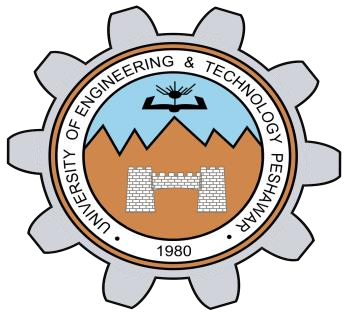
**Project Report for DBMS :**

**Online Learning Platform**



**CSE 403L: DBMS LAB**

**Submitted By:**

**Muhammad Maaz (22PWCSE2204)**

**Muhammad Talha Khan (22PWCSE2213)**

**Muhammad Ibad Khan (22PWCSE2170)**

**Section: A**

**Submitted to : Engr . Maam Sumiyyea Salahudin**

**Department of Computer Systems Engineering**

**University of Engineering and Technology Peshawar**

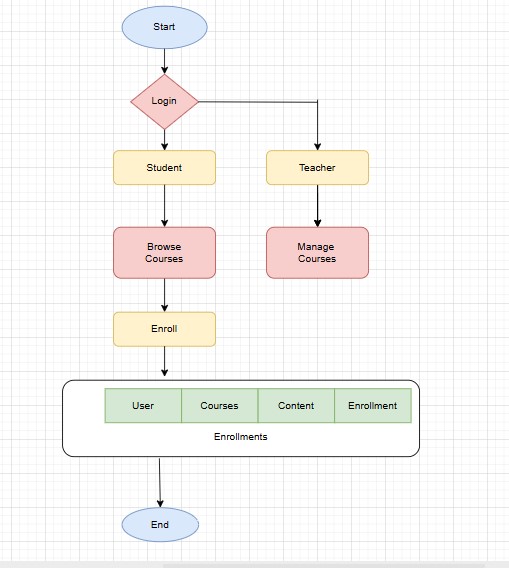
**INTRODUCTION:**

The Online E-Learning Platform is a digital education management system that provides remote access to courses for students and a management interface for teachers and administrators. It bridges the gap between learners and educators through a web-based portal. The platform supports **student and teacher login**, **course management with associated fees**, **progress tracking**, and **enrollment records**. This project simulates the core features of modern LMS (Learning Management Systems) such as Coursera or Udemy, with customization options for organizations and academic institutions.

**Implementation:**

1. **Frontend:**
   * Built using modern web technologies (likely **React.js** or **Laravel Blade**).
   * Responsive UI featuring:
     + Course progress bars
     + User details
     + Enrolled courses list
     + Instructor names
     + Enrollment date tracking o Navigation bar with **Courses** and **Logout** options.
2. **Backend:**
   * Frameworks like **Django**, **Laravel**, or **Node.js** for server-side processing.
   * Admin dashboard for managing:
     + Course content and fees
     + Enrollment records  Teacher and student accounts
3. **Database:**
   * Relational database like **MySQL** or **PostgreSQL** to store:
     + User accounts (Student/Teacher)
     + Course metadata
     + Enrollment details and payment status
     + Progress percentage tracking
4. **Authentication System:**
   * Separate login for **teachers** and **students** with role-based access control.
   * Password security through hashing algorithms (e.g., bcrypt).
5. **Other Features (inferred from UI):**
   * Admin can view all enrollments. o Teachers are assigned to specific courses. o Each student’s progress is tracked per course.

**Flowchart:**



**Programming languages/:**

1. Php (Backend)
2. Laravel 11+
3. Mysql 4. Html
4. CSS 3
5. Java script(Frontend)

**Working principle:**

The Online E-Learning Platform operates based on the interaction between users and digital learning content through structured courses and enrollment mechanisms.

1. **Users (Students, Teachers, Admins):**
   * **Student Users:**
     + Register or log in to the platform. o Browse the available courses (some may be free, others paid). o Enroll in a course by submitting payment (if required). o View enrolled courses in their personal dashboard.
     + Track their learning progress through progress bars and completion status.
     + Receive certificates upon successful course completion.
   * **Teachers/Instructors:**
     + Log in through a separate interface. o Create and manage course content (videos, PDFs, quizzes). o Monitor student enrollments and their individual progress. o Grade quizzes or assignments (if included).
     + Communicate with students through messaging or announcements.
   * **Admin:** o Access full control over the system through the admin dashboard. o Manage all user accounts (teachers and students). o Approve courses, assign instructors, and manage fees. o View analytics such as student progress, revenue reports, and course popularity.

