


Start
Input Course Details
title, description, category
Validate Teacher
Permissions
Valid
Invalid
Check Course Title
Uniqueness
Exists
Unique
Create Course Record
in Database
Initialize Lesson
Structure
Set Course Status
to 'Draft'
Generate Course
Access Code
Access Denied
Title Already Used
Return New Course ID
End

2. Content Upload Algorithm

Input: File data, course ID, content type

Process:

1. Validate file size and format
2. Scan for security threats
3. Generate unique filename
4. Upload to storage server
5. Create content metadata record
6. Link content to specific lesson

Output: Upload confirmation message

3. Assessment Grading Algorithm

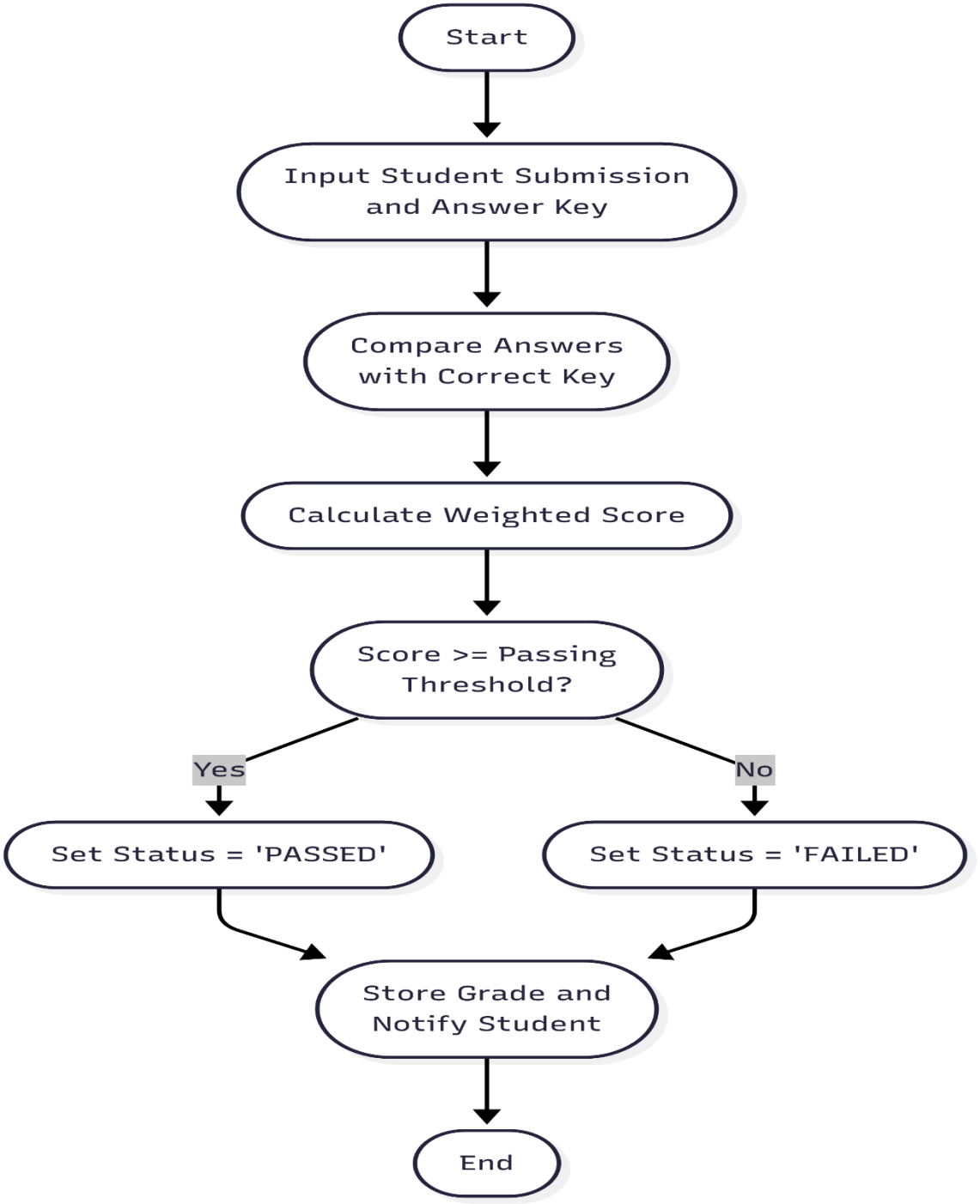
Input: Student submission, answer key

Process:

1. Compare answers with correct key
2. Calculate weighted score
3. Determine pass/fail status
4. Store grade and notify student

Output: Final grade and feedback

Flowchart:


Start
Input Student Submission and Answer Key
Compare Answers with Correct Key
Calculate Weighted Score
Score >= Passing Threshold?
Yes
No
Set Status = 'PASSED'
Set Status = 'FAILED'
Store Grade and Notify Student
End

III. Student

1. Course Enrollment Algorithm

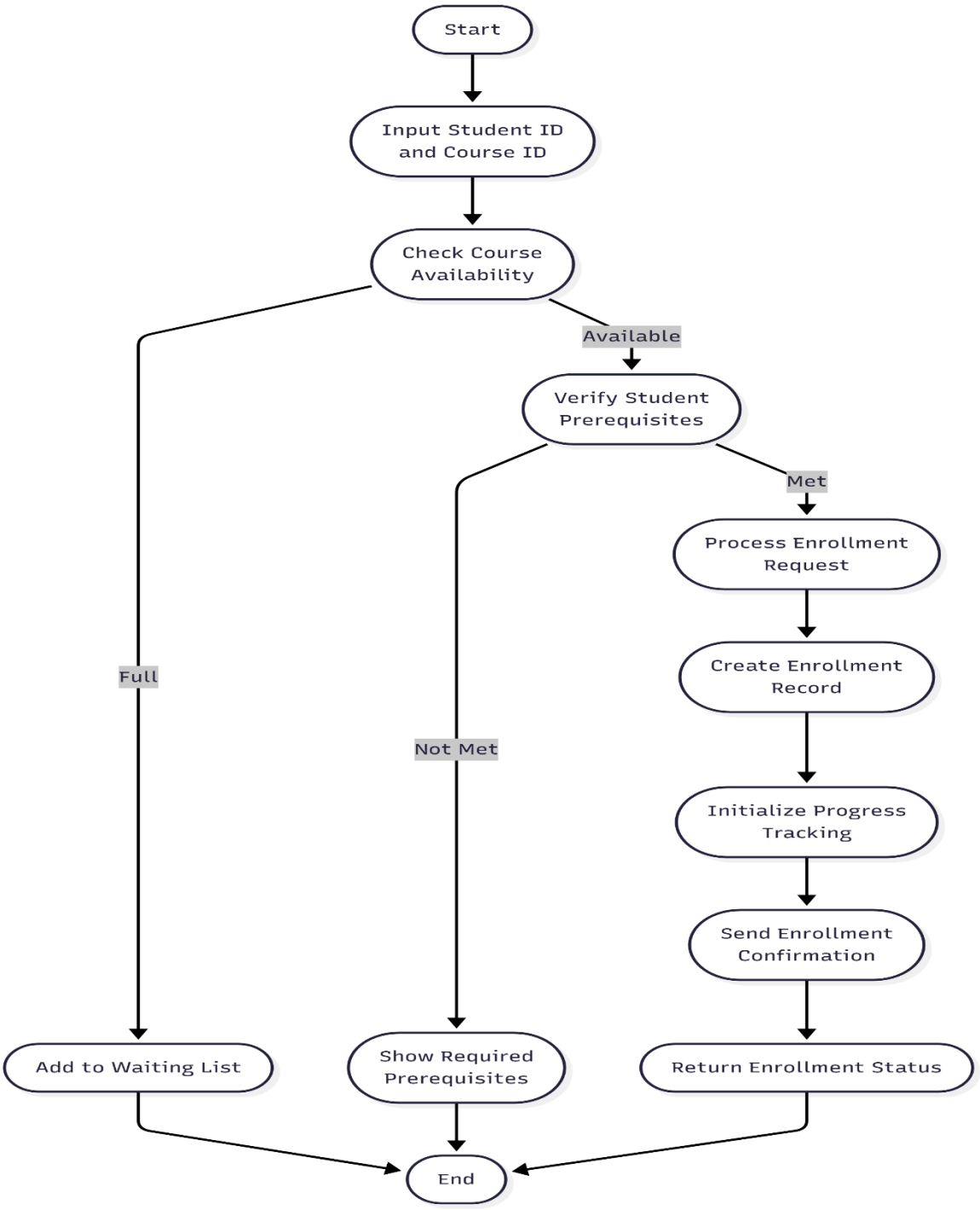
Input: Student ID, course ID

Process:

1. Check course availability and capacity
2. Verify student prerequisites
3. Process enrollment request
4. Create enrollment record
5. Initialize progress tracking
6. Send enrollment confirmation

Output: Enrollment status

Flowchart:



	Prepared by (Student)	Approved by (Faculty)
	Project Group No: 03	LE THANH NHAN
Signature		
Date		

2. Assignment Submission Algorithm

Input: Assignment file, student ID, assignment ID

Process:

1. Validate submission deadline
2. Check file format compatibility
3. Upload assignment to server
4. Create submission record
5. Update assignment status
6. Notify teacher of new submission

Output: Submission confirmation

3. Progress Tracking Algorithm

Input: Student activity data

Process:

1. Record lesson completion
2. Update course progress percentage
3. Check milestone achievements
4. Generate progress report
5. Update student dashboard
6. Send progress notifications

Output: Updated progress status

IV. System-Level Algorithms

1. Authentication Algorithm

Input: Username, password

Process:

1. Retrieve user record from database
2. Verify account status (active/inactive)
3. Compare password with stored hash
4. Generate session token if valid
5. Update last login timestamp
6. Set session expiration time

Output: Authentication result and session token

2. Notification System Algorithm

Input: User ID, message type, content

Process:

1. Check user notification preferences
2. Determine notification priority
3. Format message according to type
4. Send notification via selected channels
5. Log notification delivery
6. Store in notification history

Output: Notification delivery status

3. Database Backup Algorithm

Input: Backup type (full/incremental)

Process:

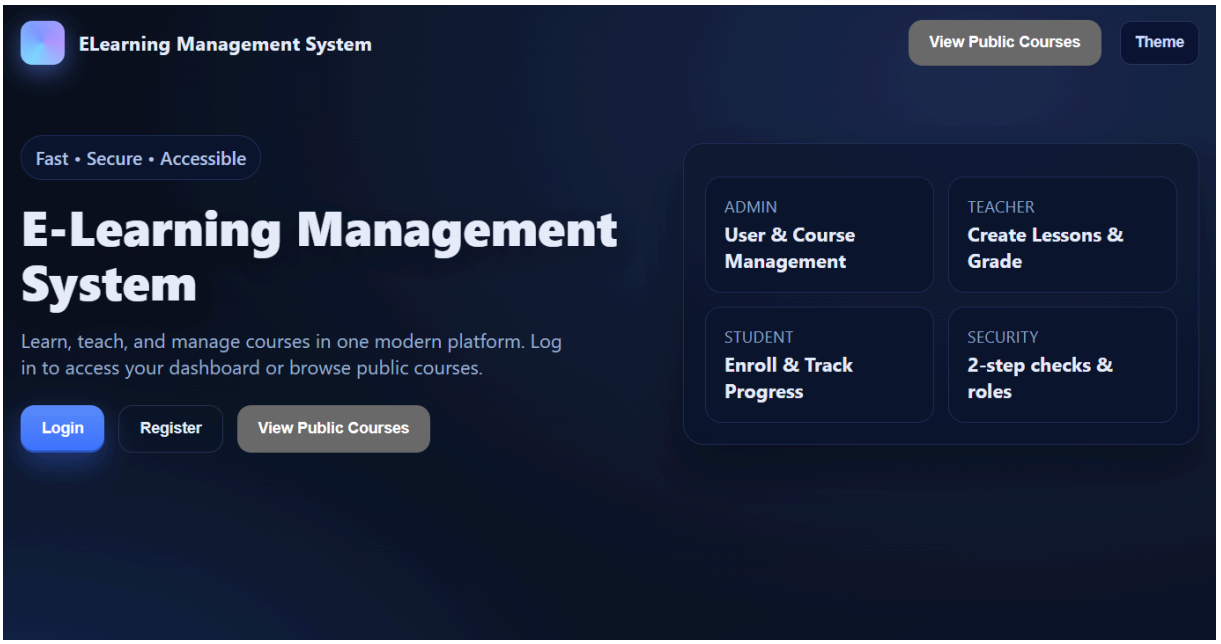
1. Check system resources availability
2. Create backup timestamp identifier
3. Export database according to backup type
4. Compress backup file
5. Upload to secure cloud storage
6. Verify backup file integrity
7. Update backup log records

Output: Backup completion status

Review 03

Screen shots

1. HomePage

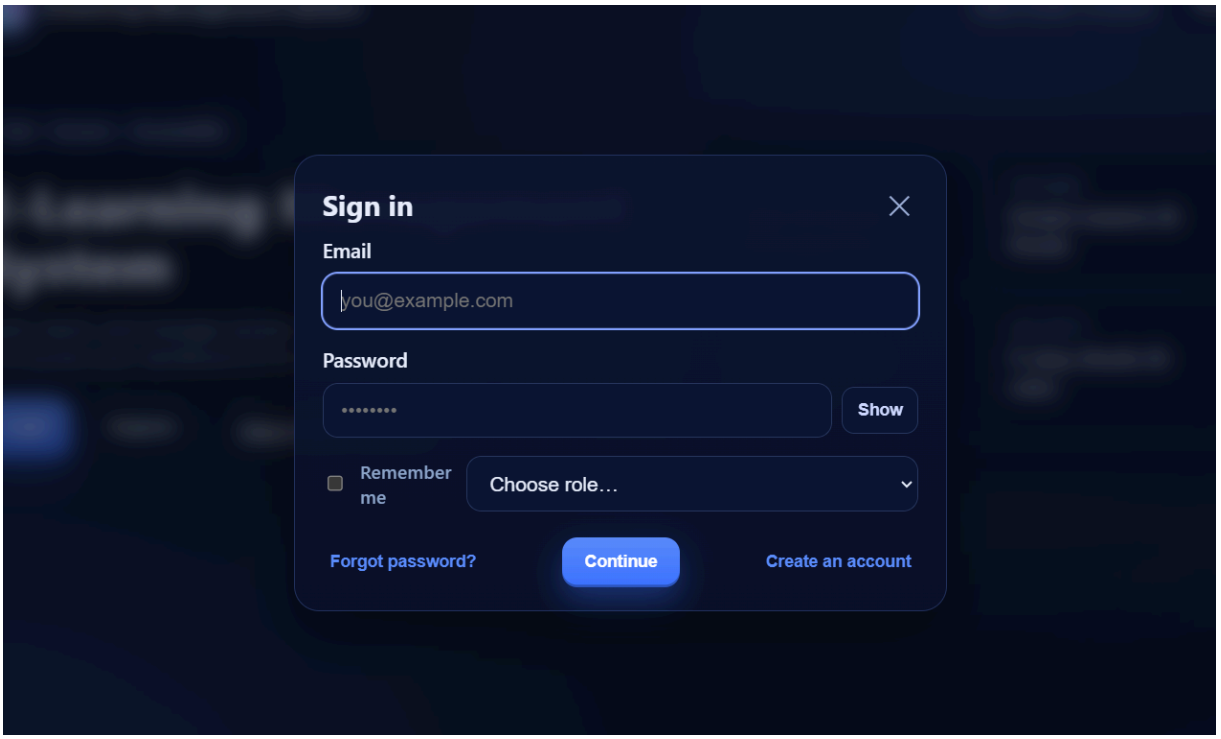


Description : This is Homepage where you can login, register, view courses…

From :

To : Login, Register, Courses….

