|  |  |  |
| --- | --- | --- |
| **Menoufia University** |  |  |
| **Faculty of Computers and Information** | **Year: Fourth Year** |
| **Computer Science Department** | **Graduation Project 2022-2023** |

**Graduation Project**

* **Team Members:**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Dept** | **Sec** |
| **1** | **أحمد فرجاني عبدالقادر حجازي** | **CS** | **1** |
| **2** | **أحمد محمد سعيد حامد رمضان** | **CS** | **1** |
| **3** | **أحمد محمد محمود علام** | **CS** | **1** |
| **4** | **محمد رضا فتحي الحناوي** | **CS** | **3** |
| **5** | **مصطفى رضا حجازي عبدالقادر** | **CS** | **4** |

**Supervised by: Dr. Sondos Magdy**

* **Project Title:**

|  |
| --- |
| **E-learning platform (Minda)** |

* **Description:**

Minda is an E-learning platform that provides students with important courses that is well-taught by professors, giving professors the ability to add courses, materials and make quizzes.

**Chapter 1**

Abstract

**E-learning platform**

This documentation is oriented to customers, manager, developer, designer, and tester

Introduction

Welcome to the documentation of the Minda, a graduation project aimed at providing a comprehensive solution for students and professors to streamline the learning process. This platform is designed to offer an interactive and engaging learning experience that allows students to take courses, quizzes, and assignments. At the same time, it provides professors with a platform to publish their courses, assignments, and quizzes.

The e-learning platform is built to cater to the needs of modern-day students who are always on-the-go. The platform is accessible from anywhere, anytime, making it easy for students to access learning materials and complete assignments at their convenience. The platform's user-friendly interface and intuitive design make it easy for students to navigate and find the resources they need to succeed.

Professors will find the e-learning platform equally useful. The platform provides professors with a centralized location to publish their courses, assignments, and quizzes, making it easy for students to find and access them. The platform's analytics tools allow professors to monitor student progress and performance, making it easier for them to provide personalized feedback and support to each student.

Overall, the e-learning platform is a powerful tool that can help both students and professors succeed in their educational endeavors. This documentation will provide you with an in-depth look at the platform's features, functionality, and architecture, so you can gain a better understanding of how it works and how it can benefit you.

**Chapter 2**

**Proposed system**

**Definition, requirements and modelling**

**Definition**:

The e-learning platform is a web-based application that aims to provide a comprehensive solution for students and professors to streamline the learning process. The platform offers a range of features such as course modules, quizzes, assignments, and analytics tools that allow students to access learning materials and complete assignments at their convenience. At the same time, professors can publish their courses, assignments, and quizzes on the platform, monitor student progress and performance, and provide personalized feedback and support to each student.

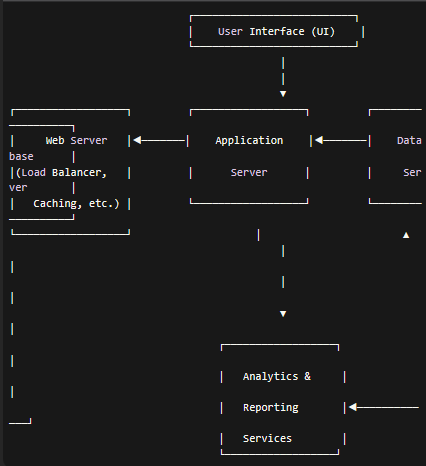
**Requirements**:

To build the e-learning platform, the following requirements must be met:

1. User Registration and Authentication: The platform should allow students and professors to register and log in to the platform using their credentials.
2. Course Management: The platform should allow professors to create, edit, and delete courses. Each course should have a name, description, and learning objectives.
3. Course Modules: The platform should allow professors to create modules for each course. Each module should have a name, description, and learning objectives.
4. Quizzes: The platform should allow professors to create quizzes for each module. Each quiz should have a set of questions, options, and correct answers.
5. Assignments: The platform should allow professors to create assignments for each module. Each assignment should have a name, description, and submission deadline.
6. Analytics: The platform should provide professors with analytics tools to monitor student progress and performance. The analytics should include data on student engagement, quiz scores, assignment submissions, and overall course performance.
7. User Interface: The platform should have a user-friendly interface that is easy to navigate and use.
8. Security: The platform should be secure and protect user data from unauthorized access.
9. Compatibility: The platform should be compatible with different web browsers and devices to ensure accessibility for all users.
10. Scalability: The platform should be scalable to accommodate a growing number of users and courses.