1. **Courses:**
   * Each course has:
     + A **title**, **description**, and **instructor** assigned.
     + A **fee** (optional) for access. o Defined **content modules** such as lectures, readings, quizzes, or assignments.
     + A **progress tracking mechanism** to update student advancement in real-time.
   * Courses are visible to all users through a "Courses" page or dashboard.

1. **Content:**
   * Course content includes:
     + **Pre-recorded video lectures**
     + **Downloadable resources** like notes, slides, and eBooks o **Interactive quizzes or assignments** o **Discussion sections** for engagement
   * Students must go through the content **sequentially or module-wise**.
   * Progress is updated after every completed module or quiz submission.

1. **Enrollments:**
   * When a student selects a course:
     + The system records an **enrollment entry** linking the student, course, and enrollment date. o A **payment gateway** (if integrated) processes the transaction. o The enrolled course becomes visible on the student’s dashboard (as shown in the screenshot). o Each enrollment tracks the **completion percentage**, helping both student and teacher monitor progress.
     + Admin can view all enrollments with details like student email, course name, progress, and instructor.

**Laravel Development, MySQL Integration & Deployment (Summary)**

**Laravel Overview**

Laravel is a PHP web framework based on the MVC architecture. It offers built-in tools for authentication, routing, migrations, Blade templating, and database access via Eloquent ORM.

**Laravel + MySQL Integration**

* **Database Config:** Set in .env file (DB name, user, password).
* **Migrations:** Create and update MySQL tables using php artisan migrate.
* **Seeding:** Insert dummy data using php artisan db:seed.
* **Eloquent ORM:** Interact with MySQL using simple PHP syntax.

**Example:**

php

CopyEdit

$course = Course::create(['title' => 'Laravel Dev']);

**Deployment Steps**

1. **Install dependencies:** composer install
2. **Push code to hosting (e.g., GitHub/Railway/cPanel)**
3. **Set environment (.env) for production**
4. **Run migrations:** php artisan migrate --force
5. **Set permissions:** chmod -R 775 storage/ bootstrap/cache
6. **Point domain to /public folder**

**Tools Used**

* **Laravel** – Backend development
* **MySQL** – Relational database
* **XAMPP / Railway** – Local + live deployment
* **GitHub** – Code versioning

**Conclusion**

Laravel and MySQL together provide a clean, scalable, and efficient full-stack development environment. Laravel’s built-in tools make it easy to manage databases, while deployment ensures production readiness

**Improvements and Future Enhancements:**

1. **Paid Course Integration:**

* + - Add secure payment gateways (PayPal, JazzCash, etc.) for course fees.
    - Auto-generate invoices and track payment status.

2. **Institutional Use (Universities, Academies, Institutes):**

* + - Adapt the platform for formal education.
    - Useful for universities, coaching academies, and technical institutes offering online courses or diplomas.
    - Can be customized with branding and official certifications.

3. **Certificates with Verification:**

* + - Auto-issue certificates on course completion. o Include institution logos, signatures, and a verification link.

4. **Dashboard Improvements:**

* + - Create separate dashboards for students, teachers, and admins.
    - Show detailed analytics on progress, completion, and performance.