2. Login

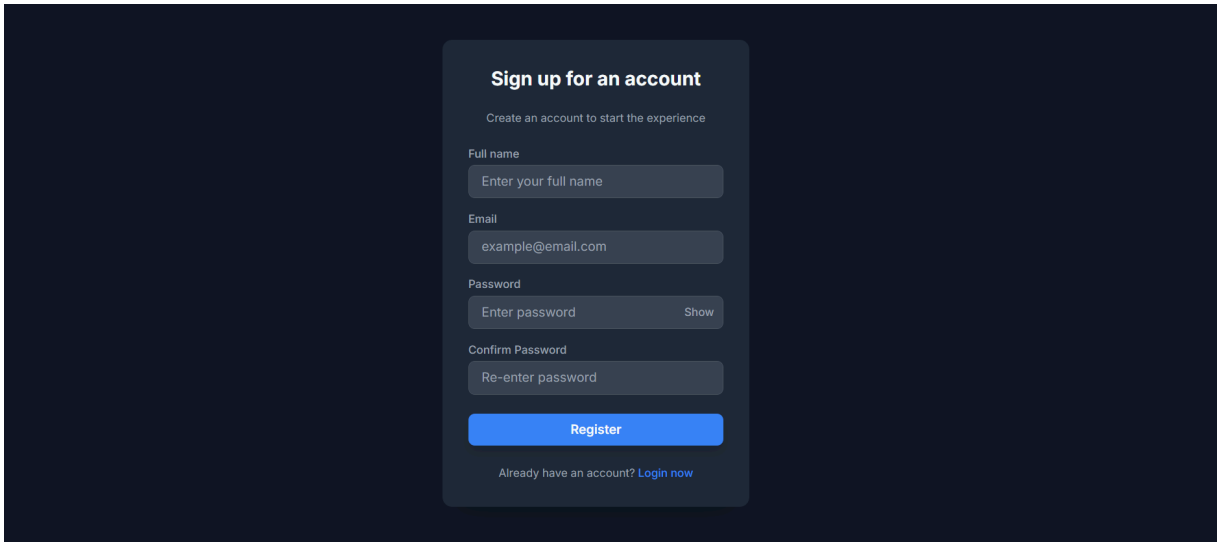


Description : This is where you can login to the web

From : HomePage

To : Public courses.

3. Register



Description : This is where you can register to the web

From : HomePage

To : Public courses

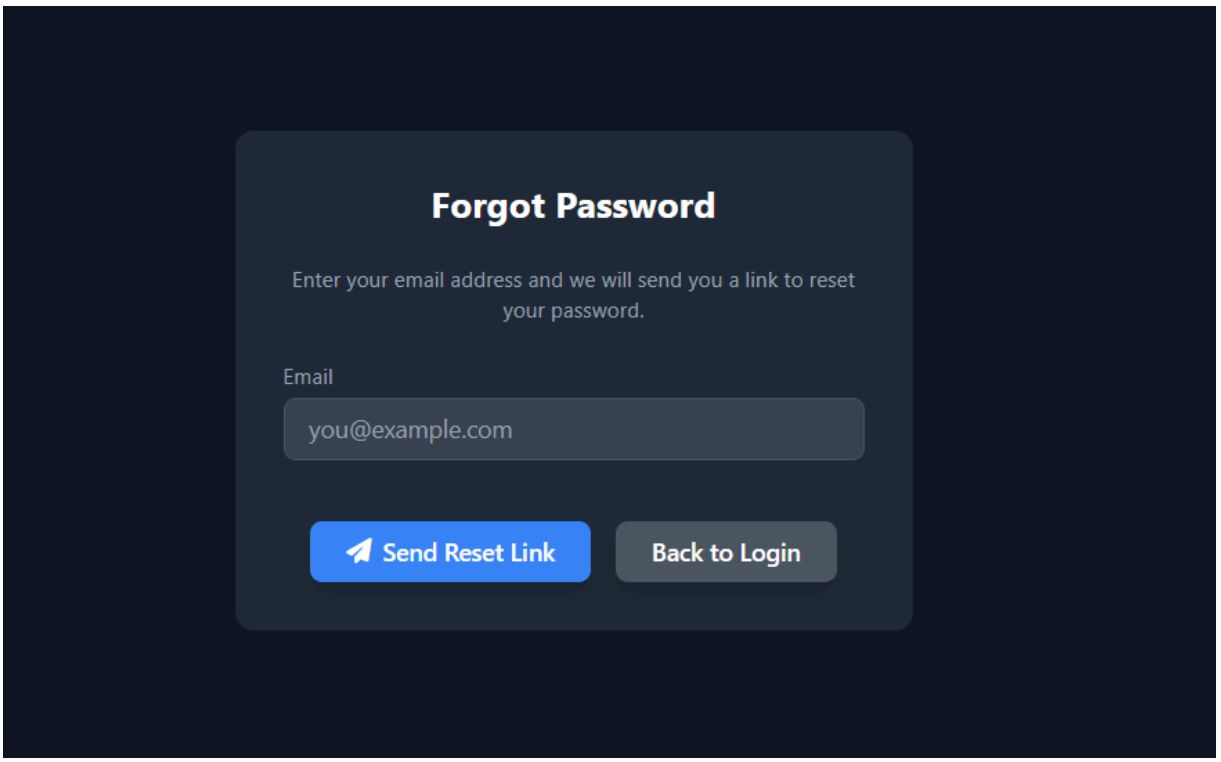
4. Admin Management

Description :System-wide administration functions, including user management, courses, assignments/exams, reports, and authorization.

From : System Access (Admin login)

To : User Management / Course Management / Assignment & Exam Management / Report Management

5. Forgot password form

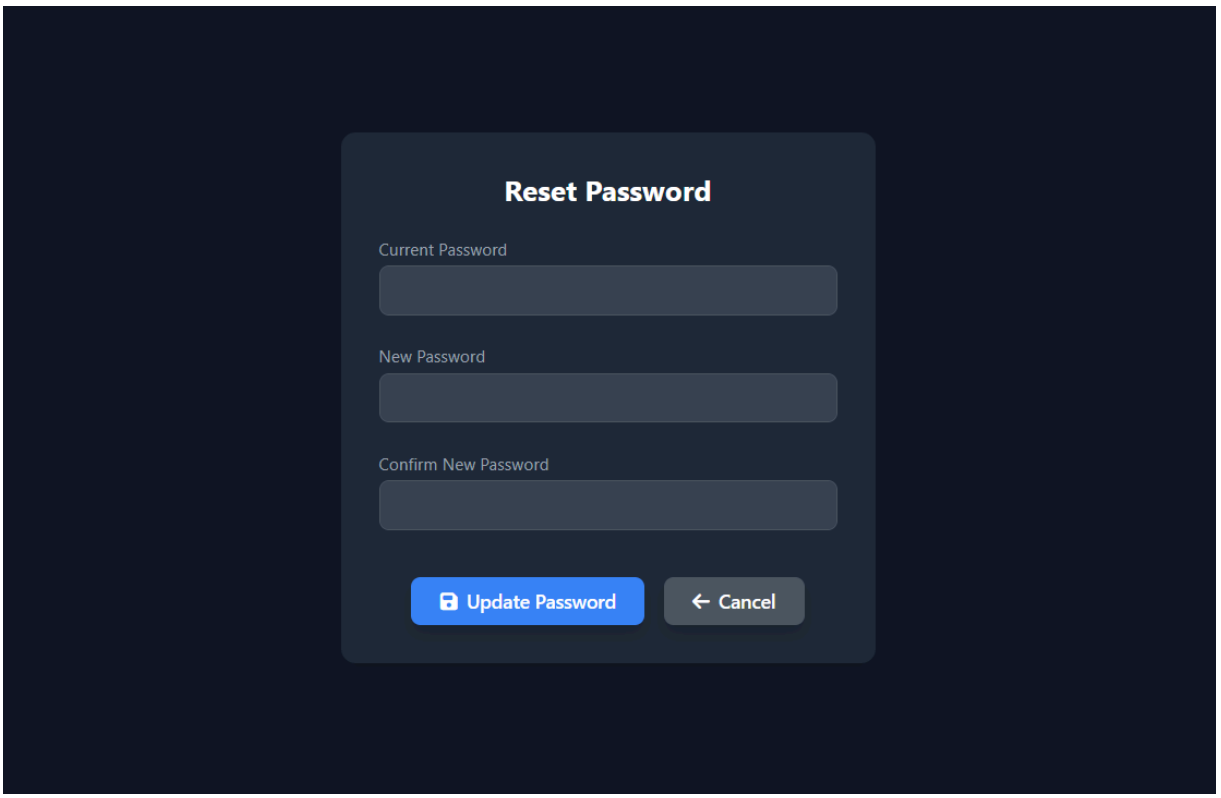


Description : The form allows users to enter email/username to receive a password reset link.

From : Public Course / Login Page

To : Reset Password Form

6. Reset password form

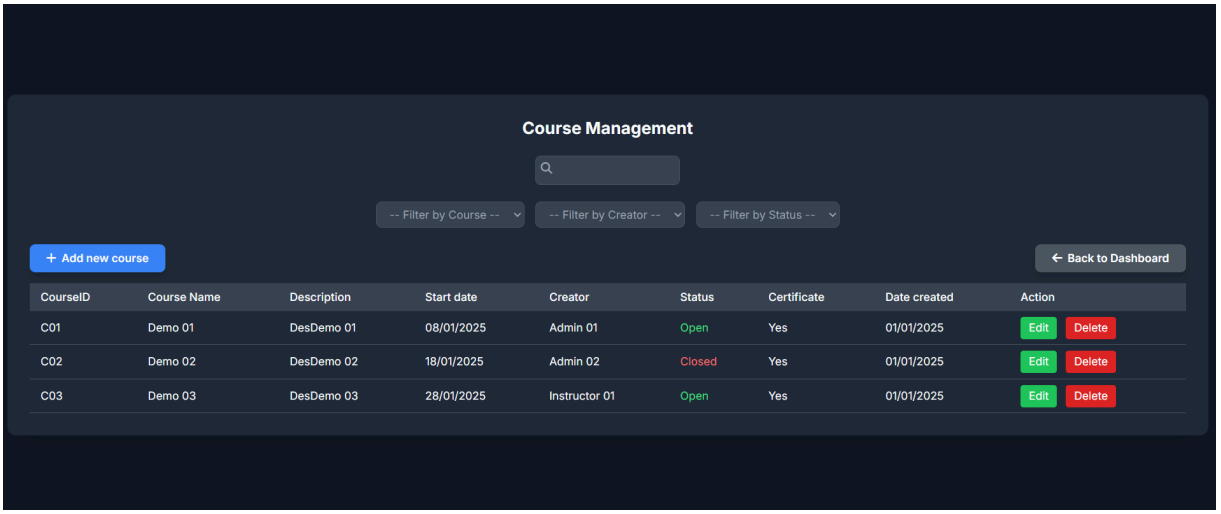


Description : Form for user or admin to change forgotten password.

