**НАЦИОНАЛНА ПРОФЕСИОНАЛНА ГИМНАЗИЯ ПО КОМПЮТЪРНИ ТЕХНОЛОГИИ И СИСТЕМИ**

**ГР.ПРАВЕЦ ПРИ ТЕХНИЧЕСКИ УНИВЕРСИТЕТ – СОФИЯ**

**ДИПЛОМНА РАБОТА**

ДЪРЖАВЕН ИЗПИТ ЗА ПРИДОБИВАНЕ НА ТРЕТА СТЕПЕН НА ПРОФЕСИОНАЛНА КВАЛИФИКАЦИЯ – ЧАСТ ПО ТЕОРИЯ НА ПРОФЕСИЯТА

на

Ученик: *Мартин Кръстев Йорданов*

ученик/чка от (XII) клас

Тема: **Информационен уебсайт за университети**

**ПРОФЕСИОНАЛНО НАПРАВЛЕНИЕ:** 481 КОМПЮТЪРНИ НАУКИ

**ПРОФЕСИЯ:** 481020 СИСТЕМЕН ПРОГРАМИСТ

**СПЕЦИАЛНОСТ:** 4810201 СИСТЕМНО ПРОГРАМИРАНЕ

Ученик: ………………….. Ръководител: ………………………

*Ивелин Михайлов*

Правец, 2024

**Съдържание**

[**Увод 2**](#_jktz1f8apxs2)

[**Глава 1: Литературен обзор 3**](#_lilitklxixrc)

[1.1 Методи за разработка на уеб - приложения 3](#_zarwnqb9axit)

[1.1.1. Agile Методология 3](#_wdzsa1mt5prr)

[1.1.2 Използване на Фреймуърк и Библиотеки 3](#_dgp00yspcucx)

[1.1.3 Waterfall Методология 4](#_d7pbu3a0ytnz)

[1.1.4 DevOps Практики 4](#_3u6zgqwjh7ci)

[1.1.5 Continuous Integration (CI) и Continuous Deployment (CD) 4](#_1r7l5nb3ft01)

[1.2 Среди за разработка 5](#_xvhmawjrzalb)

[1.2.1 Visual Studio 5](#_obbpkm824zta)

[1.2.2 Visual Studio Code 7](#_8r1866aoj7nc)

[1.2.3 MySQL Workbench 7](#_mlrwr6qdyp8i)

[1.3 Предимства пред други уеб сайтове 8](#_ip0sd783w2dw)

[**Глава 2: Теоретична част 10**](#_ahxo5jvpuko)

[2.1 Използвани технологии и среди за разработка 10](#_easac59r0yw6)

[2.1.1 HTML(HyperText Markup Language) 10](#_q90kdwb0v48w)

[2.1.2 CSS(Cascading Style Sheets) 11](#_qe0yddysvghk)

[2.1.3 PHP(Hypertext Preprocessor) 12](#_1imd6bld12ef)

[2.1.4 MySQL 13](#_y71l74a2ud4p)

[2.1.5 Bootstrap 13](#_86har39vo3u7)

[2.1.6 Xampp 14](#_zduw8zdo2nlf)

[2.1.7 Visual Studio Code 14](#_sc6rrfl6oy0b)

[2.1.8 MySQL Workbench 14](#_rp9ztk47rpp3)

[2.2 Файлова структура на приложението 16](#_cno5jcgvk5yg)

[**Глава 3: Инженерно решение на поставената задача 17**](#_q42dofhtu6ey)

[3.1 Описание на кода 17](#_bz9m4xlsby89)

[3.2 Описание на база данни 28](#_juulpusfaz3f)

[**Глава 4: Анализ на получените резултати 29**](#_93stg19u3r4q)

[**Заключение 37**](#_3g2s1oz5hp)

[Какво може да се се подобри в приложението? 37](#_kt38ffsp94qa)

[**Използвана литература 38**](#_lkp72vrs285a)

[**Списък с използвани съкращения и означения 39**](#_jk3yv1qat7me)

# 

# **Увод**

Темата на дипломната ми работа е информационен уеб сайт за университетите в България. Целта на сайта е да предостави лесен и функционален начин да се търсят университети в страната чрез града, в който са, и специалностите, които прегладат. За всеки университет се представя кратка информация за в съответния прозорец, както и линк с препратка към официалния сайт на университета, където потребителите могат да се информират допълнително. Като функционалност сайтът също разполага с форум, в който могат да се търсят публикации чрез ключови думи.

Сайтът е с Responsive(адаптивен) дизайн за мобилни телефони. Предоставена е възможност за автентикация на потребителския достъп с имейл и парола, като чрез потребителски профил потребителите получават достъп до форум, в който могат да публикуват и да четат публикации по дадена тема.

Сайтът разполага с начална страница, която насочва потребителя към страницата за създаване на профил или към основната страница, в която се съдържа информацията за университетите. Също има страница за влизане, страница с форум и страница с информация и контакти за връзка с нас. Има и страница за всеки един университет, която се изкарва от базата данни.

Уебсайтът е удобен и лесен за ползване. Той има проста структура и дизайн, които улесняват потребителите в търсенето и намирането на нужната информация. Софтуерът е напълно на български език, за да може да достигне по-голям брой от българската аудитория, посещаваща сайта.

Една от основните причини да избера точно тази тема за дипломната ми работа е липсата на качествени сайтове, които изпълняват тези функции.

# Глава 1: Литературен обзор

## 1.1 Методи за разработка на уеб - приложения

Разработката на уеб приложения е сложен процес, който изисква комбинация от различни методи, технологии и стратегии. Това са някои от най-използваните методи за разработка на уеб приложения.

### **1.1.1. Agile Методология**

Agile методът е популярен подход за разработка на софтуер, който се фокусира върху гъвкавостта и адаптивността. Този метод насърчава сътрудничеството между разработчиците, клиентите и потребителите през целия процес на разработка. Agile методологията включва редовни итерации и инкрементално развитие на продукта, като се отчитат обратните връзки и се правят корекции спрямо тях. Въпреки че Agile често се свързва с работа в екип, принципите му за гъвкавост и адаптивност могат да бъдат полезни и за индивидуален разработчик. Agile насърчава итеративен подход към разработката, който може да помогне на разработчика да се адаптира към променящите се изисквания и да подобри своето производство. Гъвкавите методи разбиват задачите на малки стъпки с минимално планиране, без да засягат дългосрочното планиране на проекта. Итерациите (етапите) стават на кратки срокове (timeboxes), които обикновено траят от една до четири седмици. През всяка итерация екипът работи по всяка една от функциите: планиране, анализ на изискванията, проектиране, разработка, тестване и проверка при приемане.

### **1.1.2 Използване на Фреймуърк и Библиотеки**

За разработката на уеб приложения се използват различни фреймуърки и библиотеки за различните аспекти на приложението - от фронтенда до бекенда. Някои от популярните фреймуърки включват React.js, Angular, Vue.js за фронтенд разработка и Django, Ruby on Rails, Express.js за бекенд разработка. Една от най-популярните CSS фреймуърки е Bootstrap.

Най-общо казано в програмирането “framework” наричаме софтуер, който носи в себе си базови функционалности в зависимост от типа приложение, който искаме да изграждаме. Това е така, тъй като приложенията от даден тип отговарят на определени нужди и така в един момент те започват да предлагат сходни решения и възможности. Разработчикът взема този общ софтуер и чрез добавяне на необходимия му код, той създава собственото приложение от което има нужда.

Казано иначе, един фреймуърк (или наричан още „технологична рамка“ или „софтуерна рамка“) представлява универсална, преизползваема софтуерна среда, която е стандартизиран способ за изграждане и разполагане на приложения. По този начин технологичните рамки ускоряват разработката на софтуерни приложения, продукти и различни решения. За целта технологичните рамки могат да включват и допълнителни приложения и инструменти като компилатори, библиотеки, application programming interface-и (API-та), като всичко това ви позволява пълноценна разработка.

Технологичните рамки имат няколко основни свойства, които ги различават от софтуерните библиотеки: инверсията на контрол (при което фреймуъркът „диктува“ на разработчика как да структурира програмата си и „следи“ поведението на потребителя), добавянето на допълнителен код (разработчикът може да модифицира технологичната рамка, чрез добавяне на външни модули код, които да му дадат нови възможности) и немодифицируем код (въпреки, че разработчикът може да добавя външни модули, той не може да редактира основния код на framework-а по никакъв начин).

### **1.1.3 Waterfall Методология**

Waterfall методологията е по-традиционен подход, който включва последователност от фази - анализ, проектиране, имплементация, тестване и поддръжка. Всяка фаза завършва преди да започне следващата, като резултатът се предава на следващата фаза. Този метод е подходящ за проекти с ясно дефинирани изисквания и стабилна спецификация.

### 1.1.4 DevOps Практики

DevOps е практика, която се фокусира върху автоматизацията, колаборацията и комуникацията между разработчиците и операционния екип. Целта на DevOps е да се подобри скоростта на разработка и доставка на софтуер, като се намалява времето за изграждане, тестване и доставка на нови функции.

### 1.1.5 Continuous Integration (CI) и Continuous Deployment (CD)

Continuous Integration и Continuous Deployment са практики, които автоматизират процесите на тестване, интеграция и доставка на софтуер. Целта им е да се осигури постоянно качество на кода и бърза доставка на нови функции към потребителите.

В случая за проект като този, разработен от един човек, най-много помагат agile методологията и фреймуърките.

## 1.2 Среди за разработка

### 1.2.1 Visual Studio

Microsoft Visual Studio е мощна интегрирана среда за разработка на софтуерни приложения за Windows и за платформата .NET Framework. Използва се за разработка на конзолни и графични потребителски интерфейс приложения, както и Windows Forms или WPF приложения, уеб сайтове, уеб приложения и уеб услуги на всички поддържани платформи от Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework и Microsoft Silverlight.

Visual Studio предоставя мощна интегрирана среда за писане на код, компилиране, изпълнение, дебъгване (както за високо така и за машинно ниво), тестване на приложения, дизайн на потребителски интерфейс (форми, диалози, уеб страници, визуални контроли и други), моделиране на данни, моделиране на класове, изпълнение на тестове, пакетиране на приложения и стотици други функции. Могат да се добавят и плъгини, които повишават функционалността на почти всяко ниво – включително добавянето на поддръжка за source-control системи (като Subversion и Visual SourceSafe), добавяне на нови инструменти като редактори и визуални дизайнери за domain-specific languages или инструменти за други аспекти (като например: Team Foundation Server, Team Explorer).

Visual Studio поддържа различни езици за програмиране (например: C#, VB.NET, C/C++, F#, XML/XSLT, HTML/XHTML, JavaScript, CSS и други) и различни технологии за разработка на софтуер (Win32, COM, ASP.NET, ADO.NET Entity Framework, Windows Forms, WPF, Silverlight и още десетки други Windows и .NET технологии). Съществуват и отделни езикови версии на Visual Studio, които обаче предоставят по-ограничени услуги за потребителя: Microsoft Visual Basic, Visual J#, Visual C#, and Visual C++.

Microsoft предлага безплатни „Express“ издания на някои от компонентите на Visual Studio 2010. Като например: Visual Basic, Visual C#, Visual C++, Visual Web Developer и други. Чрез програмата DreamSpark, Microsoft предоставя на студенти напълно безплатно някои от своите продукти. Чрез нея могат да бъдат изтеглени Visual Studio 2012, 2010, 2008 и 2005 Professional Editions, заедно със специфичните езикови версии (Visual Basic, C++, C#, J#) на Visual Studio Express 2010 и други.

Visual Studio не поддържа нито един език за програмиране, решение или вътрешен инструмент, вместо това функционалността от плъгини се кодира като VSPackage. Когато се инсталира, функционалността се предлага като услуга. Самото IDE предлага три услуги: SVsSolution, което дава възможност да се изброят проекти и решения; SVsUIShell, който предвижда прозоречна и UI функционалност (включително и табовете, ленти с инструменти и Windows-ските инструмент) и SVsShell, който се занимава с регистрация на VSPackages. В допълнение, IDE е отговорен за координирането и осигурява комуникация между услугите. Всички редактори, графиките, видовете проекти и други инструменти са внедрени като VSPackages. Visual Studio използва COM за достъп до VSPackages. Visual Studio SDK включва и Managed Package Framework (MPF), който е набор от управлявани обвивки около COM-интерфейси, които позволяват на пакети да бъдат написани на всеки съвместим език CLI. Въпреки това, MPF не предоставят цялата функционалност изложени от COM интерфейсите на Visual Studio. Услугите могат да бъдат използвани за създаването и на други опаковки, които добавят функционалност към Visual Studio IDE.

Поддържането на езици за програмиране се добавя със специален VSPackage, който се нарича Language Service. Тази услуга определя различните интерфейси, които VSPackage предоставя, за да се добавят различни функционалности. Функционалностите, които могат да бъдат добавени по този начин са синтактичното оцветяване, завършване на писаното, съчетаването на скобите, помощна информация с пояснения за параметрите, списъците на елементите и маркери за грешка при компилацията. Ако интерфейс се реализира, функционалността ще бъде на разположение за езика за програмиране. Езиковите услуги трябва да бъдат приложени в основата на всеки език за програмиране. Приложенията може да използвате повторно кода от анализатора или компилатор на дадения езика. Езикови услуги могат да се изпълняват в собствения (native) или управляемия код. На свой ред, собствения код може да се използва съответно за интерфейсите COM или Babel Framework (част от Visual Studio SDK). За управляемия код, MPF включва опаковки (слоеве) за управляване на писането услуги език.

Във Visual Studio не се включва каквато и да е поддръжка на source control, но се определят два алтернативни начина на системи за source control да се интегрира в IDE. Един VSPackage Control Source може да осигури собствен персонализиран потребителски интерфейс. За разлика от това, плъгин за source control се ползва от MSSCCI (Microsoft Source Code Interface Control), предвижда набор от функции, които се използват за изпълнение на различна функционалност за контрол на кода (source control) със стандартен интерфейс. MSSCCI първо е трябвало да интегрира Visual SourceSafe в Visual Studio 6.0, но е по-късно направено чрез Visual Studio SDK. Visual Studio. NET 2002, използва MSSCCI 1.1 и Visual Studio. NET 2003 използва MSSCCI 1.2. Visual Studio 2005, 2008 и 2010 г. използването MSSCCI Версия 1.3, която добавя поддръжка за преименуване и изтриване (propagation), както и асинхронен отваряне.

Visual Studio поддържа работа с няколко копия на околната среда (всяка със собствен набор от VSPackages). Копията използват различни раздели на регистъра за да съхранява конфигурацията на тяхното състояние и са диференцирани (различават се) по тяхното AppId (Application ID) Идентификационен номер на приложението. Копията са стартирани от AppId-specific .exe, което избира AppId, определя основния кошер (root hive) и стартира IDE. VSPackages са регистрирани за едно AppId и са свързани с други VSPackages за това AppId. Различните издания (едишъни) на продукти на Visual Studio са създадени с помощта на различни AppIds. На Visual Studio Express Edition продуктите се инсталират с техните собствени AppIds, но Standard, Professional и Team Suite продукти споделят същият Идентификационен номер на приложението. Следователно, може да се инсталира Express редом с други едишъни, за разлика от останалите, които обновяват една и съща инсталация. Професионалната версия включва множество VSPackages в стандартната версия и Team Suite включва множество VSPackages в двете други версии. AppId на системата е отнесена от Visual Studio Shell в Visual Studio 2008.

### 1.2.2 Visual Studio Code

Visual Studio Code е редактор на програмен код за Windows, Linux и OS X. Това е първият редактор на Microsoft, който може да се ползва под Linux и macOS. Поддържа богат набор от инструменти за разработване като дебъгване, вграден Git Control, IntelliSense, „Side-by-Side Editing“ (позволява работа едновременно върху 2 файла отворени един до друг) и др. Той също така дава възможност за персонализиране, което означава, че потребителите могат да променят темата на редактора, клавишните комбинации, настройките и др. Все още не е известно дали потребителите ще могат да разширяват функциите на редактора чрез разработване на различни разширения, както е във Visual Studio. Редакторът е продукт на Microsoft и е безплатен, публично достъпен за преглед.

Visual Studio Code е базиран на Electron, който е базиран на Chromium, използван да разгръща io.js приложения за десктопа. Visual Studio Code използва Blink layout engine, за да направи интерфейса.

Той е с отворен код и предлага богати възможности за персонализация и разширяемост, което го прави подходящ както за начинаещи, така и за напреднали програмисти. Все по-популярен е сред разработчиците, които работят с различни езици и технологии, включително JavaScript, TypeScript, Python, Java, и много други.