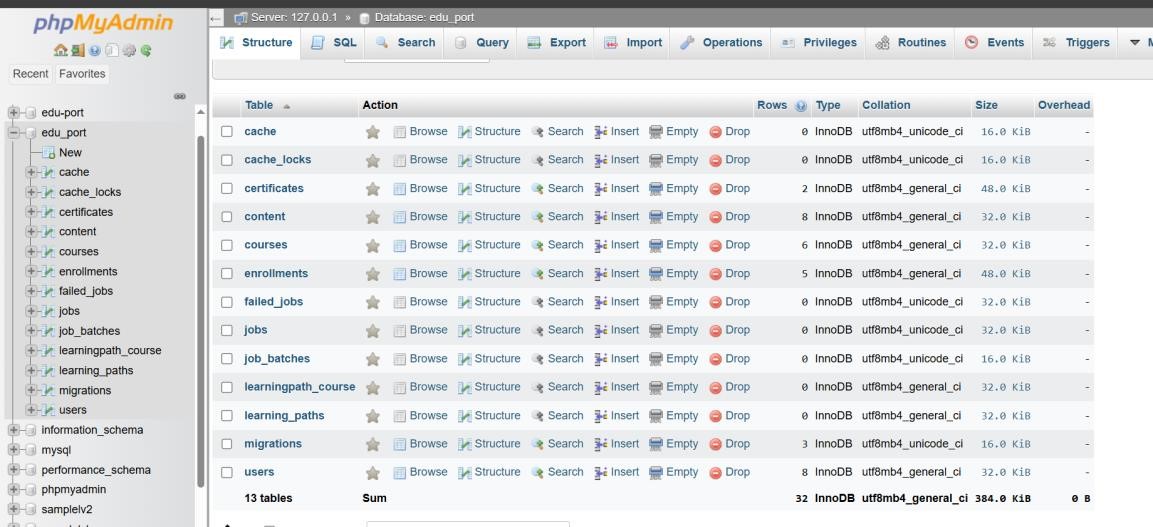
1. **Mobile App & Real-time Features:**

o Add mobile support for learning on the go. o Integrate live classes via Zoom/Google Meet and enable in-app chat or discussion forums.

