

Functional Requirements:

1. User Management

1.1 User Registration

- **Create Account:** The system shall allow users to create a new account using their email address.
- **Email Confirmation:** The system shall send a confirmation email to users after registration. Users must click the link in the email to activate their account.

1.2 User Login

- **Login:** The system shall allow users to log in using their email and password or through their linked social media accounts.
- **Password Recovery:** The system shall provide a password recovery feature where users can reset their password by clicking a link sent to their registered email.

1.3 User Profile Management

- **Edit Profile:** The system shall allow users to edit their personal information, such as name, profile picture, and contact details.
- **User Dashboard:** The system shall provide each user with a personalized dashboard displaying their enrolled courses, learning progress, and notifications about new activities.

2. Course Management

2.1 Course List

- **Show Courses:** The system shall display a list of all available courses. Users shall be able to filter courses by subject, level, or other criteria.
- **Course Preview:** The system shall display key details for each course, including title, description, instructor name, rating, and duration.

2.2 Course Creation (for Instructors/Admins)

- **Course Creation Tools:** The system shall provide instructors with tools to create, edit, and manage courses, including adding lessons, quizzes, and other resources.
- **Upload Content:** The system shall allow instructors to upload various types of content, such as videos, PDFs, and quizzes.

2.3 Course Enrollment

- **Enroll in Courses:** The system shall allow users to enroll in courses with a single click.

- **Send Enrollment Request:** If a course requires approval, the system shall allow students to send a request to join the course. The instructor or admin shall review and approve or reject the request.
- **Waitlist Feature:** If a course is full, the system shall allow users to join a waitlist and notify them if a spot becomes available.

3. Content Delivery

3.1 Section Structure

- **Sequential Learning:** The system shall require users to complete each Section and its associated quiz before moving to the next Section.
- **Different Content Types:** The system shall support various content types, including videos, reading materials, quizzes, and assignments.

3.2 Multimedia Support

- **Video Playback:** The system shall support high-quality video playback for all lessons.
- **Downloadable Resources:** The system shall allow users to download additional resources, such as PDFs and study guides.

4. Assessment and Feedback

4.1 Quizzes and Assignments

- **Mandatory Quizzes:** The system shall require users to complete quizzes linked to each lesson to unlock the next Section.
- **Instant Feedback:** The system shall provide immediate feedback to users after completing a quiz, showing their scores and the correct answers.

4.2 Certificates

- **Digital Certificates:** The system shall issue a digital certificate to users upon completing a course. Users shall be able to share these certificates on their profiles or job applications.

5. Support and Feedback

5.1 Customer Support

- **Help Center:** The system shall provide a help center with frequently asked questions (FAQs) and guides to assist users.

- **Contact Support:** The system shall allow users to contact support via a form or live chat for additional assistance.

5.2 Feedback Collection

- **User Surveys:** The system shall send regular surveys to users to collect feedback for improving the platform.

6. Analytics and Reporting

6.1 User Analytics

- **Track Learning Progress:** The system shall allow users to view their learning progress, including completed lessons and quiz scores.
- **Engagement Reports:** The system shall provide admins with reports on user engagement, such as login frequency and course popularity.

6.2 Course Performance

- **Performance Dashboard:** The system shall provide admins with a dashboard to monitor course performance and identify areas for improvement.

7. Instructor-Specific Features

7.1 Instructor Dashboard

- **Course Management:** The system shall provide instructors with a dashboard to create, edit, and manage their courses.
- **Student Management:** The system shall allow instructors to view and manage students enrolled in their courses, including tracking progress and sending notifications.
- **Student Request Management:** The system shall allow instructors to approve or reject student requests to enroll in their courses.

Non-Functional Requirements:

1. Performance

- **Response Time:** The system shall load pages within 2 seconds under normal usage conditions.
- **Concurrent Users:** The system shall support up to 10,000 concurrent users without degradation in performance.
- **Scalability:** The system shall be scalable to handle a 50% increase in users and data over the next two years.

2. Usability

- **User Interface:** The system shall have an intuitive and user-friendly interface, ensuring that new users can navigate the platform with minimal guidance.
- **Accessibility:** The system shall comply with WCAG 2.1 Level AA standards to ensure accessibility for users with disabilities.
- **Mobile Compatibility:** The system shall be fully functional and responsive on mobile devices, including tablets and smartphones.

