

Документация на индивидуален проект по курс

# **CITB549 Проект: Клиент-сървър технологии**

Изготвен от: Емилия Стоянова,  
с факултетен номер: F108556



## 1. Въведение

Текущият проект е изготвен съобразно изискванията по курс „Проект: Клиент-сървър технологии“ със сигнатура CITB549 и преподавател в курса –гл. ас. д-р Лъчезар Томов. Проектът представлява приложение с клиент-сървър архитектура, в което потребител може да оставя, редактира и/или изтрива коментари. Темата на проекта е автобиографичен сайт (тип CV), в който да се съдържа галерия със снимки и да бъде responsive.

## 2. Базата данни на проекта.

Базата данни към проекта е реализирана с SQL Server Management Studio 20. Сървъра е локален, достъпва се чрез Windows Authentication. Базата в този сървър към проекта е с име cvProgram. В нея се съдържат 2 таблици, които се използват от приложението: dbo.messages и dbo.userRegistration. И двете таблици са създадени с първа колона първичен ключ със стойност uniqueidentifier. Всяка следваща колона има за цел да удовлетвори изискванията за дадена функционалност. Таблицата за съобщенията съдържа информация освен за уникалния идентификатор на всяко съобщение също и за името на човека, имейл адреса му и съдържанието на оставеното съобщение. Таблицата за регистрация съдържа освен уникален идентификатор за всеки регистрирал се потребител също и за неговото име, фамилия, имейл адрес и парола.

### 3. Сървърна част

Сървърната част на проекта е реализирана в Visual Studio IDE с темплейтен проект ASP.NET Core Web API. За да бъдат успешно имплементирани функциите и връзката с базата в проекта са допълнително конфигурирани следните NuGet пакети:

Microsoft.EntityFrameworkCore,

Microsoft.EntityFrameworkCore.SqlServer както и EF Core Power Tools. В Programs.cs файлът са добавени имплементации във връзка успешното стартиране на проекта, установяване на връзка с базата, съобразяване на CORS policy както и наличие на Swagger с цел тест. Стрингът на връзка с базата е също конфигуриран в appsettings.json файл. Данните от базата за обработени чрез cvContext файл, който е автоматично генериран при всяка промяна на инстанцията на базата с помощта на EF Core Power Tools, в който се декларират полетата от всяка таблица. Класовете в проекта също представляват обектите от всяка таблица. Тъй като има имплементирани отделни заявки за Log in и регистрация за отделните страници е добавен допълнителен клас LoginRequest.cs за улеснение и по-добра четимост на заявките в контролер файла. Наличните класове са:

- LoginRequest.cs
- Message.cs
- userRegistration.cs

Заявките за обработване на данните в базата за реализирани в отделна папка под името Controllers. В нея се съдържат два отделни файла, които са MessagesControllers.cs и UserRegistrationControllers.cs. И двата файла съдържат обект от

класа `cvContext`, който е `_cvContext`, с цел достъпване на данните в базата. Файлът контролер на съобщенията съдържа 4 HTTP заявки, които са: GET, POST, PUT, DELETE. Всяка заявка, освен GET, съдържа в себе си try-catch конструкция и проверка които да служат за Error handling на потенциални проблеми, които могат да възникнат по време на изпълнение на заявката. Файлът контролер на потребителските заявки съдържа 2 POST заявки, като едната служи за Log In а другата за регистрация в платформата. Освен че двете заявки съдържат в себе си Error handling behavior, в заявката за регистрация съществува валидация на сървърно ниво за дублицирани имейли, която не позволява съществуването на два записа с един и същи имейл адрес.

#### 4. Клиентска част

Данните достъпват до клиентската част чрез извличането им през URL от Swagger. Логиката по обработването на GET заявката за съобщенията, POST заявката за съобщенията и POST заявките за регистрация и за вход в платформата са имплементирани в отделни JavaScript файлове и са добавени към HTML структурата само на конкретната страница. Освен логиката за обработването на данните на клиентско ниво е също имплементирана Slider функционалност, която да има ролята на галерия в сайта, както и зареждането на header, navigation bar и footer като темплейти в по-голямата част от HTML файловете. Стилизирането на клиентската част е реализирано почти изцяло с Bootstrap framework като с него е постигнат и responsiveness на сайта с подхода mobile first. Всяка

страница е отделена в собствен HTML файл, като са налични 3 темплейтни HTML-а – за header, footer и navigation bar. Страниците с проекта са: homePage, about, getInTouch, allMessages, loginPage, registrationPage.

- homePage – достъпва се през навигацията, представлява уводна страница с кратка обобщена лична информация.
- about – достъпва се през навигацията, разширява информацията от началната страница, като съдържа и галерия под формата на слайдър.
- getInTouch – достъпва се през навигацията, съдържа контактна форма за оставяне на съобщение, данни за връзка извън контактната форма и бутон за страница с всички съобщения.
- allMessages – достъпва се само през getInTouch страницата. Съдържа всички съобщения, запазени в базата. Всяко съобщение има опцията да бъде изтрито или да се редактира неговото съдържание.
- loginPage – съдържа контактна форма за вход с полета за въвеждане на потребителско име и парола. Съдържа линк към страницата за регистрация.
- registrationPage – съдържа контакта форма с полета за потребителско име, име, фамилия, имейл адрес, парола и потвърждение на паролата. На клиентско ниво е имплементирана валидация за правилен формат на имейл адреса и съвпадение на паролите. Съдържа линк към страницата за вход.