VS Code се отличава с интуитивен потребителски интерфейс, богат екосистем от разширения и интеграция с различни инструменти за разработка като Git и Docker. С поддръжка за множество операционни системи, включително Windows, macOS и Linux, той предоставя консистентно изживяване на всички потребители.

Visual Studio Code разполага с прост и лесен за използване интерфейс, който предлага интелигентни функции като автоматично довършване на кода и интуитивни команди за работа с файлове и папки. Има богата екосистема от разширения, със стотици разширения, които покриват широк спектър от нужди на програмистите, като синтаксис подчертаване за различни езици, инструменти за отстраняване на грешки и интеграция с различни сервиси. Има и вградена поддръжка за Git, която улеснява работата със системата за управление на изходния код, като предоставя възможности за преглед на промените, управление на бранчове и извършване на комити. Множество функционалности: Включващи работа с дебъгери, терминал, вградени команди за работа с файлове и папки, теми и цветни схеми за персонализация на интерфейса.

### 1.2.3 MySQL Workbench

MySQL Workbench е унифициран инструмент за дизайн на релационни бази данни с отворен код, поддържан от Oracle. Предназначен е за работа с MySQL бази данни, предоставяйки графичен интерфейс за управление на бази данни и инструменти за разработка и администриране.

MySQL Workbench разполага с:

* Дизайн на база данни: Позволява на потребителите да създават и моделират релационни бази данни с помощта на визуални инструменти за проектиране на таблиците, връзките и ключовете.
* Работа със скриптове: Предоставя възможност за изпълнение на SQL заявки и скриптове директно във вграденото средство за редактиране на код.
* Управление на бази данни: Позволява на потребителите да управляват сървъри, бази данни и таблици, включително създаване, редактиране и изтриване на обекти.
* Възможност за визуализация на данни: Предлага графичен преглед на структурата на базата данни и резултатите от заявките.
* Миграция на данни: Поддържа миграция на данни от различни източници, включително от Microsoft SQL Server, Microsoft Access и други.

## 1.3 Предимства пред други уеб сайтове

Разработката на уеб сайт предоставя редица предимства. Първо, индивидуално разработените уеб сайтове предлагат гъвкавост и персонализация, което гарантира по-добро съответствие на нуждите на клиента и по-добър контрол върху съдържанието. Второ, те осигуряват уникална идентичност и запомнящо се преживяване за потребителите, което може да допринесе за изграждането на конкурентно предимство. Друго предимство е възможността за гъвкаво разширяване с нови функции и модули, което подпомага постоянното развитие и растеж на уеб сайта.

Информационният уеб сайт за университети се отличава от други. Първо, той предлага цялостна и структурирана информация за университета, включително академични програми, курсове. Информационният уеб сайт предоставя удобен начин за представяне на университета пред потенциални студенти и за поддържане на връзка с партньори. Той предоставя и препратки към официалните страници на всеки университет в неговата база данни.

Няма особено много сайтове на български език, които могат да се сравнят с този като тематика. Количеството сайтове за образование в Българие е оскъдно, което е още едно предимство на сайта.

В сравнение с другите такива сайтове, този разполага с много качества, които го отличават, например:

* Модерен и добре-изглеждащ дизайн, който улеснява работата на потребителя като предоставя лесен и интуитивен начин за навигация, търсене на информация и използване на различните функционалности.
* Адаптивен дизайн за компютър и телефон.
* Функционалност за търсене на университети, която позволява да се филтрират всички висши заведения в базата данни по град и специалностите, които предлагат. Така потребителя може да намери точно къде се предлагат специалностите, от които той се интересува, както и какво предлагат университетите в областта, в която живее.
* Включва секции като форум и информация за контакти с администраторите на уеб сайта.

Тези фактори правят инфо а. Той предоставя много по-качествен начин за потребителя да се образова относно това какво предлагат университетите в страната като обучение и да намери точно това, което търси, отколкото други сайтове с тази информация.

# 

# Глава 2: Теоретична част

## 2.1 Използвани технологии и среди за разработка

В настоящия проект са използвани следните технологии, езици за програмиране и среда за разработка : HTML, CSS, PHP/MySQL, Bootstrap, XAMPP, MySQL Workbench и Visual Studio Code.

### 2.1.1 HTML(HyperText Markup Language)

HTML е основният маркиращ език за описание и дизайн на уеб страници. HTML е стандарт в интернет, а неговите стандарти се определят от международния консорциум W3C. Текущата версия на стандарта е HTML 5.0 (от 28 октомври 2014 г.), а предходната стабилна версия е HTML 4.1.

Описанието на документа става чрез специални елементи, наречени HTML елементи или техните маркери, които се състоят от тагове и съответстващите етикети (HTML tags) и ъглови скоби (като например елемента <html>). HTML елементите са основната градивна единица на кода, който изграждат уеб страниците. Чрез тях се форматира, графично оформя текста и неговите отделните части в рамките на една уеб страница, като например заглавия, цитати, текстови раздели, хипертекстови препратки и т.н. Най-често HTML елементите са групирани по двойки <h1> и </h1>.

В повечето случаи HTML кодът е написан в текстови редактори, с файлов формат .html, .htm, dhtml и се качва и хоства на сървъри, които са онлайн в интернет или са част от www мрежата. Тези .html файлове съдържат програмно на таговете на HTML и текстово съдържание със маркери и коментари – също инструкции за браузъра, за това какъв точно тип е .html страницата, а също за това как да се показва текстът, особено що се отнася до езиковите характеристики. За да се илюстрира как се включва текст в HTML код: <маркер> Някакъв текст. </край на маркера>. уеб браузърите са програмирани от своя страна така, в повечето случаи, макар че някои браузъри могат да имат съответно проблеми на версията, за да могат да прочетат HTML документите и да ги покажат на екрана като уеб страници. Браузърите не показват самите HTML тагове, освен ако не се отиде в менюто за да се направи това, така че те „интерпретират“ (тоест парсват) съдържанието на страницата като код и текст за да могат след работа на процесора да покажат желаното уеб съдържание.

Основното предимство на HTML е, че уеб страниците, които са го включват в кода си, могат да се разглеждат чрез показването им от браузъра на екрана на повечето устройства. Уебстраницата може да има дизайн, който дори изглежда с добър дизайн с помощта на CSS или „правилно оформен“ (например с помощта на C#), както върху монитора на персоналния компютър, но също и върху миниатюрния дисплей на пейджър или дисплея на мобилен телефон.

HTML може да прикрепя скриптове писани на езици като JavaScript, който е помощен за HTML, и това променя поведението на дадена уеб страница. Cascading Style Sheets (CSS) се използват, като това се прави за да се определя изгледа и оформлението на текста и други включени в страницата изображения и илюстриращи материали. World Wide Web Consortium (W3C) поддържа както HTML, така и CSS, и насърчава използването на CSS в HTML страниците още от 1997. Това допринася за разделяне съдържанието и структурата на уеб страниците от тяхното визуално представяне.

HTML 5 е най-новият проект на HTML стандарта, като той все още е в процес на разработка. Той въвежда много нови свойства. Някои от тях са <video>, <audio>, <canvas>. Основната цел на тези подобрения е да се улесни употребата на мултимедийни и графични елементи, без да е необходимо да се ползват външни апликации. Други нововъведения, като <section>, <article>, <header>, <nav>, целят да обогатят съдържанията на документите. Някои от досегашните елементи, като <a> и <menu>, са променени и стандартизирани.

HTML 5 е поддържан от по-стари браузъри, тъй като е направен така, че новите му функции просто да се игнорират от тях.

HTML 5 бързо набира популярност и проучванията показват, че през есента на 2011 година над 30 от 100-те водещи сайтове вече ползват HTML 5, а през лятото на 2013 година 153 от водещите 500 компании вече са имплементирали HTML 5 в сайтовете си.

### 2.1.2 CSS(Cascading Style Sheets)

CSS е език за описание на стилове (език за стилови файлове, style sheet language) – използва се основно за описание на онлайн представянето на уеб базиран документ, който написан на език за маркиране. Най-често се използва допълнително към чистия HTML, но се прилага и върху XML уеб страници и документи. Спецификацията на CSS официално се поддържа от W3C.

CSS още в началото на развитието на www започва да се добавя към стандартния HTML с цел да бъдат разделени съдържанието и структурата на уеб страниците отделно от тяхното визуално представяне. Преди стандартите за CSS, установени от W3C през 1995 г., съдържанието на сайтовете и стила на техния дизайн са писани в една и съща HTML страницата. В резултат на това HTML кодът се превръща в сложен и нечетлив, а всяка промяна в проекта на даден сайт изисквала корекцията да бъде нанасяна в целия сайт страница по страница. Използвайки CSS, настройките за форматиране могат да бъдат поставени в един-единствен файл и тогава промяната ще бъде отразена едновременно на всички страници, които използват този CSS файл.

SS позволява да се определя как да изглеждат елементите на една HTML страница – шрифтове, размери, цветове, фонове, и др. CSS кодът се състои от последователност от стилови правила, всяко от които представлява селектор, последван от свойства и стойности

Селекторите в CSS се използват като наименуват типа селекция и покажат съответно към кои елементи на HTML документа трябва да бъде прилаган съответният стил. Съществуват много видове селектори. Някои селектори позволяват постигане и на динамичност на страницата, макар и в определена степен. Например само с помощта на CSS могат да бъдат направени падащи (drop-down) или пък изскачащи менюта (за мобилни устройства), и особено за хипервръзки, при които може да се задава цвета на линка, и също при посочване на линка, той да променя цвета си и др.

Свойство представлява конкретна характеристика на елемент от HTML документа, а стойност е стойността, която се дава на това свойство. Например на свойство за размер може да се даде стойност число в дадена мерна единица, а за свойство цвят – стойност, която представлява стандартното наименование на цвета или hex стойности. от рода на #ffffff (бяло) до #000000 (черно).

Съществуват 3 основни версии на езика CSS + една, която още не е поддържана от основните браузъри. За всяка от тях е разработен специален тест, чрез който може да се провери дали даден web браузър поддържа съответната версия на езика. Всяка следваща версия разширява възможностите на езика. Могат да се използват и разширения на CSS (Sass, Less) или фреймуърци(Foundation Zurb), които улесняват работата и позволяват допълнителна функционалност като наследяване, задаване на променливи и др.

### 2.1.3 PHP(Hypertext Preprocessor)

PHP е скриптов език със синтаксис, базиран на C и Perl. Използва се предимно в интернет среда за изпълнение на широк кръг от услуги и е един от най-популярните езици за програмиране в интернет.

PHP се разпространява под отворен лиценз (PHP License), който по своята същност е BSD лиценз и който позволява безплатно разпространяване на програмния код на интерпретатора на езика, както и създаването на производни интерпретатори под други лицензи с уговорката, че тези интерпретатори не могат да включват PHP в името си. Фактът, че PHP се разпространява свободно, го прави удачен избор за изграждане на уеб сървър, базиран изцяло на свободни продукти – GNU/Linux, Apache, MySQL/PostgreSQL и др.

При поискване кодът, който е написан на PHP, се интерпретира от уеб сървъра, на който е качен, и резултатът се връща на уеб браузъра. Потребителят не може да види чистия PHP код, без да има достъп до самия файл, в който той е записан. По този начин се осигурява защитата. PHP файловете могат да съдържат текст, HTML, CSS, JavaScript и PHP код и имат разширение \*.php.

Самият език е преносим на много изчислителни архитектури и операционни системи като GNU/Linux, UNIX, macOS, Windows.

### 2.1.4 MySQL

MySQL е многопоточна, многопотребителска, SQL система за управление на бази данни (СУБД) с повече от шест милиона инсталации. MySQL AB разпространява MySQL като свободен софтуер под GNU General Public License (GPL), но също така под традиционните за комерсиален софтуер лицензи за случаи, когато използването е несъвместимо с GPL. Подобно разпространение е известно като двойно лицензиране.

MySQL се разработва, разпространява и поддържа от Шведската компания MySQL AB, която държи авторските права за голяма част от програмния код. Подобен е моделът на JBoss, а също и начинът, по който Free Software Foundation работи с авторските права по нейните проекти в отличие от Apache проекта, където софтуерът е обществено разработван, а авторските права за програмния код принадлежат на индивидуалните автори.

Компанията разработва и поддържа системата, продава поддръжка и сервизни договори, както и комерсиални лицензи за MySQL, и наема хора от целия свят, които работят съвместно с помощта на интернет. MySQL AB е основана от Давид Аксмарк, Алан Ларсон, и Михаел „Монти“ Видениус през 2001.

Достъпни са програмни интерфейси, позволяващи програми, написани на различни програмни езици да имат достъп до MySQL бази данни. Такива са: C, C++, C#, Delphi (чрез dbExpress), Eiffel, Smalltalk, Java (с директна поддръжка), Lisp, Perl, PHP, Python, Ruby, REALbasic (Mac), FreeBasic, и Tcl, като всеки от тях има специфичен програмен интерфейс. Интерфейс тип ODBC наречен MyODBC позволява на други програмни езици, които поддържат ODBC интерфейс да комуникират с MySQL база данни.

MySQL е популярна за интернет приложения като MediaWiki или Drupal и е база данни в LAMP, MAMP и WAMP платформи, и за софтуера с отворен код Bugzilla – приложение за проследяване на грешки. Нейната популярност като Интернет приложение е тясно свързана с популярността на PHP, като често, комбинирани с MySQL, са наречени Динамичното дуо. Лесно е да се намерят много източници, които ги комбинират в интернет статии или книги (PHP and MySQL for Dummies, PHP and MySQL Bible, Beginning PHP and MySQL, и други). В тези книги се твърди, че MySQL е по-лесен за изучаване от други бази данни.

### 2.1.5 Bootstrap

Bootstrap е платформа за уеб разработка, която се използва за създаване на модерни и адаптивни уеб сайтове и уеб приложения. Bootstrap представлява HTML, CSS и JavaScript библиотека, която предоставя набор от готови компоненти, стилове и скриптове, които могат да се използват за бързо и лесно разработване на уеб интерфейси.

Предимствата на Bootstrap включват бързото прототипиране на уеб сайтове и приложения, респонсивния дизайн, крос-браузърната съвместимост и гъвкавостта и модуларността на платформата. Този фреймуърк се използва широко от уеб разработчиците по целия свят поради своите множество предимства и лесното му използване.

### 2.1.6 Xampp

XAMPP е безплатен и свободен софтуерен пакет, който съдържа Apache HTTP Server, MySQL база от данни и други необходими инструменти за използване на езиците за програмиране PHP (с PEAR) и Perl. Програмата се разпространява с GNU General Public License и служи за безплатен и лесен за употреба уеб сървър, който има възможност да борави с динамични страници. XAMPP има версии за Windows, Linux, Sun Solaris и Mac OS X. Отличава се с много лесна и бърза инсталация.

Името на XAMPP е акроним от X (коя да е от четирите операционни системи), Apache, MySQL, PHP и Perl.

XAMPP се обновява често, за да включва в себе си последните разработки на Apache/MySQL/PHP и Perl. В пакета са включени и няколко допълнителни модула като OpenSSL, FTP-Server ProFTPd, Webalizer и phpMyAdmin.

### 2.1.7 Visual Studio Code

Visual Studio Code (VS Code) е редактор на програмен код, разработен от Microsoft, който е наличен за операционните системи Windows, Linux и macOS. Той се отличава с лекота, мощ и гъвкавост, като предоставя богати възможности за разработка на софтуер. Включва вградена поддръжка на много езици за програмиране, интеграция с Git за контрол на версиите, автоматично завършване на кода, дебъгери и много други разширяеми функционалности чрез плъгини.

VS Code е популярен сред програмистите заради своята бързина, отзивчивост и обширната общност от разработчици, която предлага множество разширения и теми за персонализация. Той се използва за разработка на уеб приложения, мобилни приложения, десктоп приложения и други видове софтуерни проекти.

### 2.1.8 MySQL Workbench

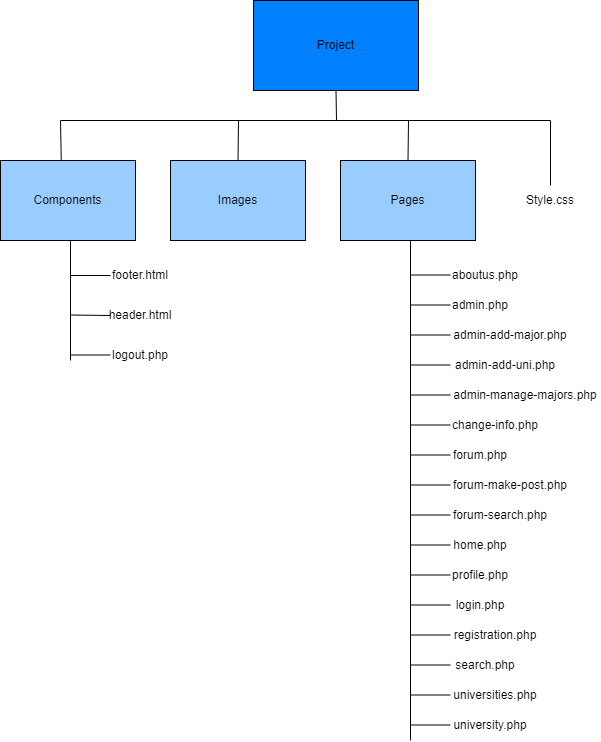
MySQL Workbench е интегриран визуален инструмент за проектиране на бази данни, разработен за архитекти на бази данни, програмисти и администратори на бази данни. Този софтуер предоставя набор от инструменти за моделиране на данни, разработка на SQL заявки, администриране на бази данни и поддръжка на бази данни през единен потребителски интерфейс.