3. Security

- **Data Encryption:**
 - All sensitive data (e.g., passwords, payment information) shall be encrypted using industry-standard encryption protocols (e.g., AES-256).
 - Passwords shall be hashed using ASP.NET Identity's secure hashing algorithms (e.g., PBKDF2 with HMAC-SHA256) before storage in the database.
- **Authentication:**
 - The system shall enforce strong password policies, including minimum length, complexity requirements, and regular password expiration.
 - The system shall use ASP.NET Identity to manage user authentication, including password hashing, validation, and user session management.
 - The system shall support multi-factor authentication (MFA) for added security.
- **Data Privacy:**
 - The system shall comply with GDPR and other relevant data protection regulations to ensure user privacy.
 - Sensitive user data shall be stored securely, and access shall be restricted to authorized personnel only.

4. Reliability

- **Uptime:** The system shall have an uptime of 99.9% annually, excluding scheduled maintenance.
- **Error Handling:** The system shall provide meaningful error messages and recover gracefully from failures without data loss.
- **Backup and Recovery:** The system shall perform daily backups and ensure data can be restored within 1 hour in case of failure.

5. Maintainability

- **Modularity:** The system shall be designed with modular components to allow easy updates and maintenance.
- **Documentation:** The system shall include comprehensive technical documentation for developers and user guides for end-users.
- **Logging:** The system shall log all critical actions (e.g., login attempts, course enrollment) for auditing and troubleshooting purposes.

6. Compatibility

- **Browser Support:** The system shall support the latest versions of major browsers, including Chrome, Firefox, Safari, and Edge.
- **Operating Systems:** The system shall be compatible with Windows, macOS, and Linux operating systems.

7. Scalability

- **Load Handling:** The system shall handle a 50% increase in user load without requiring significant architectural changes.
- **Database Scalability:** The database shall be capable of scaling horizontally to accommodate growing data storage needs.

8. Availability

- **Disaster Recovery:** The system shall have a disaster recovery plan in place to ensure continuity of operations in case of major failures.
- **Maintenance Windows:** Scheduled maintenance shall be communicated to users at least 24 hours in advance and shall not exceed 2 hours per month.

9. Legal and Compliance

- **Data Protection:** The system shall comply with all applicable data protection laws, including GDPR.
- **Intellectual Property:** The system shall ensure that all uploaded content (e.g., course materials) respects intellectual property rights and copyright laws.

10. Localization and Internationalization

- **Language Support:** The system shall support multiple languages, with English as the default language.
- **Localization:** The system shall allow for localization of content, including date formats, currencies, and regional settings.

High-Level Design:

1. System Architecture

The system will follow a **3-tier architecture**:

1.1. Presentation Layer (Frontend):

- Built using **React/Next.js** for a dynamic and responsive user interface.
- Communicates with the backend via **RESTful APIs**.

1.2. Application Layer (Backend):

- Built using **ASP.NET Core**.
- Implements business logic, authentication, and authorization.
- Uses **ASP.NET Identity** for user management, password hashing, and role-based access control.
- Exposes **RESTful APIs** for the frontend to interact with.

1.3. Data Layer (Database):

- Uses **SQL Server** for structured data storage.
- Stores user data, course content, progress tracking, and analytics.
- Supports database migrations for schema updates.

2. Key Components

2.1. User Management Module:

- Handles user registration, login, and profile management.
- Uses **ASP.NET Identity** for authentication and authorization.

2.2. Course Management Module:

- Allows instructors to create, update, and manage courses.
- Enables students to browse, enroll, and track progress in courses.

2.3. Content Delivery Module:

- Delivers course content (videos, PDFs, quizzes) in a structured manner.
- Ensures sequential learning by locking sections until prerequisites are completed.

2.4. Assessment and Feedback Module:

- Manages quizzes, assignments, and instant feedback.
- Issues digital certificates upon course completion.

2.5. Analytics and Reporting Module:

- Tracks user progress and engagement.
- Provides dashboards for admins and instructors to monitor course performance.

2.6. Support and Feedback Module:

- Includes a help center, live chat, and feedback collection mechanisms.

3. Workflow

3.1. User Registration and Login:

- Users register via email or social media.
- **ASP.NET Identity** handles password hashing and authentication.

