

MEGAMINDS: Personalized Curriculum Recommendation System

Laila Amgad, Nouran Hassan, Roaa Khaled,
Yahia Tamer, Malak Mohamed
Supervised by: Dr. Esraa Afify

December 21, 2024

Table 1: Document Version History

Version	Date	Reason for Change
2.0	21-Dec-2024	SRS Final version's specifications are defined.

GitHub Repository

<https://github.com/Nouran246/MegaMinds-Course-Recommendation-System>

Contents

1	Introduction	4
1.1	Purpose of this document	4
1.2	Scope of this document	4
1.3	Business Context	4
2	Business Plan	5
2.1	Target Audience	5
2.2	Key Benefits	5
2.2.1	Enhanced Student Experience:	5
2.2.2	Improved Instructor Support:	5
2.2.3	Valuable Data for Educational Institutions:	5
3	Business Goal	6
3.1	Vision	6
3.2	Mission	6
3.3	Key Objectives	6
4	Problem Statement for MegaMinds	6
5	Design View Points	7
5.1	Use Case Diagram	7
5.2	Sequence Diagram	8
5.3	Context Diagram	10
5.4	Activity Diagram	11
5.5	Class Diagram	12
6	Similar Systems	12
7	System Description	13
7.1	Problem Statement	13
7.2	System Overview	13
7.2.1	Main Features	13
8	Functional Requirements	14
8.1	Detailed Functional Specification	14
9	Non-Functional Requirements	15
9.1	Reliability	16
9.2	Maintainability	16
9.3	Scalability	16
9.4	Security	16
9.5	Usability	16
9.6	Assurance	16
10	Data Design	17
10.1	Database	17
11	Human Interface Design	18

12 Requirement Matrix	28
13 Test Cases (Unit Testing)	29
14 Project Plan	31
14.1 Team Members:	31
14.2 Task Distribution Table	31
15 Appendices	31
15.1 Definitions, Acronyms, Abbreviations	31

Abstract

The increasing emphasis on personalized learning and student-centered methodologies has underscored the necessity of effectively guiding students through their academic experiences. Conventional educational frameworks frequently overlook the diverse learning preferences, career goals, and personal interests of individual students. This document outlines the software specifications for the Personalized Curriculum Recommendation System, a platform that utilizes machine learning techniques to deliver customized curriculum suggestions tailored to the distinct profiles of students. By examining data related to students' interests, strengths, and objectives, this system seeks to enrich educational experiences and assist students in making well-informed decisions regarding their academic trajectories. With features such as real-time recommendations, resource integration, progress monitoring, and feedback systems, the Personalized Curriculum Recommendation System aims to render education more pertinent and impactful for each learner, ultimately fostering enhanced satisfaction and academic achievement.

1 Introduction

1.1 Purpose of this document

The aim of this Software Requirements Specification (SRS) document is to outline the essential features, objectives, and aims of the Personalized Curriculum Recommendation System. This document serves as a detailed resource to support the processes of development, implementation, and deployment. The main audience for this SRS comprises the development team, stakeholders, project managers, and educational institutions that have an interest in the system.

1.2 Scope of this document

This Software Requirements Specification (SRS) delineates the functionalities, requirements, and design elements of the Personalized Curriculum Recommendation System, referred to as "MEGAMINDS." It provides comprehensive descriptions of both functional and non-functional requirements, foundational user interface components, data architecture, and the primary classes along with their interrelations. This document is intended to act as a guiding reference throughout the development process, facilitating a shared understanding of the system's capabilities and the desired results.

1.3 Business Context

As the landscape of education increasingly embraces personalized and student-focused methodologies, there is an escalating demand for technological solutions that tailor educational trajectories to the unique interests and professional ambitions of each student. The Personalized Curriculum Recommendation System aims to address this need by improving students' learning experiences and facilitating informed decision-making through the provision of tailored curriculum recommendations. This innovative approach not only aids educational institutions but also empowers students, thereby rendering their academic pursuits more efficient and closely aligned with their prospective career objectives.

2 Business Plan

The Personalized Curriculum Recommendation System aims to support students, educators, and educational administrators by offering a unified platform that delivers tailored curriculum advice and learning materials. This business plan delineates the intended audience, advantages, and criteria for evaluating success.

2.1 Target Audience

The target audience for the system includes:

The target audience for the system includes:

- **Students:** Undergraduate and graduate students primarily seek assistance in choosing courses, engaging in activities, and accessing resources that correspond with their unique learning styles and professional aspirations.
- **Instructors:** Faculty members seeking to gain a deeper insight into the needs of their students and provide tailored support and resources to facilitate their academic development.
- **Educational Administrators:** University or school officials tasked with the development of curricula, analysis of student engagement metrics, and enhancement of educational programs.

2.2 Key Benefits

2.2.1 Enhanced Student Experience:

- **Personalized Guidance:** Each student receives a tailored curriculum that matches their personal goals, learning preferences, and career aspirations.
- **Clearer Academic Pathways:** Recommendations allow students to visualize a structured, purpose-driven academic journey, which boosts confidence and satisfaction.

2.2.2 Improved Instructor Support:

- **Insight into Student Needs:** Instructors gain insights into student engagement and interests, allowing them to adapt their teaching methods.
- **Opportunity for Mentorship:** By understanding student profiles, instructors can offer more personalized guidance to help students make informed decisions.

2.2.3 Valuable Data for Educational Institutions:

- **Analytics on Curriculum Effectiveness:** Administrators gain data-driven insights into which courses and resources students are engaging with the most, helping optimize curriculum offerings.
- **Enhanced Program Development:** Feedback and performance data provide a foundation for curriculum improvement and strategic decision-making in course offerings.

3 Business Goal

The Personalized Curriculum Recommendation System seeks to enhance the educational journey for students by offering tailored guidance that corresponds with their unique interests, learning preferences, and professional goals. This system is intended to meet the demand for individualized educational trajectories within a progressively varied and evolving academic environment. The objectives of the system from a business perspective are delineated as follows:

3.1 Vision

To develop a sophisticated educational platform that empowers students to make enlightened decisions regarding their academic and professional trajectories. This system aspires to a future where tailored learning experiences inspire students to realize their utmost potential.

3.2 Mission

The objective is to create a comprehensive and intuitive recommendation system that leverages sophisticated algorithms and data analytics to connect students with courses, resources, and activities that correspond to their individual academic and career aspirations. This initiative aims to assist students in constructing a significant and productive educational journey.

3.3 Key Objectives

1. **Increase Student Engagement:** Offer tailored guidance to students, ensuring they stay inspired and actively involved in their educational journeys.
2. **Facilitate Career-Driven Learning:** Harmonize the suggested curricula with the long-term aspirations of students, ensuring they are endowed with the essential skills and knowledge for their future careers.
3. **Enhance Decision-Making in Education:** Assist students in navigating their educational journey by providing them with tailored, data-informed choices regarding courses and extracurricular activities.
4. **Deliver Continuous Improvement:** 1. Leverage insights from student feedback and performance metrics to meticulously refine the recommendation algorithms, thereby elevating the system's accuracy as time progresses.

4 Problem Statement for MegaMinds

In the contemporary digital environment, there is a growing demand among individuals to acquire new skills and knowledge in order to remain competitive and adapt to the evolving requirements of various industries. However, the extensive selection of online courses offered across multiple platforms can often be overwhelming, complicating the process of identifying the most pertinent and high-quality educational resources that align with individual goals, skill levels, and learning preferences. This challenge is particularly

significant for students and lifelong learners who seek a more streamlined and personalized educational experience but often lack the necessary guidance to effectively navigate the available options.

In the absence of a tailored recommendation system, learners may find themselves investing excessive time in searching for suitable courses, which can result in frustration, disengagement, and lost opportunities for learning. Existing solutions that aggregate courses or offer recommendations frequently do not adequately address the specific learning styles, time limitations, and interests of users.

5 Design View Points

5.1 Use Case Diagram

The below use case diagram in figure below demonstrates the system functional requirements. The system is composed of two user types: student and admin.