MySQL Workbench позволява създаване на модели на бази данни чрез вградения си инструмент за моделиране на данни, който позволява на потребителите да създават и управляват различни обекти на базата данни като таблиците, връзките между тях и ограниченията на данните. Освен това, интегрираните инструменти за SQL разработка позволяват на програмистите да създават, редактират и изпълняват SQL заявки и скриптове директно от средата на Workbench.

MySQL Workbench е също така предназначен за администриране на бази данни, като предоставя функционалности за управление на потребители, настройка на сървърни параметри, мониторинг на работната нагрузка и изпълнение на рутинни административни задачи.

## 

## 2.2 Файлова структура на приложението

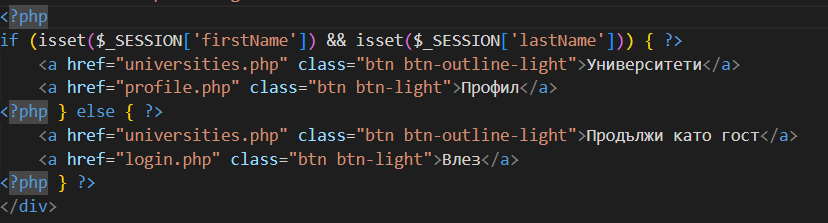
****

*Фигура 2.1 - Блок схема на файловата структура на приложението*

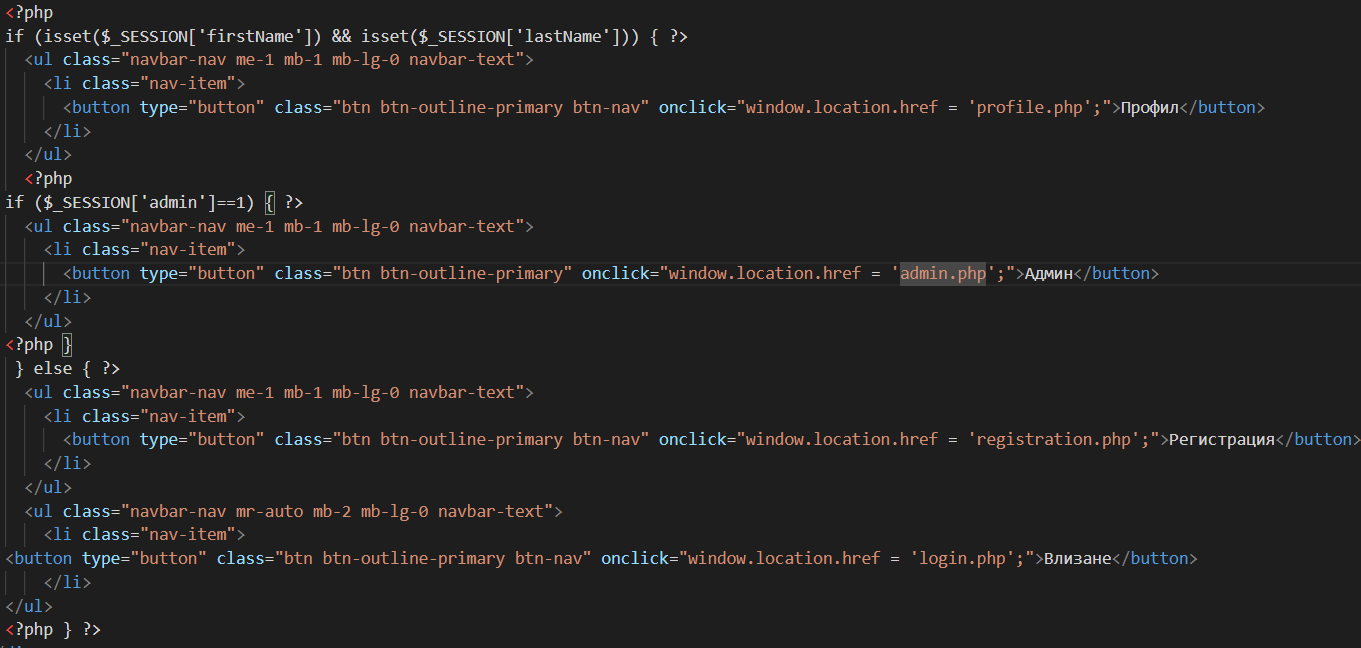
# Глава 3: Инженерно решение на поставената задача

## 3.1 Описание на кода

За изпълнение на задачата избрах да пиша във Visual Studio Code. Когато потребителят влезе в сайта, първото нещо, което вижда, е начална страница, която показва различни бутони в зависимост от това дали потребителят е влязъл в своя профил, или не. Това става чрез if проверка за това дали има зададени първо име и фамилия в сесията с php(фиг. 3.1). Същата проверка съществува и в header-а на страницата.



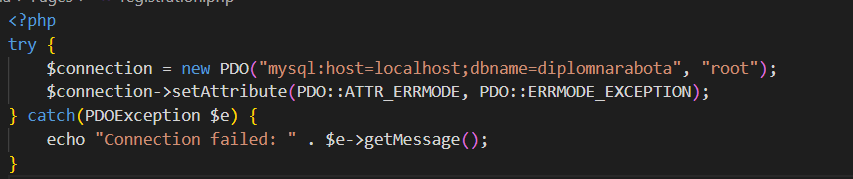
*Фигура 3.1 Проверка за сесия*



*Фигура 3.2 Проверка за сесия*

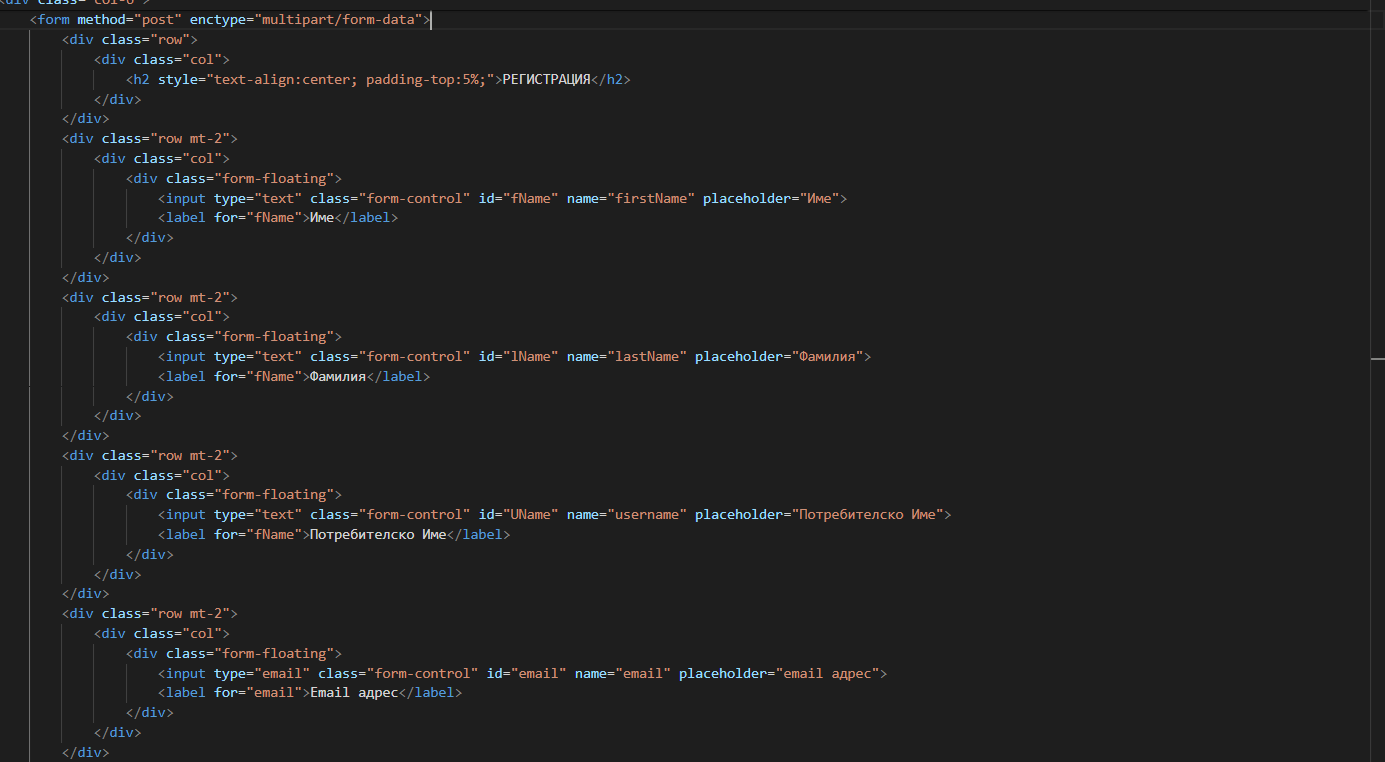
От тази страница потребителя бива пренасочен или към главната страница, или, в зависимост от това дали е влязъл в профила си, или към страницата за влизане или страницата с информация за неговия профил.

Страницата за регистрация и входната страница са направени чрез HTML форми и връзка с MySQL база данни чрез PHP(фиг. 3.3).



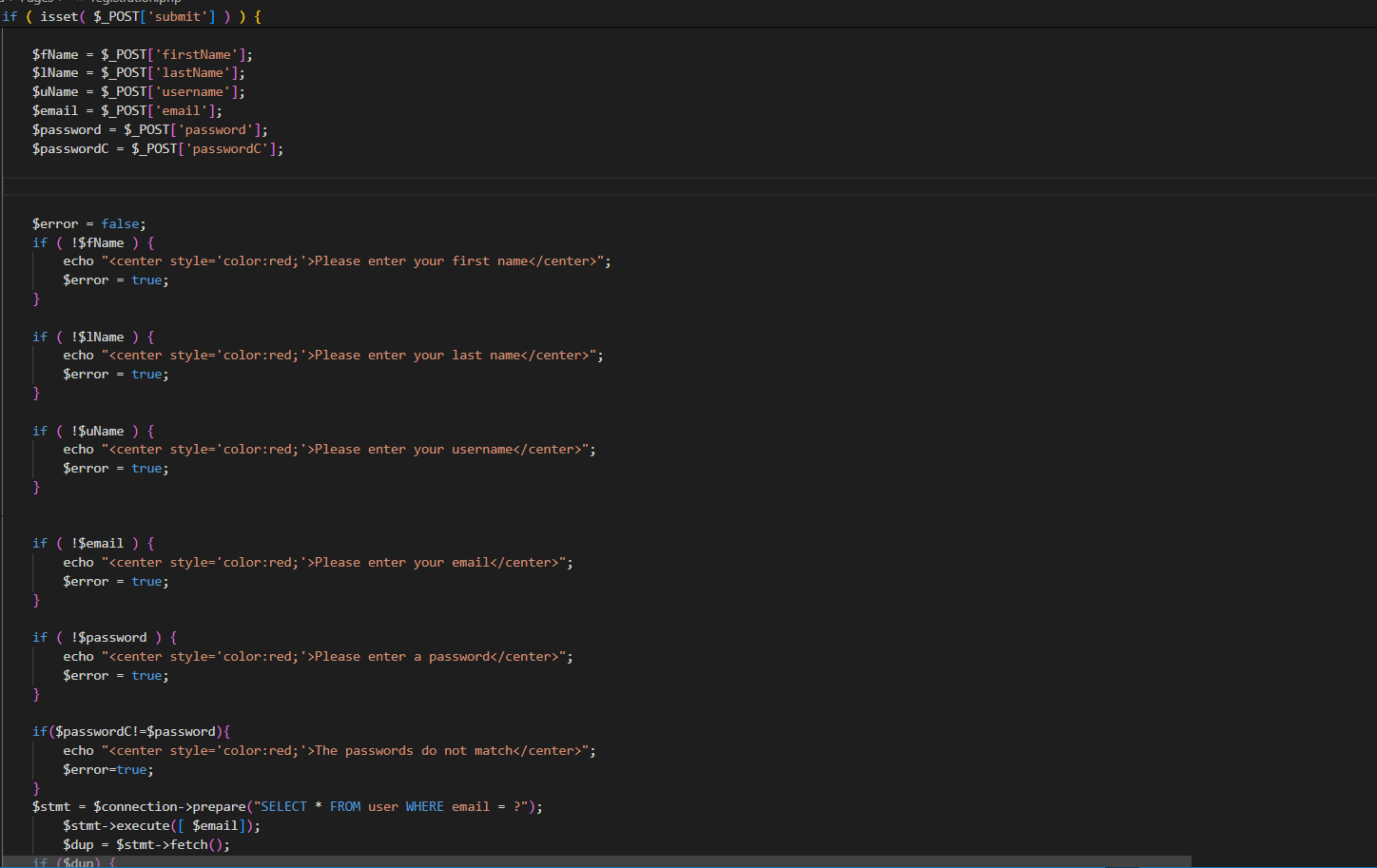
*Фигура 3.3 Връзка с базата данни за проекта*

HTML формата(фиг. 3.4.1 и фиг. 3.4.2 е дефинира с метод POST. От там в бекенда на страницата(фиг. 3.5.1 и 3.5.2) за всяко поле от формата се дефинира променлива, която съдържа въведената от потребителя информация чрез атрибута name на всеки ред от формата. Извършва се проверка за всяко поле дали е празно, както и за това дали имейла на потребителя вече съществува в базата данни(фиг. 3.6), след което се прави INSERT заявка в базата данни(фиг. 3.8).

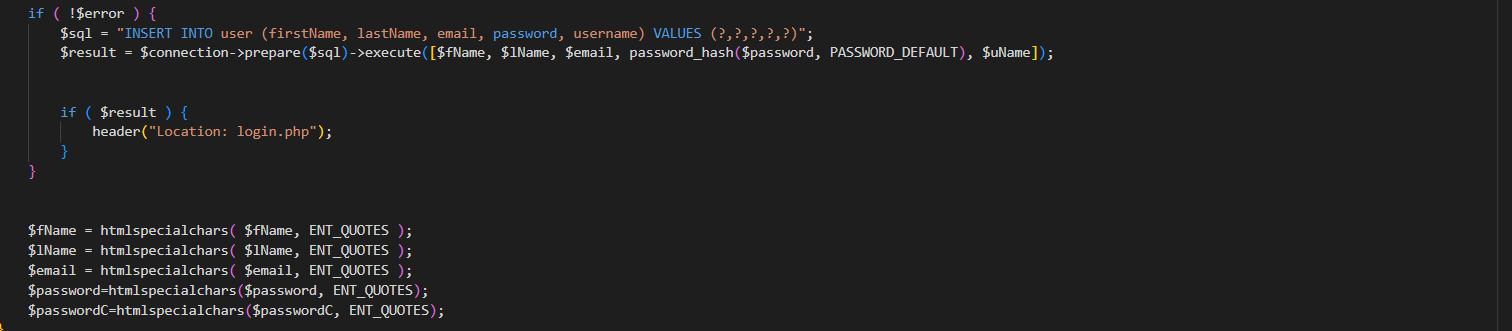
*Фигура 3.4.1 HTML форма за регистрация*



*Фигура 3.4.2 HTML форма за регистрация*



*Фигура 3.5.1 Back-end на формата за регистрация*



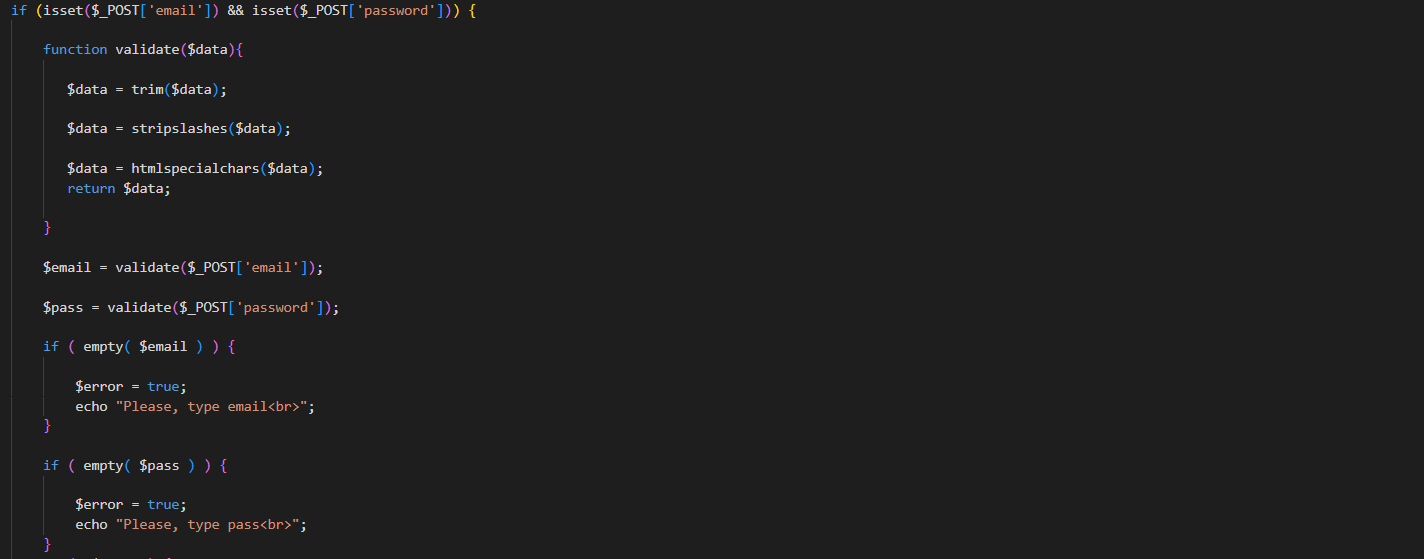
*Фигура 3.5.2 Back-end на формата за регистрация*

При вкарване на информацията в базата данни, паролата се криптира чрез хеширане, за да може администраторите и всеки, който получи достъп до базата данни, да няма достъп до паролите на потребителите. Това е важно за сигурността, тъй като много хора използват една и съща парола на повече от едно място.