3.2. Course Enrollment:

- Students browse courses, send enrollment requests (if required), and get approved by instructors.

3.3. Content Delivery and Progress Tracking:

- Students complete lessons and quizzes sequentially.
- Progress is tracked and displayed on the user dashboard.

3.4. Assessment and Certification:

- Students complete quizzes and assignments.
- Digital certificates are issued upon course completion.

3.5. Analytics and Reporting:

- 1.1. Admins and instructors view engagement and performance reports.

4. Technology Stack

- **Frontend:** React/Next.js
- **Backend:** ASP.NET Core
- **Database:** SQL Server
- **Authentication:** ASP.NET Identity, JWT Token
- **APIs:** RESTful APIs
- **Hosting:** Cloud-based (Monster ASP, Vercel)

Detailed Design (DD):

1. Introduction

- **Purpose:** The document outlines the detailed design of the system, describing how the database schema and components will work together to support the e-learning platform.
- **Scope:** This design document focuses on the database schema and its role in managing courses, users, tests, and related functionalities.

2. Database Schema Overview

- The database consists of the following tables:
 - **Authentication and Authorization:**
 - AspNetUsers, AspNetRoles, AspNetUserRoles, AspNetUserClaims, AspNetRoleClaims, AspNetUserLogins, AspNetUserTokens
 - **Course Management:**
 - Courses, Categories, Sections, Lessons, Section Tests
 - **User Interactions:**
 - Enrollments, Students, Instructors, InstructorJoinRequests, Comments, CourseRequests
 - **Tests and Feedback:**
 - Questions, ChoicesQuestions, UserAnswers, TestResults, courseFeedbacks
 - **Files and Tokens:**

- Files, RefreshTokens

3. Detailed Design of Key Tables

- **AspNetUsers:** Handles user authentication and stores user details.
 - Fields: Id, Username, Email, etc.
- **Courses:** Represents courses available on the platform.
 - Fields: Id, Name, Description, CategoryId, etc.
 - Relationships: Linked to Categories (via CategoryId), Sections, Files, Instructors and Enrollments.
- **Sections:** Represents sections within a course.
 - Fields: Id, CourseId, Title, Order, etc.
 - Relationships: Linked to Courses, Lessons, and SectionTests.
- **SectionTests:** Stores test information for sections.
 - Fields: Id, SectionId, TestName, etc.
 - Relationships: Linked to Sections and Questions.
- **Questions:** Stores questions for tests.
 - Fields: Id, TestId, QuestionText, etc.
 - Relationships: Linked to SectionTests and ChoicesQuestions.
- **ChoicesQuestions:** Represents multiple-choice answers for questions.
 - Fields: Id, QuestionId, ChoiceText, IsCorrect, etc.
- **Enrollments:** Tracks students enrolled in courses.
 - Fields: Id, StudentId, CourseId, etc.
 - Relationships: Linked to Students and Courses.

4. System Workflows

- **User Registration and Login:**
 - Tables: AspNetUsers, AspNetUserRoles, RefreshTokens
 - Workflow: A user registers or logs in, and their roles and tokens are managed.

- **Course Enrollment:**
 - Tables: Courses, Students, Enrollments
 - Workflow: Students enroll in courses, creating an entry in the Enrollments table.
- **Course Progression:**
 - Tables: Sections, Lessons, SectionTests, TestResults
 - Workflow: Students complete sections and pass tests (TestResults) to unlock new sections.
- **Test Submission:**
 - Tables: SectionTests, Questions, ChoicesQuestions, UserAnswers, TestResults
 - Workflow: Students submit answers for questions, and results are stored in TestResults.

5. ERD Relationships

- Courses have multiple sections, and sections have multiple lessons and tests.
- Tests are linked to questions, which may have multiple-choice answers.
- Users (instructors and students) interact with the system through enrollments, course creation, and test submissions.

6. Future Enhancements

- Adding support for multimedia files in lessons (Files table).
- Implementing advanced analytics for course performance and student progress.

System Integration Plan:

1. Introduction

This section outlines the process for integrating the e-learning platform components with the database, front-end, back-end, and external services to ensure seamless functionality. The integration plan aims to achieve a unified system where users, courses, tests, and other entities operate cohesively.