From : Public Course

To : User Management

7. Account Management

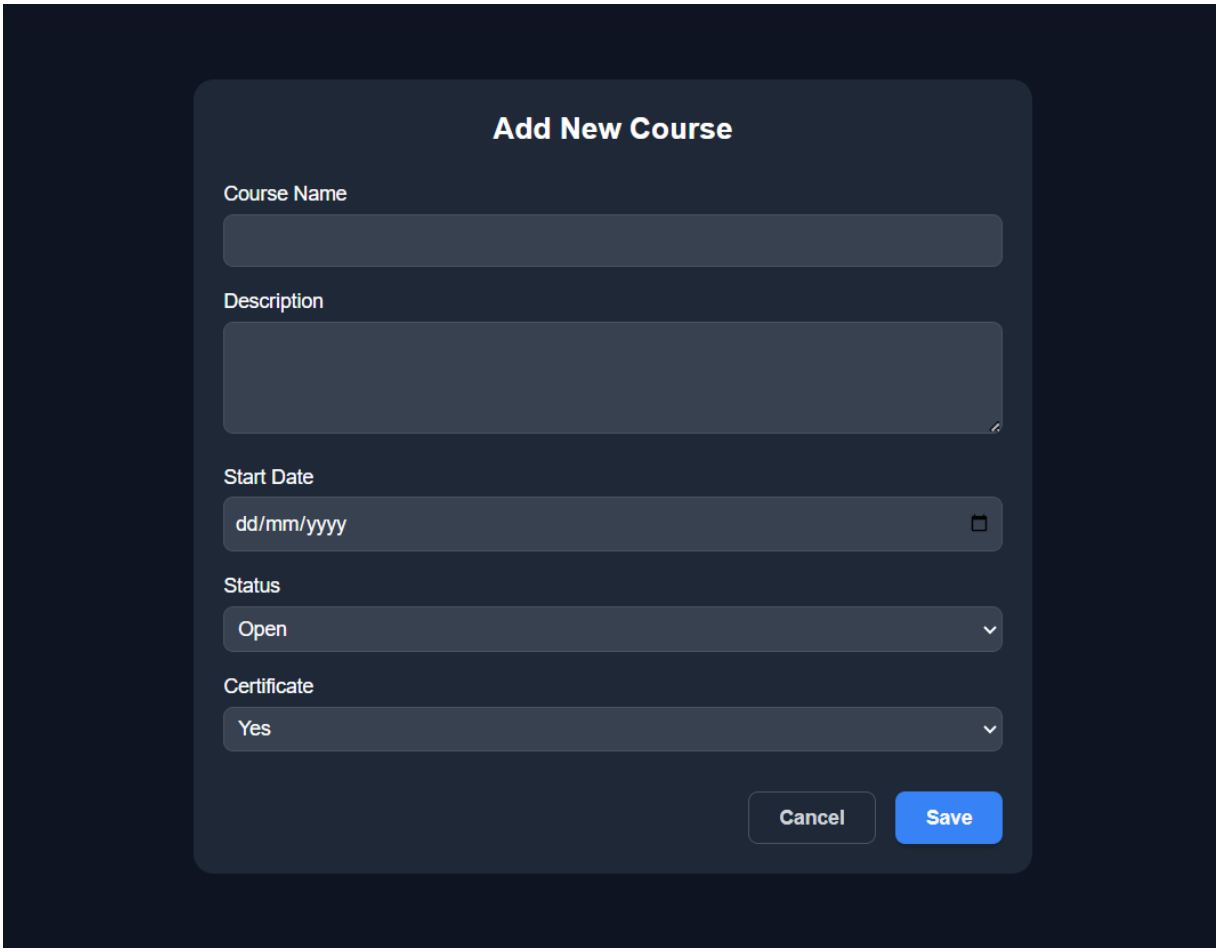


Description : This is where you manage accounts(Admin role)

From : Public course

To : User Management

8. Account create form

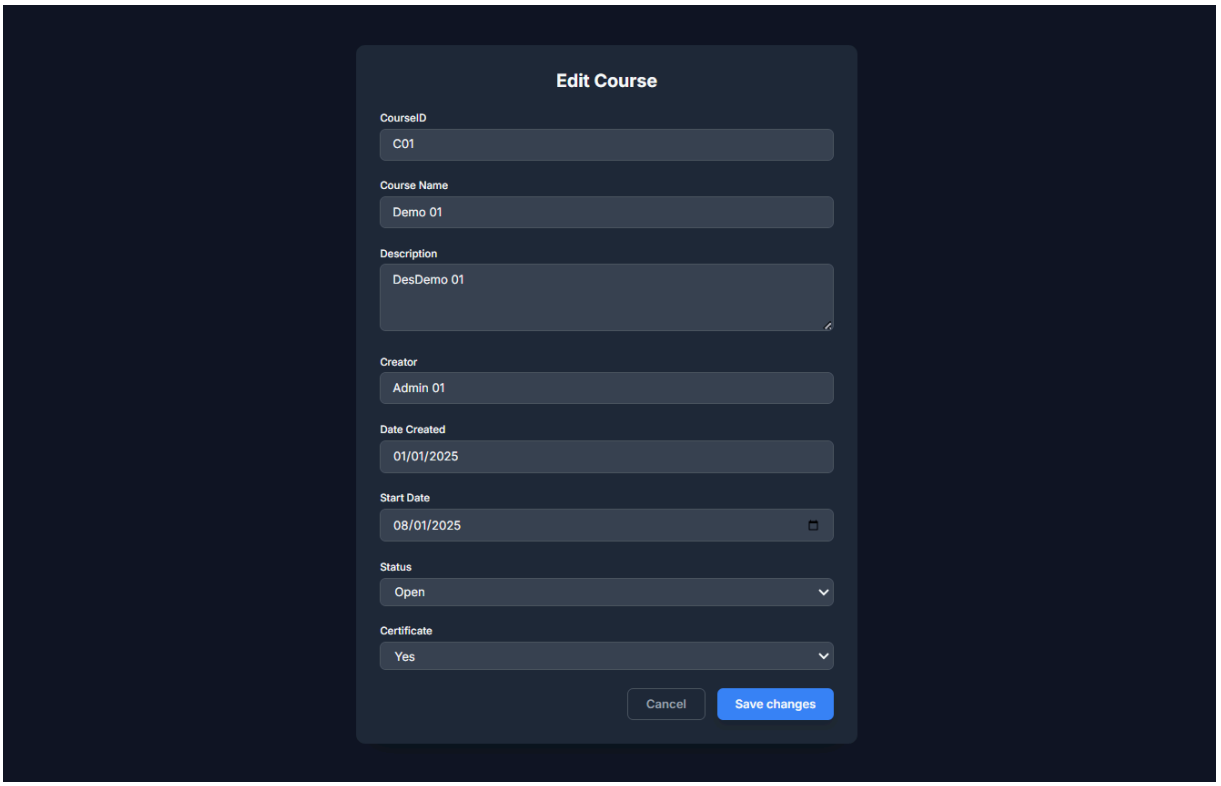


Description : Form for admin to create new user account.

From : Account Management

To : User Management

9. Account edit form

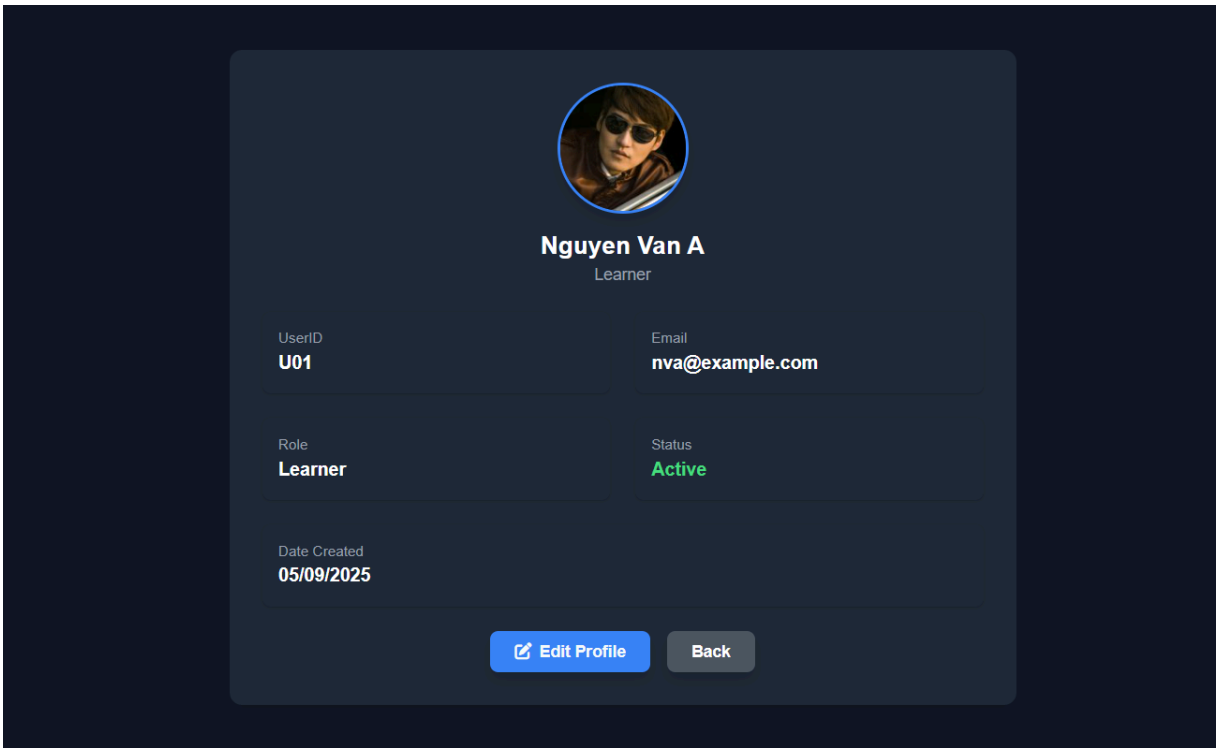


Description : Form to edit existing account information.

From : Account Management

To : User Management

10. User Profile

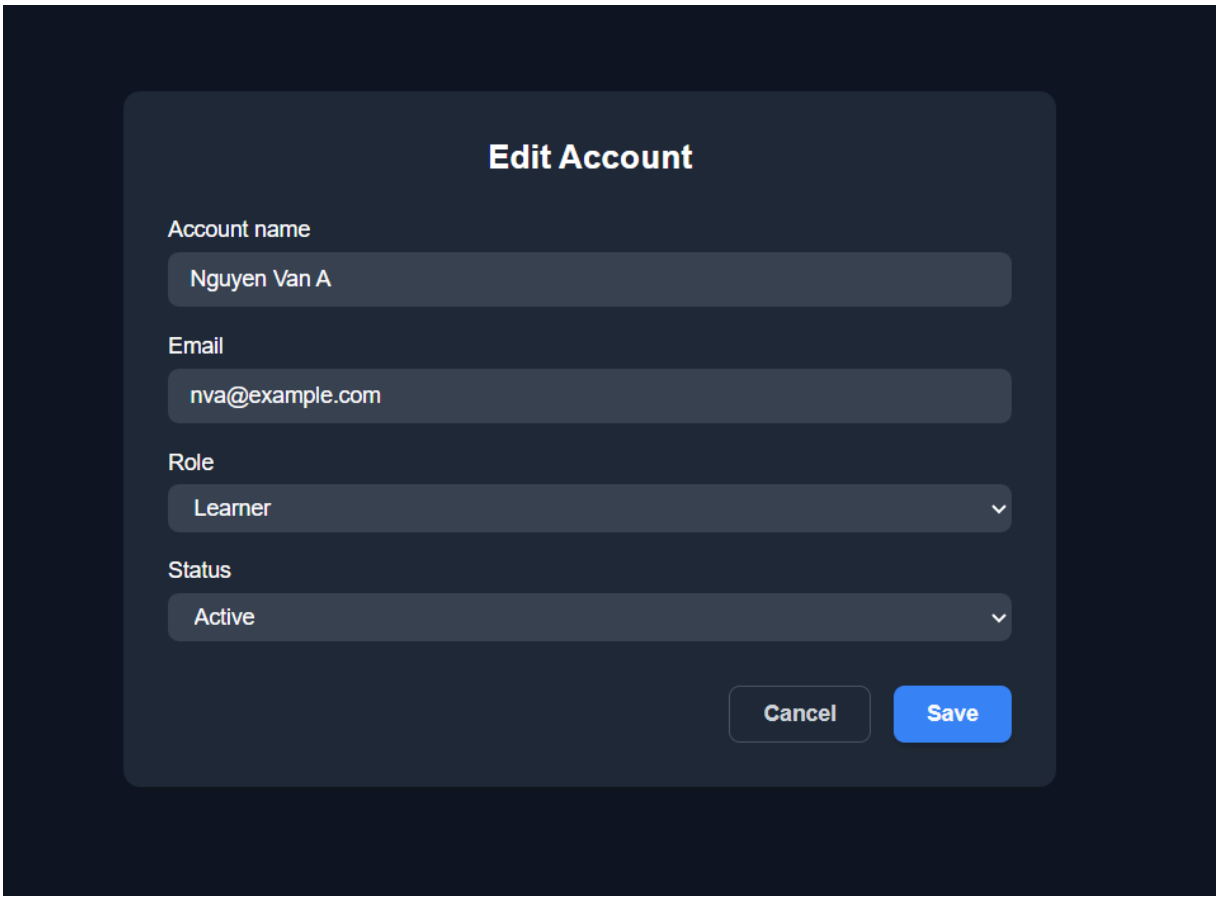


Description : This is where you change your profile infomantion and password

From : Public course

To : User-Profile-Management

11. Edit User Profile

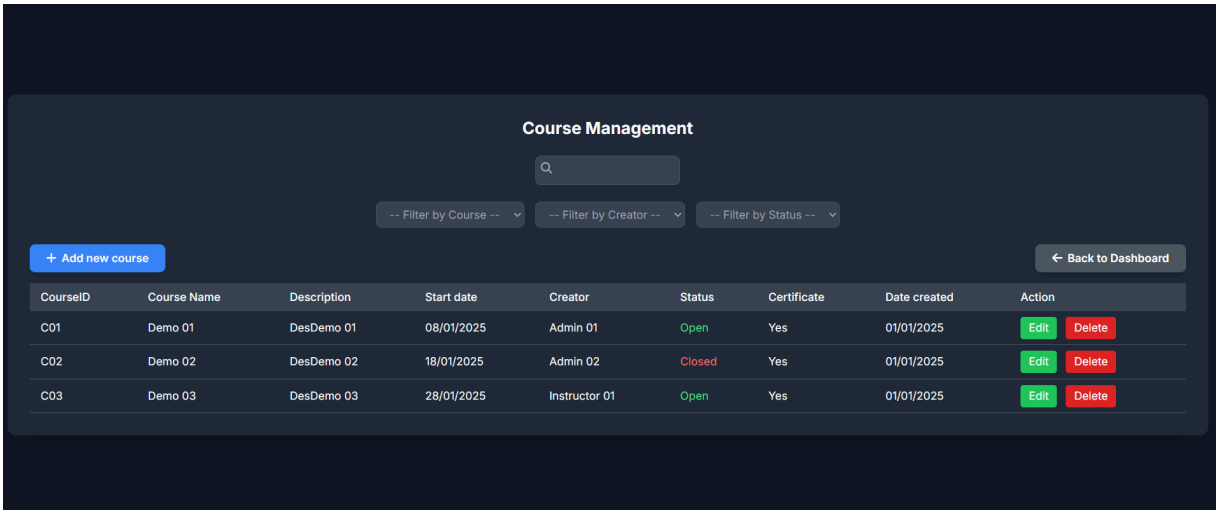


Description : Allows users to edit personal information such as avatar, email, password