В страницата за вход отново е дефинира HTML форма с метод POST с полета за имейл и парола(фиг. 3.6). В бекенда на страницата(фиг. 3.7.1 и 3.7.2) се дефинират променливи за двете полета, след което се оформя текста и се извършват проверки за валидация на профила в базата данни. При валидацията на профила се използва метод password\_verify(), която сравнява въведената парола от формата с хешираната от базата данни. При успешно влизане се създава сесия, в която се дефинират полета с информацията за потребителя от базата данни, за да може да се ползва по време на сесията и потребителя се препраща към главната страница. Сесия се стартира в началото на всяка страница.



*Фигура 3.6 HTML форма за регистрация*

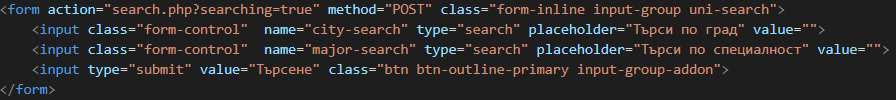


*Фигура 3.7.1 Back-end на страницата за вход*



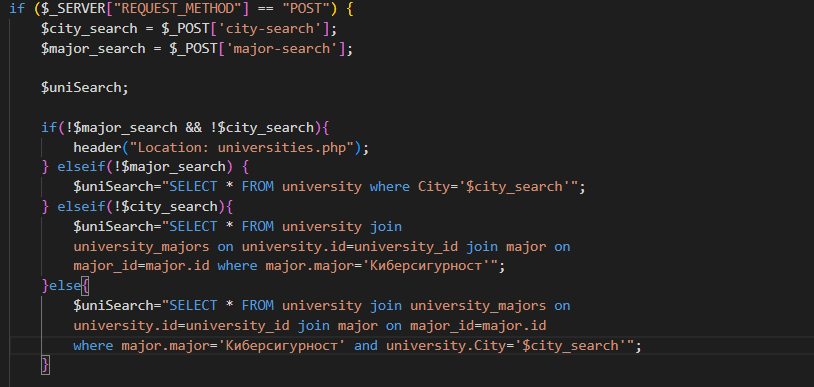
*Фигура 3.7.2 Back-end на страницата за вход*

Когато потребителят влезе във своя профил, той бива пренасочен към основната страница на сайта - тази, която показва всички университети. В нея отново се прави връзка с базата данни, като именно от нея се изкарва информацията за университетите. Страницата разполага и търсачка. Потребителят има възможност да филтрира през университетите от базата данни по град и специалност. Това става чрез форма(Фиг. 3.8), която изпраща потребителя към отделна страница, която изкарва резултатите от търсенето.



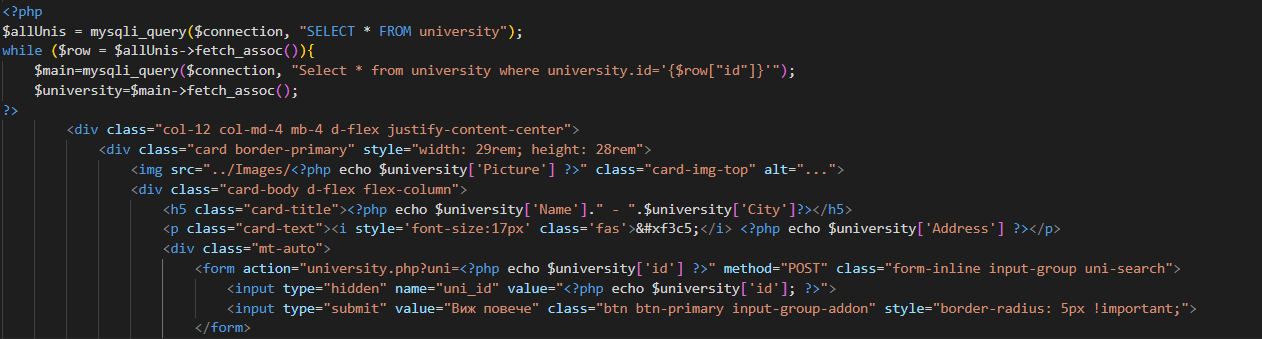
*Фигура 3.8 HTML форма за търсене*

В самата страница за търсене се получават въведените от потребителя данни и се записват в променливи. Извършва се проверка дали и двете полета са запълнени, и в зависимост от това дали са запълнени и ако не са, кое от двете е запълнено се дефинира заявка, която да изкара резултатите от търсенето:



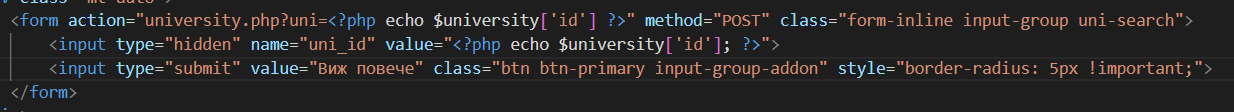
*Фигура 3.9 Back-end на страницата с резултатите от търсенето*

Когато полетата за търсене са празни, потребителя се връща на страницата с всички университети. Функционалността на тази страницата е чрез for цикъл да изкара всички входове в таблицата за университети в стилизираната страница чрез SQL заявки. Първо се създава променлива в цикъла, която съдържа индекса на елементите, след което се обработва SQL заявката и се създава нова променлива - масив, който съдържа всички колони на таблицата. Чрез тази променлива се изкарва нужната информация в HTML кода.

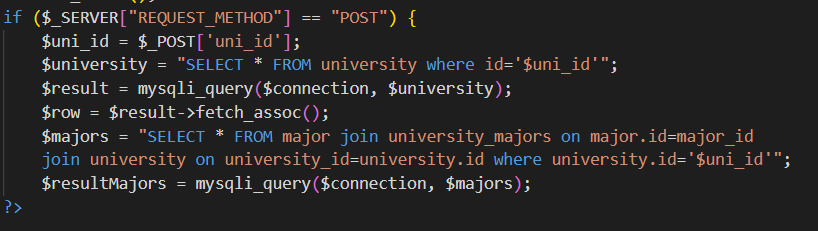


*Фигура 3.10 Изкарване на университетите от базата данни*

От тази страница потребителя може да достъпи и отделна страница, която изкарва информация за самия университет, на който е кликнал. Това става чрез POST форма(Фиг. 3.11), която изпраща id-то на университета към другата страница.

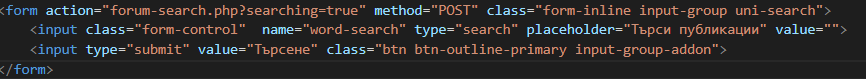


*Фигура 3.11 Форма за id на университета*

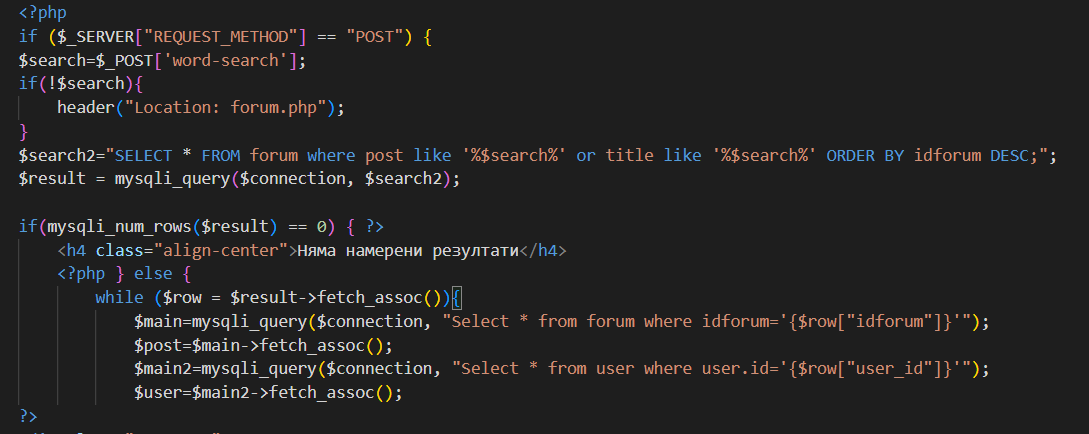
**

*Фигура 3.12 Back-end изкарване на информация за избрания университет*

При регистрация и влизане в потребителски профил, потребителят получава достъп до форум. Форумът се състои от 3 страници - една, която показва всички публикации, една с резултати от търсене и една за писане на публикации. На основната страница на форума има форма(Фиг 3.13), подобна на тази за търсене на университетите(Фиг 3.8). Бекенда за изкарване на резултати(Фиг 3.14) използва същата логика, която се използва и за изкарване на резултати при филтриране на университетите(Фиг. 3.9).

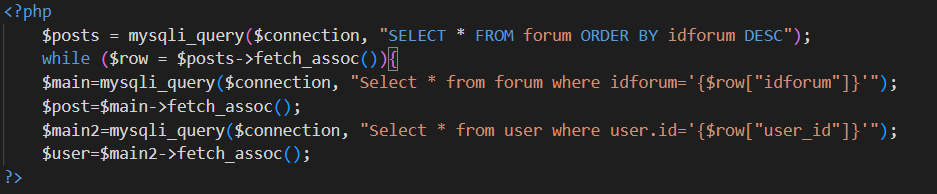


*Фигура 3.13 Форма за търсене във форума*

**

*Фигура 3.14 Изкарване на резултати от търсене във форума*

За публикациите във форума има отделна таблица в базата данни. Тя се обхожда чрез for цикъл в основната страница на форума и от нея се изкарват всички публикации, които са били качвани някога във форума, почвайки от последните качени. Изкарват се чрез SQL заявка.



*Фигура 3.15 Изкарване на всички публикации във форума.*

За качване на публикации за форума, потребителя отваря друга страница, в която има форма за писане на публикацията (Фиг 3.16). При изпращане на формата, данните от нея се запазват в променливи, чрез които се генерира SQL заявката за вкарване на публикацията в базата данни.

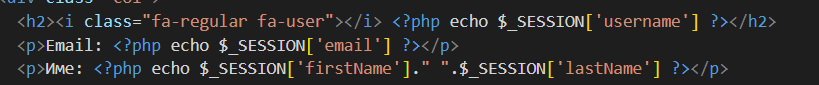


*Фигура 3.16 HTML форма за качване на публикация*

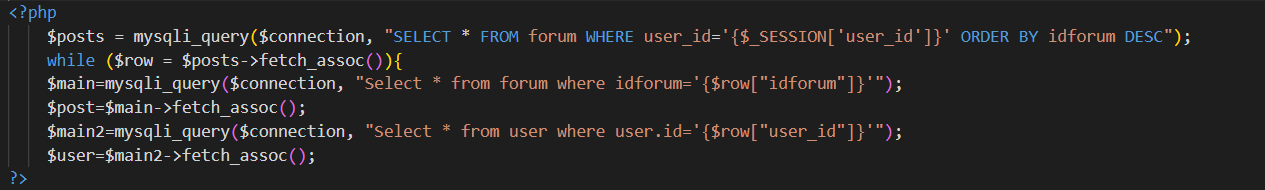
**

*Фигура 3.17 Back-end на формата за качване на публикация*

В потребителския интерфейс има и страница с информацията от профила на потребителя. В нея от променливите, дефинирани в сесия(Фиг. 3.7.2) се изкарва информация за потребителя и неговите публикации във форума.