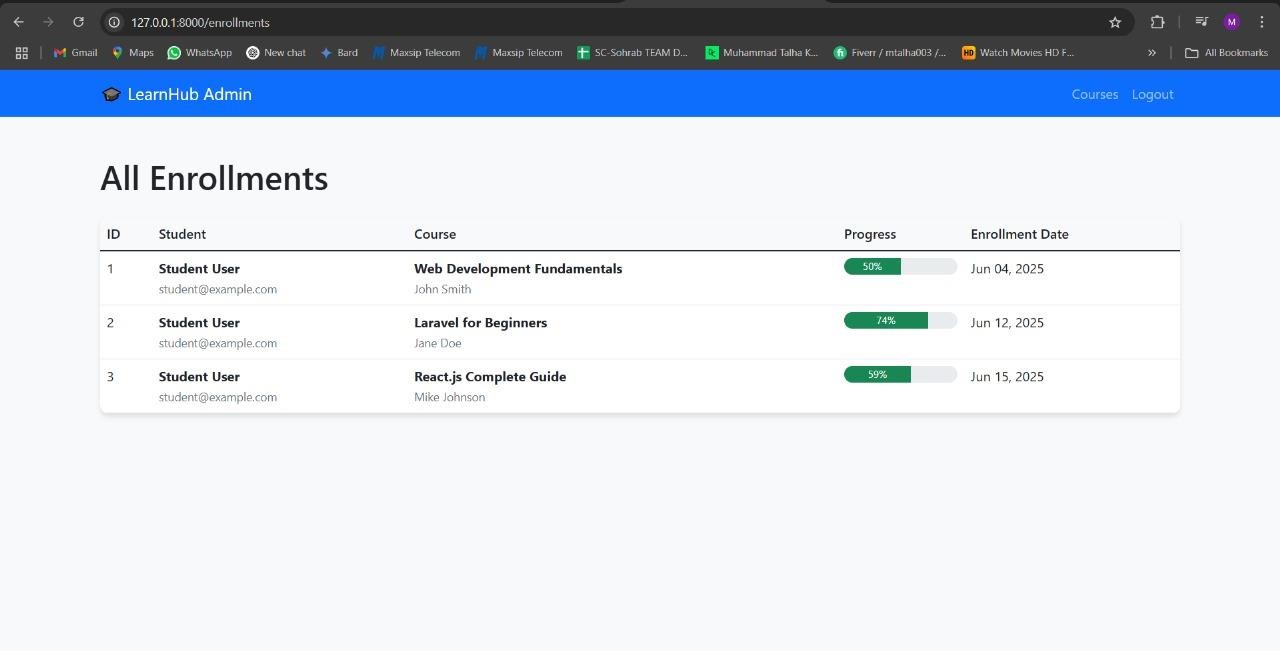
**References:**

* + - * Chatgpt
      * Laraval guide
      * Github
      * Youtube