2. Integration Objectives

- 1.2. Ensure smooth communication between the front-end, back-end, and database.
- 1.3. Securely manage user authentication and authorization.
- 1.4. Allow efficient interaction between users and platform features such as courses, tests, and progress tracking.
- 1.5. Support scalability and performance for concurrent users.
- 1.6. Facilitate integration with third-party services like email notifications, file storage, and analytics.

3. Integration Components

3.1. Front-End Integration

- **Description:** The front-end is responsible for providing a user-friendly interface for students, instructors, and administrators.
- **Technologies:** React, Angular, or any modern front-end framework.
- **Integration Steps:**
 - Connect to the back-end API using HTTP/HTTPS requests (e.g., REST API).
 - Authenticate users via the back-end authentication endpoints (AspNetUsers).
 - Dynamically fetch and display data (e.g., courses, lessons, and tests) from the database through API calls.

3.2. Back-End Integration

- **Description:** The back-end manages business logic, processes user requests, and interacts with the database.
- **Technologies:** ASP.NET Core, .NET Web API.
- **Integration Steps:**
 - Implement API endpoints for:
 - User management: Login, registration, role assignment.

- Course operations: Fetch, create, update, and delete courses.
- Test operations: Submit answers, calculate results, unlock next sections.
- Secure API communication using HTTPS and token-based authentication (e.g., JWT stored in RefreshTokens).
- Handle error responses and status codes to ensure smooth front-end interaction.

3.3. Database Integration

- **Description:** The database stores all persistent data, including users, courses, tests, and progress.
- **Technologies:** SQL Server.
- **Integration Steps:**
 - Design a schema migration plan using EF Core Migrations.
 - Optimize queries for frequent operations like fetching courses, lessons, and test results.
 - Use stored procedures or database triggers where needed for critical operations (e.g., unlocking sections upon test completion).

3.4. Authentication and Authorization

- **Description:** Ensure secure user access and role-based functionality.
- **Components:**
 - ASP.NET Identity for user management (AspNetUsers, AspNetRoles, AspNetUserRoles).
 - Token-based authentication (e.g., JWT).
- **Integration Steps:**
 - Implement role-based access control (e.g., only instructors can create courses).
 - Secure sensitive data (e.g., encrypt passwords, protect access tokens).

3.5. Third-Party Services

- **Description:** Integration with external services enhances functionality and user experience.
- **Components:**
 - **Email Notifications:**
 - Use cases: Send confirmation emails, test results, and course updates.
 - **File Storage:**
 - Technology: Monster-ASP Cloud Storage.
 - Use cases: Store and retrieve multimedia files for courses and lessons.
 - **Analytics:**
 - Technology: custom analytics solution.
 - Use cases: Track user activity, course engagement, and test performance.

UML Use Case Description for the Codixa E-Learning Platform

1. Actors

- Student: A user who enrolls in courses, takes tests, and tracks progress.
- Instructor: A user who creates and manages courses, sections, lessons, and tests.
- Administrator: A user responsible for managing system-wide configurations, users, and roles.
- System: Represents the platform itself, including database and external services.

2. Use Cases

The following are the main use cases:

2.1. Register and Log In

- **Actor(s):** Student, Instructor, Admin
- **Description:** Allows users to create an account or log in to access platform features.
- **Steps:**

1. User provides email, password, and other details.
 2. System validates inputs and stores user information in the AspNetUsers table.
 3. User receives confirmation and is logged into the system.
- **Extensions:**
 - Forgot Password: User can reset their password via an email link.

2.2. Browse and Enroll in Courses

- **Actor(s):** Student
- **Description:** Allows students to view courses and enroll in them.
- **Steps:**
 1. Student browses courses by category or search.
 2. System fetches courses from the Courses table.
 3. Student selects a course and enrolls, creating an entry in the Enrollments table.

2.3. Create and Manage Courses

- **Actor(s):** Instructor
- **Description:** Allows instructors to create, update, or delete courses.
- **Steps:**
 1. Instructor provides course details, including title, description, and category.
 2. System stores course details in the Courses table.
 3. Instructor adds sections and lessons to the course.

2.4. Take Tests and Submit Answers

- **Actor(s):** Student
- **Description:** Enables students to take tests associated with sections of a course.
- **Steps:**
 1. Student selects a section and starts a test.
 2. System fetches questions and choices from the Questions and ChoicesQuestions tables.
 3. Student submits answers, which are stored in the UserAnswers table.
 4. System calculates the score and stores it in the TestResults table.
- **Extensions:**
 - Students can review incorrect answers and retry tests.