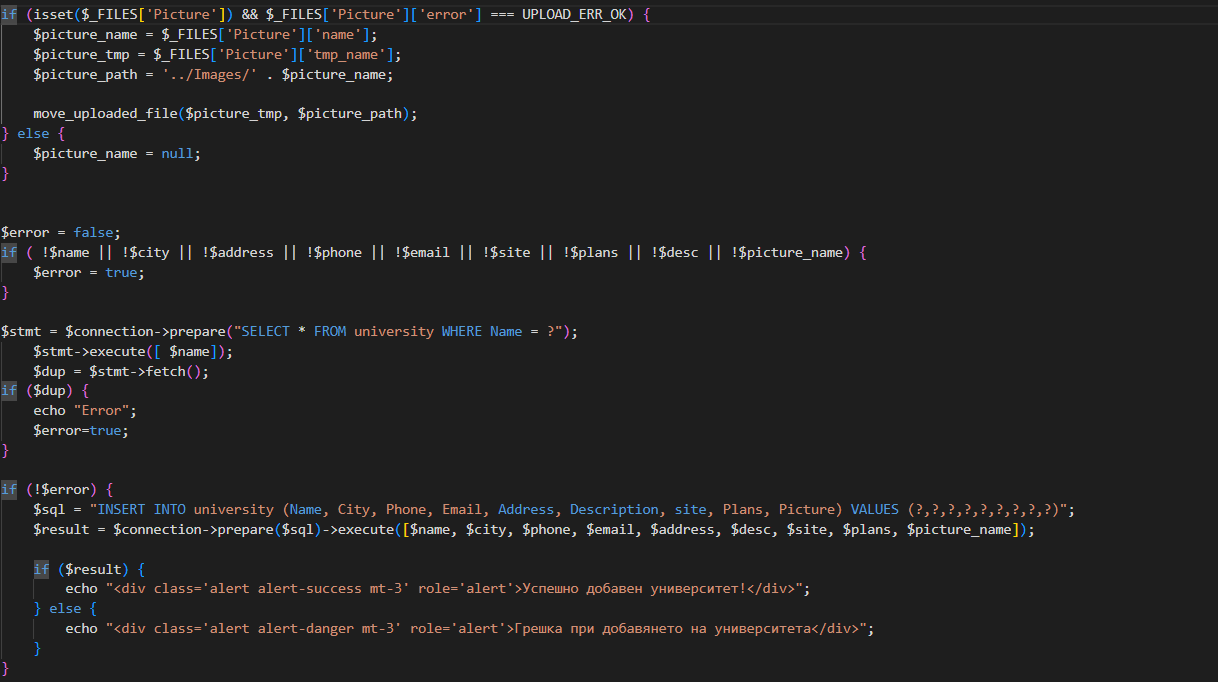


*Фигура 3.18 Изкарване на информация за потребителя от сесия*

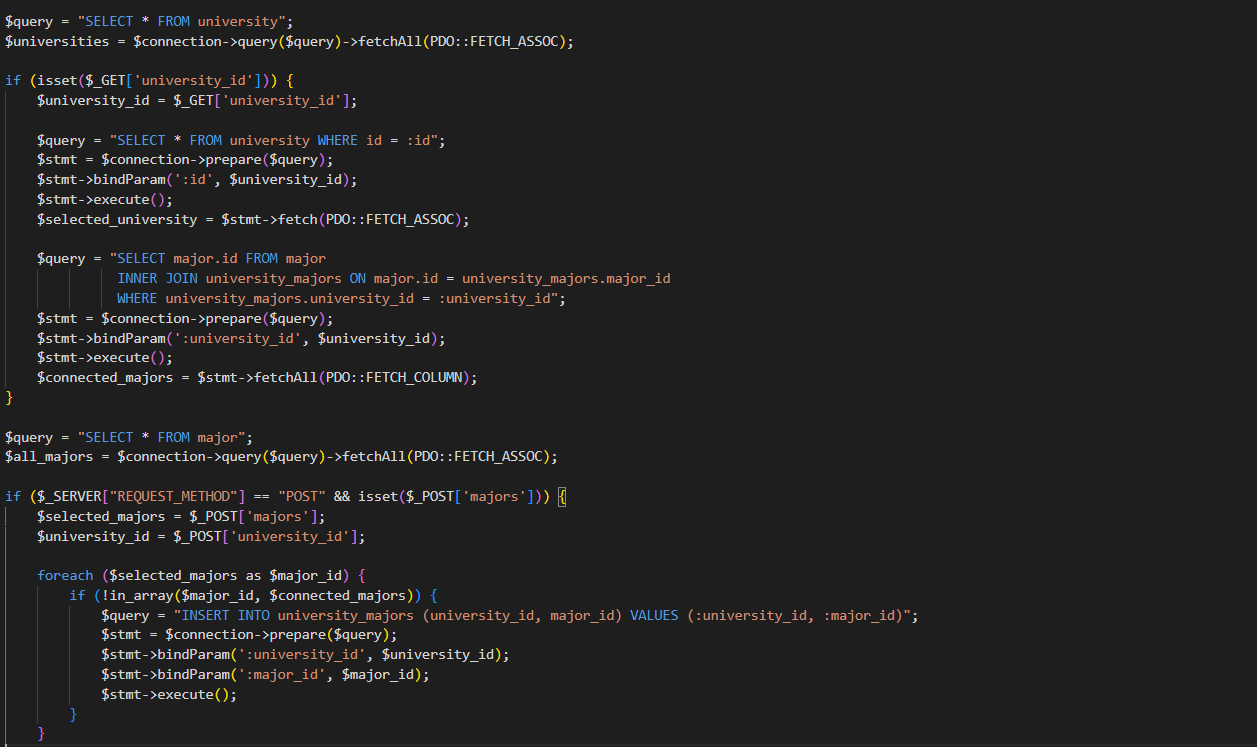
**

*Фигура 3.19 Изкарване на публикациите на потребителя от базата данни.*

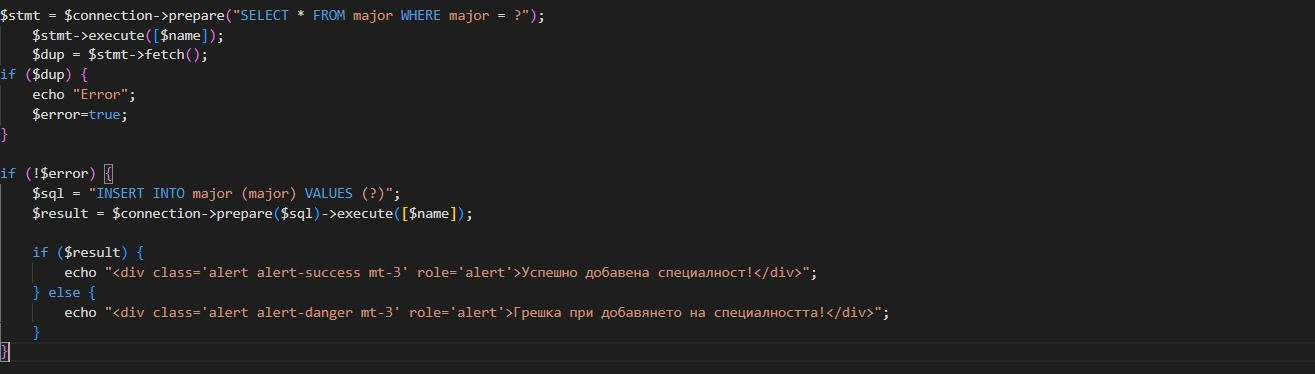
Освен с потребителския интерфейс, сайтът разполага и с администраторски панел, който служи за добавяне на университети(Фиг. 3.20), добавяне на специалности(фиг. 3.22) и добавяне на специалности към университет(Фиг. 3.21), отново чрез форми.



*Фигура 3.20 Вкарване на университет в базата данни*



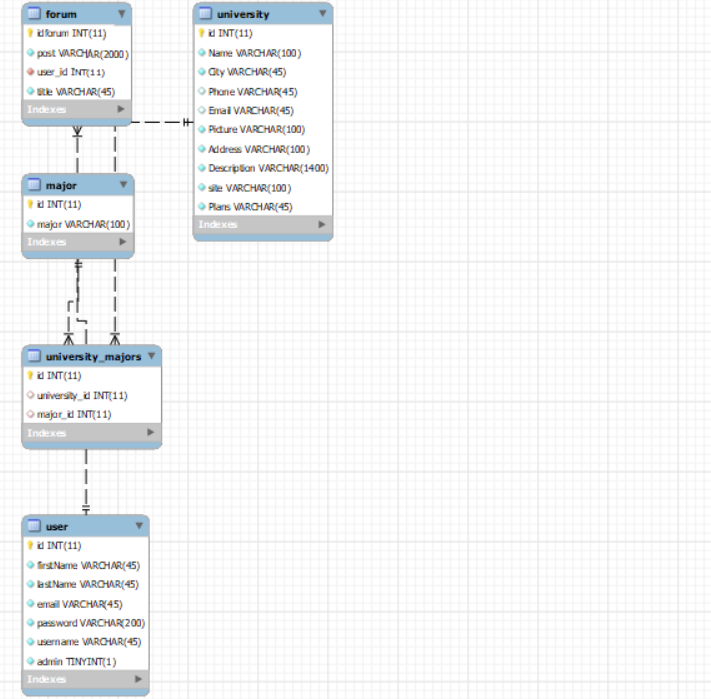
*Фигура 3.21 Свързване на специалности с университет*



*Фигура 3.22 Вкарване на специалност в базата данни*

## 3.2 Описание на база данни

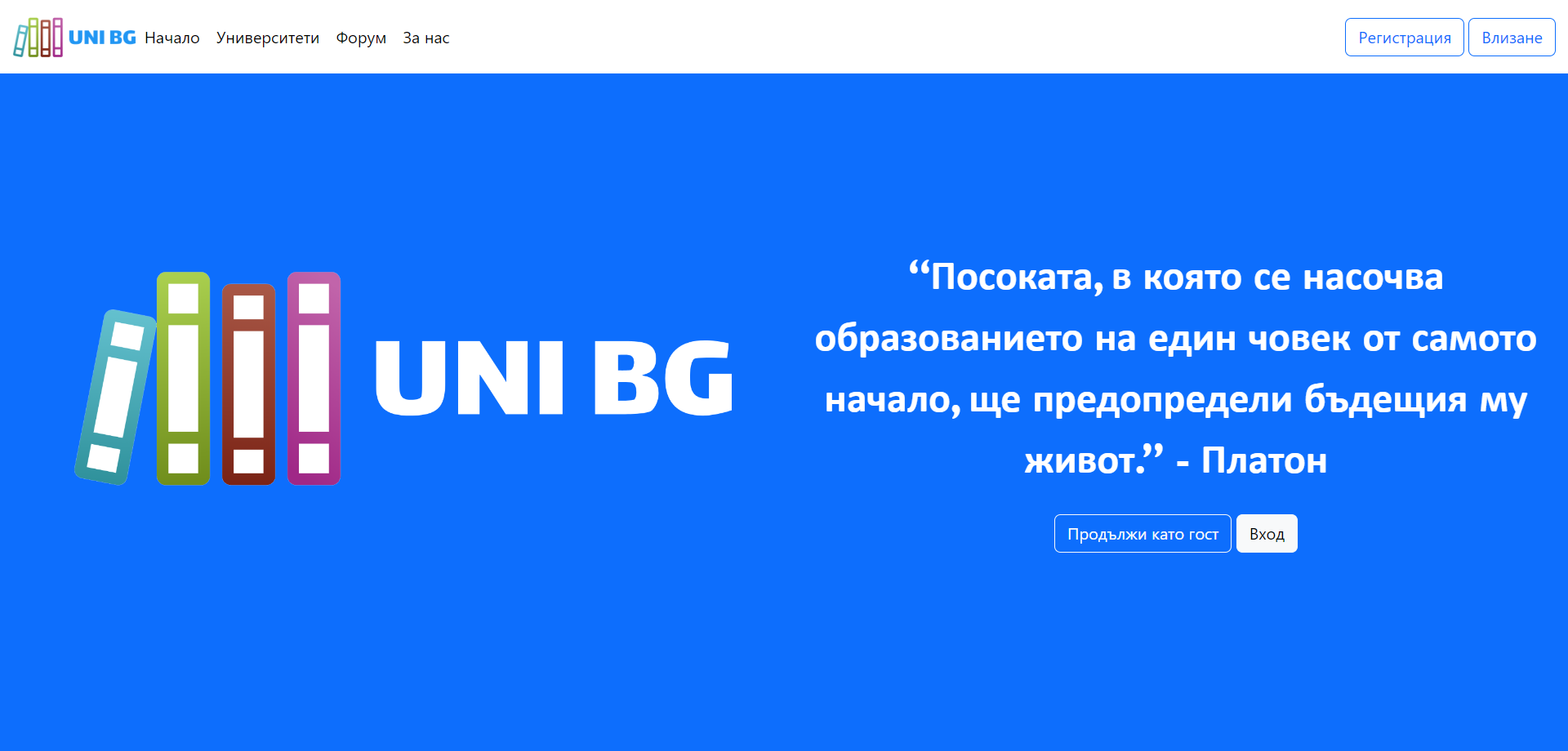
В базата данни на приложението има 5 таблици. Първата е за потребители, в нея има колони за първо име, фамилия, имейл, парола и потребителско име. Също и колона за това дали са администратор или не, която може да има само 2 стойности - 0 и 1. Следващата таблица е тази за университетите. В нея се съдържа цялата информация за университета, която излиза на сайта. Тя е свързана с таблицата за специалности чрез свързваща таблица, която съдържа само id на университет и на специалностите. Самата таблица със специалности има поле само за името на самата специалност. Последната таблица в базата е тази, която съдържа публикациите във форума. В нея има полета за самата публикация и за нейното заглавие, и външен ключ към таблицата с потребители, чрез който се изкарва потребителското име на автора на публикацията.



*Фигура 3.23 Изкарване на публикациите на потребителя от базата данни.*

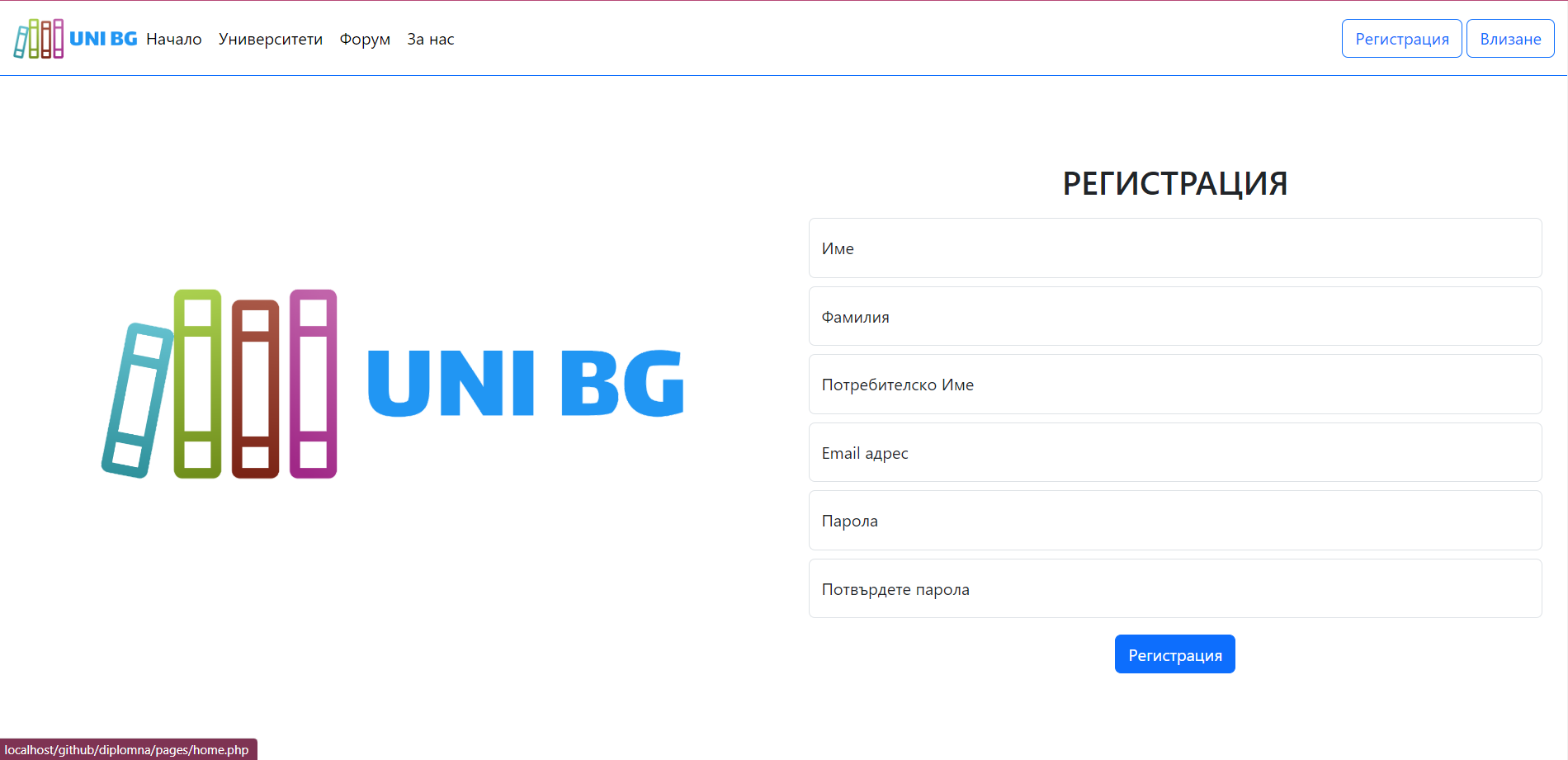
# Глава 4: Анализ на получените резултати

При отваряне на сайта, първото нещо, което потребителят вижда е начална страница(Фиг. 4.1), която го подтиква да влезе в профила си. На нея е показано логото на сайта до цитат и препратки към страницата за влизане и, в случаи че е решил да продължи без профил, към основната страница.

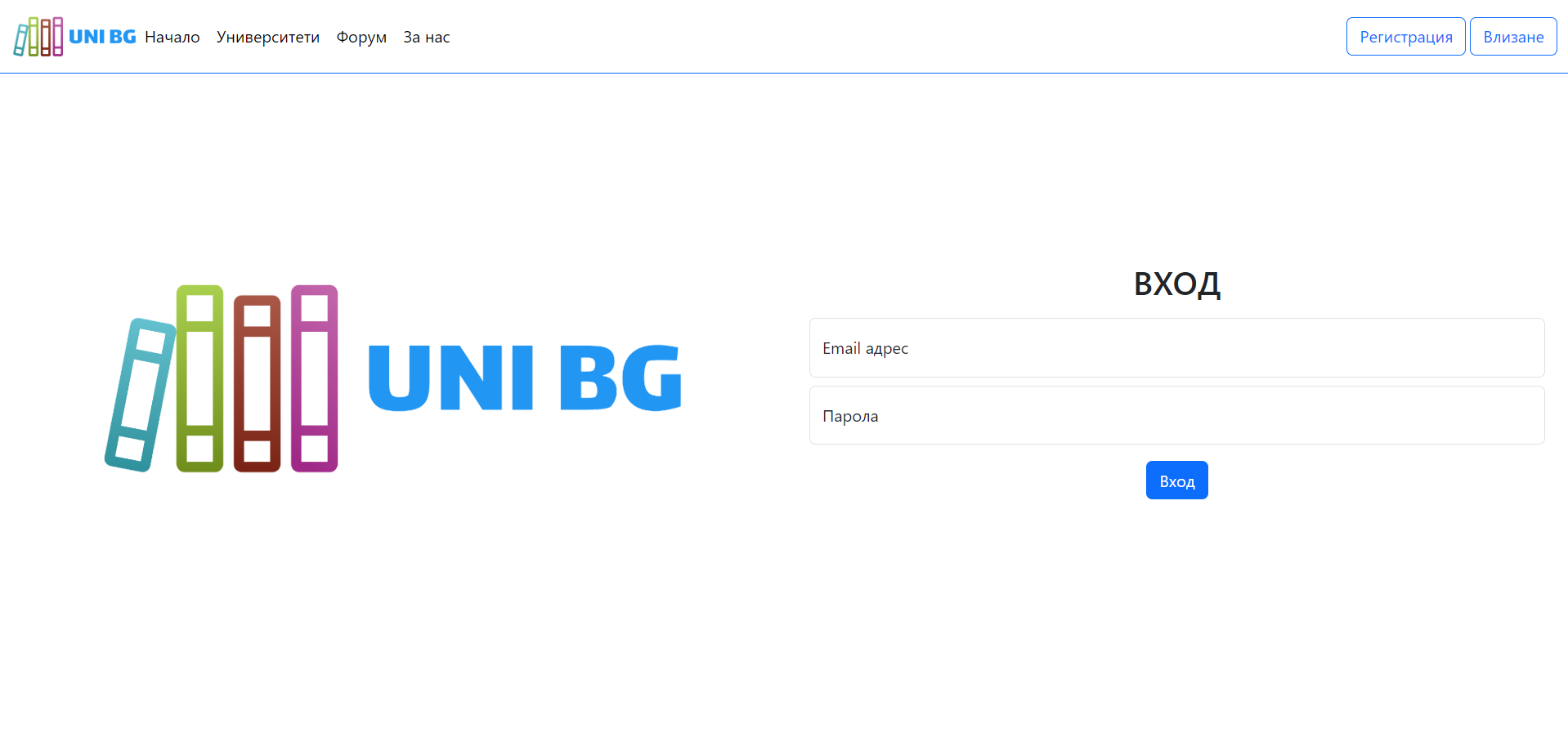


*Фигура 4.1 Начална страница*

Ако потребителят избере вход, той бива пренасочен към страницата за влизане(Фиг. 4.3). От там ако няма профил може да влезе в страницата за регистрация(Фиг. 4.2), която след успешно регистриране го препраща отново към страницата за вход.