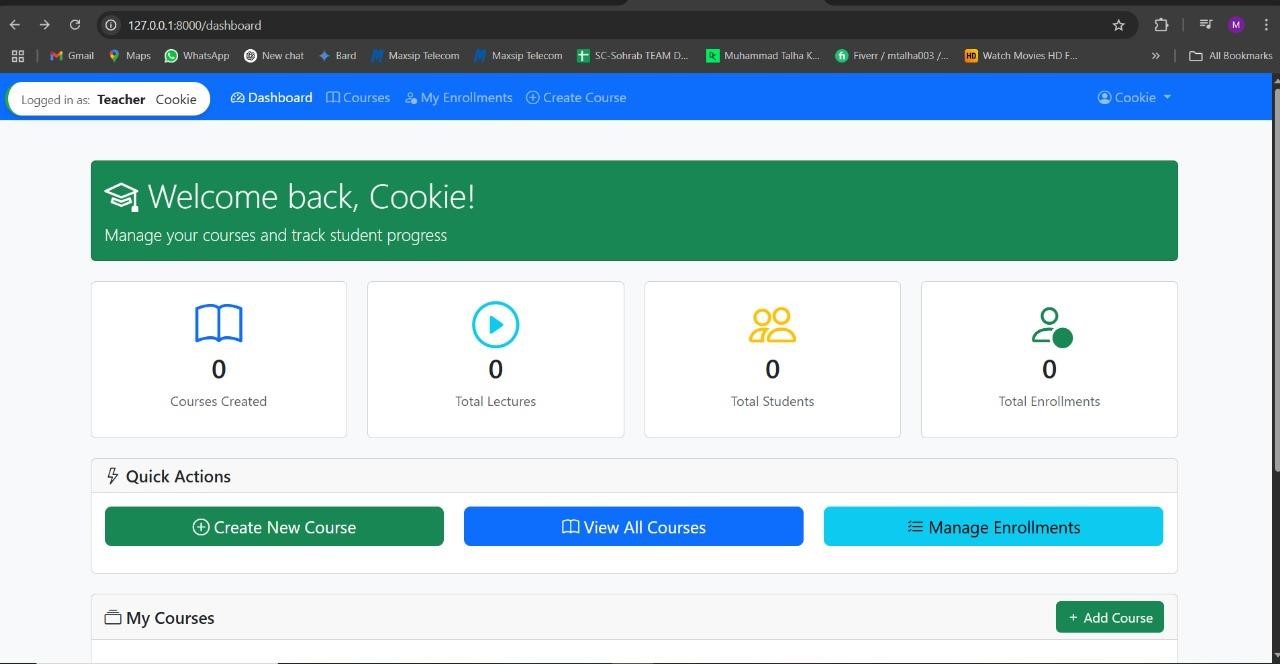
**SQL Database Table and Queries:**



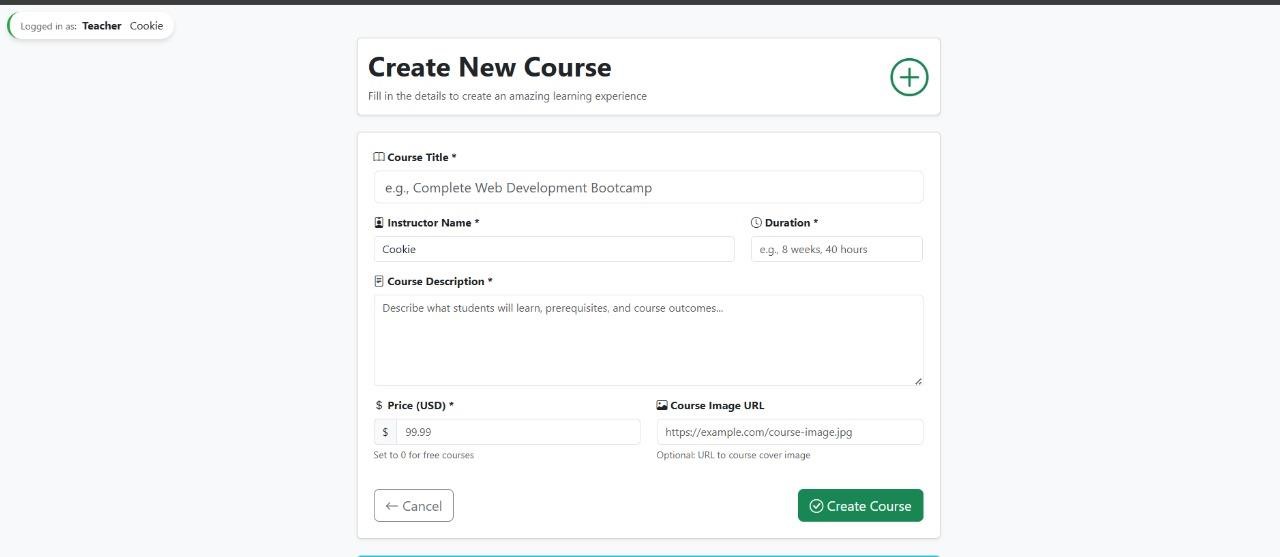
**Admin panel for all Enrollments:**