**Modelling:**

**For web-side**

****

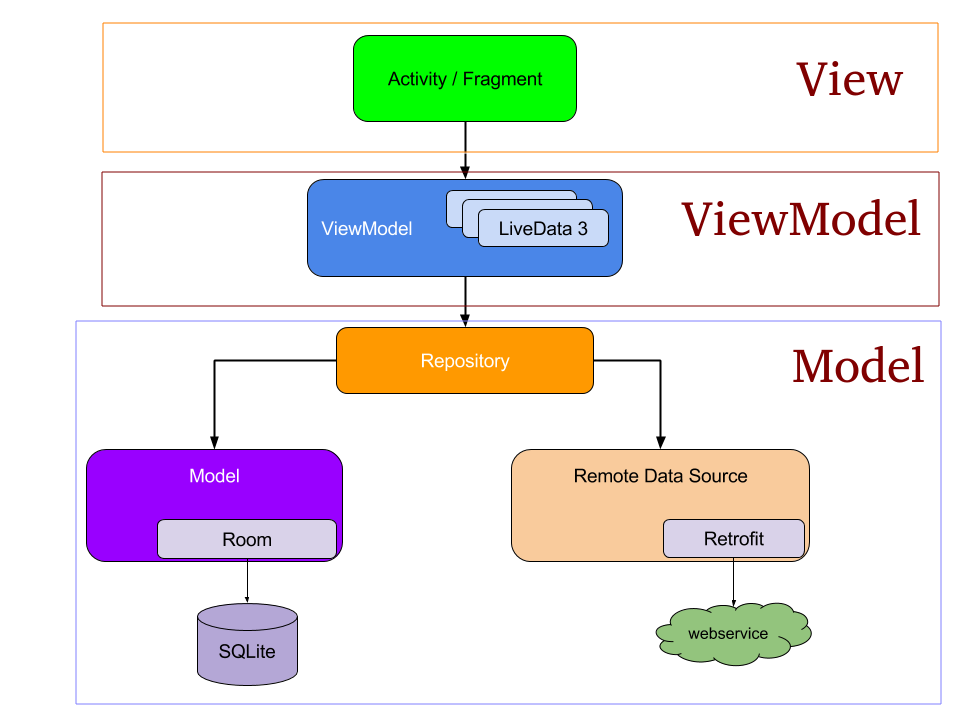
In this diagram, we have four main components:

1. User Interface (UI): This is the component that users interact with to access the e-learning platform. The UI could be a web application, mobile application, or a combination of both.
2. Web Server: This component handles incoming user requests and distributes them across multiple application servers using load balancing. It may also perform caching to reduce the load on the application servers.
3. Application Server: This is where the core business logic of the e-learning platform resides. It handles user authentication, course management, quiz and assignment creation, and other platform-specific functionality.
4. Database Server: This is where all the data related to the e-learning platform is stored. It includes data on courses, modules, quizzes, assignments, user profiles, and more.

Additionally, there is an Analytics & Reporting Services component that provides analytics and reporting functionality to the platform. This component collects data from the database and generates reports on student progress, performance, and engagement.

By modeling the e-learning platform in this way, we can better understand how the different components of the system work together to provide a seamless user experience and ensure that the platform is scalable and secure.

For Mobile-Side



MVVM (Model-View-ViewModel) is an architectural pattern commonly used in mobile app development to separate the presentation logic from the business logic of the application. In MVVM, the user interface is split into three main components: the Model, the View, and the ViewModel.

Here are the three layers of the MVVM architecture:

1. Model: The Model layer represents the data and business logic of the application. This layer is responsible for fetching and updating data, and it provides an abstraction layer for the data source, such as a database or API. The Model layer typically consists of data models and services that interact with the data source.
2. View: The View layer represents the user interface of the application. This layer is responsible for displaying data and responding to user input. The View layer consists of views, which are UI components such as screens, dialogs, and widgets.
3. ViewModel: The ViewModel layer acts as an intermediary between the View and Model layers. It is responsible for preparing and managing data for presentation in the View layer. The ViewModel layer exposes data and commands that the View can bind to, and it also handles user input and updates the Model layer accordingly. The ViewModel layer typically consists of view models, which are classes that implement the logic for a specific view.