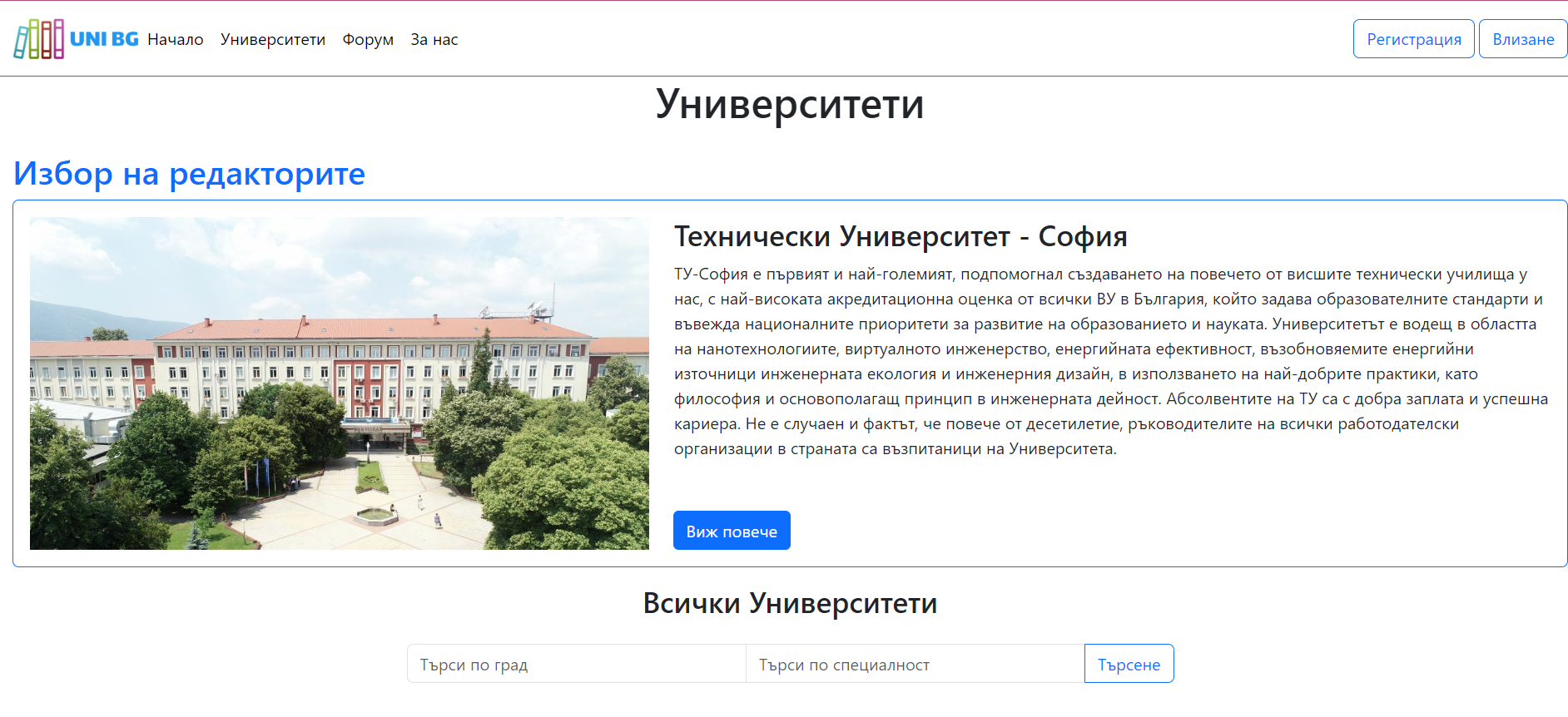


*Фигура 4.2 Страница за регистрация*

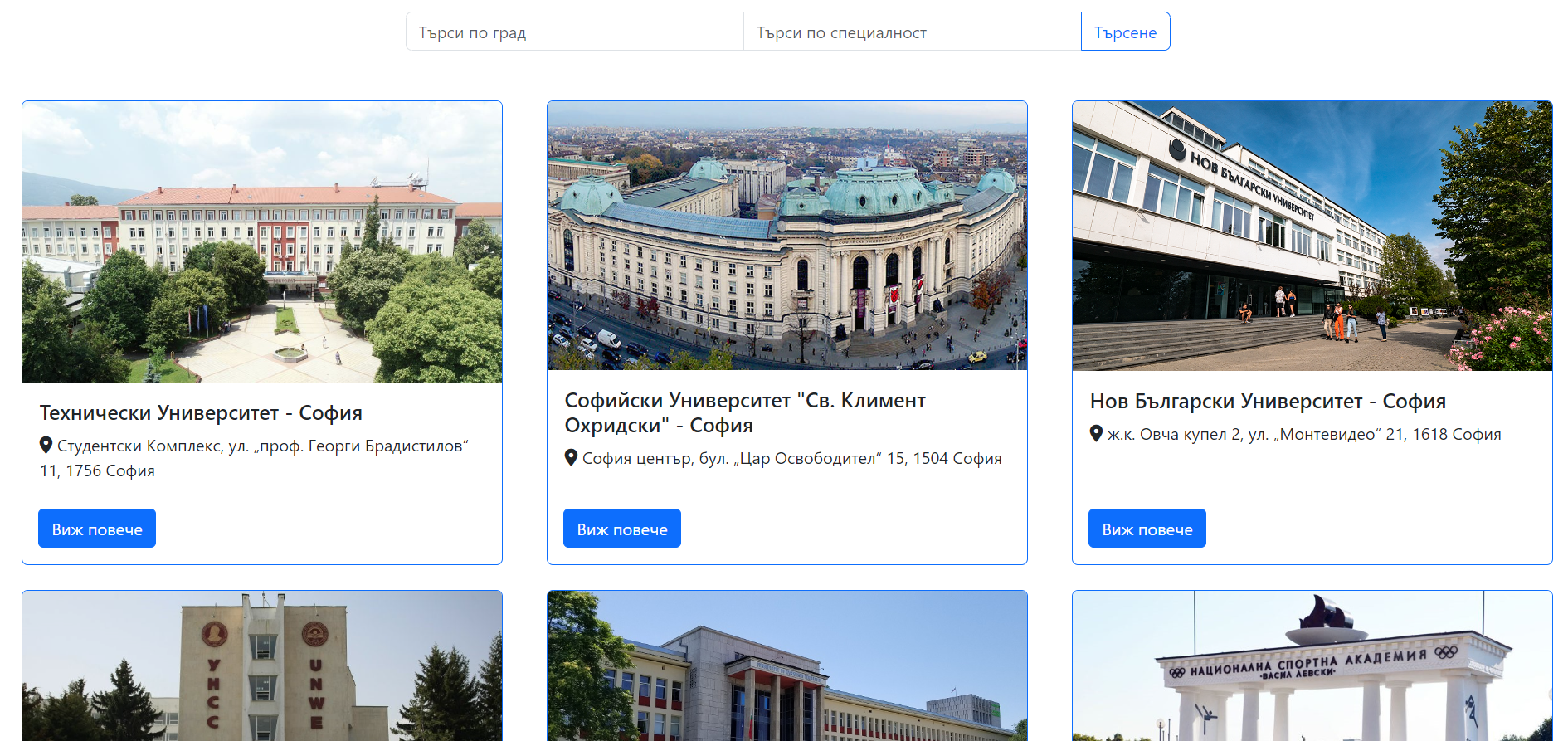


*Фигура 4.3 Страница за вход*

Когато потребителят влезе в профила си, той бива препратен към основната страница. Това е страницата, която съдържа лист с всички университети. На нея най-отгоре има избор на редакторите, който е университетът, който препоръчват администраторите, под него има търсачка и под нея са всички университети.

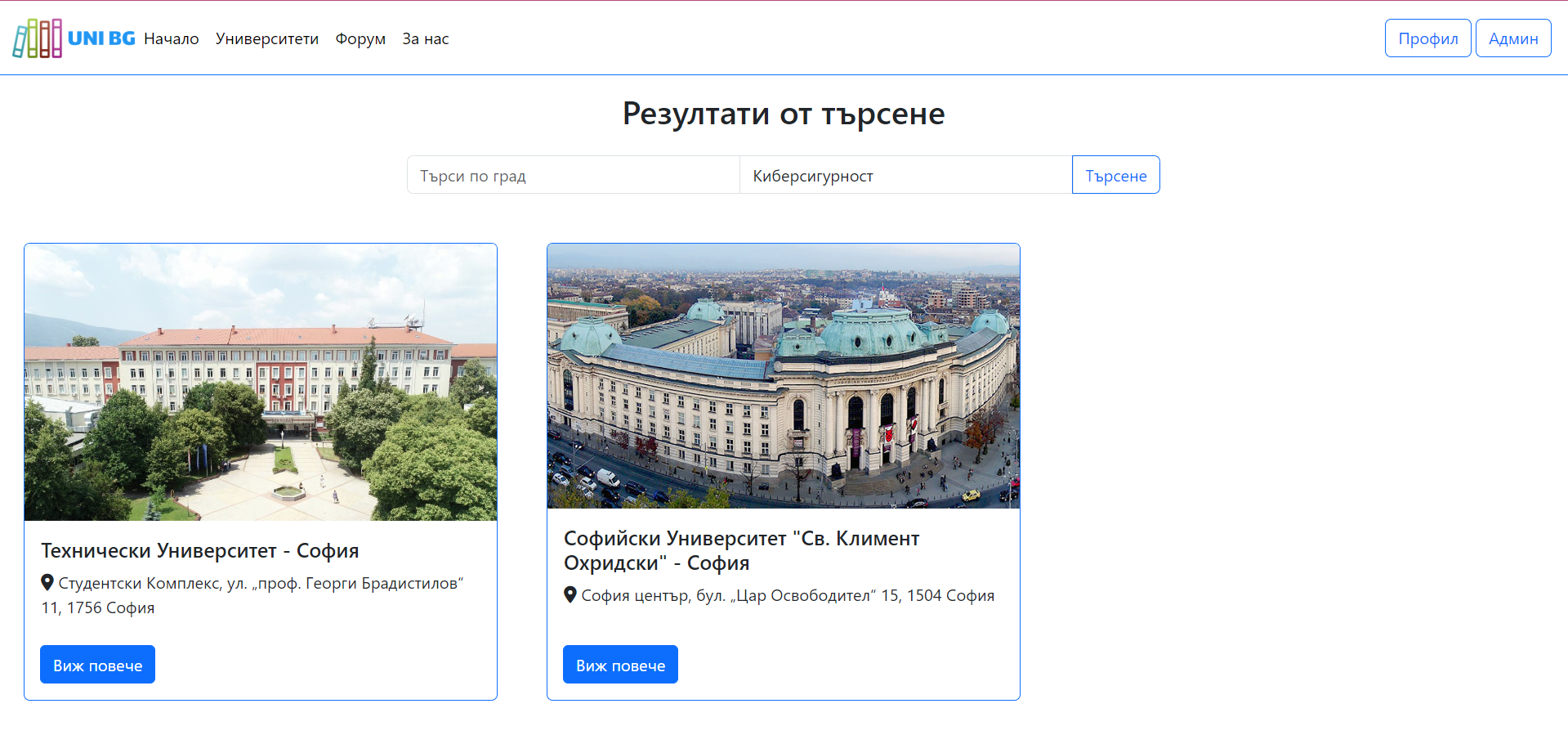


*Фигура 4.4.1 Страница с университети*



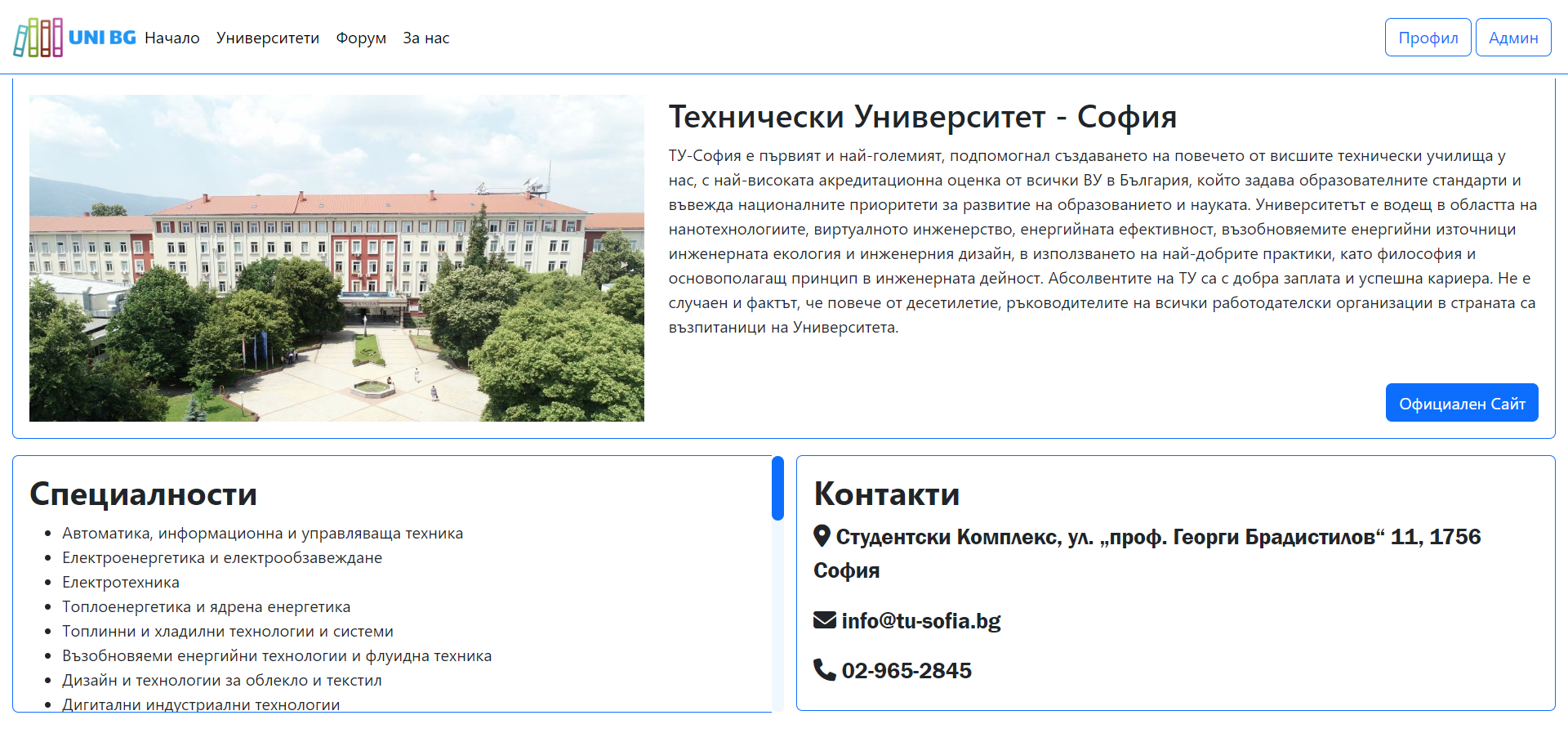
*Фигура 4.4.2 Страница с университети*

При търсене чрез търсачката потребителят се препраща към друга страница, на която вижда резултатите от търсенето. Например, се търси с въведена ключова дума “киберсигурност”, излизат само университетите, които предлагат тази специалност(Фиг. 4.5).



