**План за дипломна теза**

# Описание

## Софтуерна част:

Софтуерна част:

1. От какво е съставена?
2. Подробно разглеждане на архитектурата
3. Файлова организация :

* Снимка на Solution Explorer
* Коя папка за какво отговаря и какво съдържа

1. Архитектура на базата от данни
2. Схема на базата от данно

* От какво се състои базата от данни

1. Описание на всяка една таблица

* Таблица Х представя ....
* Какви атрибути има
* Снимка на таблицата с описание на атрибутите(пояснение)

### ASP .NET MVC Pattern

**Обяснение на ASP.NET MVC + снимка на файловата ни архитектура (малко техн. Обяснение + ):**

### Microsoft SQL Server

**Изображение на таблиците от базата + релацията им**

**Таблица Х представлява ....**

**Има следните атрибути ...**

**Изображените/таблица за атрибутите:**

**Име на атрибут / Тип / Пояснение (уникално идентифициране на всеки един потребител)**

## Програмна Реализация

**На фигура Х /Solution Explorer/ се вижда базовата файлова структура на проекта. Тя се състои от множество пакети, които групират класове и интерфейси, за да обособят модулите на проекта. В директорията wwwroot се съдържат ...**

**Пакета Controllers ....**

* **Структура на Controllers пакета + снимка на Solution Explorer**

**Пакета Models ……..**

1. Create Program page
2. Create Course page

* Course Name
* Course Description
* Course Program
* Inheritance
* Parent Course Id
* Course Criteria
  + Start Date
  + Skip Course End Date

1. Create Lecture page

* Lecture Name
* Lecture Desc
* Inheritance
  + Corresponding Course
  + Parent Lecture
* Lecture Materials
  + Lecture materials
  + Homework materials
  + Video URL
* Lecture Criteria
  + IsFinal
  + IsExam
  + Required Grade
  + Date/Time to show
  + Date/Time for HW end
* !! On create – If Lecture is final – check if some students have passed the course by exam lectures

Профилна страница:  
е разделена на две части, като първата предоставя възможност на студента да промени личните си данни и предпочитания. Той може да добави или промени имената си, които ще се използват за идентификация при оценяване на предадените задания. Освен това, в профила студентът може да направи избор какъв вид учебни материали предпочита - текстови, видео или и двата вида. Този избор ще бъде взет предвид при показването на лекциите за курсовете.

Във втората част на профилната страница се предоставя информация за всички текущи и завършени курсове, към които студентът е записан или е преминал. Фигура 28 показва списък с текущите курсове, които ученикът е записал и му предоставя бърз достъп до тях. Фигура 29 показва списък с преминатите курсове и оценките за тях, както и дава достъп до курса и материалите към него.

Описание на: • Course service o CreateCourseAsync(CreateCourse input, ApplicationUser user) - създаване на нов курс в базата данни, добавяне на задание за пропускане на курса o AddSkippingExamSolutionToCourse - добавяне на решение на заданието за пропускане в базата o SetNextCourseId - създаване на наследяване на курса o RemoveCourseInheritance - премахване на наследяването от курса при неговото изтриване o GetAllAsKeyValuePairs - създаване на списък с името на програмата и курса o GetNextCourseSuggestion - показва кой е следващият препоръчителен курс за записване/преминаване o DeleteSkippingCourseMaterial - изтриване на материалите за пропускане на курс o DeleteSkippingAssignment - изтриване на предадени задания за пропускане на курс o DeleteCourse - изтриване на курс от базата и премахване на наследяване

# Използвани технологии

ASP.NET Core is an open-source web application framework that is used to build modern, cloud-based web applications. It is a cross-platform framework that runs on Windows, macOS, and Linux. ASP.NET Core is built on top of .NET 5.0, and it provides a modular and high-performance architecture for building web applications.

One of the key features of ASP.NET Core is its support for views and partial views. Views are used to generate the HTML markup that is sent to the client's web browser. Partial views, on the other hand, are reusable views that can be included within other views. This allows developers to build modular and reusable UI components.

Another feature of ASP.NET Core is its support for areas. Areas allow developers to logically group related functionality within a web application. This makes it easier to organize and manage large web applications. Each area can have its own controllers, views, and other components.