MVVM architecture provides several benefits, including:

1. Separation of Concerns: MVVM separates the presentation logic from the business logic of the application, making the code easier to maintain, test and modify.
2. Testability: MVVM makes it easy to test the ViewModel layer in isolation from the View and Model layers.
3. Reusability: MVVM allows for the reuse of ViewModel classes for different Views and Models.
4. Better User Experience: MVVM can improve the user experience by providing a responsive and interactive user interface that is easy to use.

**Chapter 3**

**Technical work**

**Configuration and practical steps**

**The configuration** of the e-learning platform project involves setting up the necessary software and environment to develop, deploy, and run the application. Here are some of the key configuration steps involved in setting up the e-learning platform:

1. Development Environment: The development environment should be set up with the necessary software tools and frameworks to develop the e-learning platform. This typically involves installing a web server such as Nginx, a programming language such as Kotlin, Typescript or Java, a database such as MongDB, and any additional libraries or frameworks required for the project.
2. Front-end Development: The front-end development involves creating the user interface of the e-learning platform. This typically involves using HTML, CSS, and JavaScript for web, Kotlin for Android to design and develop the user interface of the platform. Front-end frameworks such as Angular can also be used to simplify the development process.
3. Back-end Development: The back-end development involves creating the application logic and server-side functionality of the e-learning platform. This typically involves using a web framework such as Node JS to handle user authentication, course management, quiz and assignment creation, and other platform-specific functionality.
4. Database Configuration: The database configuration involves setting up the database server and creating the necessary tables and relationships to store data related to courses, modules, quizzes, assignments, user profiles, and other platform data. This typically involves using a database management system such as MongoDB and Firebase.
5. Deployment Configuration: The deployment configuration involves setting up the necessary infrastructure to deploy the e-learning platform to a production environment. This typically involves configuring a web server such as Cloudinary and Filestack, load balancer, caching, and other components to handle high levels of traffic and ensure the platform is scalable and reliable.
6. Security Configuration: The security configuration involves implementing security measures to protect the e-learning platform from potential threats such as unauthorized access, data breaches, and malware attacks. This typically involves implementing authentication and authorization mechanisms such as JWT and JOI, securing the application and database servers, and implementing encryption for sensitive data.

Overall, the configuration of the e-learning platform involves setting up a robust development environment, implementing front-end and back-end functionality, configuring the database, deploying the application to a production environment, and ensuring the platform is secure and scalable.

**Practical Steps**

1. Define the Requirements: Define the requirements for the e-learning platform, including user registration and authentication, course management, course modules, quizzes, assignments, analytics, user interface, security, compatibility and scalability.
2. Choose the Technology Stack: Choose the technology stack for the e-learning platform, including the programming language, web framework, front-end framework, database management system, and other tools and libraries needed for development.
3. Set Up the Development Environment: Set up the development environment with the necessary software tools and frameworks to develop the e-learning platform.
4. Design the User Interface: Design the user interface of the e-learning platform using HTML, CSS, and JavaScript, and a front-end framework if applicable.
5. Develop the Back-end Functionality: Develop the back-end functionality of the e-learning platform using a web framework such as Laravel, Django, or Node JS to handle user authentication, course management, quiz and assignment creation, and other platform-specific functionality.
6. Configure the Database: Configure the database server and create the necessary tables and relationships to store data related to courses, modules, quizzes, assignments, user profiles, and other platform data.
7. Implement Security Measures: Implement security measures to protect the e-learning platform from potential threats such as unauthorized access, data breaches, and malware attacks. This typically involves implementing authentication and authorization mechanisms, securing the application and database servers, and implementing encryption for sensitive data.
8. Test the Application: Test the e-learning platform for functionality, performance, and security, and fix any issues or bugs that are discovered.
9. Deploy the Application: Deploy the e-learning platform to a production environment, including configuring a web server, load balancer, caching, and other components to handle high levels of traffic and ensure the platform is scalable and reliable.
10. Monitor and Maintain the Application: Monitor and maintain the e-learning platform to ensure it is secure, stable, and performs well. This involves regular updates, backups, and security patches, and responding to any issues or incidents that arise.