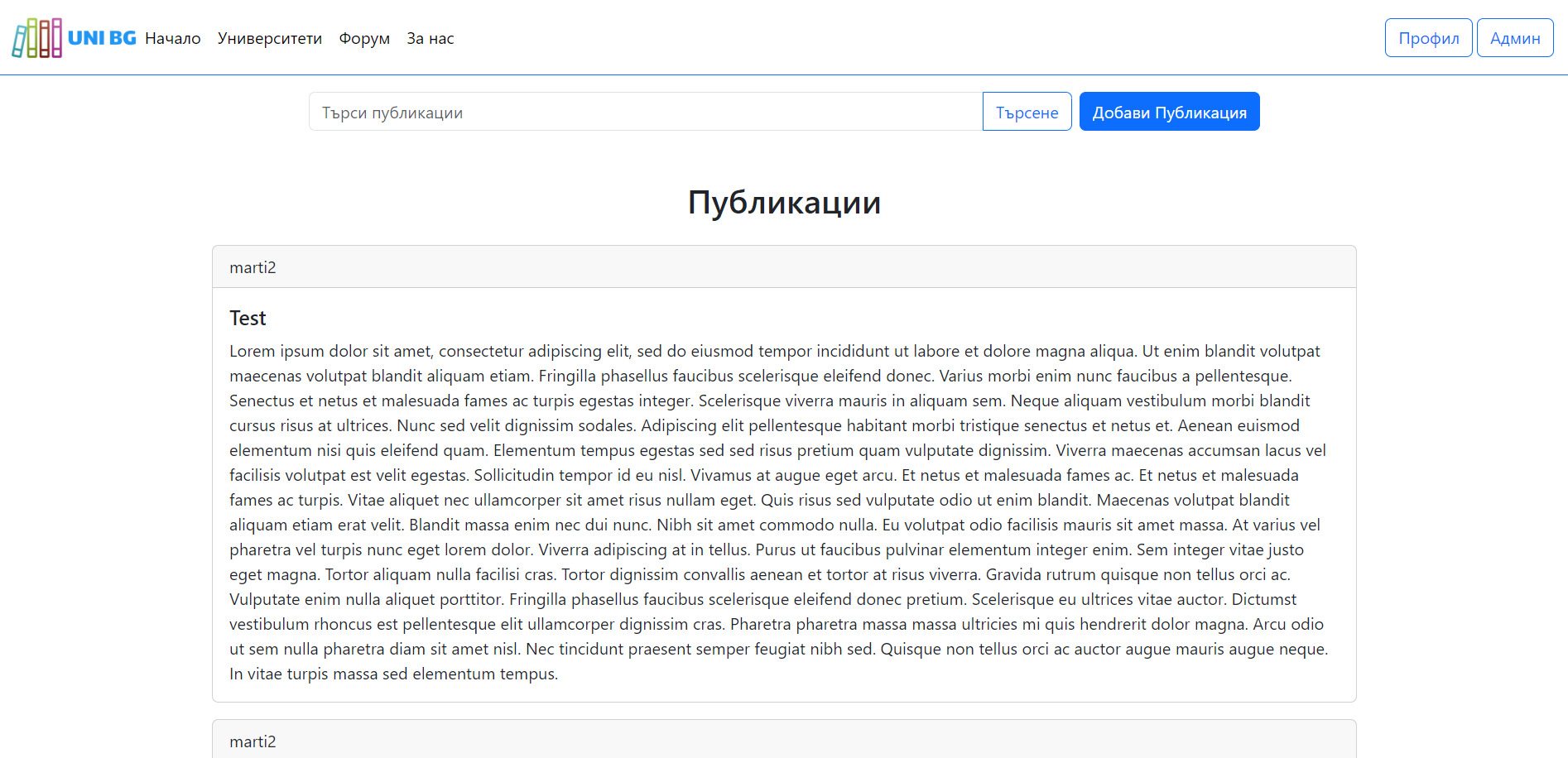
*Фигура 4.5 Резултати от търсене*

Ако потребителят избере “Виж Повече”, ще му бъде изкарана страница с още информация за дадения университет. Тя се състои от описание, лист със специалности, контакти с университета и препратки към официалния сайт на институцията.

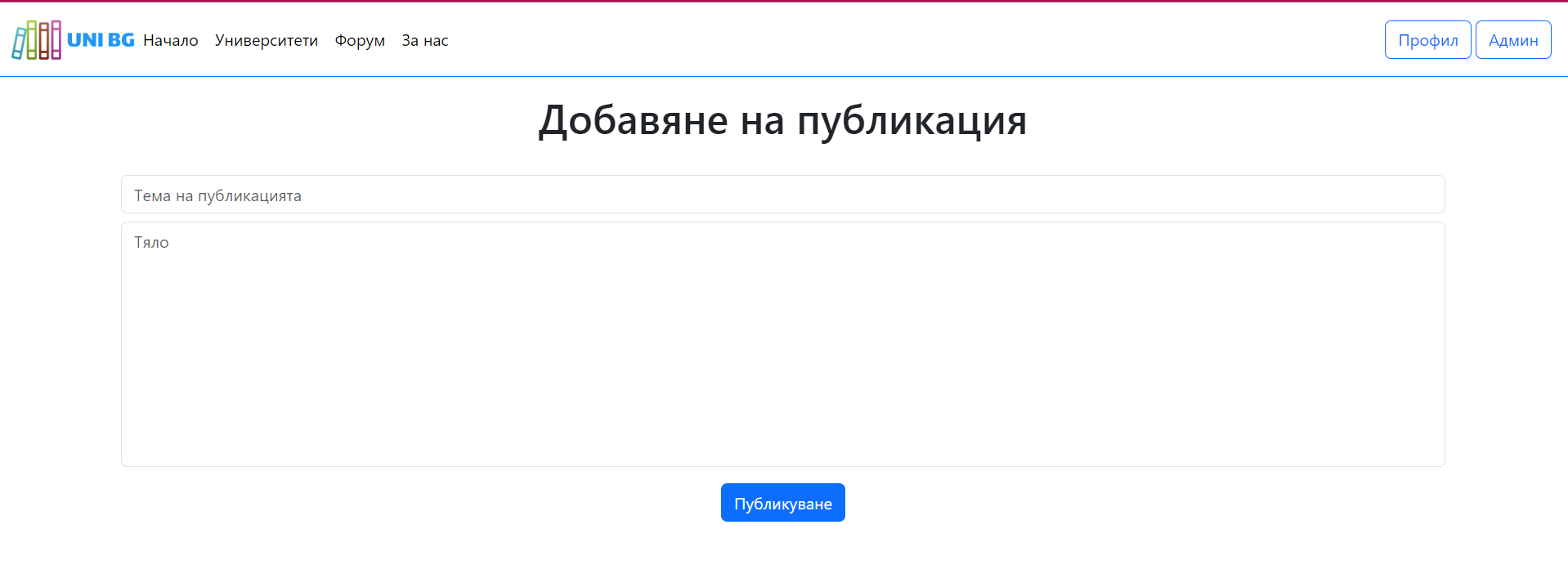


*Фигура 4.6 Страница на университет*

Ако потребителят използва сайта през профил, той има достъп до страницата с форум(Фиг. 4.7), в която всички регистрирани потребители могат да пишат публикации. От нея може да избере на напишат публикация, което г препраща към друга страница за писане на публикация(Фиг. 4.8).

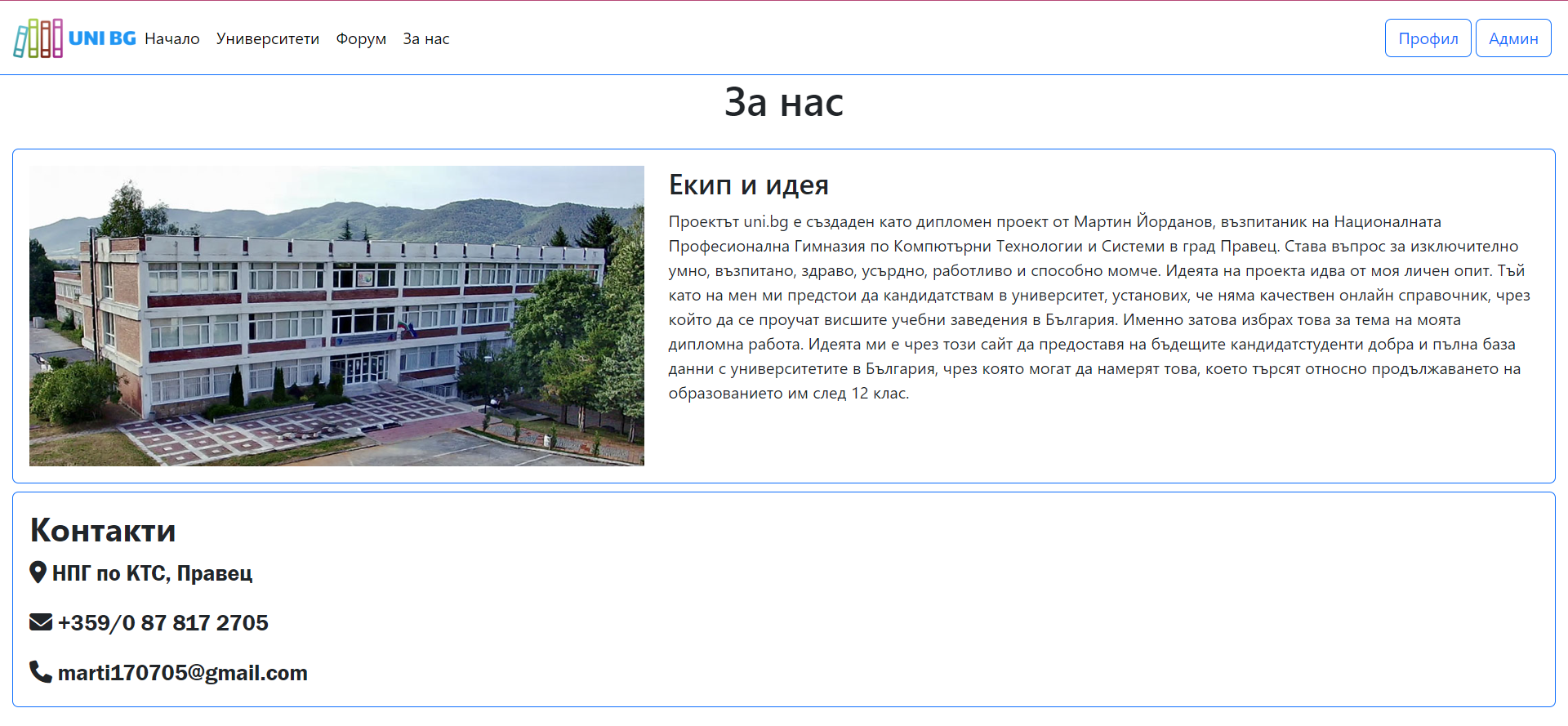


*Фигура 4.7 Форум*

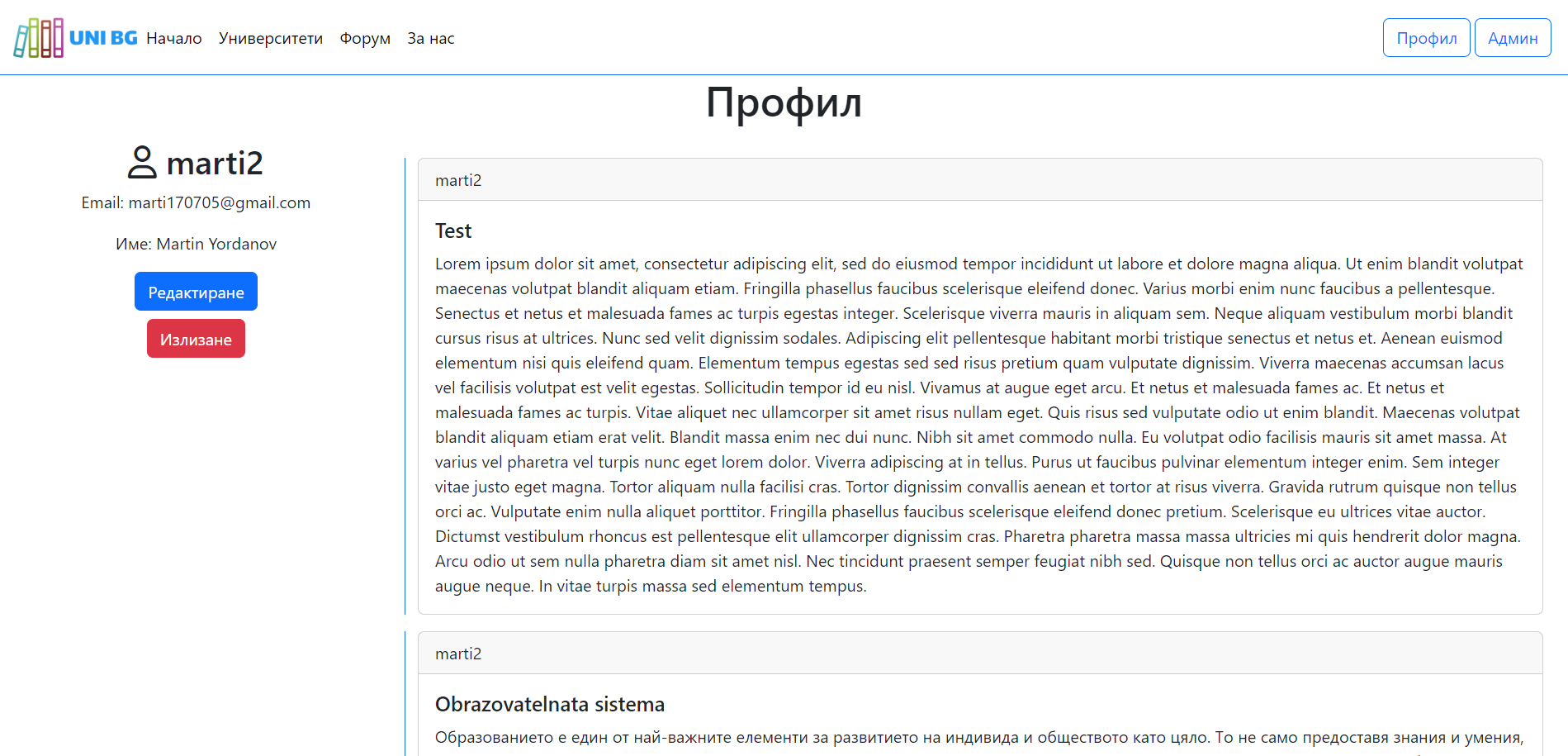
**

*Фигура 4.8 Добавяне на публикация във форум*

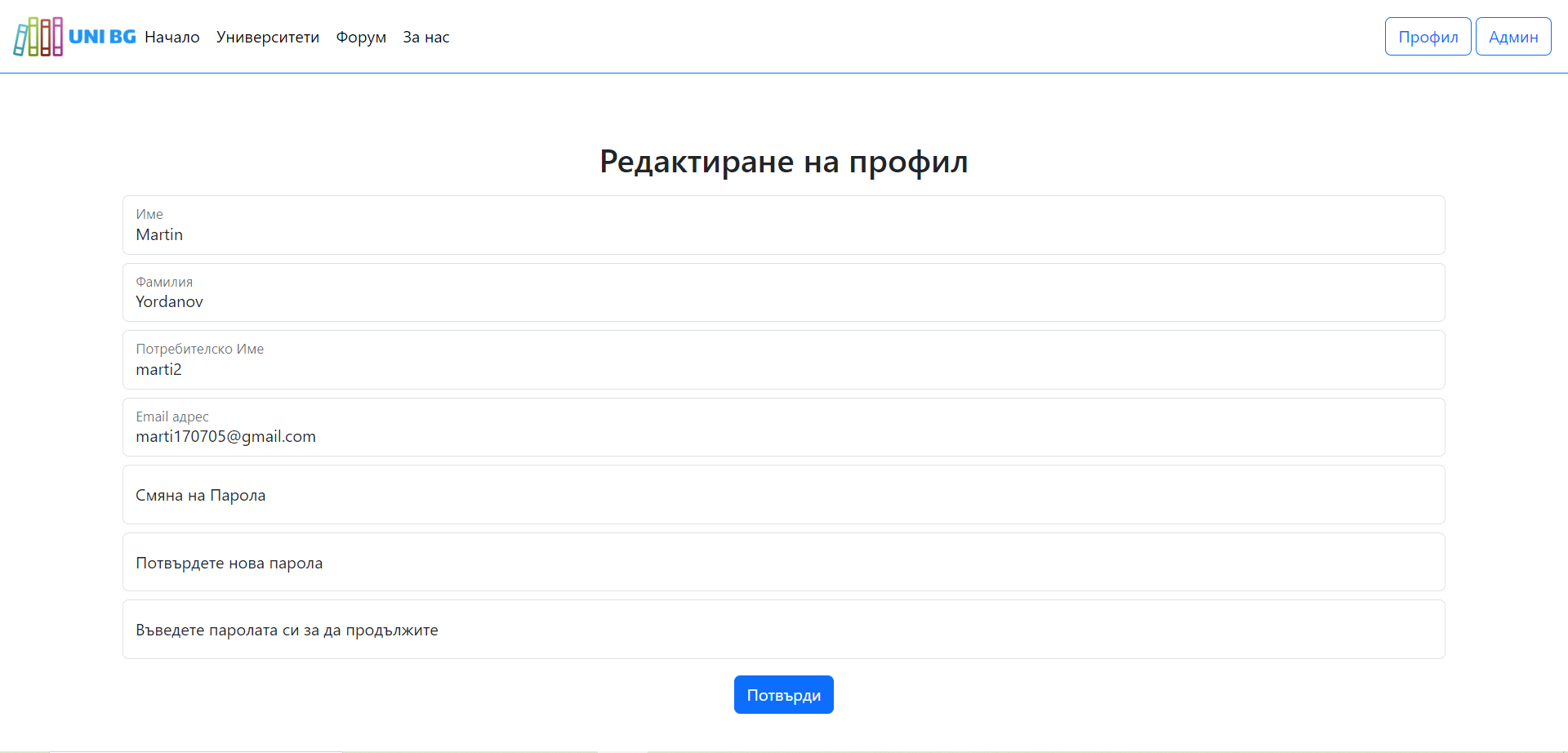
Сайтът също разполага със страница за контакти с хора от сайта и информация за създателите и идеята на сайта(Фиг. 4.9). Разполага и със страница “Профил”(Фиг. 4.10), която съдържа информация аз потребителя: Име, потребителско име, имейл и опции за редактиране на профил(Фиг. 4.11) и излизане от профила.