From : User Profile

To : User Management

12. Course Management

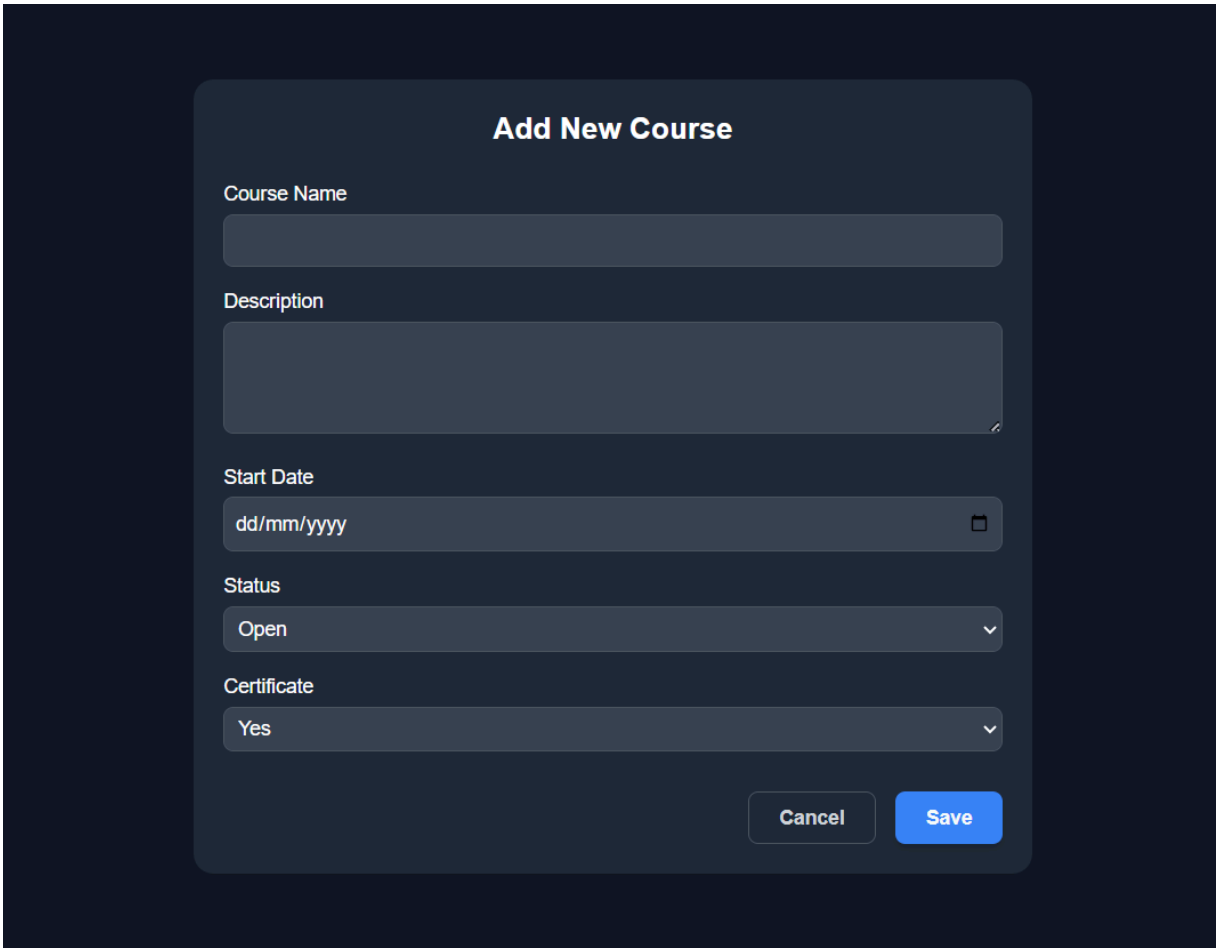


Description : This is where you manage courses (Admin, Teacher role)

From : Public course

To : Course Management

13. Course create form

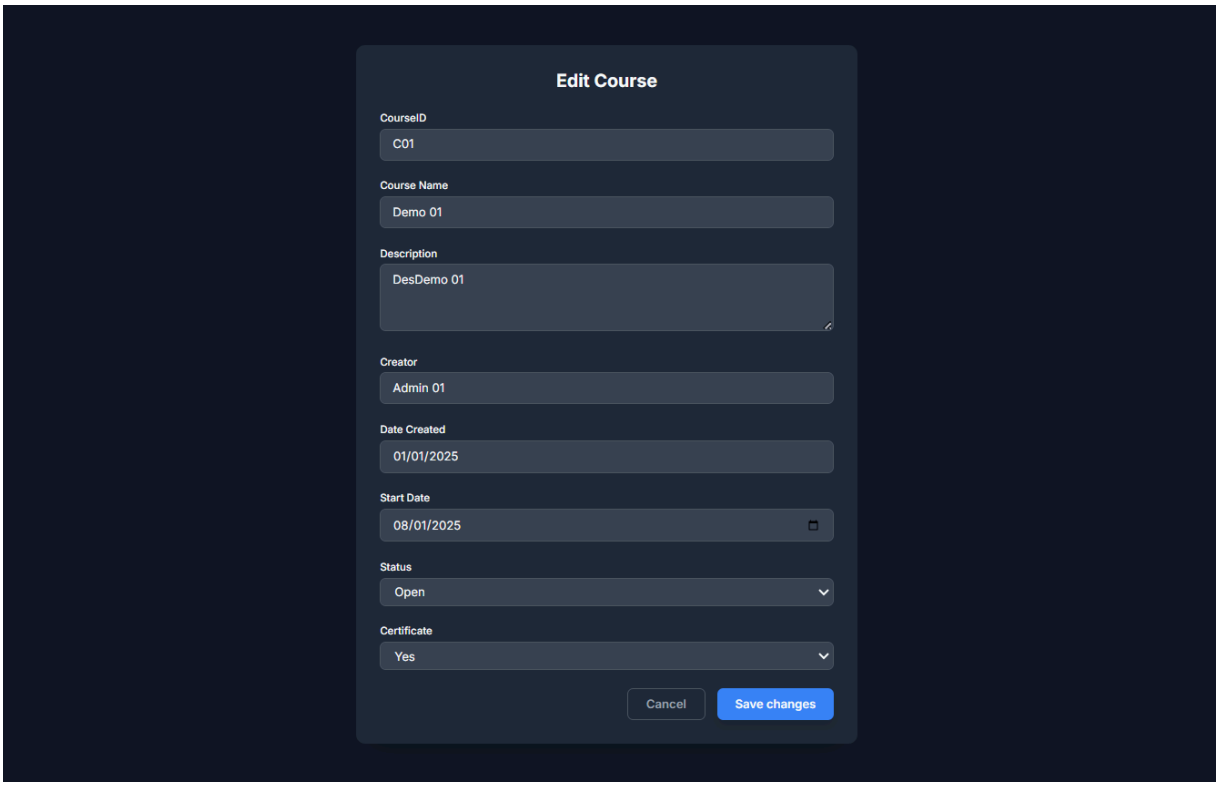


Description : Form allows admin/instructor to create new course (enter name, description, syllabus, resources).

From : Course Management (Admin/Instructor role)

To : Course Database

14. Course edit form

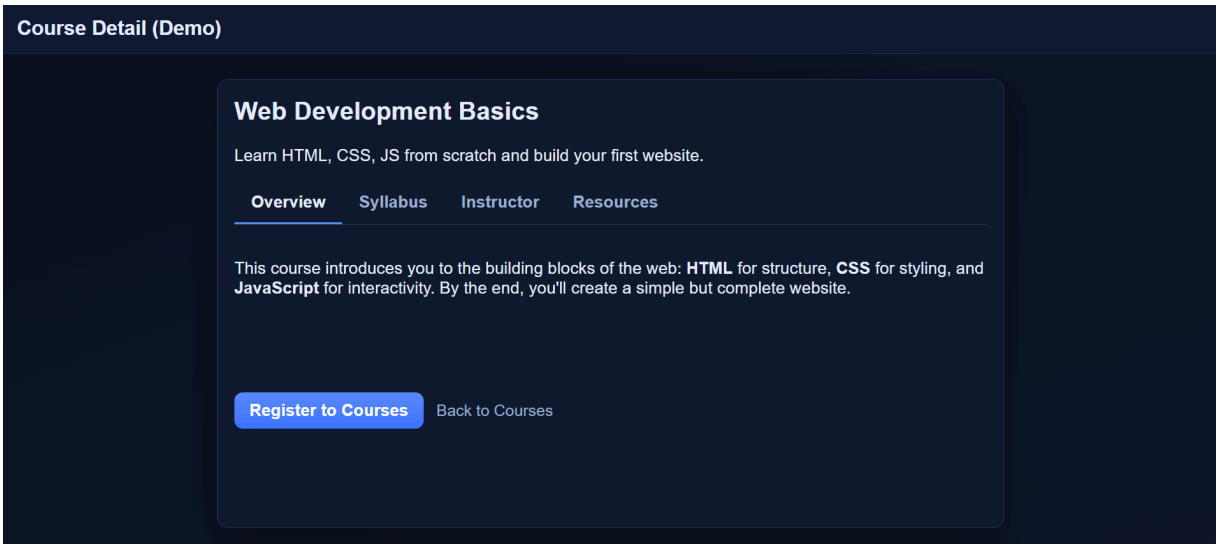


Description :Form allows admin/instructor to edit course information (name, description, syllabus, resources).

From : Course Management (Admin/Instructor role)

To : Course Database

15. Course detail

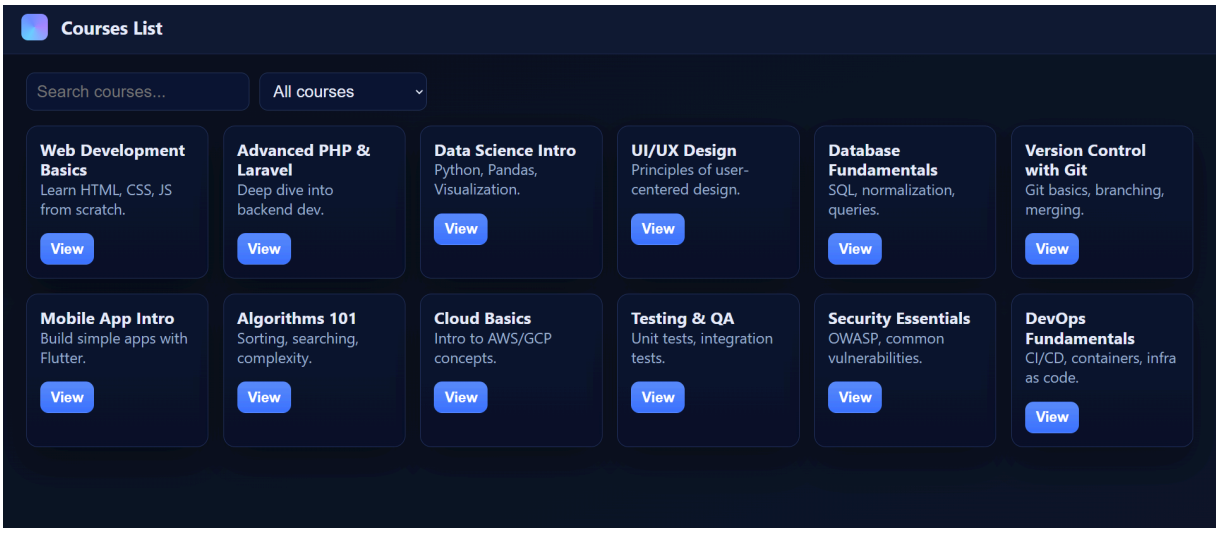


Description : Page that shows details of a course to the learner (Overview, Syllabus, Instructor, Resources, Lessons, Quiz).

From : Course List

To : Lesson View / Enrollment / Resources

16. Public courses

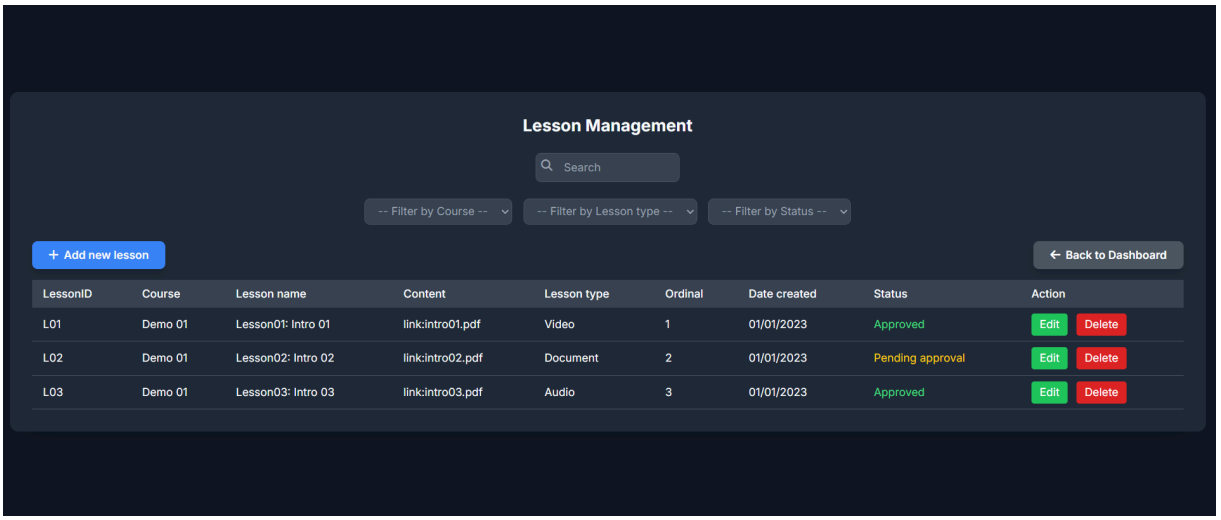


Description : This is where you view the courses you are taking

From : HomePage, login, register

To : Each course progress (Student), or course management (Teacher or Admin)