2.5. Unlock Next Sections

- **Actor(s):** Student
- **Description:** Unlocks the next section when the student passes a test.

- **Steps:**
 1. System checks the student's test results in the TestResults table.
 2. If the score meets the criteria, the next section is unlocked.

2.6. Provide Feedback

- **Actor(s):** Student, Instructor
- **Description:** Allows users to provide feedback on courses or instructors.
- **Steps:**
 1. User submits feedback.
 2. System stores feedback in the courseFeedbacks table.

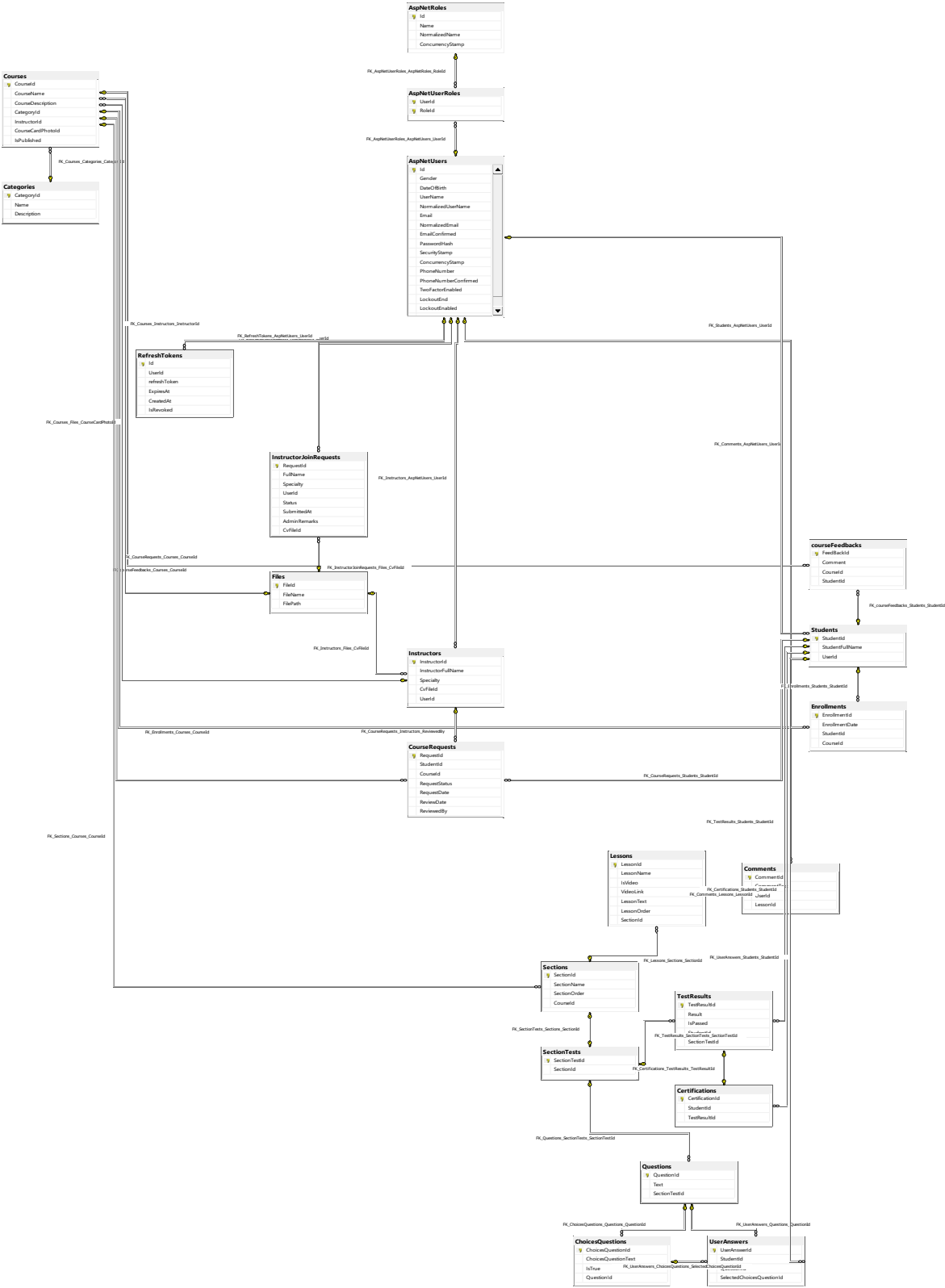
2.7. Manage Users and Roles

- **Actor(s):** Administrator
- **Description:** Allows administrators to manage platform users and roles.
- **Steps:**
 1. Admin assigns roles (e.g., student, instructor) to users via the AspNetUserRoles table.
 2. Admin updates or removes user accounts as needed.