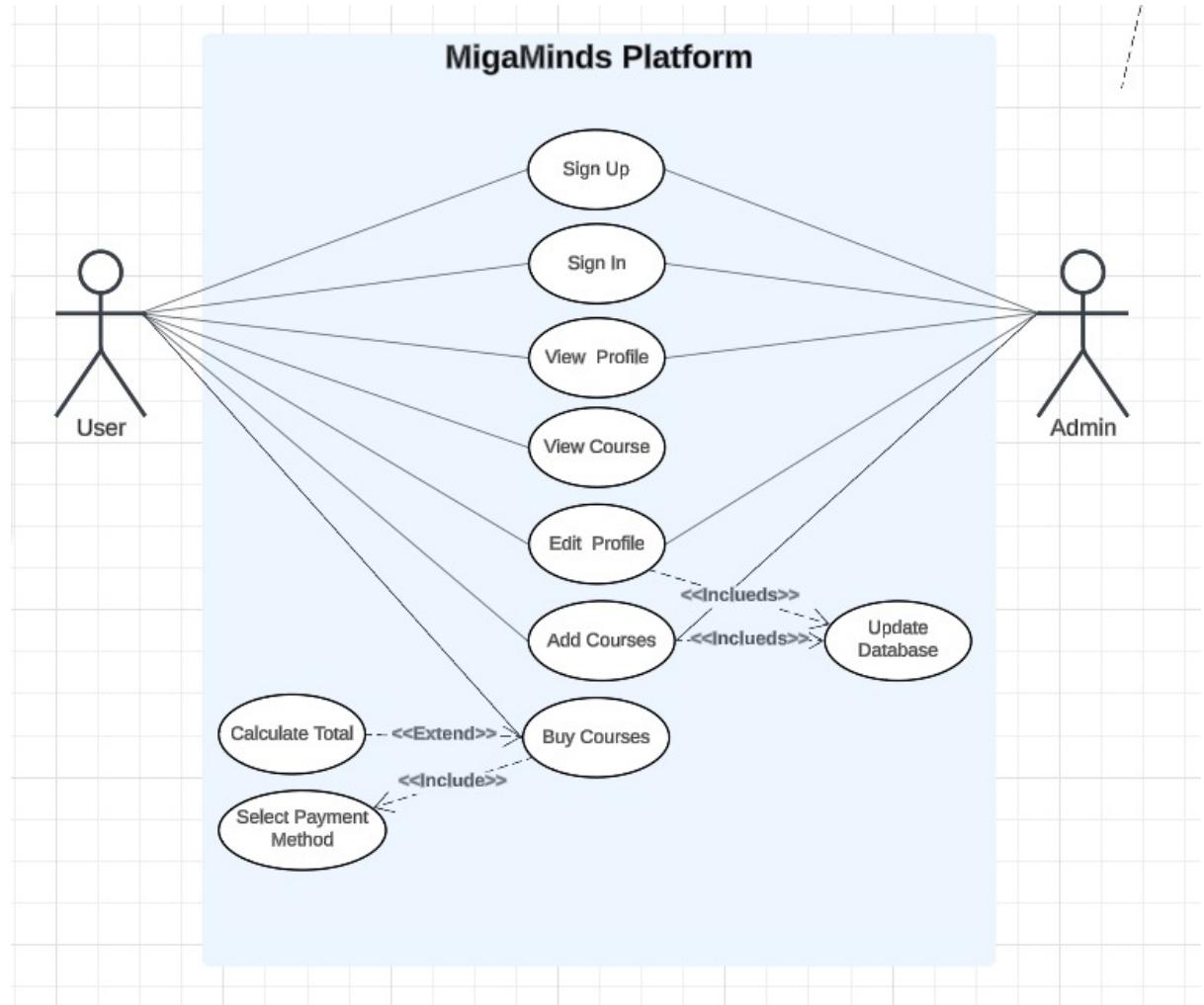
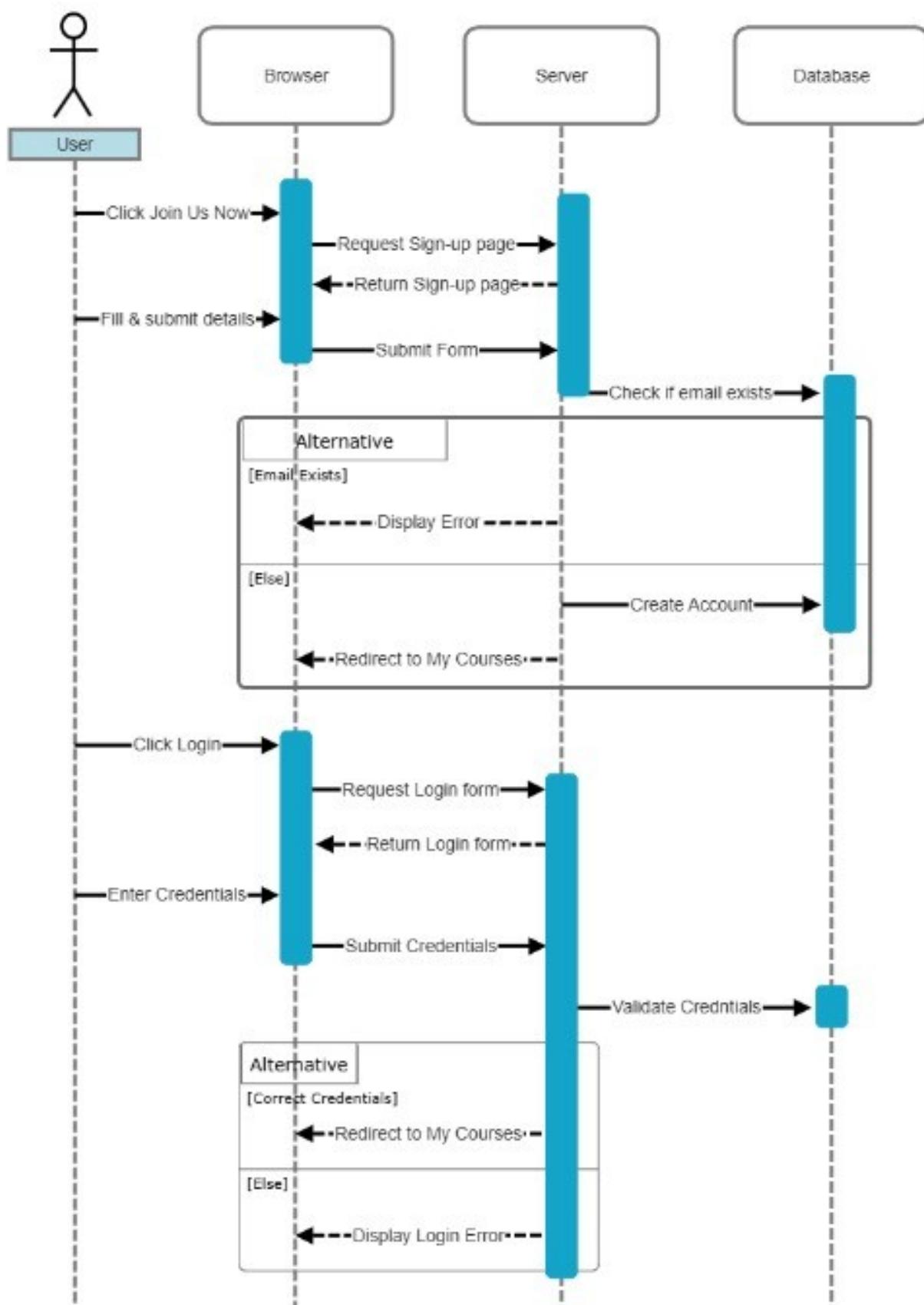