17. Lesson Management

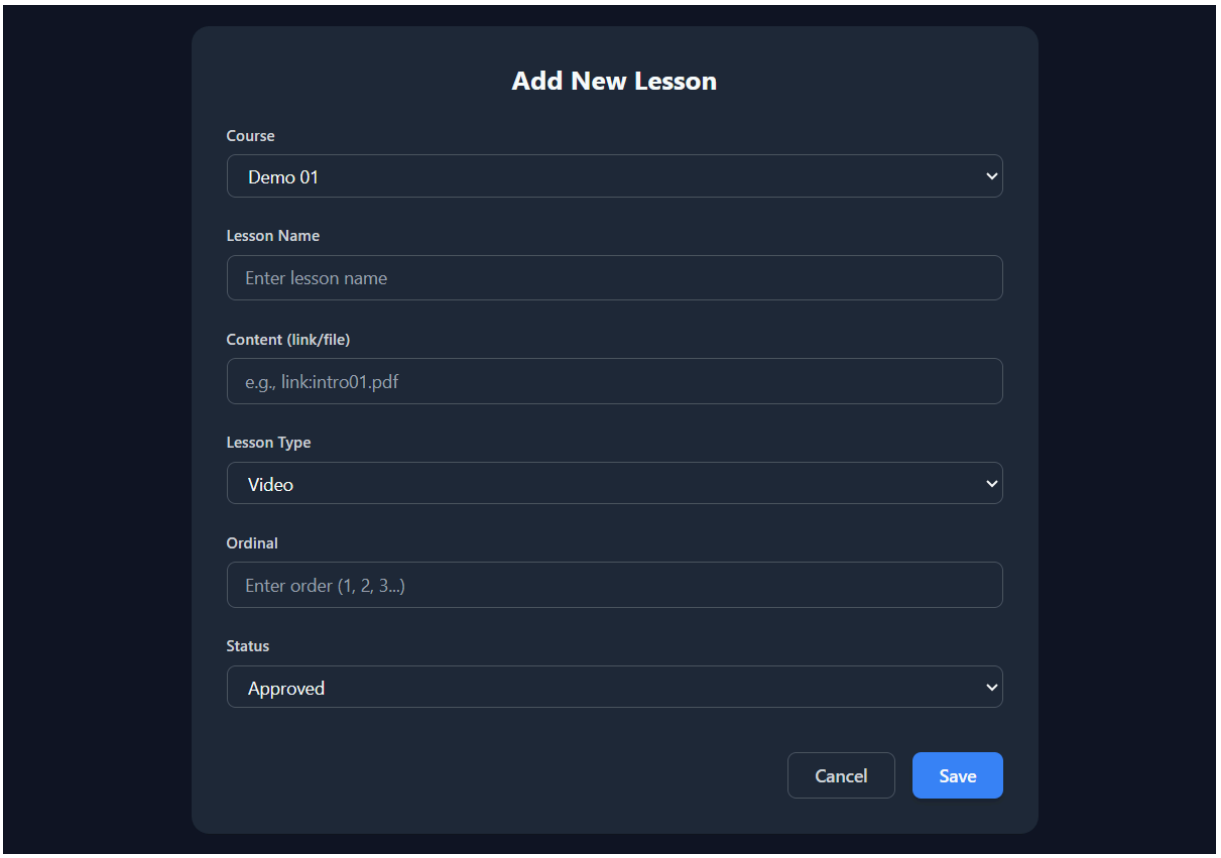


Description : Manage all lessons of courses (add, edit, delete, assign to courses).

From : Course Management (Admin/Instructor role)

To : Lesson Database

18. Lesson create form

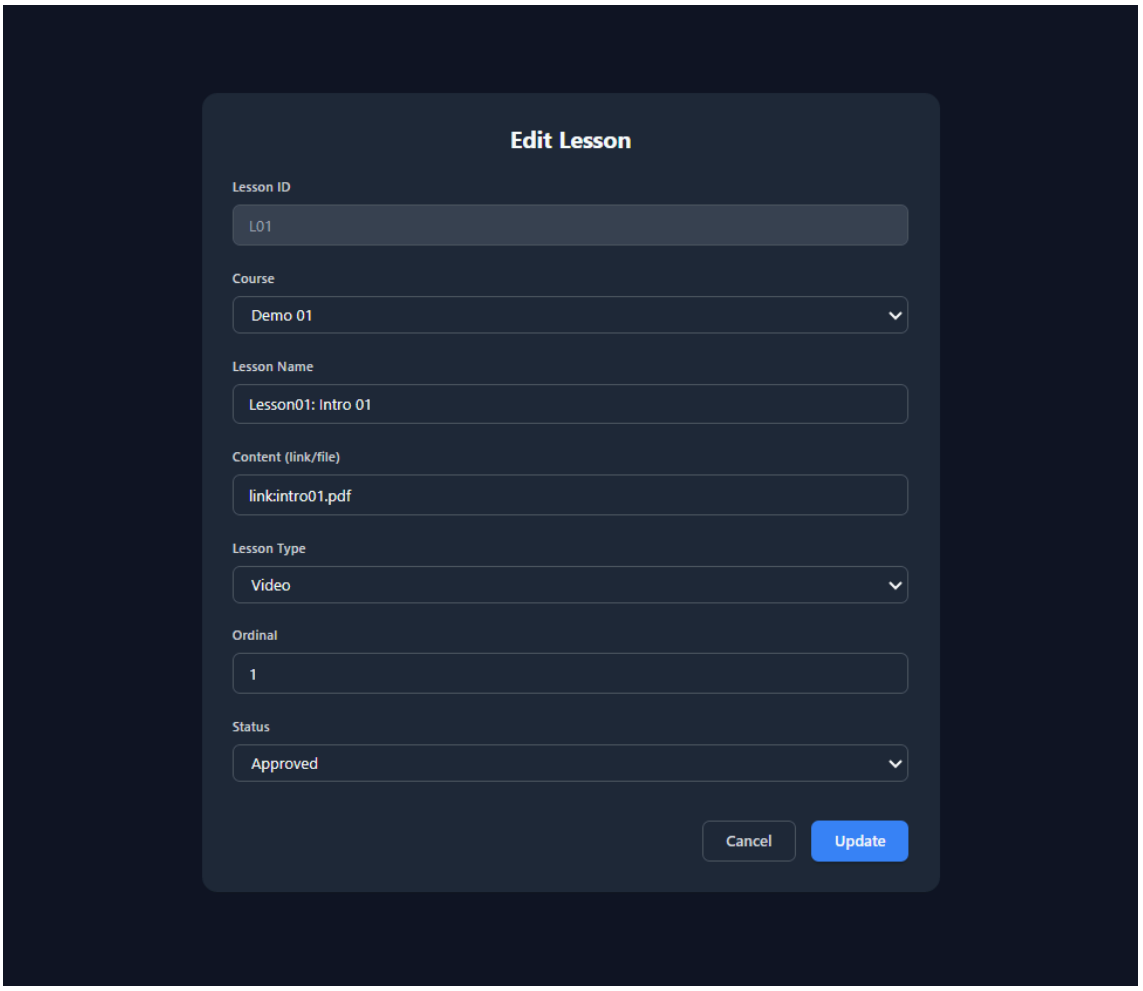


Description : Form allows admin/instructor to create new lesson (title, content, documents).

From : Lesson Management

To : Lesson Database

19. Lesson edit form

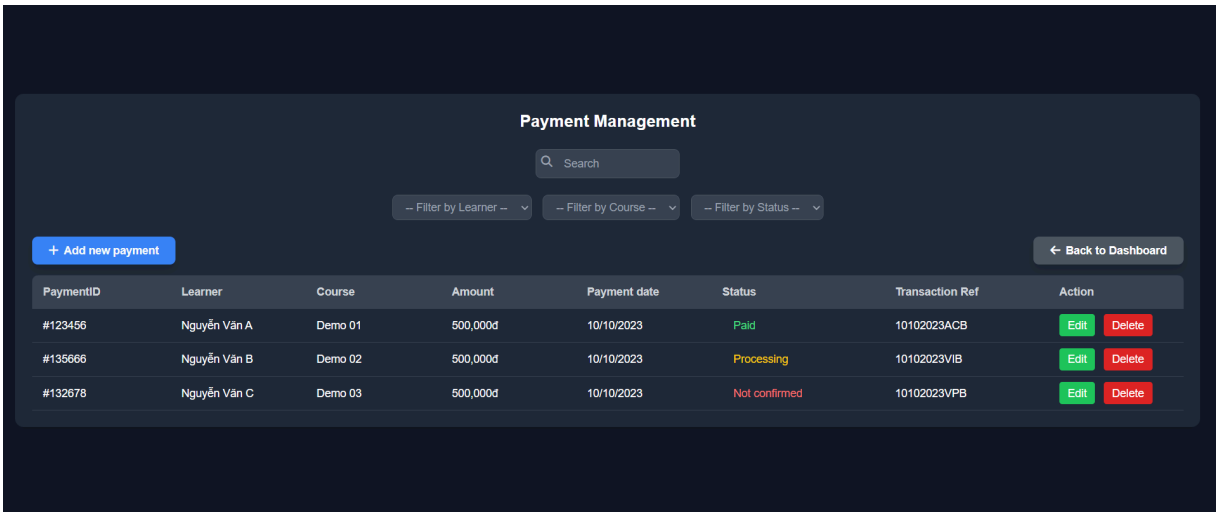


Description : Form allows admin/instructor to edit lesson information.

From : Lesson Management

To : Lesson Database

20. Payment Management

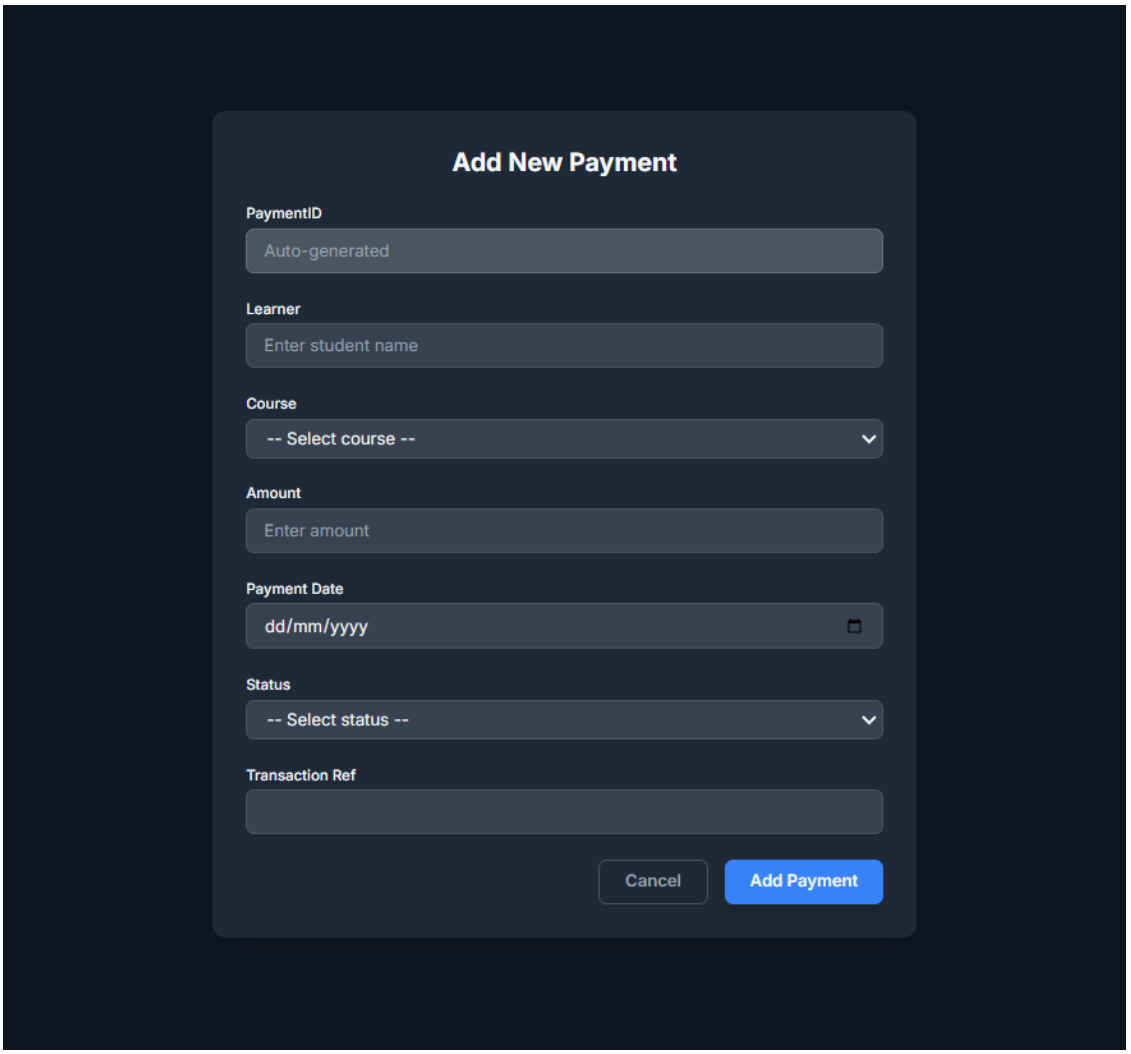


Description : Manage payment transactions (by user, by course).

From : Admin/Finance Dashboard

To : Payment Database

21. Payment create form

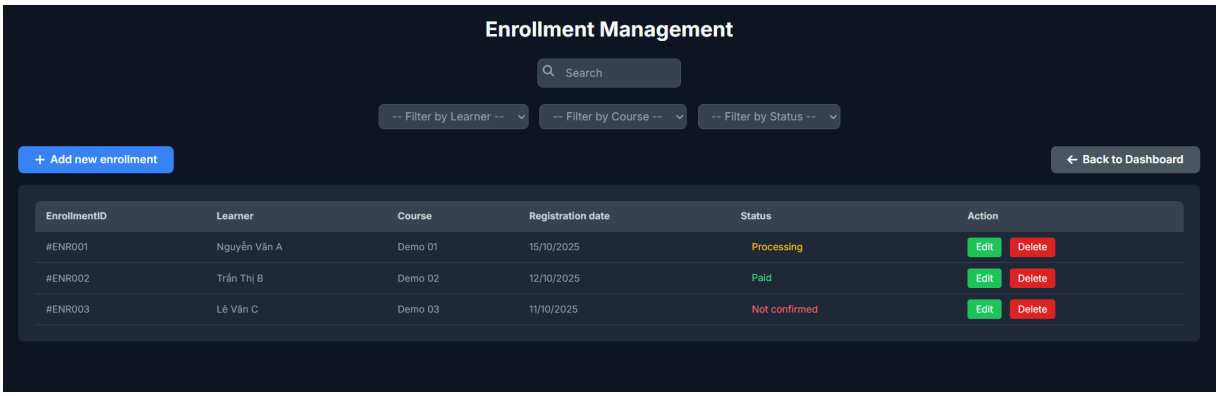


Description : Form for admin to add a new transaction/payment manually.