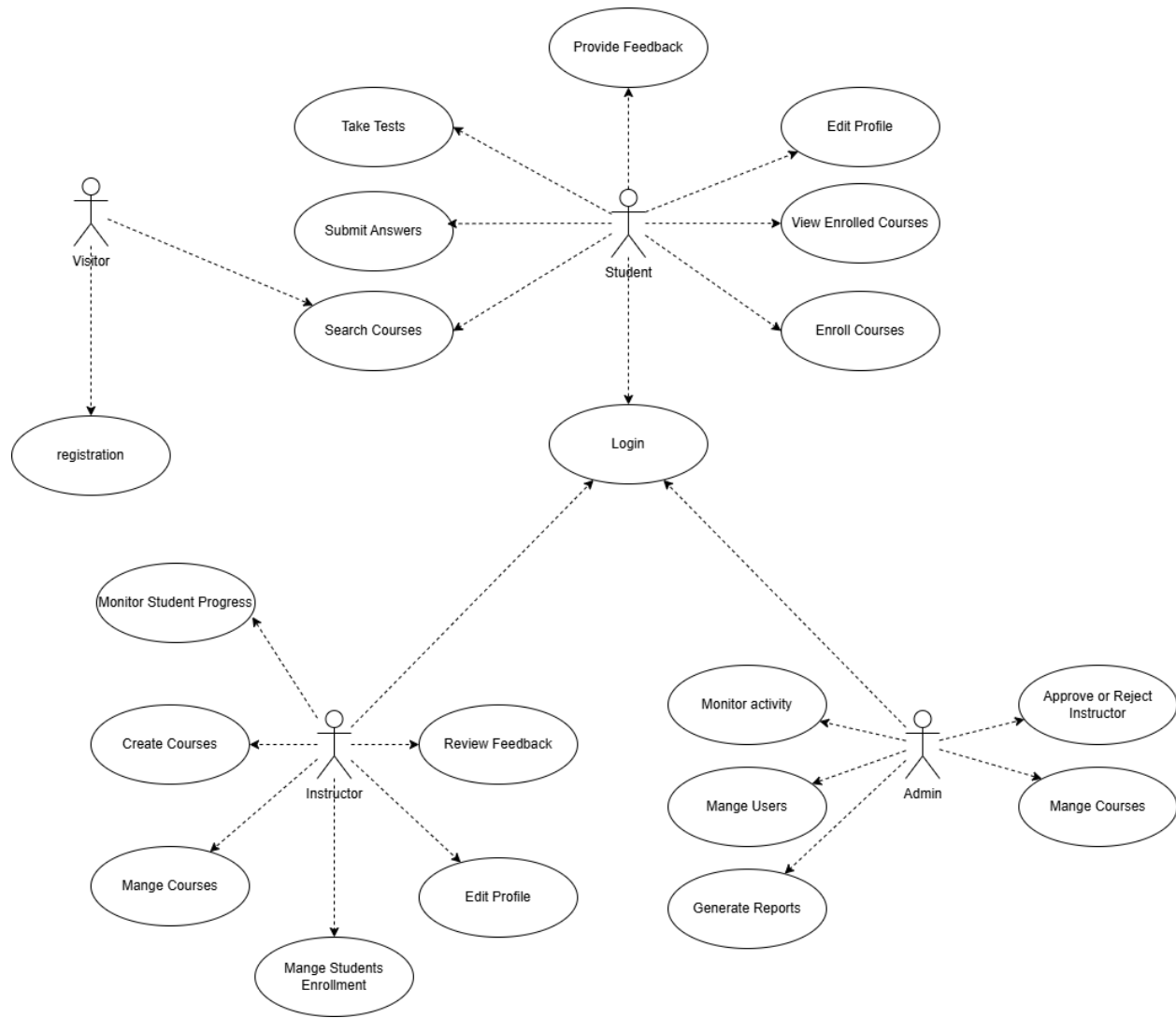
2.8. File Management


- **Actor(s):** Instructor
- **Description:** Instructors can upload files or multimedia for lessons.
- **Steps:**
 1. Instructor uploads a file.
 2. System stores the file reference in the Files table and saves the file in external storage.