Figure 1: Use-case diagrams

5.2 Sequence Diagram



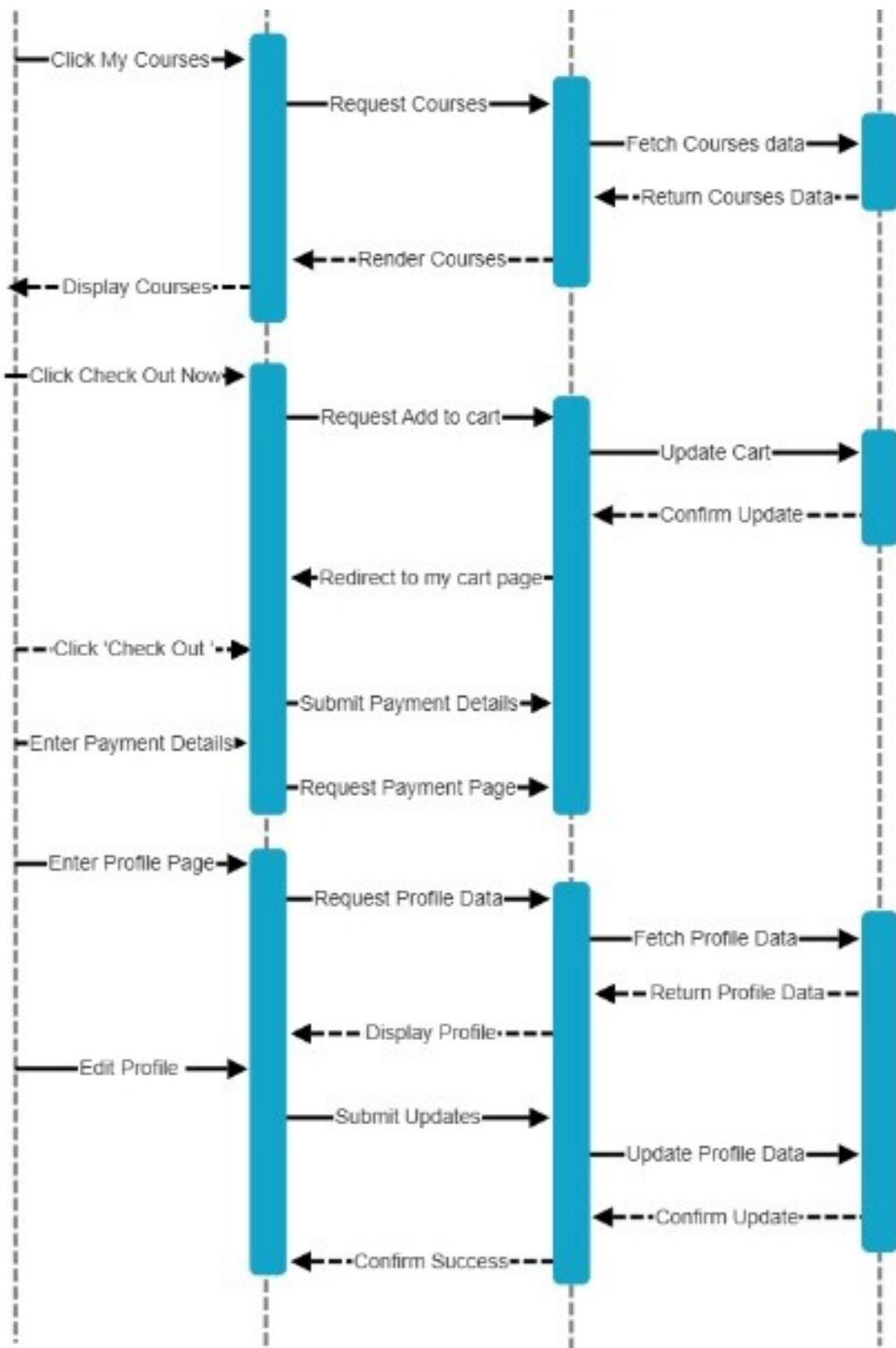


Figure 2: Sequence diagram

5.3 Context Diagram

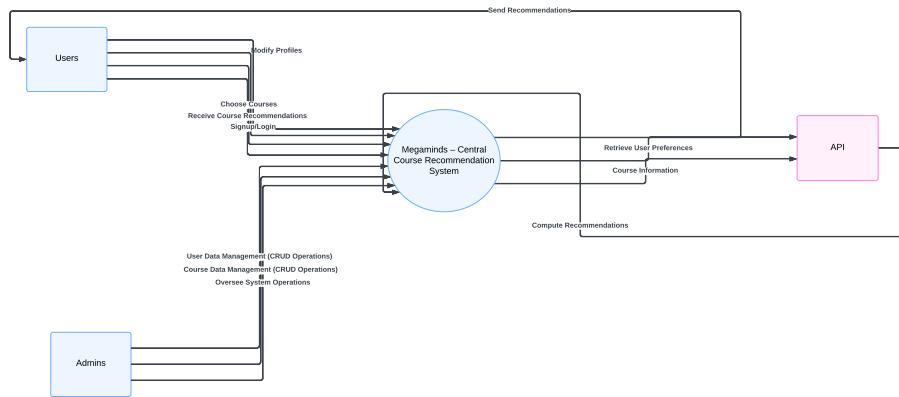


Figure 3: Context diagram

5.4 Activity Diagram

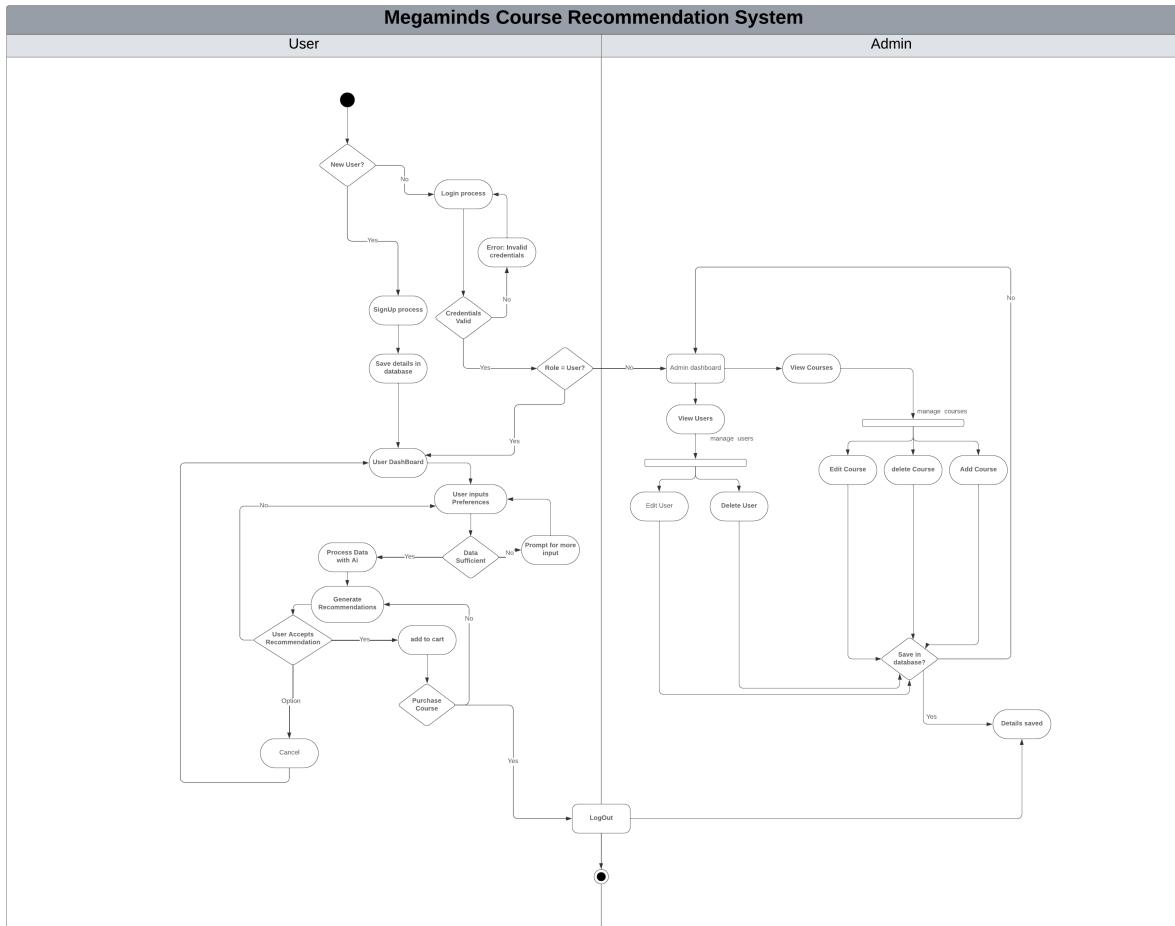


Figure 4: Activity diagram

5.5 Class Diagram

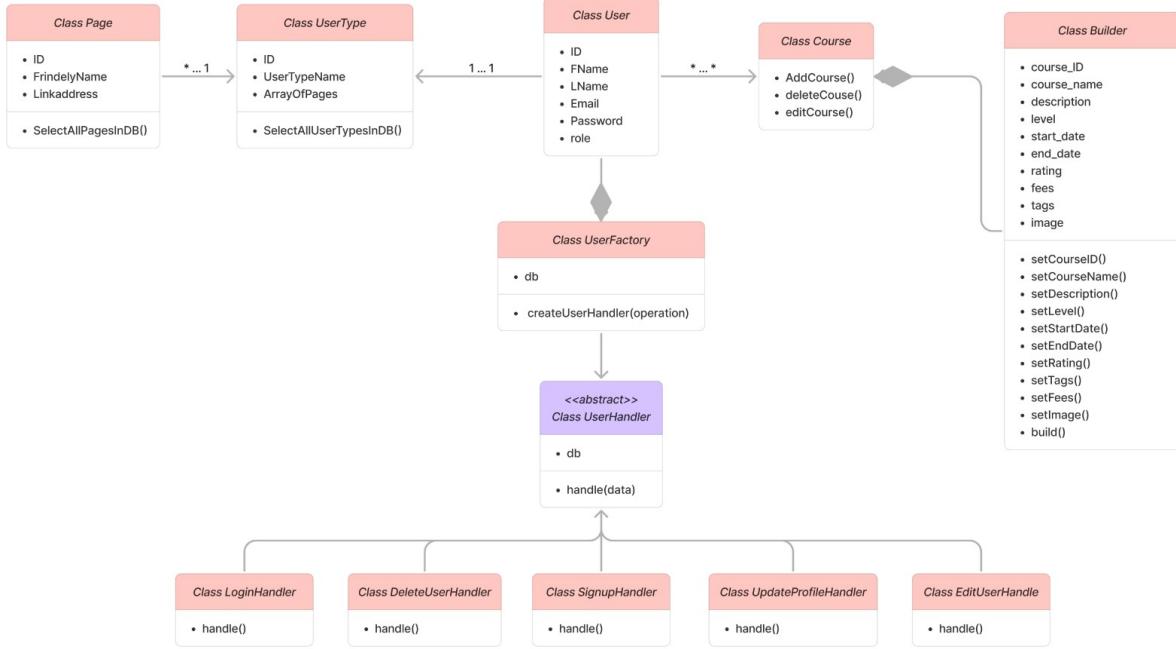


Figure 5: UML OOP Design

6 Similar Systems

The Personalized Curriculum Recommendation System can be likened to various established platforms that deliver customized educational experiences. Notable examples of such systems include:

- **Coursera:** This platform provides course recommendations that are tailored to individual user interests and their historical learning activities [1].
- **edX:** It employs learner profiles to recommend pertinent courses and educational programs [2].
- **Khan Academy:** This system modifies learning trajectories in response to student progress and performance metrics [3].
- **Udacity:** It emphasizes career-focused courses and offers personalized support for skill enhancement [4].
- **Northeastern University Tips for Online Classes:** A resource offering tips for effective learning in online courses [5].
- **Harvard Gazette Study on Active Learning:** A study highlighting the benefits of active learning strategies in classrooms [6].
- **The Raptors Nest Tips for Challenging Courses:** A blog post on how to overcome challenges in difficult courses [7].

- **Faculty Focus Course Planning:** Advice for planning courses that you've never taught before [8].
- **Quality Matters Resources for Improving Courses:** Quality Matters provides a list of ways to improve course design [9].
- **Inside Higher Ed on Improving Online Courses:** Tips for enhancing online courses in higher education [10].

7 System Description

7.1 Problem Statement

In today's fast-evolving digital landscape, individuals increasingly seek to acquire new skills and knowledge to stay competitive and adapt to changing industry demands. However, the vast array of online courses available across various platforms often overwhelms users, making it challenging to find the most relevant, high-quality learning resources tailored to their unique goals, skill levels, and learning preferences. This issue is particularly pronounced among students and lifelong learners who desire a streamlined and personalized learning experience but lack the guidance to navigate available options effectively.