From : Payment Management

To : Payment Database

22. Enrollment Management

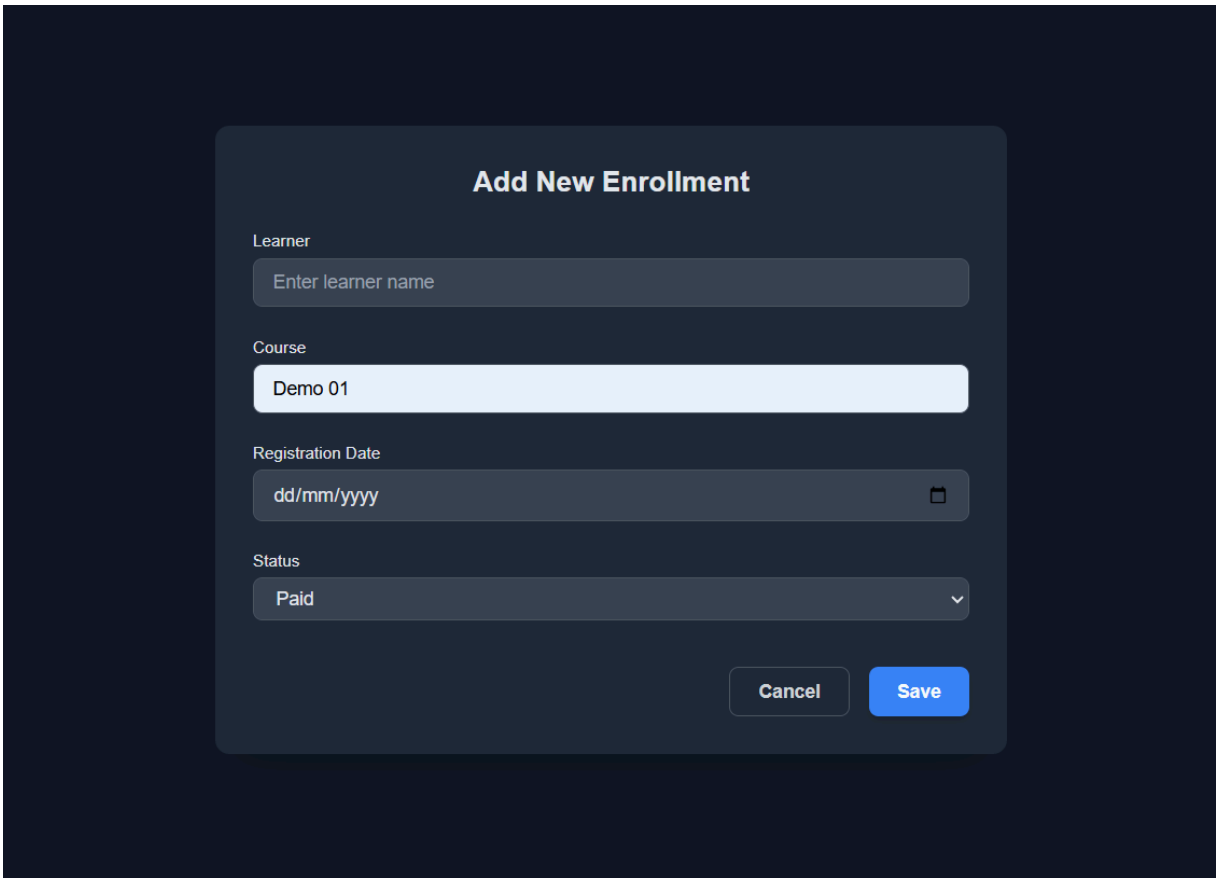


Description : Manage list of students registering for courses.

From : Course Detail / Course Management

To : Enrollment Database

23. Enrollment create form

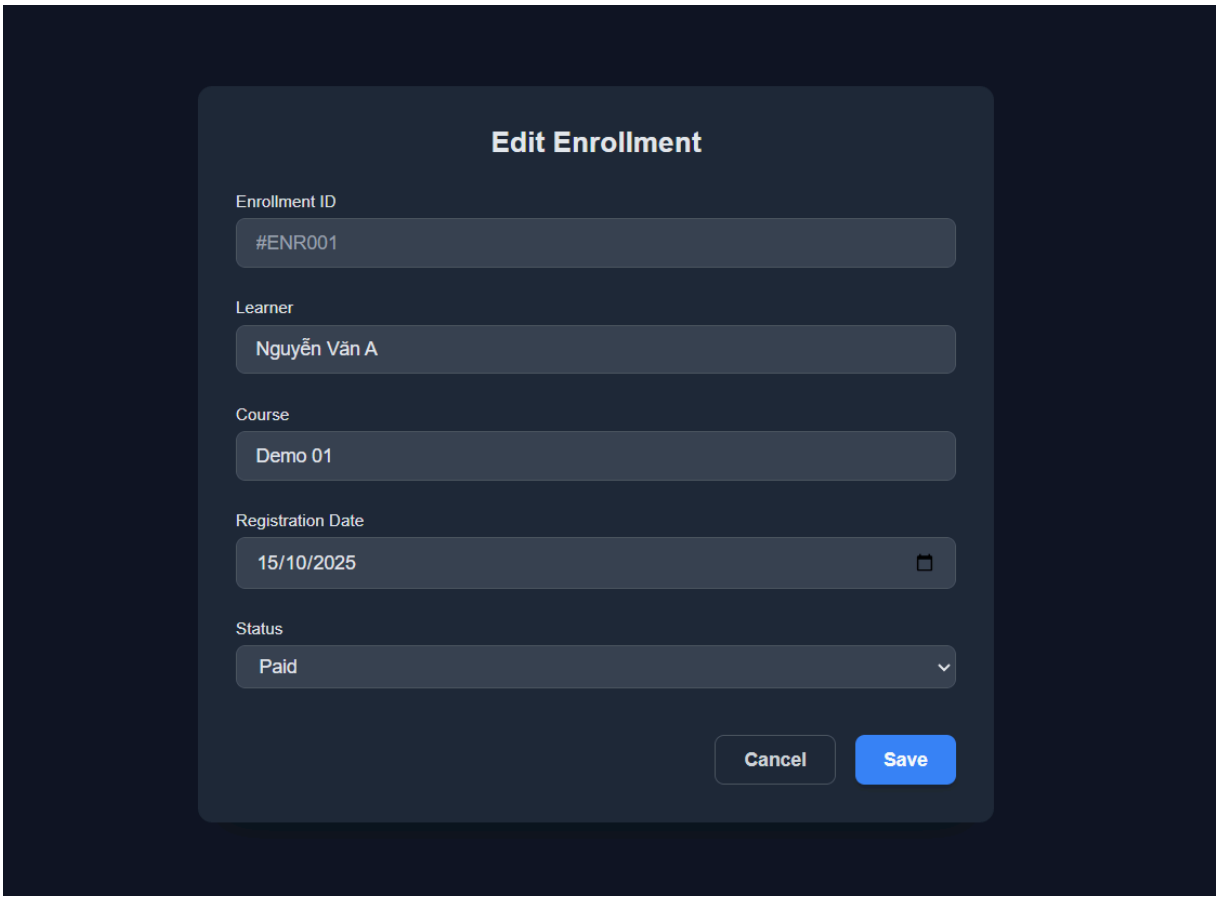


Description : Form to record new students registering for the course.

From : Course Detail / Enrollment Management

To : Enrollment Database

24. Enrollment edit form

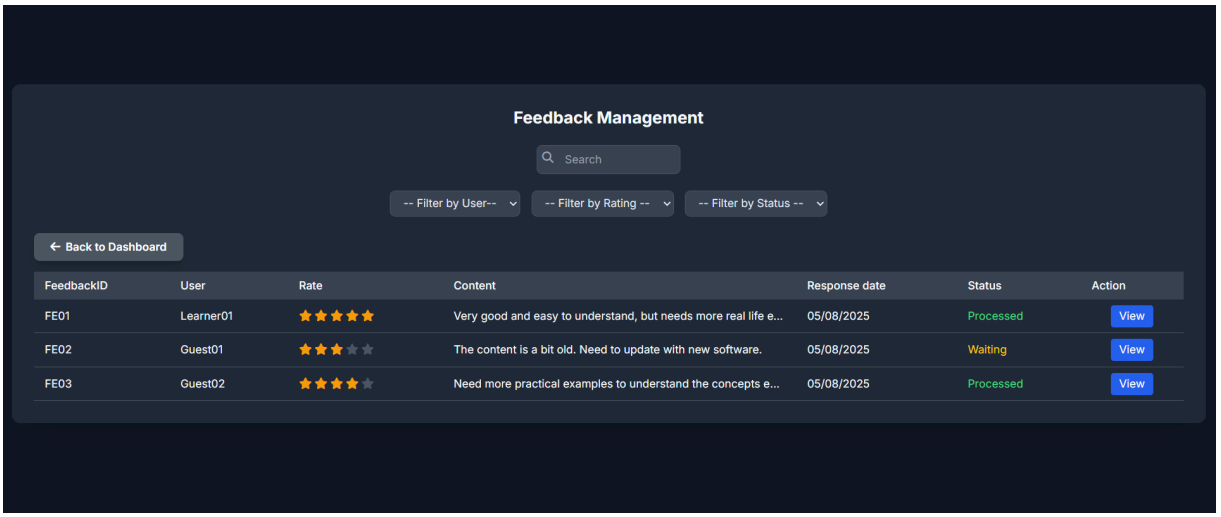


Description : The form allows editing of registration status (Active, Cancelled, Completed).

From : Enrollment Management

To : Enrollment Database

25. Feedback Management

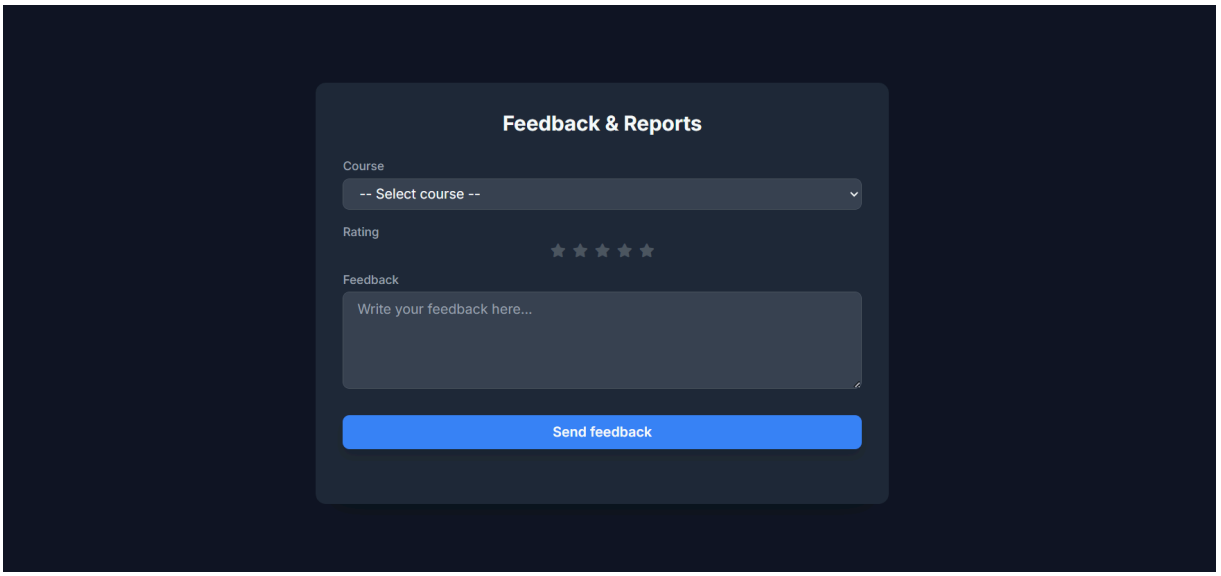


Description : Manage all feedback from learners about courses, lessons, instructors.

From : Learner Dashboard / Course Detail

To : Feedback Database

26. Feedback create form

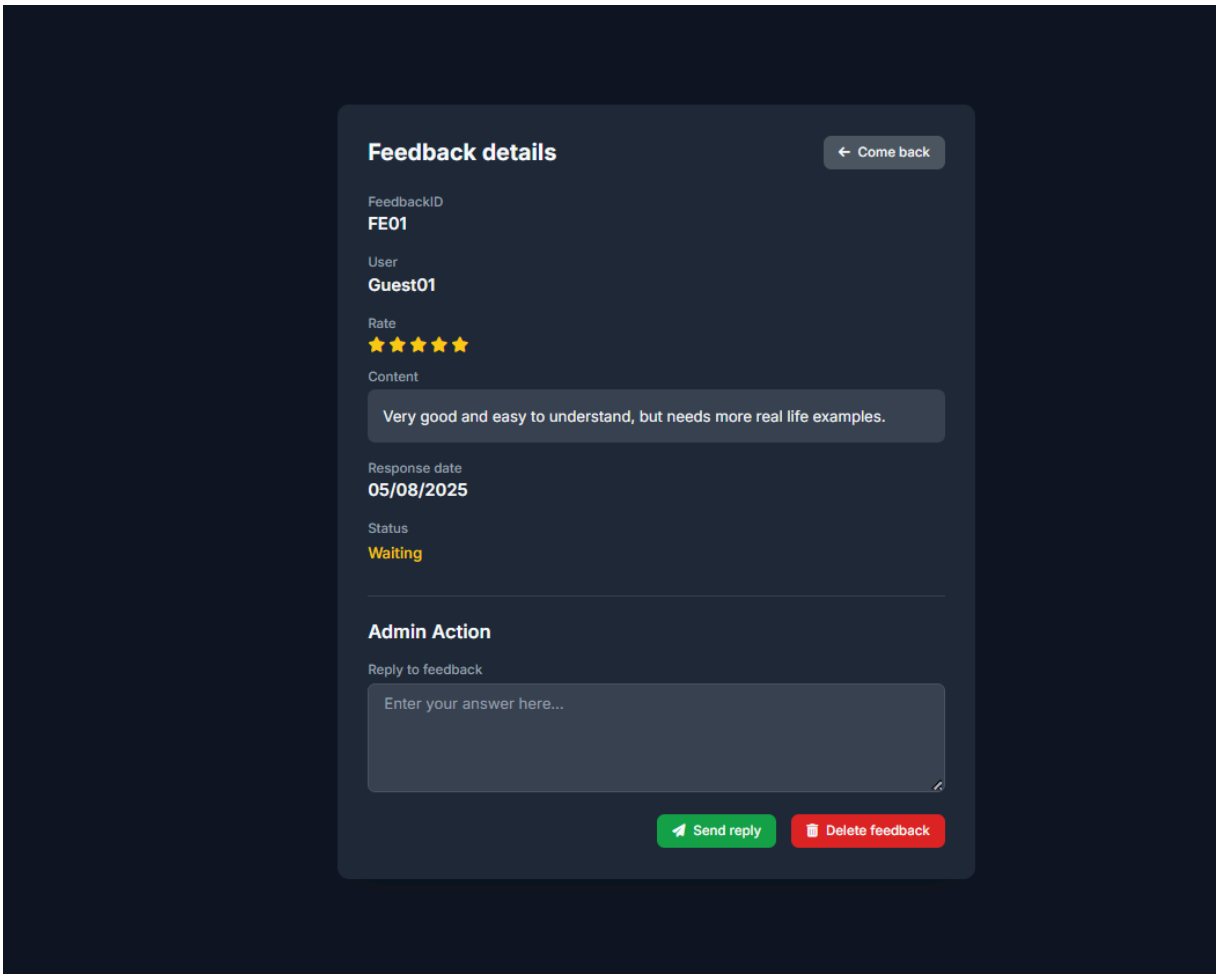


Description : Form for learners to send feedback (rating, comments, suggestions).