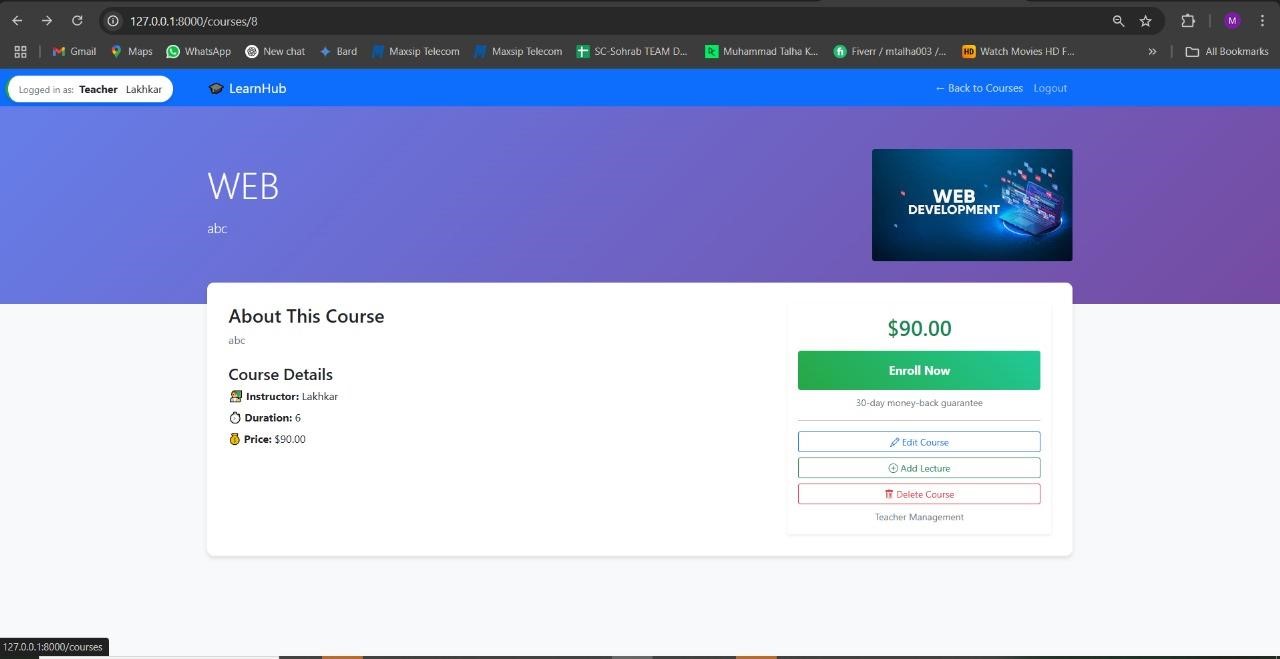
**Dashboard:**



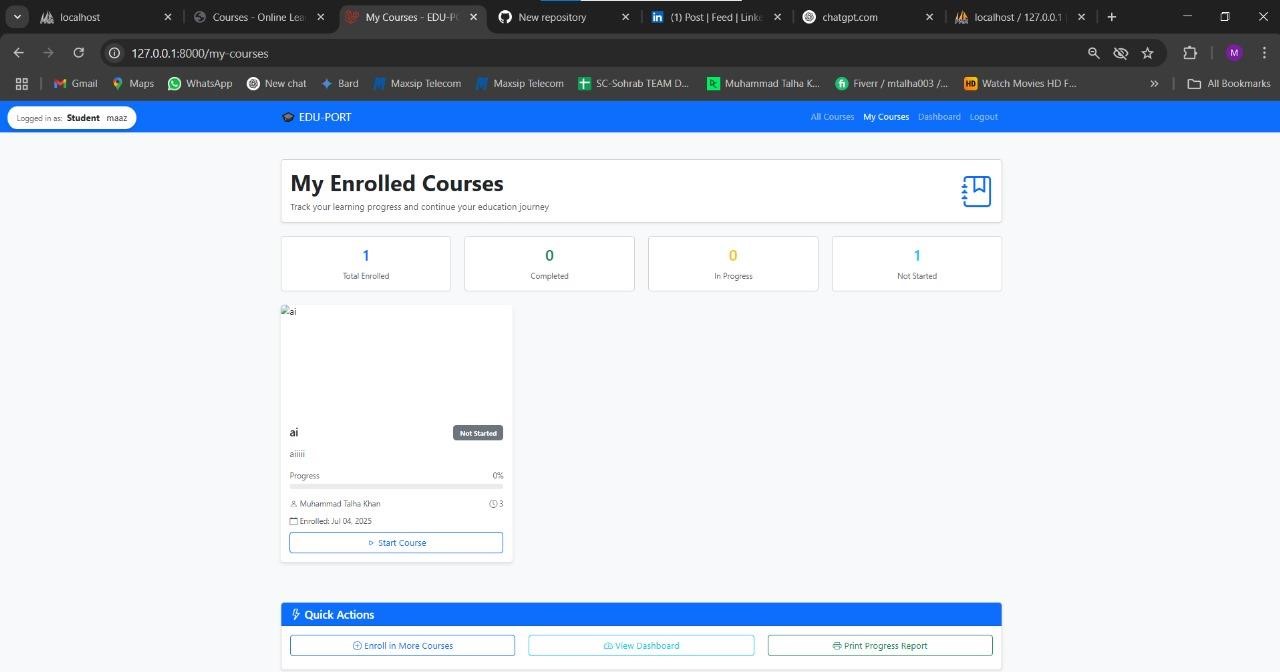
**Teacher login for Creating the Course:**