Without a personalized recommendation system, learners are at risk of spending excessive time searching for appropriate courses, potentially leading to frustration, disengagement, and missed learning opportunities. Current solutions that aggregate courses or provide recommendations often fail to cater to users' specific learning styles, time constraints, and areas of interest.

MegaMinds addresses this gap by using AI-driven recommendations to provide users with curated course suggestions tailored to their individual preferences and educational goals. By leveraging user-provided survey data, MegaMinds aims to simplify the course selection process, enhance engagement, and support users in achieving their learning objectives efficiently and effectively.

7.2 System Overview

MegaMinds is a platform that focuses on suggesting courses tailored to users through the use of intelligence technology to create a more personalized learning journey, for individuals seeking to expand their skills and knowledge base.

7.2.1 Main Features

1. **User Survey for Personalized Recommendations:** This feature tailors course recommendations to match the skill levels and preferences of users:
 - **Skill Level:** Courses are aligned with beginner, intermediate, or advanced learners based on their indicated skill level.
 - **Topics of Interest:** Users can select topics or areas of interest using the "Topics of Interest" feature to personalize their learning experience.

- **Learning Style:** Course recommendations adapt to individual learning preferences, offering options such as video-based lessons for visual learners, hands-on activities for practical learners, and text-based resources for those who prefer reading.
 - **Time Availability:** Course suggestions take into account the user's available study time to ensure alignment with their schedule.
2. **User Profile and Information Storage:** Each user has a dedicated profile where data from surveys and personal preferences are securely stored. This profile acts as a central hub for users to review suggested courses and make any necessary adjustments to their preferences.
 3. **Admin-Managed Course Content:** Administrators can add and manage recommended courses via an intuitive interface, allowing them to continuously update and refresh course selections for quality control without relying on external data inputs.
 4. **Frontend and Backend Architecture:**
 - **Frontend:** The user interface is developed using HTML, CSS, JavaScript, and the Bootstrap framework, ensuring a responsive and user-friendly experience.
 - **Backend:** The backend is implemented in PHP with an SQL database, efficiently handling user profiles, course information, and preference data.
 5. **AI-Assisted Recommendations:** MegaMinds currently uses rule-based recommendations based on survey criteria. Future updates may incorporate advanced algorithms to enhance personalization.

8 Functional Requirements

8.1 Detailed Functional Specification

1. The **user** shall be able to create an account with their information upon accessing the system.
2. The **user** shall be able to log in using their credentials and manage their session within the system.
3. The **user** shall be able to view a list of available courses with descriptions and other relevant details.
4. The **user** shall be able to receive course recommendations based on their interests, preferences, or course history.
5. The **user** shall be able to search for courses by category, difficulty level, or keywords to explore additional options.
6. The **user** shall be able to bookmark courses for later reference.

7. The **user** shall be able to view a history of completed or previously enrolled courses.
8. The **user** shall be able to add, update, or delete personal information in their account.
9. The **user** shall be able to view detailed course recommendations, including prerequisites and related courses.
10. The **admin** shall be able to create, edit, and delete courses within the system.
11. The **admin** shall be able to view a list of all users and manage user accounts, including adding, editing, and deleting users.
12. The **system** shall validate all user input on the client side to ensure proper data entry and provide immediate feedback.
13. The **system** shall manage session timeouts and user authentication to maintain secure access control.
14. The **system** shall restrict access to administrative functionalities based on user role and permissions.
15. The **system** shall display recommendations based on predefined criteria, such as user preferences, course prerequisites, or popularity.
16. The **system** shall implement frontend and backend validations for both sign-in and sign-up processes.
17. The **system** shall store user session data securely, ensuring data integrity and confidentiality.
18. The **system** shall support CRUD (Create, Read, Update, Delete) operations for managing users and courses within the system. (CR20)

9 Non-Functional Requirements

The non-functional elements of the Course Recommendation System specify essential factors, including performance, security, usability, and scalability. These elements ensure the system is maintainable and can manage growing demands effectively. They aim to provide users with a secure and satisfying experience by offering a system that not only meets user expectations but also exceeds them in terms of reliability, scalability, and performance.

9.1 Reliability

The Course Recommendation System is designed to be highly reliable, with a focus on ensuring availability and avoiding single points of failure. Regular data backups are taken to prevent any data loss, thereby enhancing users' confidence in the system's durability and dependability.

9.2 Maintainability

The system is built to be easily maintainable, with modular and well-documented code that facilitates updates and improvements. This organizational structure simplifies debugging and supports the addition of new features, helping to ensure that future enhancements can be made with minimal disruption.

9.3 Scalability

The system is highly scalable, designed to manage an increasing load of users and data without compromising responsiveness. As the user base and course data grow, the system can effectively scale to maintain optimal performance, enhancing the user experience.

9.4 Security

Security is paramount in the system design. Passwords are securely stored using hash, and system checks verify data integrity. Session management and automatic interlocks further enhance data protection, ensuring user data remains safe and secure.

9.5 Usability

The Course Recommendation System prioritizes usability, offering a user-friendly interface that guides users clearly. Meaningful error messages provide feedback on actions, allowing users to address issues effectively and enhancing the overall user experience.

9.6 Assurance

The system is structured to deliver high performance with short response times, minimizing wait times for loading pages and executing CRUD operations. Effective session management ensures smooth interactions, providing users with a reliable and efficient application.

10 Data Design

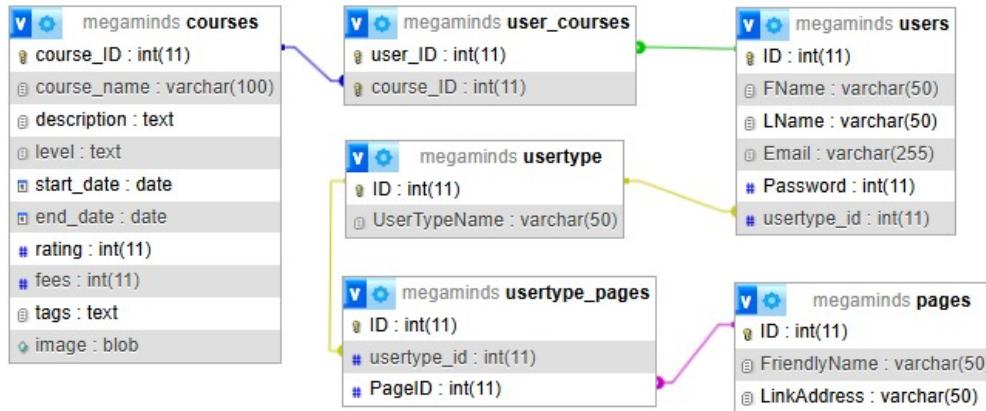


Figure 6: Screenshot of Database from XAMPP

10.1 Database

The database for the Megaminds website, as shown in Figure 2, includes tables for users, courses, pages, and user types.

- **User table:** Defines users with fields such as ID, FName, LName, Email, Password, and role, with identifiers that distinguish between the type of usage and individual users.
- **Courses table:** Defines courses with fields like course_ID, course_name, description, level, start_date, end_date, rating, fees, tags and image.
- **Pages table:** Defines pages with fields like ID, friendlyName, and LinkAddress, listing pages and their corresponding paths with a unique identifier and a name.
- **User Type table:** Defines user roles with fields such as ID and userTypeName, identifying whether a user is an admin or a regular user.
- **UserType_pages:** A join table between the pages table and user type pages to manage access permissions.
- **User_Courses table:** A join table between the users table and courses table to manage the enrollment of users in courses. It includes fields such as user_ID and course_ID, referencing the users and courses tables, respectively.
- **Menu table:** Defines the navigation menu for the website with fields like ID, name, and href, listing menu items and their corresponding hyperlinks.

11 Human Interface Design

1. Landing Page

The landing page provides a welcoming introduction with options to sign in or sign up and navigate the system's features.

2. Sign In and Sign Up

Sign In: Users log in with credentials and can recover passwords if needed.

Sign Up: New users create accounts with validated input fields.

3. Dashboard

After login, users access a dashboard featuring available courses, a chatbot for recommendations, and profile settings.

4. Available Courses and Chatbot

Users view course listings with details. An AI chatbot assists in recommending suitable courses.

5. Course Actions

Details: In-depth course information.

Add to Cart: Users add courses to their cart and proceed to checkout.

6. Payment Process

Users securely enter payment details to purchase courses, receiving a confirmation upon success.

7. My Courses

Purchased courses are listed here, allowing users to start learning.

8. Profile Management

Users can view, edit, or delete their profiles.

9. Admin Panel

Manage Users: Admins perform CRUD operations on user accounts.

Manage Courses: Admins manage course listings.

Permissions: Admins control access levels for users and administrators.

10. Permissions

User: Access to basic functionalities like course enrollment and profile management.

Admin: Elevated permissions for user, course, and system management.