From : Course Detail / Lesson View

To : Feedback Database

27. Feedback reply form

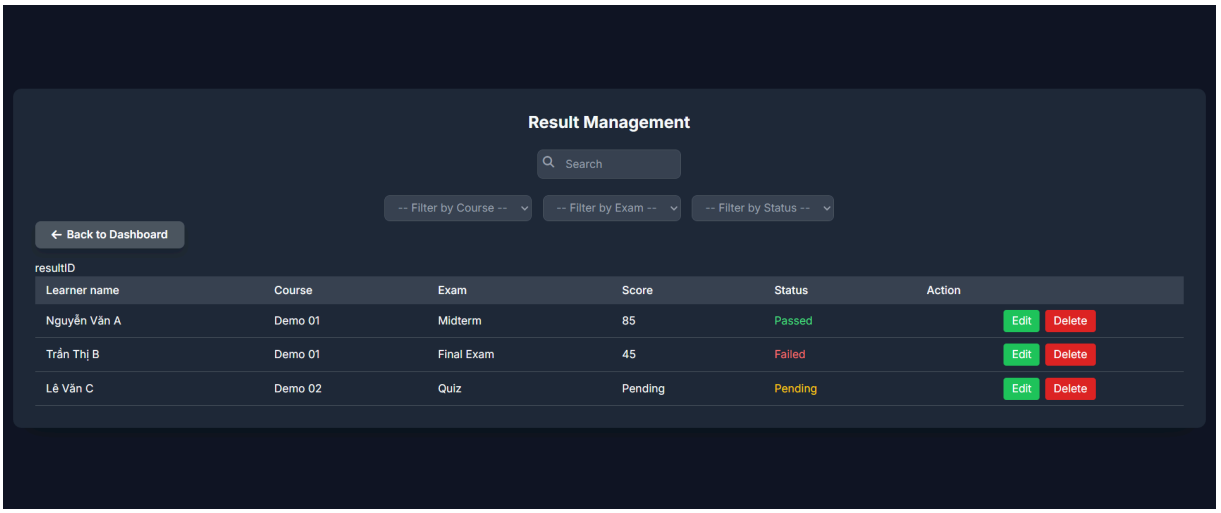


Description : Form for instructor/admin to respond to learner feedback.

From : Feedback Management

To : Feedback Database

28. Result Management

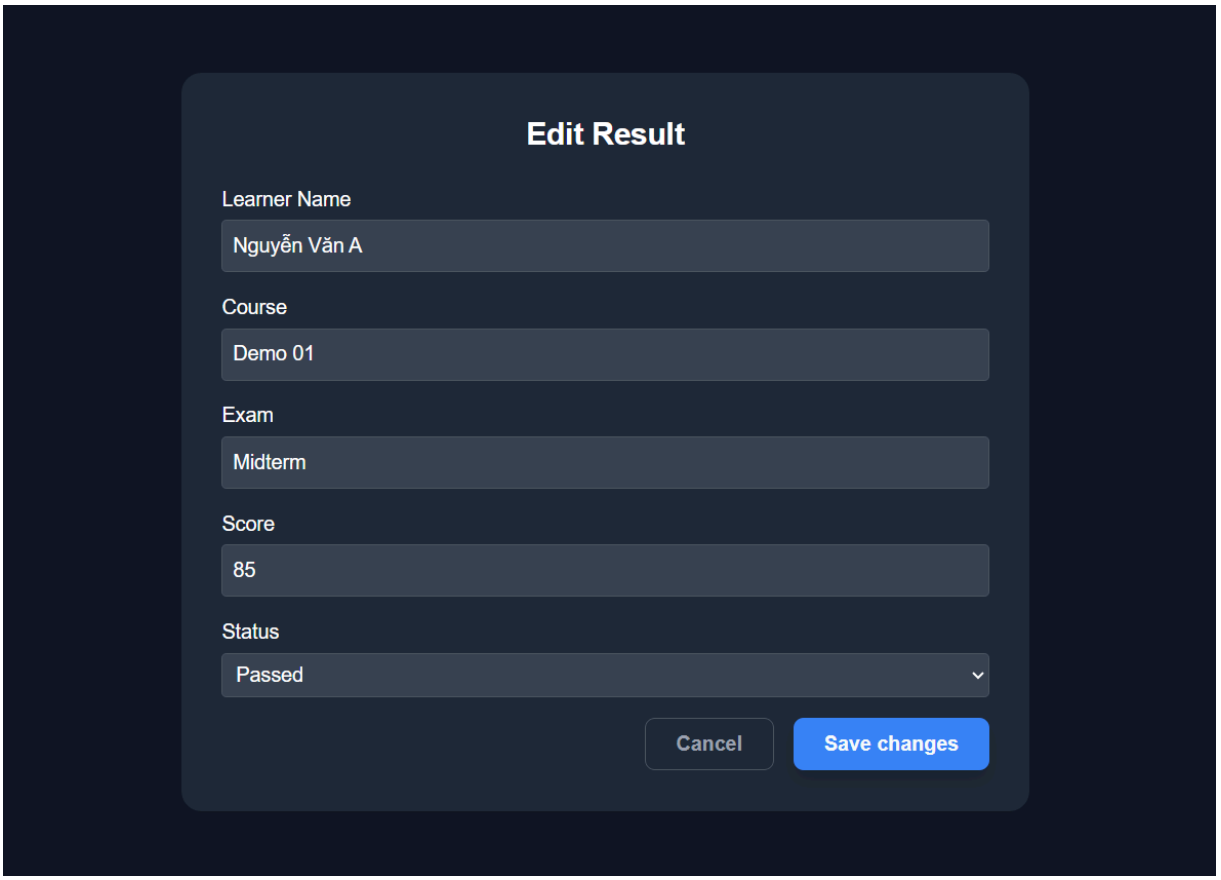


Description : Manage learners' learning results, quiz/exam scores.

From : Instructor Dashboard / Exam Submission

To : Result Database

29. Result edit form

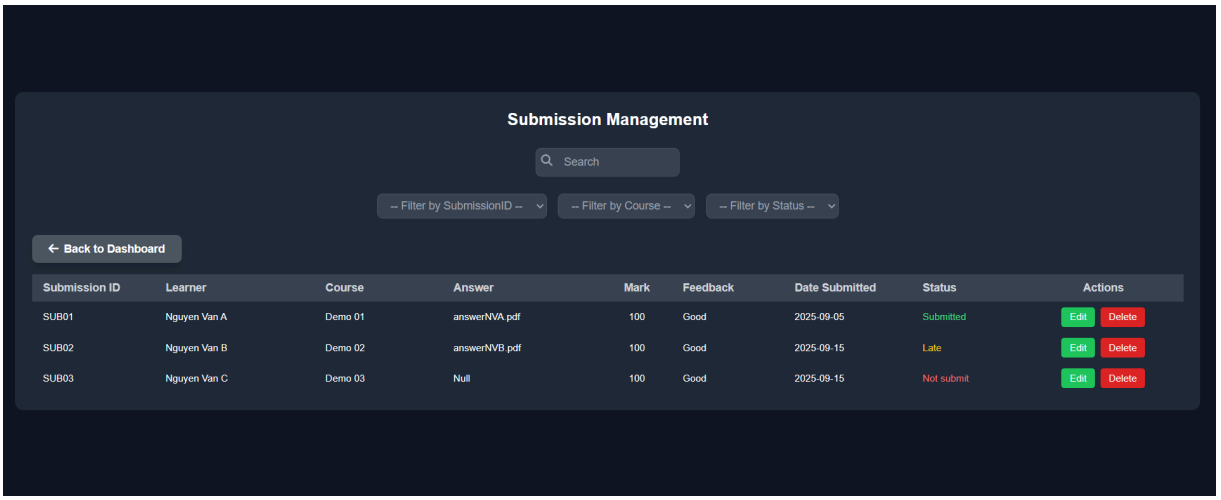


Description : Form cho phép instructor chỉnh sửa, cập nhật điểm và nhận xét.

From : Result Management

To : Result Database

30. Submission Management

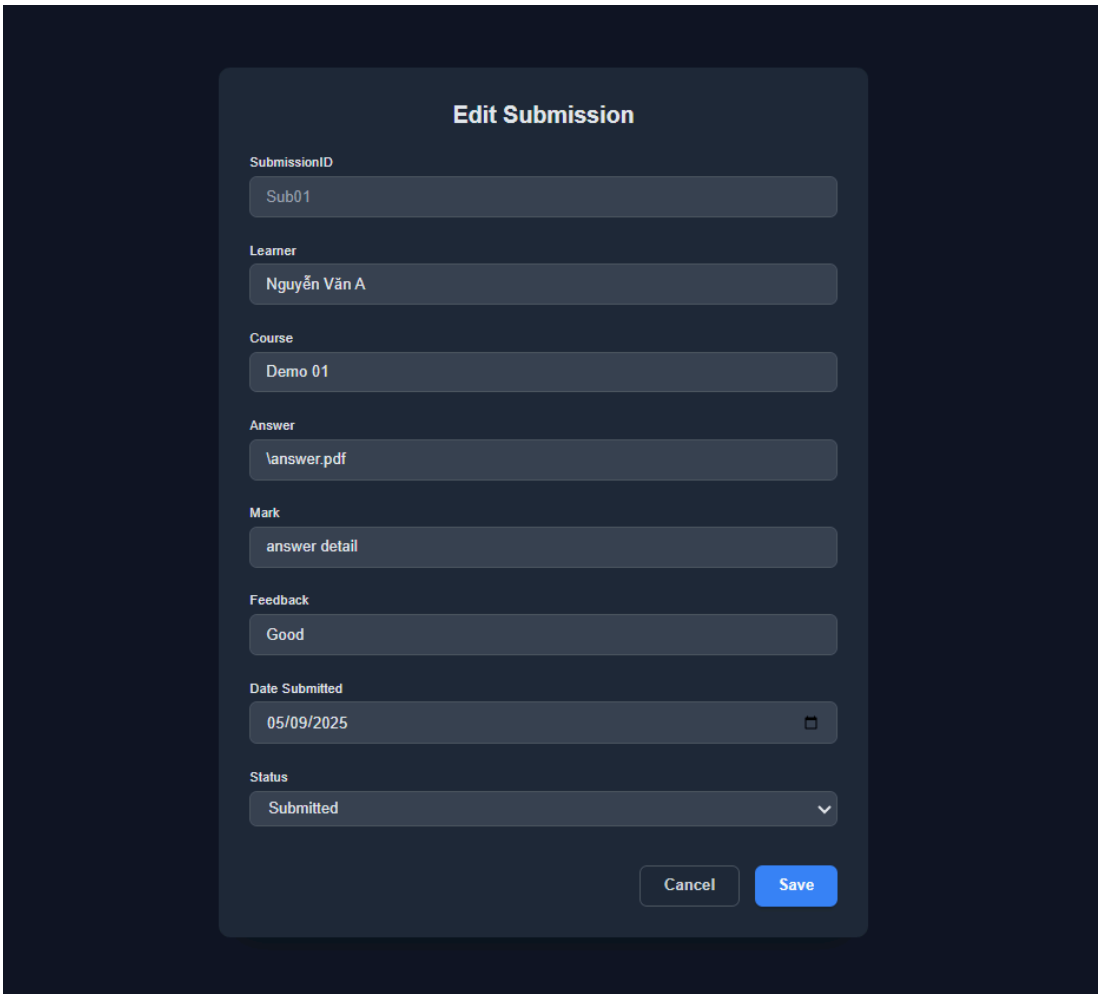


Description : Quản lý bài nộp assignment/exam của learner.

From : Learner Dashboard / Assignment Page

To : Submission Database

31. Submission edit form

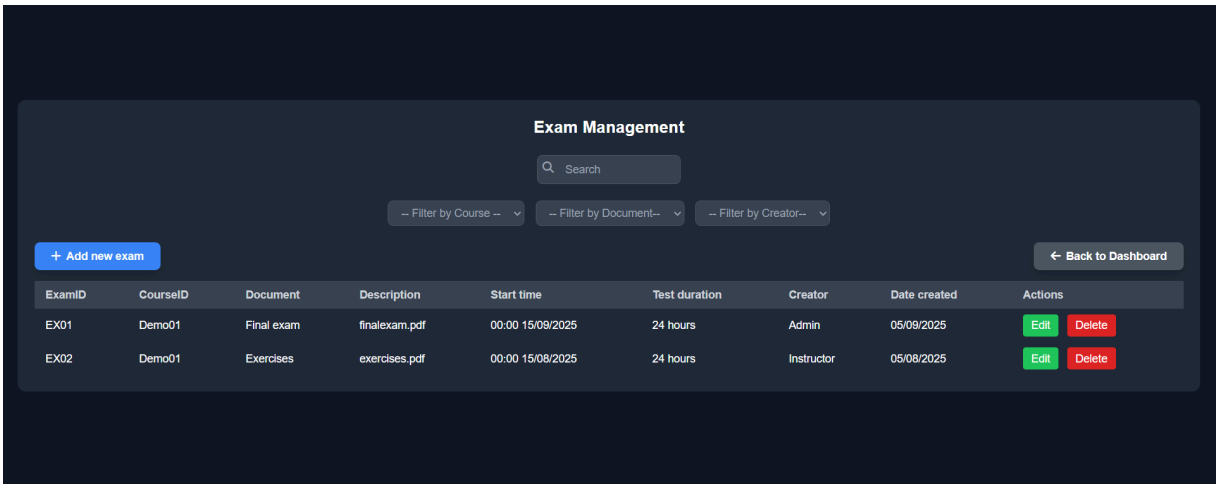


Description : Form cho phép instructor chỉnh sửa/trả lời/chấm điểm bài nộp.