**Chapter 4**

**Results**

|  |  |
| --- | --- |
| Authentication | Making sign in and sign up pages to give the ability for every user to edit, have profile, and order with his name. |
| HTML | The structure markup language to make the page skeleton. |
| CSS | Used for adding some visuals to website’s pages |
| JavaScript | Programming language used for adding some functionality and interactivity to website’s pages |
| XML | A metalanguage which allow users to define their own markup language |

|  |  |
| --- | --- |
| SQL Server | Relational database management system to manage the database, produce queries and reports and update upcoming data. |
| GUI | The graphical user interface. |
| CSS | Used for adding some visuals to website’s pages. |
| C# | Programming language used for many purposes such as creating desktop application or GUIs. |
| Java | Programming language used for many purposes such as creating desktop application or GUIs. |
| Kotlin | A modern programming language that is used and more specified in android application development. |
| MVVM | Model-View-View-Model, an architecture design pattern that is recommended by google in android development. |
| Retrofit | A library that allows developers to deal with REST APIs through android development |
| MVC | **Model–view–controller** is a software architectural pattern commonly used for developing [user interfaces](https://en.wikipedia.org/wiki/User_interface) that divide the related program logic into three interconnected elements. |
| API | Application Programming Interface that act as intermediate to bring data from server to client |
| Node JS | A back-end framework for making RESTFUL apis |
| Express JS | A back-end framework for making RESTFUL apis |
| ASP.Net Core | A cross platform framework to make web applications using C# |
| Angular | A framework for designing web-based applications |
| Typescript | Programming language for front-end purposes |
| MongoDB | Cross-platform document oriented database program |
| JWT | a [proposed Internet standard](https://en.wikipedia.org/wiki/Internet_Standard#Proposed_Standard) for creating data with optional [signature](https://en.wikipedia.org/wiki/Signature_(cryptography)) and/or optional [encryption](https://en.wikipedia.org/wiki/Encryption) whose [payload](https://en.wikipedia.org/wiki/Payload_(computing)) holds [JSON](https://en.wikipedia.org/wiki/JSON) that asserts some number of [claims](https://en.wikipedia.org/wiki/Claims-based_identity). |
| JOI | Joi validate values in JavaScript with Hapi Joi module. |
| Cloudinary | Cloudinary's Node JS SDK provides simple, yet comprehensive image and video upload, transformation, optimization, and delivery capabilities |
| Filestack | Filestack operates in three main areas of modern software file management systems: Uploads, Transformations, and Delivery |
| Firebase | Firebase is a set of backend cloud computing services and application development platforms provided by Google. |

User Requirement Definition

Functional Requirement:

* Add user by providing name, email, password.
* Log in using email and password.
* Search about Course using the name of Course.
* Home page contains Info and Goal of platform‎‎.
* Course page contains the most popular Courses.
* Settings page allows users to modify his basic Info.
* Professors can add or delete or update Course in platform.
* By completing particular courses, student will have an achievement badge.
* Student can take a quiz before enrolling to a course to determine whether he really needs to take this course or not.
* Profile page contains personal info, badges earned and contacts.
* staff can evaluate students after finishing any courses
* staff has the authority to modify the students of the course
* admin have all authorities to manage staff, students and courses

Non-Functional Requirement:

* Safety Requirements
* Software Quality Attributes
* Usability Requirement
* Performance Requirement
* Efficiency Requirement
* Product Requirement.
* Organizational Requirement.

System Architecture

System Requirement Specification

Functional Requirement:

* A user shall be able to register in system by providing name, email, password
* The system shall response to unfilled registration formation.
* The system shall contain a login page using email and password.
* Each staff member using the system shall be uniquely identified by his or her eight-digit employee number.
* Staff shall be able to add or delete book in courses.

Non- Functional Requirement:

* Safety Requirements: - The database may get crashed at any certain time due to virus or operating system failure. Therefore, it is required to take the database backup.
* Software Quality Attributes: - The Quality of the database is maintained in such a way so that it can be very user friendly to all the users of the database.
* System shall be available 24 hours per day
* Only staff shall manage all user accounts.