



Recorded Courses System

Introduction :

Dear Developer, I hope you're doing well!

Based on the following user story, you are tasked with building a database application for a **Recorded Courses System** using **Microsoft SQL Server Database**. The system should follow the development instructions and requirements outlined below to meet the user needs.

Abstract:

Hello, my name is Khalid. I run an educational platform that offers recorded courses for learners. I want to expand my platform to include a more structured system where:

- Instructors can upload courses and manage content.
- Learners can browse courses, enroll, track progress, and leave reviews.
- The platform can support subscription plans, course bundles, and detailed reporting for both instructors and admins.

My goal is to create a user-friendly system that benefits both learners and instructors, offering analytics and fostering engagement. Additionally, I would like features to reward top-performing instructors and enable learners to bookmark their favorite courses.

User Story

- **Entities and Features:**

- **Course:** Includes title, description, category, level, price, discount (if any), and total duration. Each course has multiple recorded **lessons**, and each lesson has a video file, a title, and a duration.
- **User:** Includes learners and instructors, each with email, password, profile image, and bio.
- **Instructor Features:**
 - Upload courses, add lessons, and set prices.
 - View course performance (**enrollments**, **reviews**, and ratings).
- **Learner Features:**
 - Browse, search, and filter courses.
 - Enroll in courses and track progress.
 - **Bookmark** favorite courses and leave reviews.
- **Admin Features:**
 - Manage users, courses, **categories**, and pricing plans.
 - Generate analytics **reports** on revenue, enrollments, and instructor performance.

- **Functionalities:**

- A learner can enroll in multiple courses.
- Each lesson in a course has a completion status tracked for the learner.
- Reviews and ratings are linked to courses and instructors.

Database Queries

1. **Retrieve Courses by Category:**

- Write a query to fetch all courses within a selected category, including title, price, and ratings.

2. **Search Courses:**

- Implement a query to search courses by keyword in the title or description.

3. **Track Learner Progress:**

- Create a query to retrieve the progress of a learner in a specific course, including completed lessons and remaining lessons.

4. **Top-Rated Courses:**

- Write a query to identify the top-rated courses based on average ratings and enrollment numbers.
 -
-

Database Views

1. **Active Courses View:**
 - Create a view to display all active courses, including category, instructor name, and total enrollments.
 2. **Learner Progress View:**
 - Implement a view to show progress for all learners, including course details and completion percentages.
 3. **Revenue Summary View:**
 - Create a view summarizing revenue generated by each course, instructor, and overall platform.
-

Stored Procedures

1. **Enroll Learner in Course:**
 - Write a stored procedure to handle learner enrollments, ensuring the course exists and the learner has an active account.
2. **Add a New Course:**
 - Implement a stored procedure for instructors to add courses, including lessons and metadata.
3. **Generate Revenue Report:**
 - Create a stored procedure to generate a report of total revenue per course and instructor within a specific timeframe.
4. **Manage Discounts:**
 - Write a stored procedure to apply or update discounts on courses based on promotional campaigns.

Implementation Requirements : Please Follow this instruction while developed this Project

1- Build ER – Diagram

2- Create Word Document to implement following

- a- The scope of the project
- b- The Mandatory object
- c- The Software Type and the functionality follow in the system
- d- Summarize the Objects
- e- Summarize the Objects Relationships

- 3- Build Data base System using SQL Command and Follow the Normalization Pattern
- 4- Build the necessary DB configuration and constraints depends on the system questions
- 5- Fell Free To add any additional feature to the system
- 6- remember to add dummy data for testing purpose

Acceptance Criteria

- 1- Each Entity Must have a Primary Key and Make it's Identity
- 2- Ensure about applying check constraints and Default at least for 2 properties in each table
- 3- Configure the relation using foreign key constraints and use cascade
- 4- Submit SQL Scripts
- 5- Submit SQL Backup File (Data tier applications)
- 6- Submit Word Document
- 7- Submit ER Diagram

With The best Wishes From Your Trainer **Jasser Alshaer**

*fell free to ask questions about any thing

* remember you will be the best