*Фигура 4.9 Страница “За нас”*



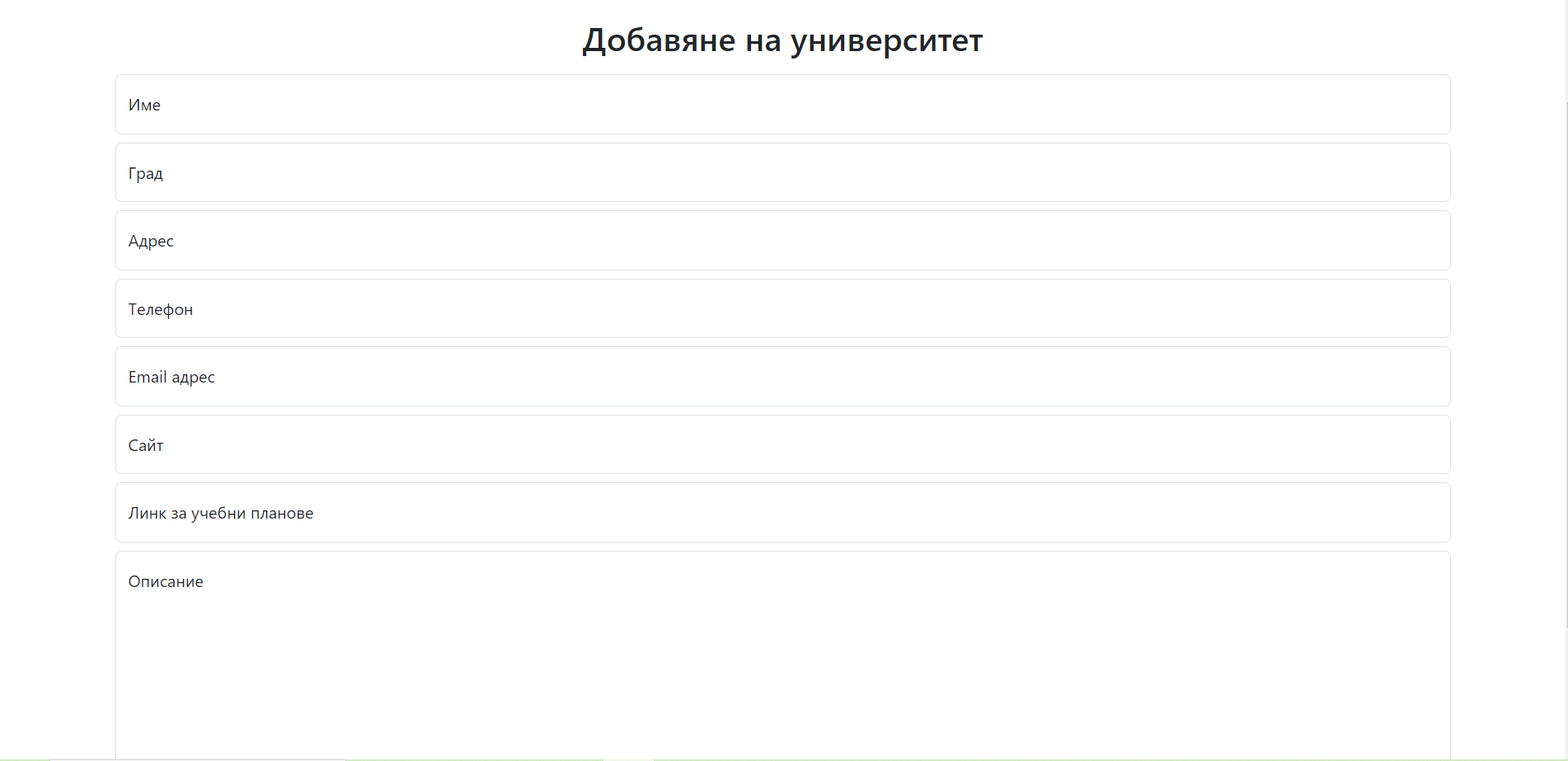
*Фигура 4.10 Страница “Профил”*



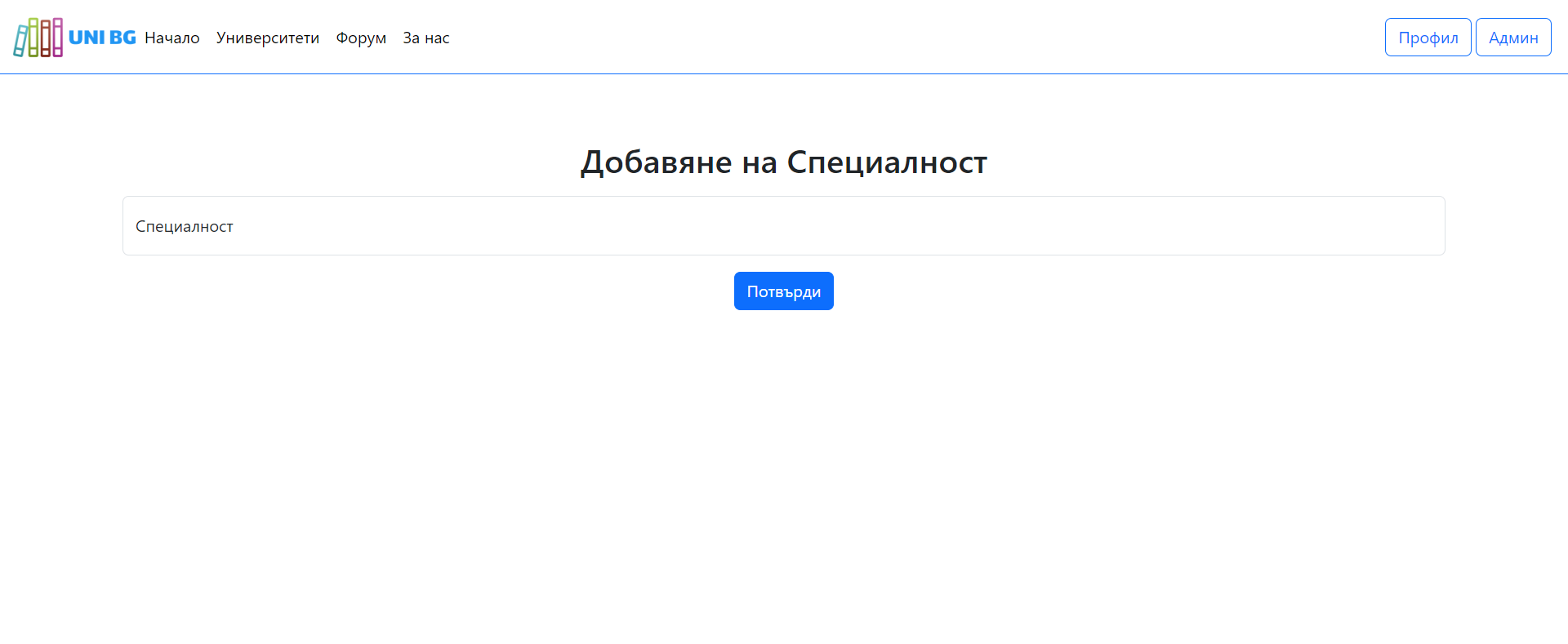
*Фигура 4.11 Страница за редактиране на профил*

Това са всички страници, достъпни за обикновения потребител. В сайта обаче има и админ панел с 3 страници, които са достъпни само за администратори:

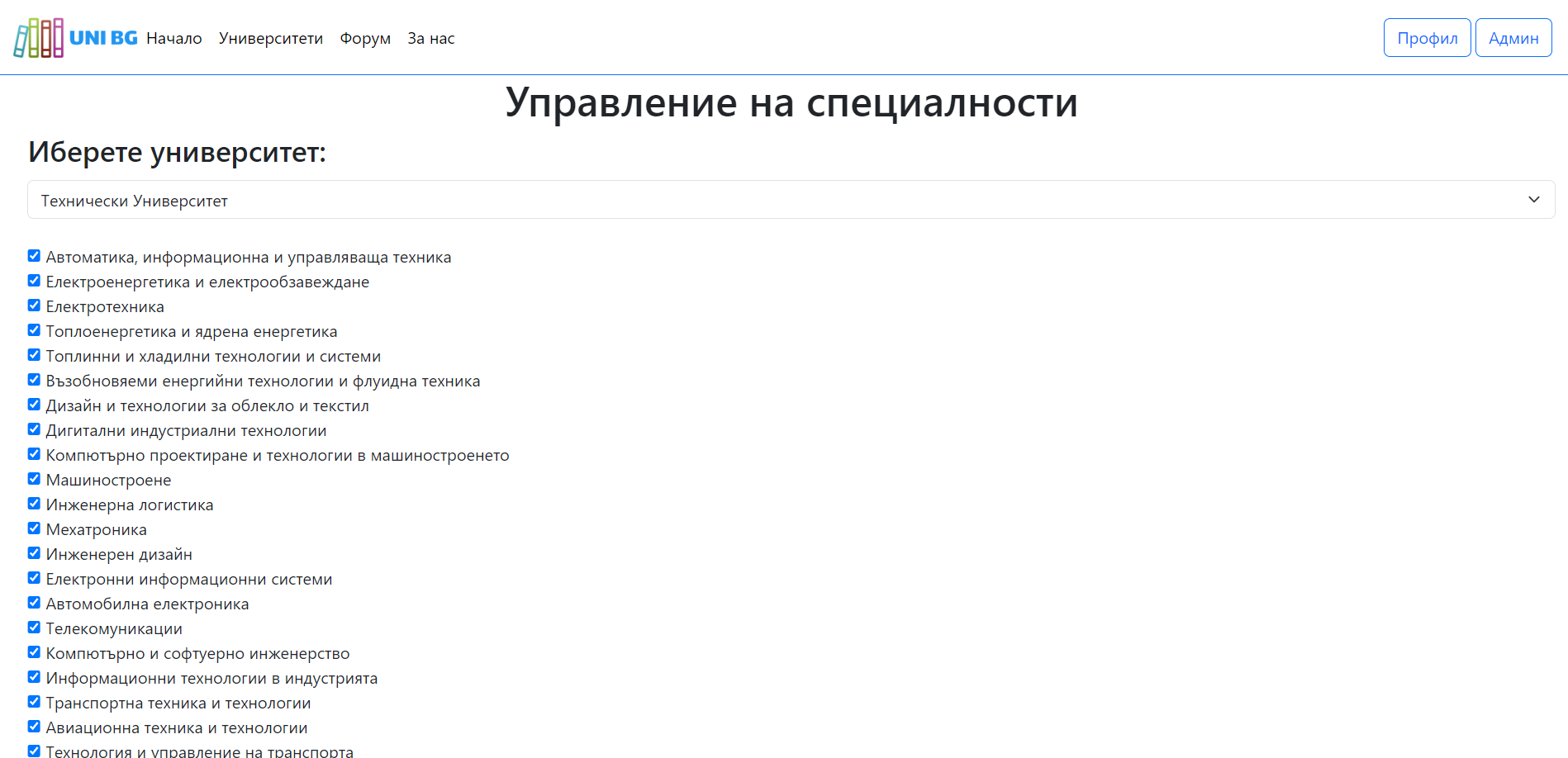
* Страница за добавяне на университети към базата данни(Фиг. 4.12)
* Страница за добавяне на специалности към базата данни(Фиг. 4.13)
* Страница за създаване на връзка между университетите и специалностите(Фиг. 4.14)



*Фигура 4.12 Страница за добавяне на университети в базата данни*

**

*Фигура 4.13 Страница за добавяне на специалности в базата данни*

**

*Фигура 4.14 Страница за управление на специалности*

# Заключение

## Какво може да се се подобри в приложението?

Уебсайтът Uni bg е в доста напреднала фаза от разработката си. Дизайнът е на ниво и е адаптивен за две устройства. Като функционалност работи добре и няма бъгове. Въпреки това има някой неща, които бих искал да променя:

1. **Допълване на базата данни**

Основното нещо, което в момента се нуждае от подобрение, е базата данни. Тя е структурирана добре, но е непълна. Целта за бъдещето е да бъде запълнена с всички висши учебни заведения в България, за да може потребителите да могат да намерят точно това, което търся, в който и край да страната да се намират и да искат да живеят.

1. **Добавяне на адаптивен дизайн за повече размери екрани**

В момента сайтът е адаптивен дизайн за компютър и мобилен телефон. Ако приложението бъде пуснато онлайн, ще е по-удобно за потребителя ако може да го използва нормално и на по-нетрадиционни резолюции и на повече устройства.

1. **Подобрение на комуникацията между потребители във форума**

Kъм момента форума на проекта е с много основна функционалност и, макар и да работи, има накъде да се подобри за да улесни комуникацията между потребители. На първо място, да бъде създадена страница за всяка публикация, в която да могат да се пишат коментари, за да може разговорът да е по-структуриран и да се задържи на едно място. Друго нещо, което може добавено към функционалността на форума е добавяне на опцията за търсене на публикации, качени от даден потребител. Това ще е нужно когато в сайта се съберат повече профили и започнат да се пишат много публикации от много потребители.

1. **Подобрения в сигурността**

Друго нещо, на което може да се обърне внимание, е сигурността на приложението. Могат да се добавят изисквания за паролите на потребителите при регистрация, за да бъдат паролите по-сложни и по-трудни за разбиване.

# Използвана литература

<https://www.browserstack.com/guide/web-application-development-guide>

<https://www.youtube.com/>

<https://www.w3schools.com/>

<https://stackoverflow.com/>

<https://ieeexplore.ieee.org/document/796139>

<https://softuni.bg/blog/what-is-framework>

<https://visualstudio.microsoft.com/>

<https://introprogramming.info/intro-csharp-book/>

<https://code.visualstudio.com/docs>

<https://www.mysql.com/products/workbench/>

<https://www.w3.org/press-releases/2014/html5-rec/>

<https://html.com/>

<https://www.php.net/>

<https://getbootstrap.com/docs/5.3/getting-started/introduction/>

<https://www.apachefriends.org/>

# Списък с използвани съкращения и означения

HTML - HyperText Markup Language

CSS - Cascading Style Sheets

PHP – Personal Home Page

SQL - Structured Query Language

HTTPS - Hypertext Transfer Protocol Secure

URL - Uniform Resource Locator

VS - Visual Studio

ODBC - Open Database Connectivity

LAMP/MAMP/WAMP - Linux/Mac/Windows, Apache, MySQL, Perl/PHP/Python