Website Photos



Figure 7: Landing Page



Figure 8: Landing Page

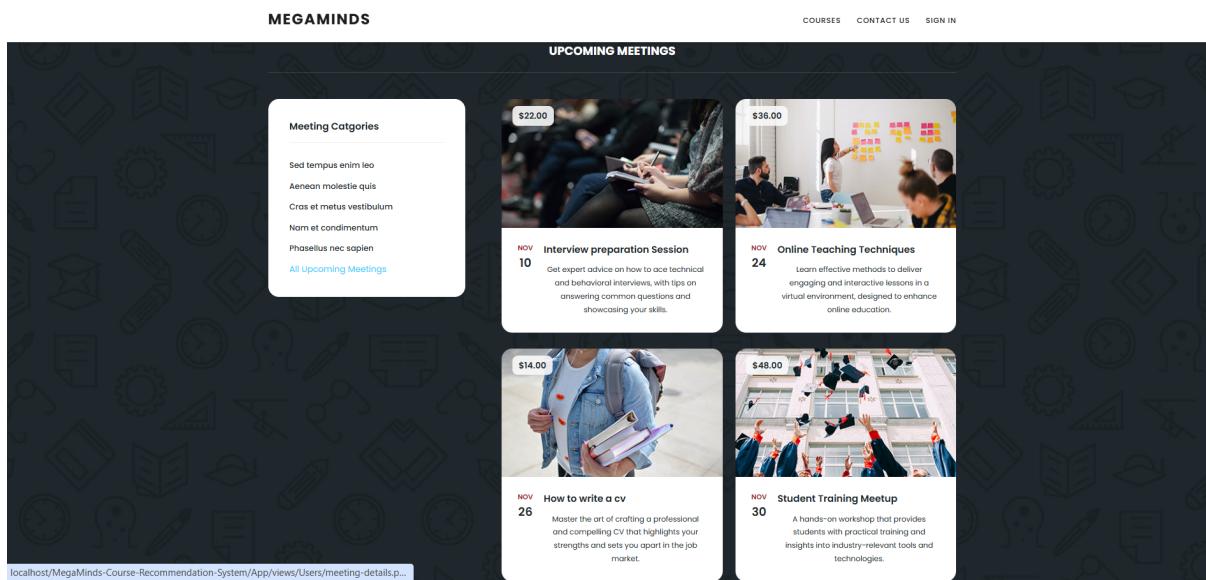


Figure 9: Landing Page

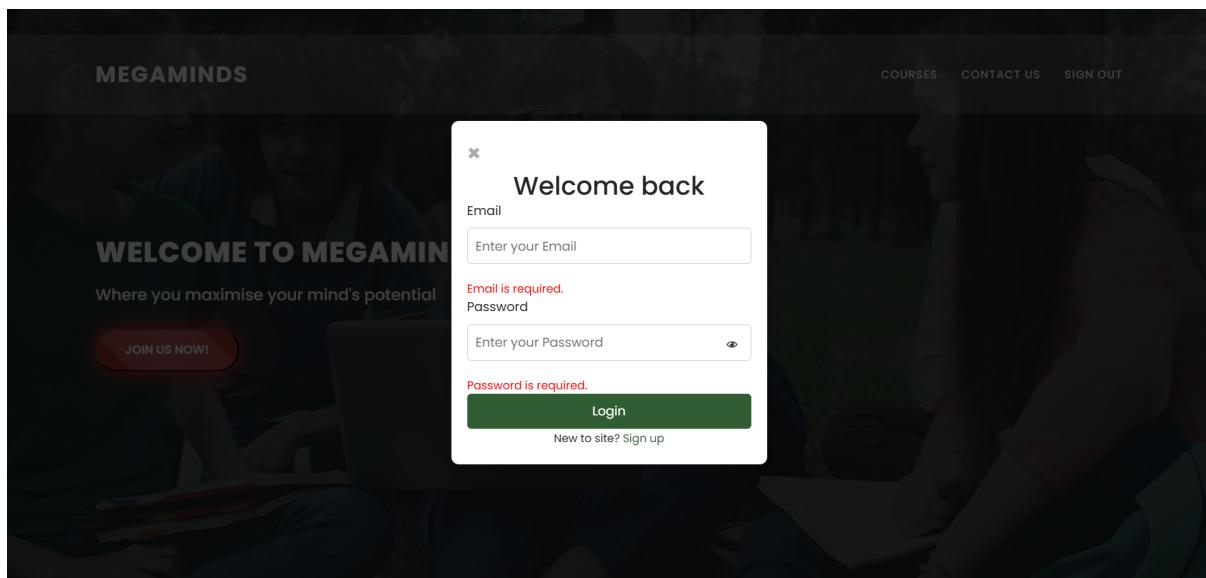


Figure 10: Sign In Page

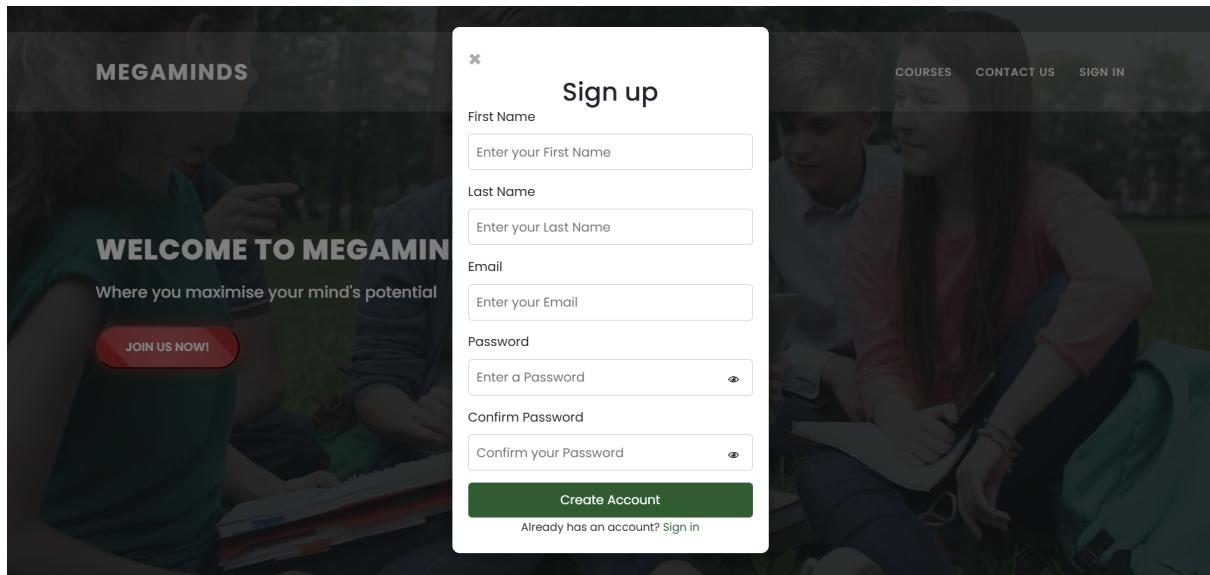


Figure 11: Sign Up Page

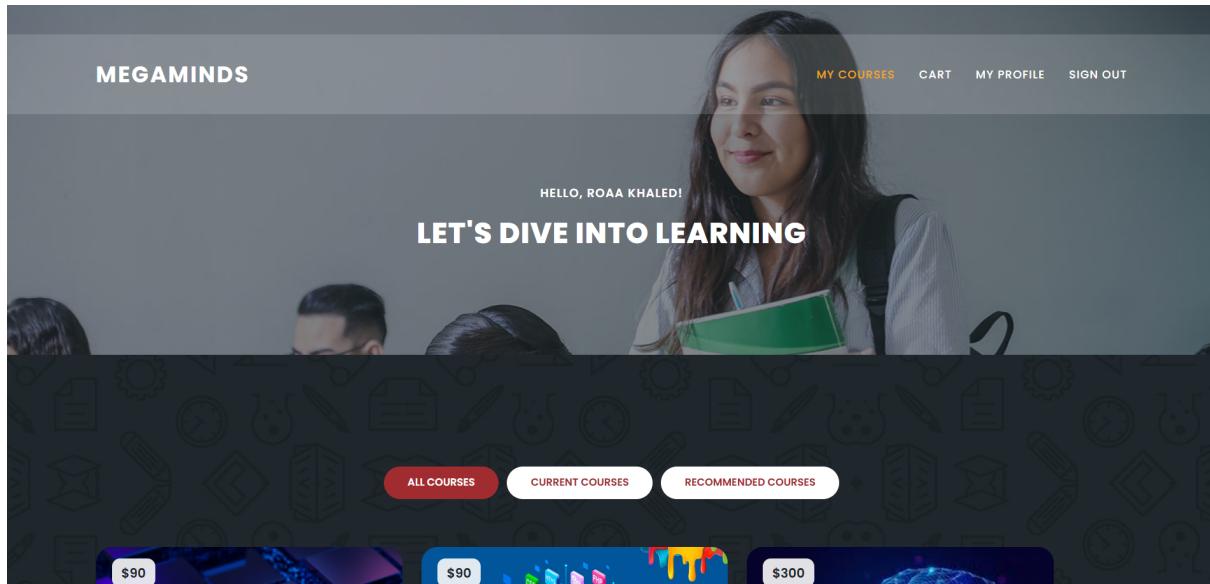


Figure 12: My courses

Chat



googles



hi



googles Bot

Who would you like to chat with?

Human

Advanced AI

What is your level in programming?

Beginner

Intermediate

Advanced

What major do you prefer to learn

Software Engineering

Cyber Security

Would you like to learn

Web development

Mobile development

We recommend you start OOP Course

🔔 Click to set your email to get notifications.

Compose your message...