From : Submission Management

To : Submission Database

32. Assignment Management

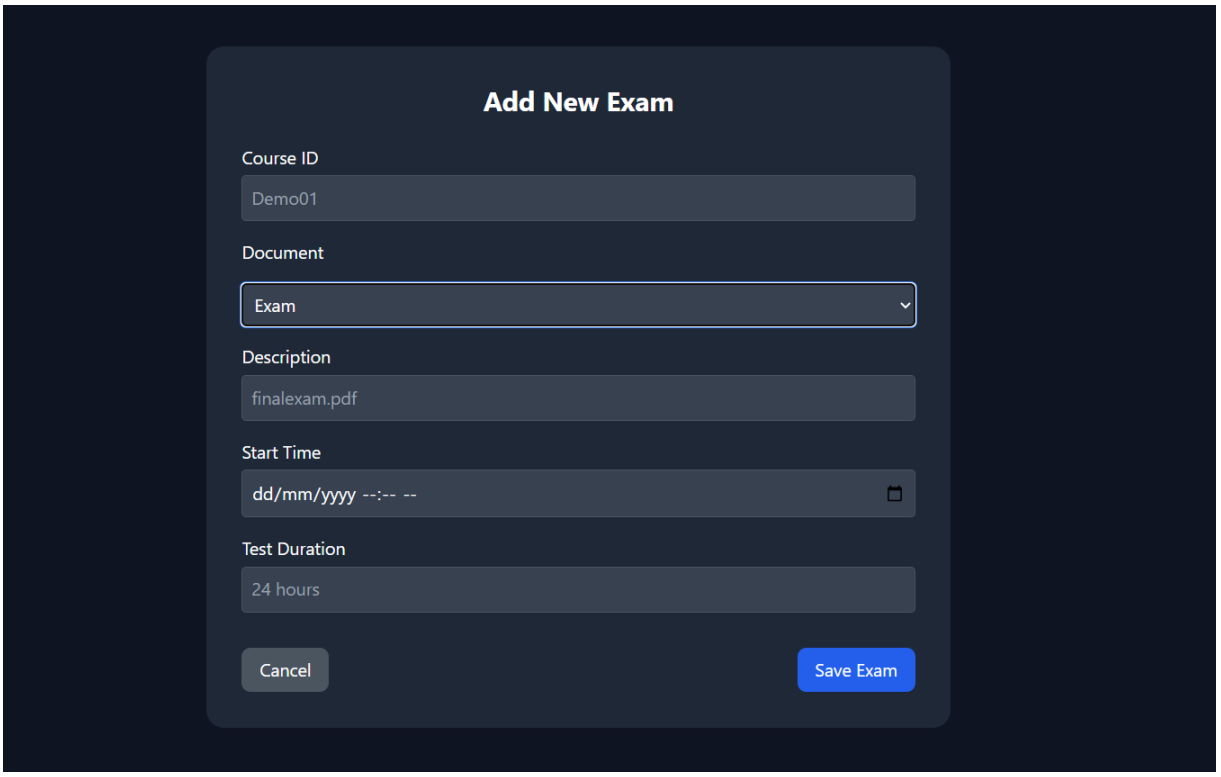


Description : This is where you manage assignment (Admin, Teacher role)

From : Public course

To : Course Management

33. Assignment create form

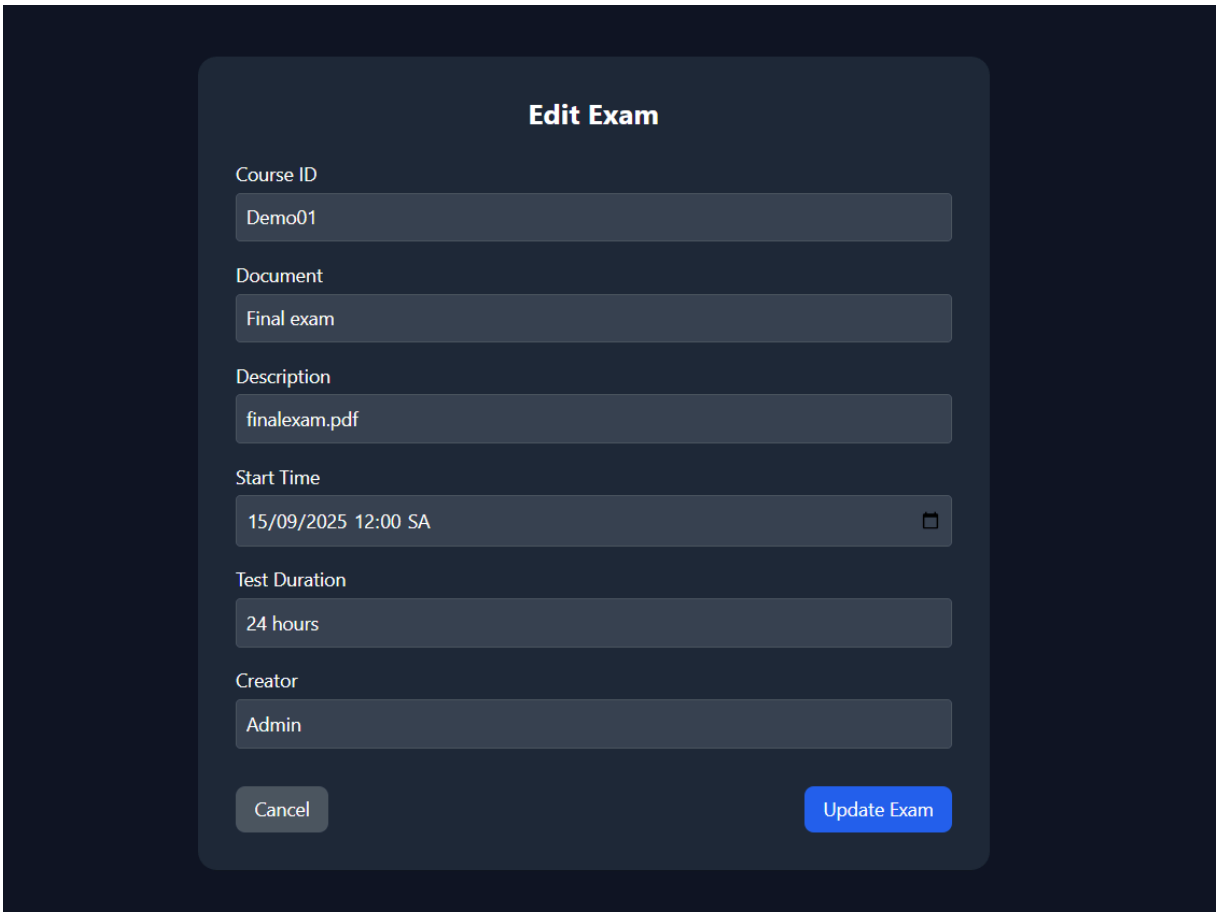


Description : Form for instructors to create a new assignment, including title, description, due date, and attached resources.

From : Assignment Management

To : Assignment Database

34. Assignment edit form



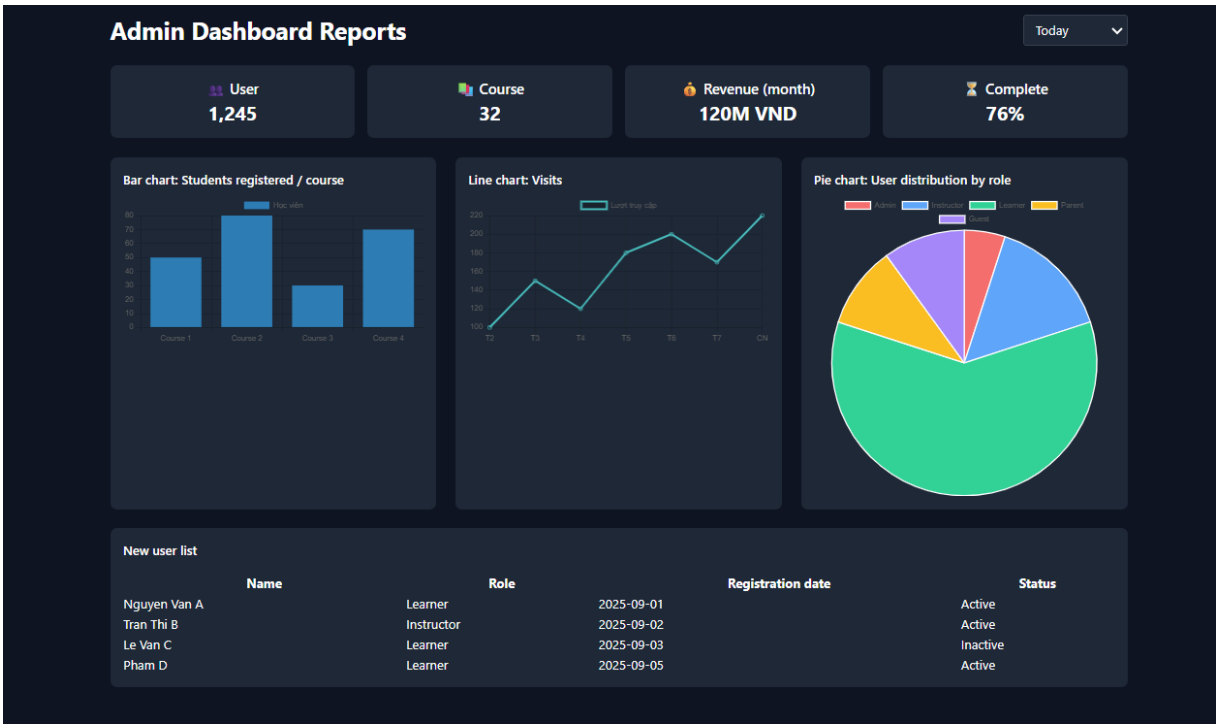
Description : Form for instructors to edit an existing assignment's details and resources.

From : Assignment Management

To : Assignment Database

35. Reports

Report_manage_admin



Report_manage_instructor

Report_manage_learner

Description : This is where you give feedback or report to Admin (Student, Teacher role)

From : Homepage, Public course,…

To : Course Management

Testing document

Sr.No	Features Tested	Remarks
1	User Registration and Login	Tested login with valid/invalid credentials
2	Admin Dashboard Access	Verified admin can access user and course management
3	Teacher Course Creation	Tested creating new courses and uploading materials
4	Student Course Enrollment	Verified students can enroll and view course content
5	File Upload Functionality	Tested uploading documents and videos
6	Basic Quiz/Assignment Features	Tested creating and submitting simple quizzes
7	User Profile Management	Verified users can update their profiles
8	Database CRUD Operations	Tested basic create, read, update, delete functions
9	Form Validation	Verified input validation on registration and login forms
10	Navigation and Routing	Tested page navigation and Laravel routing
11	Browser Compatibility	Tested on Chrome and Firefox browsers
12	Basic Security	Tested password hashing and session management
13	Responsive Design	Tested basic layout adaptation on different screen sizes
14	Course Content Display	Tested viewing lessons and materials
15	Basic Error Handling	Verified error messages display properly

Final checklist

Sr.No	Aspected Tested	Suggestion/ Remarks
1	Are all the users able to view the images and links?	Images and links load properly on main browsers
2	Have all the views, modules and controllers been properly integrated and is the site function as a single page application?	Laravel backend and ReactJS frontend integrated successfully
3	Are the GUI content devoid of spelling mistakes?	Text content reviewed for spelling errors
4	Is the application user-friendly?	Basic navigation and user interface tested
5	Is the Website launching correctly in all popular browsers?	Tested on Chrome, Firefox, Edge
6	Are all the forms validated with proper criteria?	Login, registration, and course forms validated
7	Do all text links lead to the appropriate website?	Internal navigation links working
8	Do all image links lead to the appropriate website?	Image links functional
9	Are all the images and links clearly visible on the page?	Visual elements display correctly
10	Does the Web page work properly in all the tested browsers?	Basic functionality works across browsers
11	Does the Web page take too long to be loaded fully?	Page load times acceptable
12	Is the navigation sequences correct through all the Web pages on the site?	Navigation flow tested

	Prepared by (Student)	Approved by (Faculty)
	Project Group No:03	LE THANH NHAN
Signature		
Date		

13	Is the JavaScript code working as expected in all click events?	Basic click events functioning
14	Can users login and access their respective dashboards?	Authentication and role-based access working
15	Is the course enrollment and content access working?	Students can enroll and view course materials

No	Member	Contents		Table Name	Activity Plan	Date of Preparation of Activity Plan		
					Planned Start Date	Actual Start Date	Actual Days	Status
1	**Lê Thị Hạnh Mai**	Admin	*Course Management (Add/Edit/Delete Courses)*	Courses Lessons	20/08/2025	20/08/2025		100
2		User	*View & Enroll Course, Access Lessons*		20/08/2025	20/08/2025		
1	**Cống Hiển Khoa**	Admin	*User Management (Add/Edit/Delete Users) Payment management (checking, confirming and processing transactions).*	Users Enrollment Payment	20/08/2025	20/08/2025		100
2		User	*Register, Login, Update Profile, Personal Account Management Make payments, track history, and manage personal payment information.*		20/08/2025	20/08/2025		
1	**Dương Minh Mẫn**	Admin	*Assignment/Exam & Feedback Management (Add/Edit/Delete Assignments/Exam, Manage Feedback)*	Submission Feedbacks	20/08/2025	20/08/2025		100
2		User	*Submit Assignment/Exam, Give Feedback, Personal Feedback Management*		20/08/2025	20/08/2025		

	Prepared by (Student)	Approved by (Faculty)
	Project Group No:03	LE THANH NHAN
Signature		
Date		