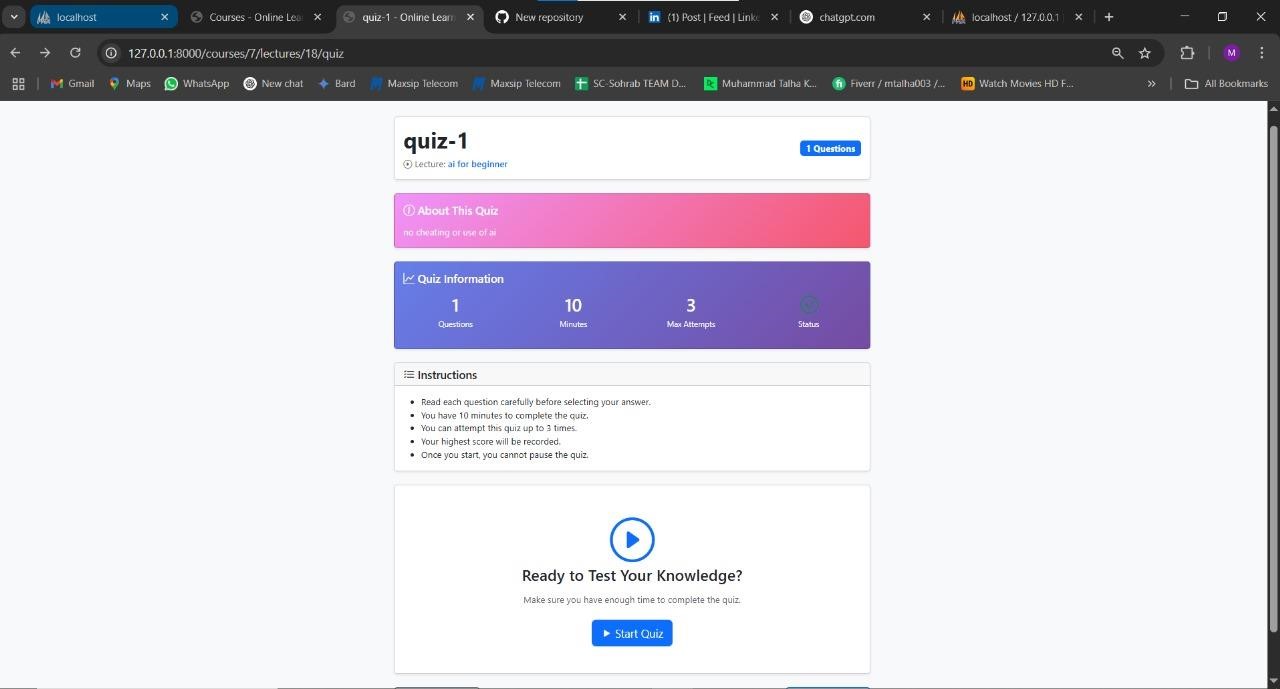
**Course details:**



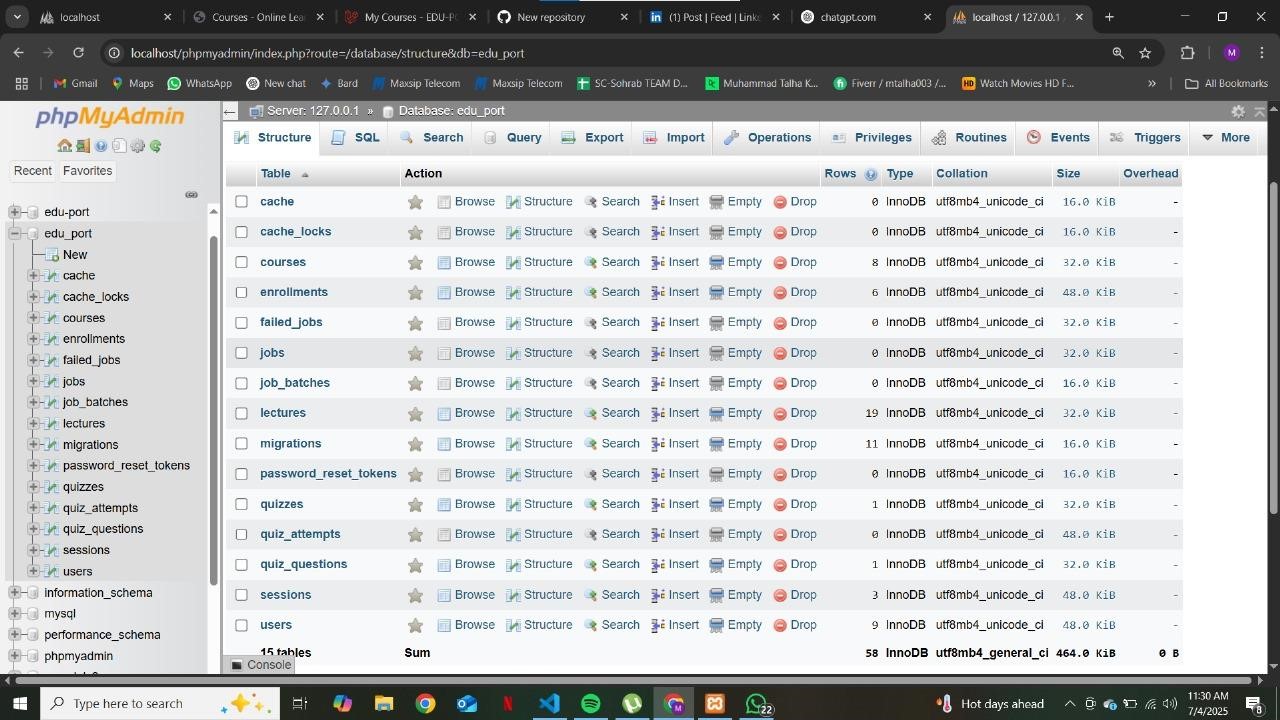
**Enrollemnt data:**



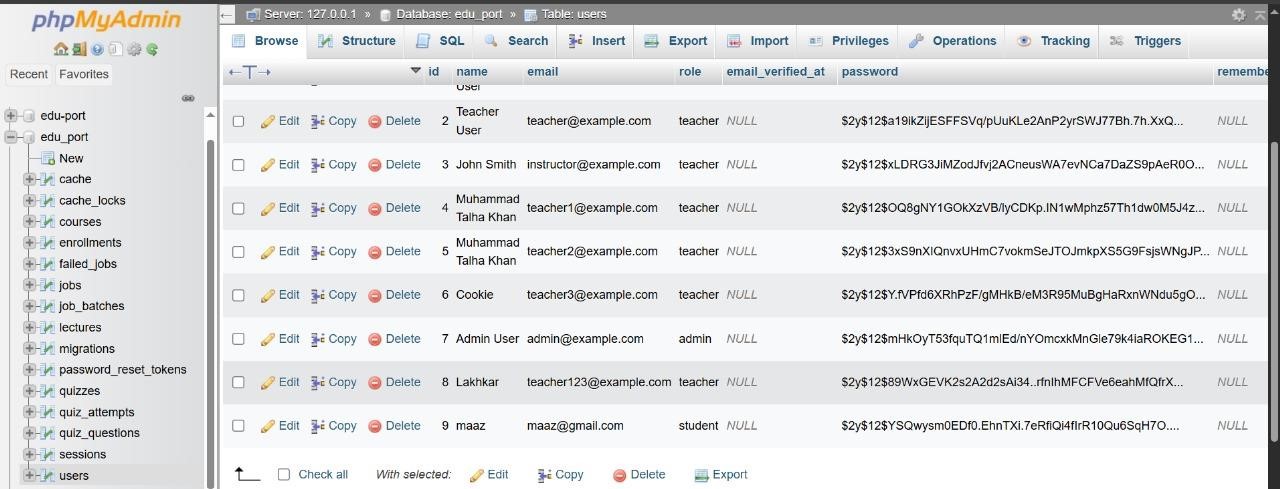
**Quizes:**



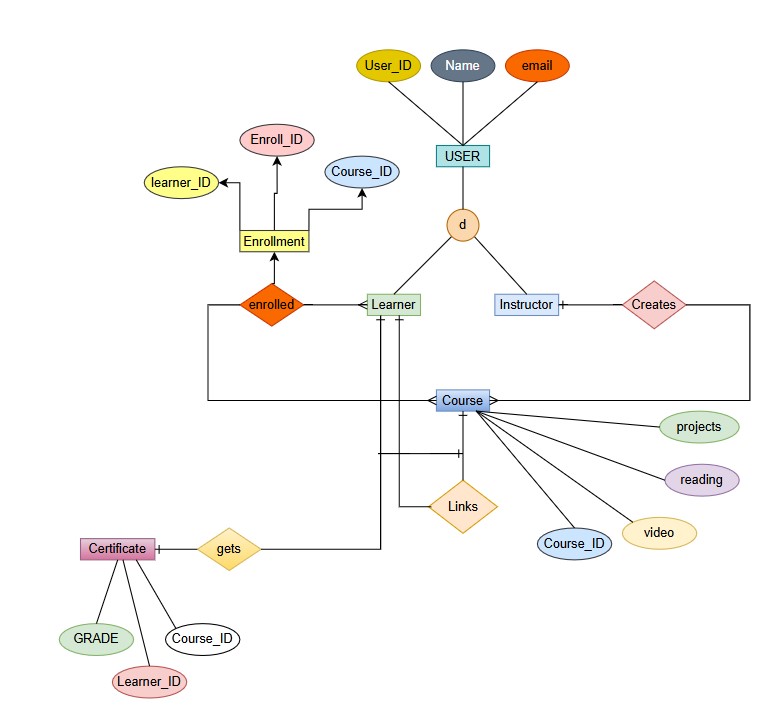
**Database View:**



**User:**



**EER Diagram:**



**Conclusion:**

This Online E-Learning Platform offers a robust, scalable, and user-friendly solution for modern education delivery. It effectively supports blended and remote learning models, making it a valuable tool for educational institutions, training centers, and individual instructors.

With well-structured course management, detailed tracking, and role-based access, it streamlines the learning process while keeping all stakeholders engaged.

*THE END*