ASP.NET Core also includes support for the Model-View-Controller (MVC) architectural pattern. This pattern separates the application into three distinct parts: the model, the view, and the controller. The model represents the application's data and business logic. The view is responsible for generating the HTML markup that is sent to the client's web browser. The controller handles incoming requests, interacts with the model, and returns the appropriate view.

Finally, ASP.NET Core includes support for Razor Pages. Razor Pages provide a simplified programming model for building web applications. They are built on top of the MVC pattern and provide a more streamlined approach for building simple web applications. Razor Pages allow developers to create web pages with minimal ceremony, making it easier to build web applications quickly and efficiently.

# Заключение

В нашето приложение за адаптивно обучение се стремим да предоставяме индивидуализирано и персонализирано обучение, което е подходящо за вашите нужди и ниво на знания. Надяваме се, че ще откриете началната страница полезна и ще продължите да използвате нашето приложение за да усъвършенствате уменията си и да развивате знанията си.

**Намери къде да добавиш:**

Проверка за освобождаване на курс при оценяване на предадени домашни задания - Тази функционалност гарантира, че студентите, които покриват изискванията от зададените като изпитни лекции, могат да бъдат освободени от курса. В същото време системата се грижи да изключи възможността за освобождаване от курса преди да сме сигурни, че всички лекции за него са добавени. /добавяне на нови изпитни лекции след края на курса, осигурявайки така правилно оценяване и постоянство в оценките/

1. Въведение:

* Обяснение на целта на онлайн платформата за адаптивно обучение и нейните основни функционалности.

1. Анализ на изискванията:

* Описание на изискванията за платформата, включително функционалностите, които ще бъдат осигурени, целевата аудитория и техните нужди.

1. Архитектура на платформата:

* Описание на архитектурата на платформата, включително базата данни, инфраструктурата, интерфейса на потребителя и други компоненти.

1. Разработка на приложението:

* Описание на процеса на разработка на приложението, включително използваните технологии и инструменти.
* Обяснение на начина, по който функционалностите на приложението работят заедно.

1. Модули на платформата:

* Описание на различните модули на платформата, включително:
* Модул за управление на потребителите и достъпа до съдържанието.
* Модул за предоставяне на адаптивни пътища за учене на базата на предпочитанията на потребителите.
* Модул за управление на учебните материали и дейности.
* Модул за събиране и анализ на данни за прогреса на потребителите.
* Модул за комуникация между потребителите и инструкторите.

1. Интерфейс на потребителя:

* Описание на интерфейса на потребителя, включително външния вид, навигацията и функционалностите, които ще бъдат налични за потребителите.
* Тестване и оценка:
* Обяснение на начина, по който платформата ще бъде тествана и оценена.
* Описание на използваните методи и инструменти за тестване.
* Описание на процеса на оценка на резултатите от тестването и възможните подобрения, които могат да бъдат направени.

1. Заключение:

План за описване на онлайн платформа за адаптивно обучение:

1. Въведение

* Описание на целите на платформата и предимствата от използването ѝ

1. Функционалности

* Създаване на курсове: описание на процеса на създаване на курсове, включително качване на материали и задания
* Оценяване на задания: описание на процеса на оценяване на заданията, качени от студентите
* Определяне на мястото на учебните обекти: описание на функционалността за определяне на мястото на учебните обекти, които имат отношение към пътя за учене
* Определяне на критерии за достъп до учебните обекти: описание на критериите за достъп до отделните обекти в курса, включително оценка от представянето на обучавания до момента и покриване на предшестващите обекта дейности
* Индивидуално обучение: описание на функционалността за индивидуално обучение чрез подбор на подходящи по трудност и формат учебни обекти
* Правила за завършване на курса: определяне на правилата за завършване на курса чрез набор от завършени дейности с определено ниво на трудност

1. Технологии

* Описание на технологиите, използвани за разработването на платформата, включително ASP.NET Core, Entity Framework Core 5.0, HTML, CSS и SQL Server.

1. Извод

* Резюмиране на функционалностите на платформата и предимствата от използването ѝ, както и бъдещите възможности за развитие.