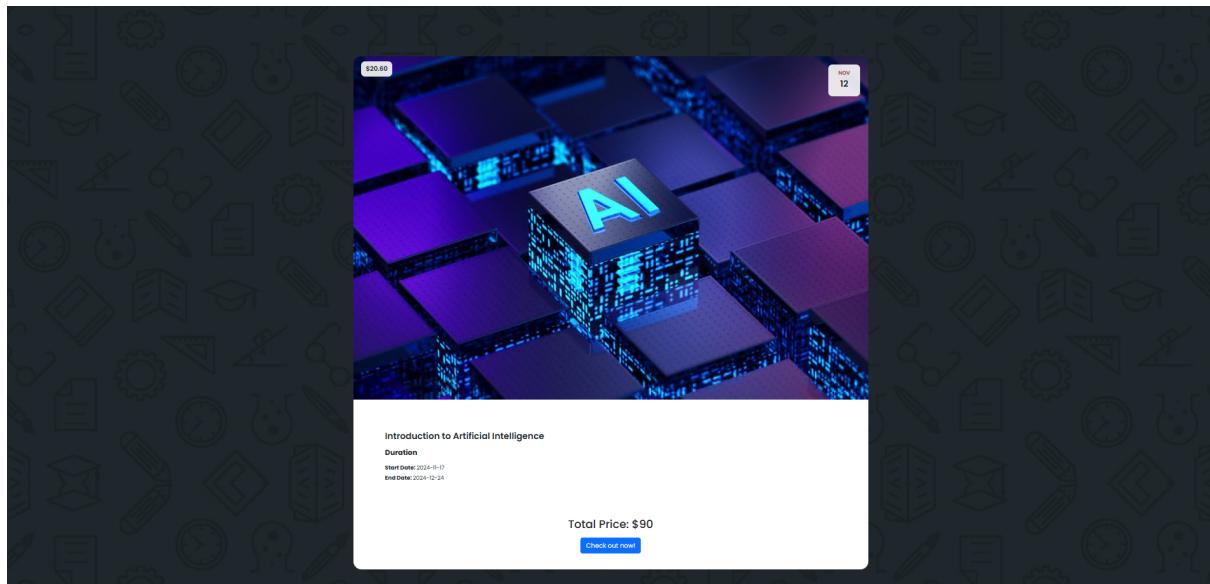


Figure 14: cart

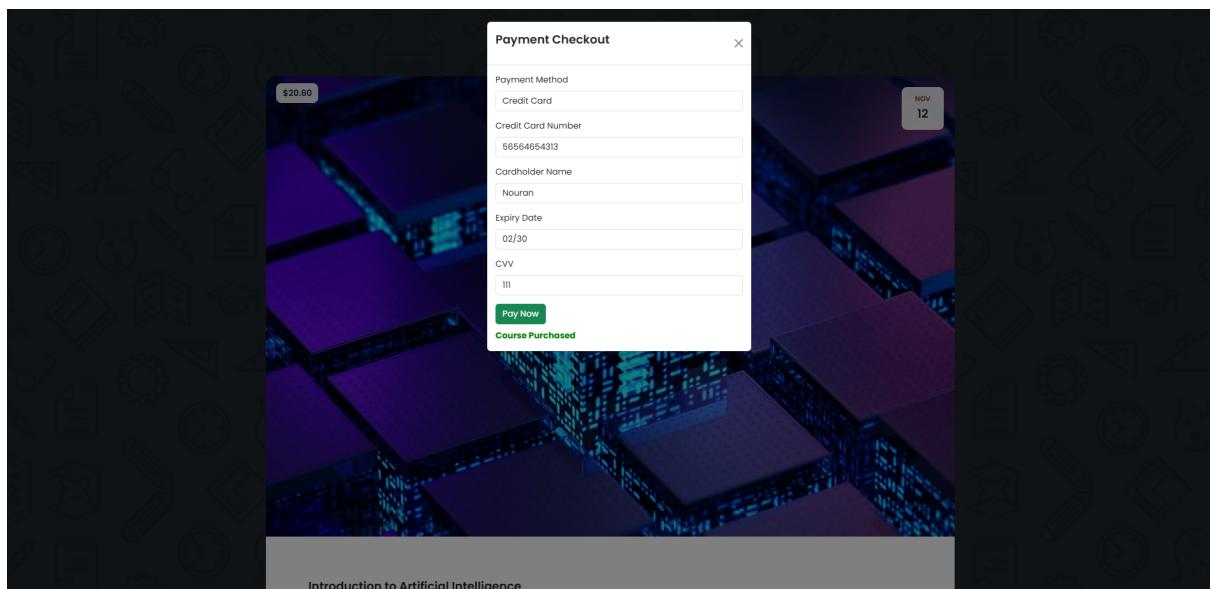


Figure 15: Cart Payment

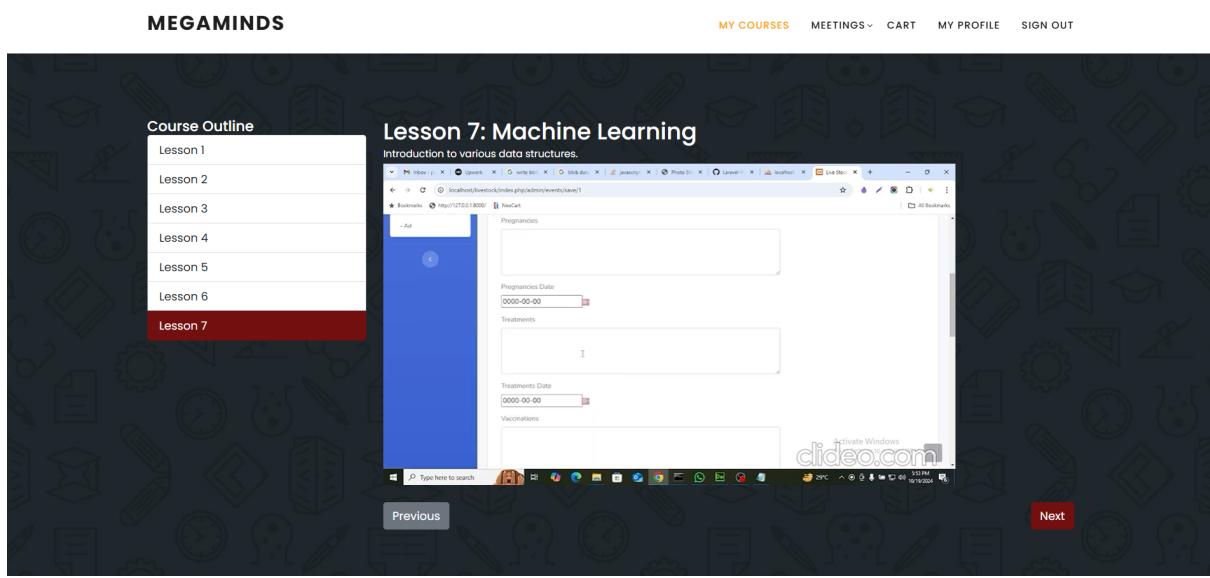


Figure 16: Course

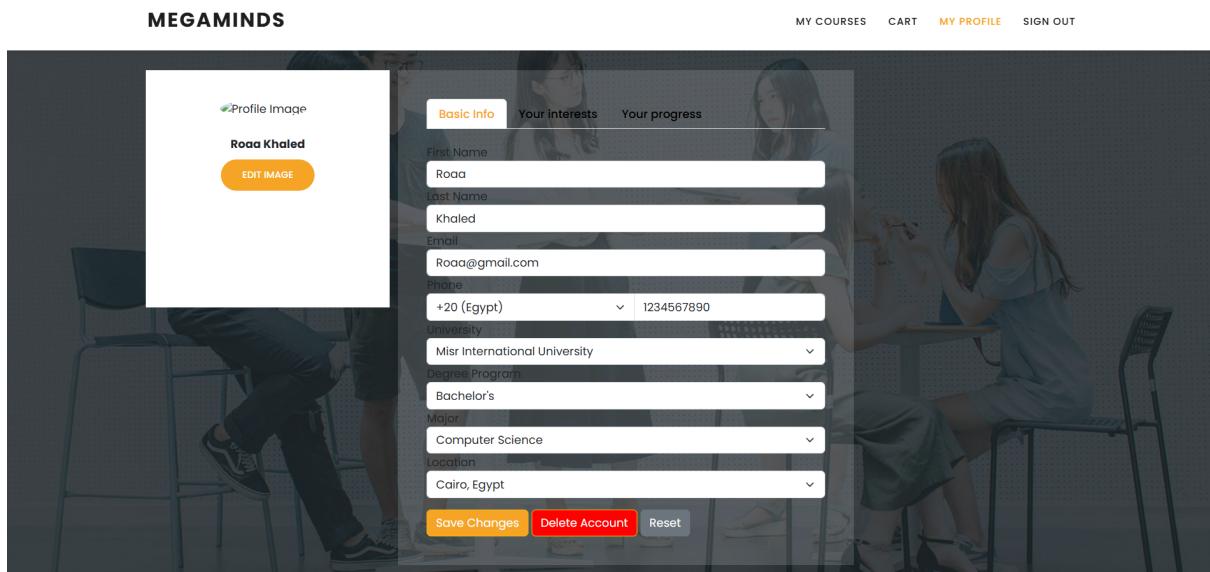


Figure 17: Profile Management

MegaMinds Admin

Manage Courses												Add New Course
Course ID	Course Name	Description	level	Start Date	End Date	Rating	Fees	Tags	Image	Operation		
2	Introduction to Artificial Intelligence	This course provides an overview of Artificial Intelligence (AI), including its history, principles, and applications. Topics include problem-solving, search algorithms, knowledge representation, reasoning, and basic machine learning techniques. Students will explore real-world applications like robotics, natural language processing, and AI ethics.	beginner	2024-11-17	2024-12-24	0	\$90	AI		Edit Delete		
3	Full Stack Developer	This course covers the complete web development process, including both front-end and back-end technologies. Students will learn HTML, CSS, JavaScript, React (or Angular), Node.js, Express, and databases like MongoDB or SQL. The course emphasizes creating fully functional, scalable web applications and understanding the deployment process.	advanced	2024-12-01	2024-12-27	0	\$90	SWE		Edit Delete		
4	Deep Learning	Deep Learning delves into advanced machine learning techniques inspired by the structure and function of the human brain. Topics include neural networks, convolutional neural networks (CNNs), recurrent neural networks (RNNs), and frameworks like TensorFlow or PyTorch. Students will apply these methods to tasks like image recognition, natural language processing, and autonomous systems.	advanced	2024-09-01	2024-12-31	0	\$300	AI		Edit Delete		
7	Networks and Security	This course introduces the fundamentals of computer networks and cybersecurity. Topics include network protocols, security threats, and mitigation strategies. Students will learn how to design secure systems and mitigate common vulnerabilities.	intermediate	2024-	2024-	0	\$70	INS		Edit		