Entity relationship diagram ERD:



Use Case Diagram:






[Home](#)[Courses](#)[About Us](#)[posts](#)[Contact](#)

Admin


By Themadbrains in **Inspiration**

Why UI should be on the radar of every mobile developer


Lorem ipsum dolor sit amet consectetur adipiscing elit. Praesentium eorum expedita nesciunt architecto quae aliquid a soluta ipsam! Vitae asperiores et tempore, consectetur magni alias.




Reading Blog List



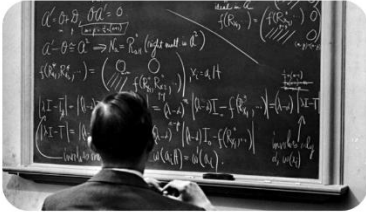
React



PHP



JavaScript



Sample Post Title


John Doe

Publisher

This is a short description for the post. It provides a brief overview of the content.

[Read More](#)

123



Sample Post Title


John Doe

Publisher

This is a short description for the post. It provides a brief overview of the content.

[Read More](#)

123




November 16, 2014

Three Pillars of Users Delight

Delight can be experienced viscerally, behaviourally, and reflectively. A great design is ...

[Research](#)[UI UX](#)

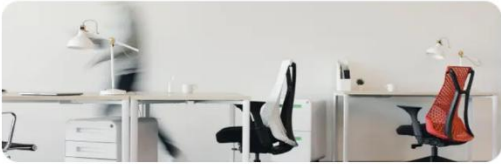


September 24, 2017

UX Mapping Methods

Visual-design principles can be applied consistently throughout the process of creating a polished UX map...

[Research](#)[UI Design](#)




March 13, 2014

Agile Development Projects and Usability

Agile methods aim to overcome usability barriers in traditional development, but past new threats to user experience quality.

[Programming](#)[Research](#)[Developments](#)



Codixa

Top learning experiences that create more talent in the world.

Product

Overview

Features

Solutions

Tutorials

Pricing

Company

About Us

Careers

Press

News

Social

Twitter

LinkedIn

GitHub

Dribbble

Legal

Terms

Privacy

Cookies

Contact

© 2024 SilhusEO. All rights reserved.

[in](#)[f](#)[G](#)[X](#)



- Topic
- Level
- Language
- Price
- Video Duration
- Features

Search Results



Vira

3 months

AWS certified solutions architect

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Delectus perferendis molestias quibusdam. Harum, reprehenderit. Exercitationem!

View Course →



John

4 months

Data Science 101

This course introduces you to data science concepts and tools. It covers everything from basic statistics to machine learning.

View Course →



Alice

6 months

Full Stack Web Development

Master both front-end and back-end technologies. Learn JavaScript, React, Node.js, and more!

View Course →



Bob

3 months

Introduction to Machine Learning

An introductory course to machine learning concepts, tools, and algorithms for beginners.

View Course →



Sara

5 months

Cyber Security Basics

Learn the fundamentals of cybersecurity and protect yourself and your data online.

View Course →



Tom

2 months

Digital Marketing Mastery

Become an expert in digital marketing strategies including SEO, content marketing, and paid ads.

View Course →

Most viewed

See all



design

3 mounthes

AWS certified solutions architect

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Delectus perferendis molestias quibusdam. Harum, reprehenderit. Exercitationem!

vira



design

3 mounthes

AWS certified solutions architect

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Delectus perferendis molestias quibusdam. Harum, reprehenderit. Exercitationem!

vira



design

3 mounthes

AWS certified solutions architect

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Delectus perferendis molestias quibusdam. Harum, reprehenderit. Exercitationem!

vira



design

3 mounthes

AWS certified solutions architect

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Delectus perferendis molestias quibusdam. Harum, reprehenderit. Exercitationem!

vira



design

AWS certif architect

Lorem ipsum consectetur c perferendis r Harum, reprel

vira



Codixa

Top learning experiences that create more talent in the world.