[Logout](#)

Figure 18: Courses Dashboard

MegaMinds Admin

Manage Courses												Add New Course
Course ID	Course Name	Description	Level	Start Date	End Date	Rating	Fees	Tags	Image	Operation		
2	Introduction to Artificial Intelligence	This course provides an overview of Artificial Intelligence (AI), including its history, principles, and applications. Topics include problem-solving, search algorithms, knowledge representation, reasoning, and basic machine learning techniques. Students will explore real-world applications like robotics, natural language processing, and AI ethics.	Select a level	mm/dd/yyyy	mm/dd/yyyy	0	\$90	AI		Edit Delete		
3	Full Stack Developer	This course covers the complete web development process, including both front-end and back-end technologies. Students will learn HTML, CSS, JavaScript, React (or Angular), Node.js, Express, and databases like MongoDB or SQL. The course emphasizes creating fully functional, scalable web applications and understanding the deployment process.	Select a level	mm/dd/yyyy	mm/dd/yyyy	0	\$90	SWE		Edit Delete		
4	Deep Learning	Deep Learning delves into advanced machine learning techniques inspired by the structure and function of the human brain. Topics include neural networks, convolutional neural networks (CNNs), recurrent neural networks (RNNs), and frameworks like TensorFlow or PyTorch. Students will apply these methods to tasks like image recognition, natural language processing, and autonomous systems.	Select a level	mm/dd/yyyy	mm/dd/yyyy	0	\$300	AI		Edit Delete		
7	Networks and Security	This course introduces the fundamentals of computer networks and cybersecurity. Topics include network protocols, security threats, and mitigation strategies. Students will learn how to design secure systems and mitigate common vulnerabilities.	Select a level	mm/dd/yyyy	mm/dd/yyyy	0	\$70	INS		Edit Delete		

[Logout](#)

Add New Course

Figure 19: Add New Courses

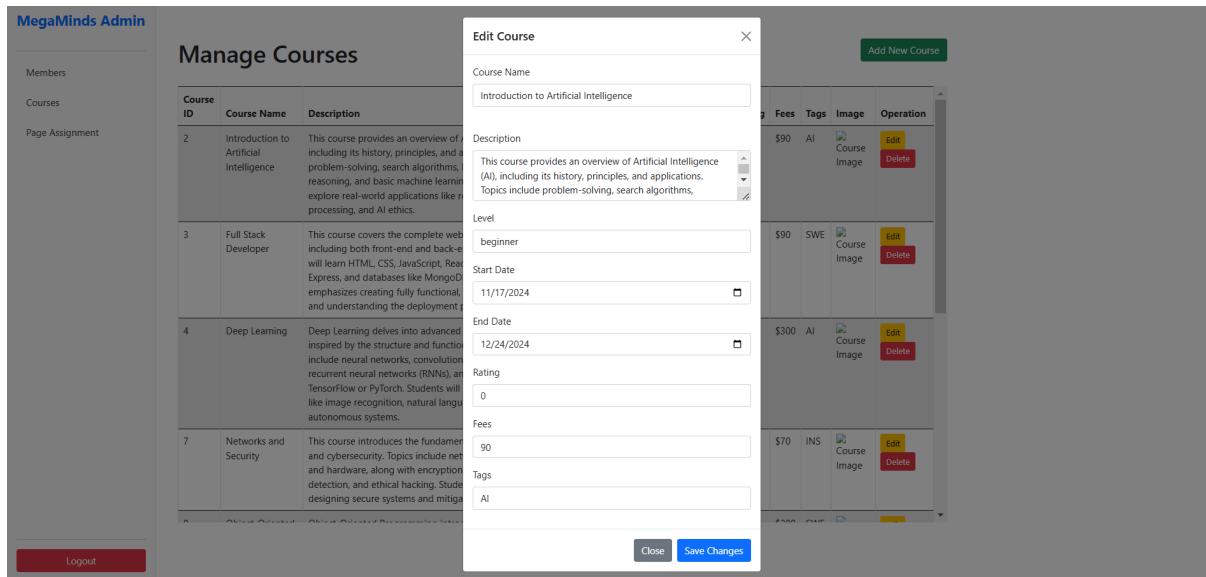


Figure 20: Edit Courses

First Name	Last Name	Email	Actions
Nouran	Hassan	Nouran@gmail.com	<button>Edit</button> <button>Delete</button>
Roaa	Khaled	Roaa@gmail.com	<button>Edit</button> <button>Delete</button>
Jana	Hassan	haha@gmail.com	<button>Edit</button> <button>Delete</button>
Salma	Ahmed	salma@gmail.com	<button>Edit</button> <button>Delete</button>
Yahia	Tamer	yahia@gmail.com	<button>Edit</button> <button>Delete</button>
Mayar	Khaled	mayar@gmail.com	<button>Edit</button> <button>Delete</button>
Laila	Amgad	laila2201298@mieugypt.edu.eg	<button>Edit</button> <button>Delete</button>

Figure 21: User Dashboard

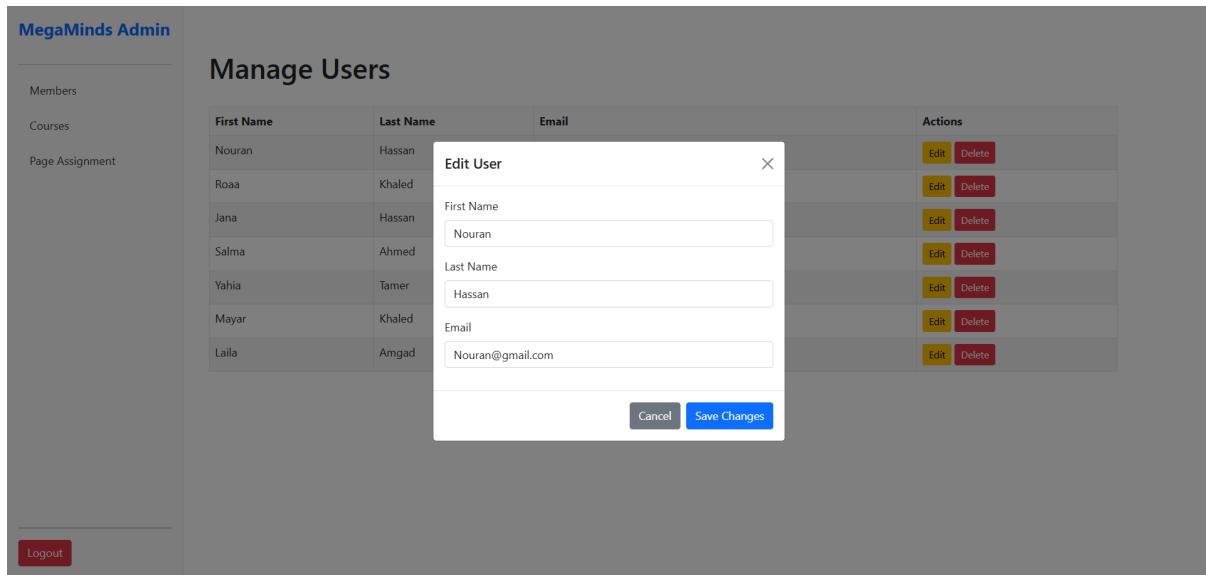


Figure 22: Edit User

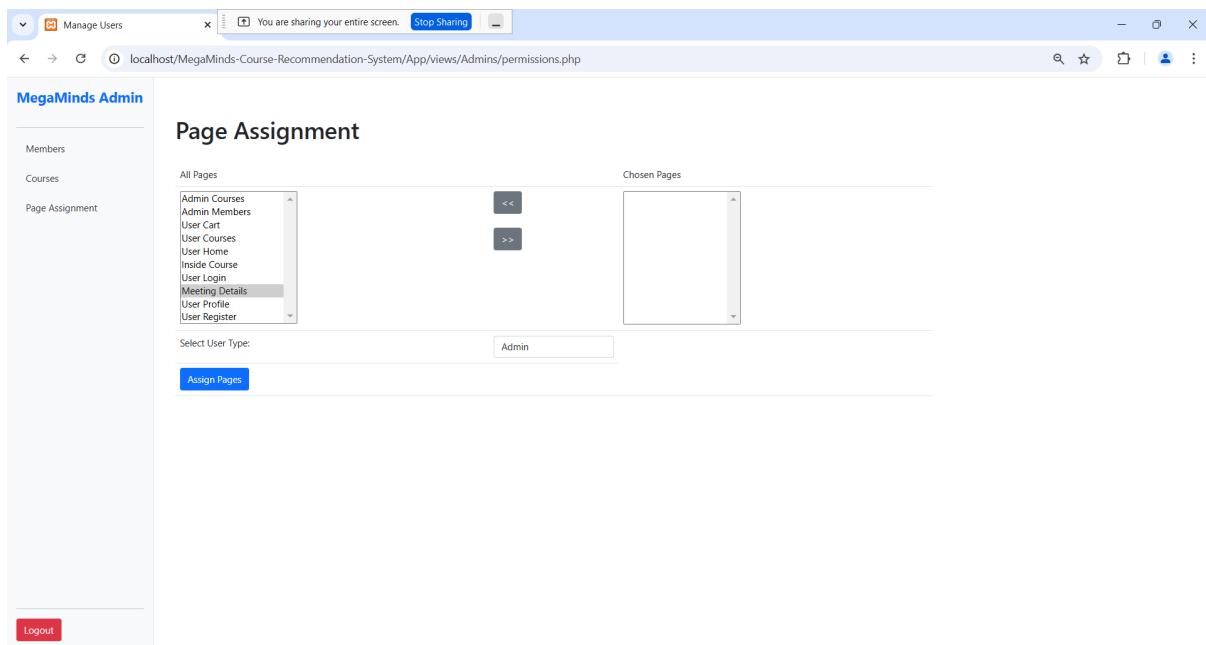


Figure 23: Permissions

12 Requirement Matrix

Req ID	Req Description	Source	Usage	Test Cases	Status
REQ-001	The user shall be able to create an account to sign in with	UserClass (Functional)	User Sign up	TC001	Done
REQ-002	The user shall be able to log in using their credentials and manage their session within the system and end their session by logging out	UserClass (Functional)	User Login Flow and log out	TC002	Done
REQ-003	The user shall be able to view a list of available courses with descriptions and other relevant details.	CourseClass (Functional)	Course Listing	TC003	Done
REQ-004	User can use Chat bot to discuss with AI to make it recommend the perfect course for the user	ChatBot	Recommendation	TC004	Done
REQ-005	The user shall be able to add, update, or delete personal information in their account.	UserClass (Functional)	Profile Management	TC005	Done
REQ-006	The admin shall be able to create, edit, and delete courses within the system.	CourseClass (Functional)	Admin Course Management	TC006	In Progress
REQ-007	The admin shall be able to view a list of all users and manage user accounts, including adding, editing, and deleting users.	UserClass (Functional)	Admin User Management	TC007	Done
REQ-008	The system shall restrict access to administrative functionalities based on user role and permissions.	Functional Requirement	Access Control	TC008	Done

Req ID	Req Description	Source	Usage	Test Cases	Status
REQ-009	The system shall implement frontend and backend validations for both sign-in and sign-up processes.	Functional Requirement	Sign-In/Sign-Up Validation	TC009	Done
REQ-010	The system shall store user session data.	Functional Requirement	Session Storage	TC010	Done
REQ-011	The system shall support CRUD (Create, Read, Update, Delete) operations for managing users within the system.	Functional Requirement	CRUD Operations for Users & Courses	TC011	Done
REQ-012	The system shall be easily maintainable with modular and well-documented code.	Non-Functional Requirement	Maintainability	TC012	Done
REQ-013	The system shall have security features, including password hashing and session management.	Non-Functional Requirement	Security	TC013	Done
REQ-014	The system shall be user-friendly, providing intuitive navigation and helpful error messages.	Non-Functional Requirement	Usability	TC014	Done
REQ-015	The system shall deliver high performance, with minimal response times for CRUD operations.	Non-Functional Requirement	Assurance	TC015	Done

13 Test Cases (Unit Testing)

- The User Class tests validate the functionalities of signup, edit, delete, and profile update handlers, ensuring reliable user management operations. All tests passed successfully, confirming their correctness.

```
$ vendor/bin/phpunit --testdox test/UserClassTest.php
PHPUnit 11.5.1 by Sebastian Bergmann and contributors.

Runtime:      PHP 8.2.4
.
.
.
Time: 00:00.021, Memory: 8.00 MB
.
.
.

User Class
✓ Signup handler
✓ Edit user handler
✓ Delete user handler
✓ Update profile handler

OK (4 tests, 4 assertions)
```

Figure 24: User Class Test Case

- The Course Class PHPUnit tests ran successfully, with all 4 tests executed and assertions made. However, minor errors were encountered due to a missing DB.php file and issues with method mocking (MethodCannotBeConfiguredException). These errors do not affect the test execution but should be addressed for smoother operation.

```
20111@Laila_s_Legion MINGW64 /c/xampp/htdocs/Megaminds-Course-Recommendation-System (main)
$ ./vendor/bin/phpunit test/CoursesClassTest.php
PHP Warning: include_once(..../.../public/includes/DB.php): Failed to open stream: No such
Warning: include_once(..../.../public/includes/DB.php): Failed to open stream: No such file
PHP Warning: include_once(): Failed opening '.../.../public/includes/DB.php' for inclusion
Warning: include_once(): Failed opening '.../.../public/includes/DB.php' for inclusion (in
PHPUnit 9.6.22 by Sebastian Bergmann and contributors.

E..E                                         4 / 4 (100%)

Time: 00:02.571, Memory: 6.00 MB
.
.
.

There were 2 errors:

1) CoursesClassTest::testAddCourseSuccess
PHPUnit\Framework\MockObject\MethodCannotBeConfiguredException: Trying to configure method
C:\xampp\htdocs\MegaMinds-Course-Recommendation-System\test\CoursesClassTest.php:27
C:\xampp\htdocs\MegaMinds-Course-Recommendation-System\vendor\phpunit\phpunit\phpunit:107

2) CoursesClassTest::testAddCourseDuplicate
PHPUnit\Framework\MockObject\MethodCannotBeConfiguredException: Trying to configure method
C:\xampp\htdocs\MegaMinds-Course-Recommendation-System\test\CoursesClassTest.php:88
C:\xampp\htdocs\MegaMinds-Course-Recommendation-System\vendor\phpunit\phpunit\phpunit:107

ERRORS!
Tests: 4, Assertions: 4, Errors: 2.
```

Figure 25: Courses Class Test Case

14 Project Plan

14.1 Team Members:

- Laila Amgad
- Roaa Khaled
- Nouran Hassan
- Yahia Tamer
- Malak Mohamed

14.2 Task Distribution Table

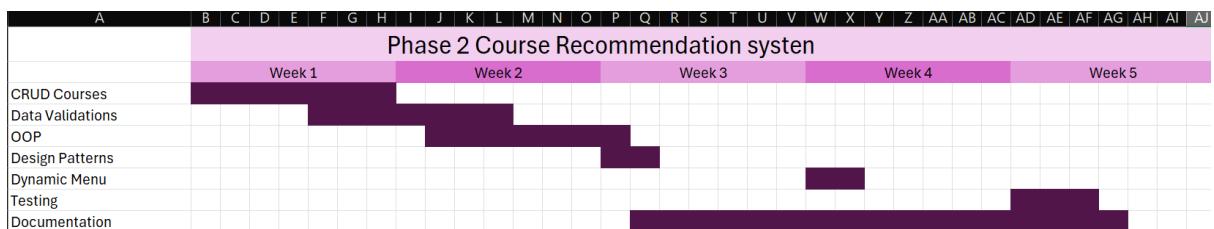


Figure 26: Task Distribution Table

15 Appendices

15.1 Definitions, Acronyms, Abbreviations

- CRUD: Create, Read, Update, Delete.
- SQL: Structured Query Language.

References

- [1] Coursera, *How Coursera Works*, [Online]. Available: <https://www.coursera.org/about>. [Accessed: December 19, 2024].
- [2] edX, *What is edX?*, [Online]. Available: <https://www.edx.org/>. [Accessed: December 19, 2024].
- [3] Khan Academy, *How Khan Academy Works*, [Online]. Available: <https://www.khanacademy.org/about>. [Accessed: December 19, 2024].
- [4] Udacity, *What is Udacity?*, [Online]. Available: <https://www.udacity.com/about>. [Accessed: December 19, 2024].
- [5] Northeastern University, *Tips for Taking Online Classes*, [Online]. Available: <https://graduate.northeastern.edu/resources/tips-for-taking-online-classes> [Accessed: December 19, 2024].

- [6] Harvard Gazette, *Study Shows That Students Learn More When Taking Part in Classrooms That Employ Active Learning Strategies*, [Online]. Available: <https://news.harvard.edu/gazette/story/2019/09/study-shows-that-students-learn-more-when-taking-part-in-classrooms-that-employ-active-learning-strategies/?utm> [Accessed: December 19, 2024].
- [7] The Raptors Nest, *Make the Most Out of a Challenging Course*, [Online]. Available: <https://theraptorsnest.camden.rutgers.edu/2019/08/make-the-most-out-of-a-challenging-course> [Accessed: December 19, 2024].
- [8] Faculty Focus, *Planning a Course You've Never Taught Before*, [Online]. Available: <https://www.facultyfocus.com/articles/course-design-ideas/planning-a-course-youve-never-taught-before> [Accessed: December 19, 2024].
- [9] Quality Matters, *Eight Ways to Improve Your Courses*, [Online]. Available: <https://www.qualitymatters.org/qa-resources/resource-center/articles-resources/eight-ways-to-improve-courses> [Accessed: December 19, 2024].
- [10] Inside Higher Ed, *8 Ways to Improve Your Online Course*, [Online]. Available: <https://www.insidehighered.com/blogs/higher-ed-gamma/8-ways-improve-your-online-course> [Accessed: December 19, 2024].