Product

Overview

Features

Solutions

Tutorials

Pricing

Company

About Us

Careers

Press

News

Social

Twitter

Linkedin

GitHub

Dribble

Legal

Terms

Privacy

Cookies

Contact



User name

Email Address

Message

Send



Codixa

Top learning experiences that create more talent in the world.

Product

Overview

Features

Solutions

Tutorials

Pricing

Company

About Us

Careers

Press new

News

Social

Twitter

LinkedIn

GitHub

Dribbble

Legal

Terms

Privacy

Cookies

Contact

© 2024 SiriusEG. All rights reserved.



WHO WE ARE

Lorem ipsum dolor sit amet consectetur adipisicing elit. Ea tempore laudantium at, assumenda velit itaque rem, labore adipisci facere ipsa hic distinctio dolorem, sequi nemo.



Lorem Ipsum is simply

Lorem ipsum dolor sit amet consectetur adipisicing elit. Ab voluptas excepturi quam deleniti nihil? Oditi!



Lorem Ipsum is simply

Lorem ipsum dolor sit amet consectetur adipisicing elit. Ab voluptas excepturi quam deleniti nihil? Oditi!



Lorem Ipsum is simply

Lorem ipsum dolor sit amet consectetur adipisicing elit. Ab voluptas excepturi quam deleniti nihil? Oditi!



Codixa

Top learning experiences that create more talent in the world.

Product

Overview

Features

Solutions

Tutorials

Pricing

Company

About Us

Careers

Press new

News

Social

Twitter

LinkedIn

GitHub

Dribbble

Legal

Terms


Privacy

Cookies

Contact

© 2024 SiriusEG. All rights reserved.





Admin
admin
alex.johnson@gmail.com

Statics

Teacher Control

Reports

Admin Control

Courses


Settings

Home

Logout

Dashboard

Dashboard Content



Statics

Statics

Teacher Control

Reports

Admin Control


Courses

Settings

Home

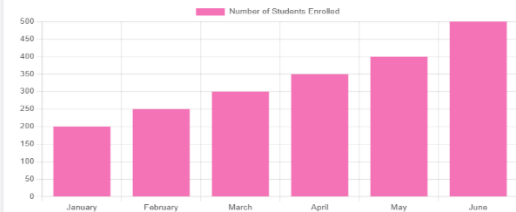
Logout

Hours of Courses Watched




Month	Hours of Courses Watched
January	50
February	75
March	100
April	150
May	200
June	300

Number of Students Enrolled



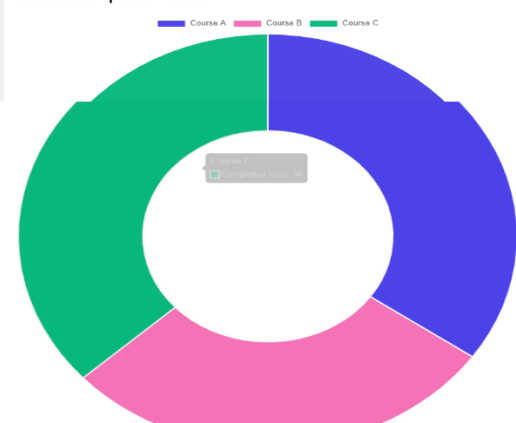
Month	Number of Students Enrolled
January	200
February	250
March	300
April	350
May	400
June	500

Revenue Generated



Month	Revenue Generated
January	1,000
February	2,000
March	3,000
April	4,000
May	4,500
June	5,000

Course Completion Rates



Course	Completion Rate
Course A	40%
Course B	35%
Course C	25%



Welcome to Codixa

Login

Register

Continuous learning through micro-courses or certifications keeps skills relevant, driving talent development in rapidly evolving industries.

Username

Enter your username

Password

Enter your password

[Forgot Password?](#)

Login



Welcome to Codixa

Login

Register

Continuous learning through micro-courses or certifications keeps skills relevant, driving talent development in rapidly evolving industries.

Username

Enter your username

Email

Enter your email

Password

Create password

☐ Remember me

[Have an account?](#)

Register



McAfee | WebAdvisor



Your download's being scanned.
We'll let you know if there's an issue.

