

**ЧАСТНА ПРОФЕСИОНАЛНА ГИМНАЗИЯ ЗА ДИГИТАЛНИ НАУКИ „СОФТУНИ БУДИТЕЛ“, гр. София**

**ДИПЛОМЕН ПРОЕКТ**

На Ева Камен Маздрашки

ученичка от XII А клас

професия- код:  **481030**, **“Приложен програмист”**

специалност- код:  **4810301**, **“Приложно програмиране”**

**Тема:** Разработка на уеб приложение за създаване на профили за кандидастване за работа

**Ръководител-консултант:** Иванжела Динчева

**Сесия**: май-юни 2025г.

**СЪДЪРЖАНИЕ:**

**Съдържание:**.................................................................................................................................2

**Увод:**................................................................................................................................................4

**Първа Глава:** Проучвителна част................................................................................................7

* + Анализ на съществуващи системи и приложения…………………………………..
  + Силни и слаби страни на наличните решения……………………………………....
  + Все още нерешени проблеми и възможности за подобрение……………………....

**Втора Глава:** Задание за софтуерна разработка.......................................................................10

* + Формулиране на цел и задачи…………………………...............................................
  + Функционални и нефункционални изисквания…………………………..................
  + Подбор на технологии и инструменти………………………….................................
  + Определяне на основни компоненти (*Entities, Models, Controllers, Views*)...............

**Трета Глава:** Проектиране, имплементация, тестване и внедряване....................................18

* Архитектура на системата (*MVC + Services*) …………………………......................
* Проектиране на база данни (*ER-диаграма*).................................................................
* Реализация на ключови модули (*Профили, Оферти, Кандидатури, Административен панел*)..............................................................................................
* Тестване (*Unit тестове и функционално тестване*)..................................................
* Разгръщане и CI/CD процеси........................................................................................

**Четвърта Глава:** Ръководство за потребителя.........................................................................21

* Регистрация, влизане и управление на профил..........................................................
* Добавяне и управление на обяви за работа.................................................................
* Кандидатстване и управление на кандидатури...........................................................
* Административен панел (*HR/Admin*)...........................................................................

**Заключение**.................................................................................................................................28

**Използвани източници**..............................................................................................................29

**Приложения**................................................................................................................................30

* Скриптове за инициализация на БД...................................................................................
* Кодови фрагменти (*Unit тестове*) ....................................................................................
* Съдържание на конфигурационни файлове......................................................................

**Увод**

В съвременната динамична среда на пазара на труда, процесът на кандидатстване за работа постоянно еволюира, превръщайки се във водеща тема не само за търсещите работа, но и за работодателите, които се стремят да улеснят комуникацията и подбора на подходящи кандидати. Развитието на информационните технологии и масовото навлизане на интернет в ежедневието създадоха нови възможности за дигитализация на кадровите процеси и автоматизация на рутинни задачи. В този контекст разработването на уеб приложение, което позволява създаване и управление на профили за кандидатстване за работа, представлява актуална и значима задача.

**Актуалност на темата**

1. **Растящи изисквания на пазара на труда**: Работодателите търсят бързи и ефективни начини за селекция на таланти, а кандидатите – интуитивни инструменти за представяне на своите умения и опит.
2. **Дигитализация и автоматизация**: Преходът към електронни платформи за кандидатстване намалява административната тежест и минимизира риска от загуба на данни.
3. **Конкурентоспособност**: Съвременните уеб приложения дават възможност за интеграция с външни услуги (социални мрежи, имейл известия, системи за анализ), което повишава качеството на услугата и конкурентоспособността на компанията.

**Цел на дипломния проект**

Целта на настоящата дипломна работа е да разработи уеб приложение за създаване и управление на профили за кандидатстване за работа, което да осигури:

* **Интуитивен потребителски интерфейс**, улесняващ навигацията и попълването на данни.
* **Гъвкави механизми за въвеждане на информация**, които позволяват детайлно представяне на професионалния опит и личните качества на кандидатите.
* **Автоматизирана система за уведомяване** на кандидатите и работодателите за нови събития и статуси на подадените кандидатури.
* **Безопасност и сигурност** на личните данни чрез защита на потребителската информация и контрол на достъпа.

**Задачи на дипломната работа**

1. **Анализ на съществуващите решения** – проучване на популярни платформи и идентифициране на добри практики.
2. **Дефиниране на функционални и нефункционални изисквания** – формулиране на точни спецификации, които приложението трябва да покрива.
3. **Проектиране на архитектура** – избор на технологични компоненти, моделиране на данни и структуриране на слоевете на приложението (MVC, services, data).
4. **Имплементация на основните модули** – разработка на потребителски интерфейс, логика за създаване на профили, управление на кандидатури и уведомления.
5. **Тестване на приложението** – провеждане на unit тестове и функционални тестове за гарантиране на качеството.
6. **Внедряване и документация** – деплоймънт в интернет среда и подготовка на потребителско ръководство.

**Структура на дипломната работа**

Дипломната работа е структурирана в четири основни части:

1. **Теоретична част** – представя обзор на литературата, анализ на налични решения и формулиране на задание за разработка.
2. **Задание за софтуерна разработка** – дефиниране на изисквания, избор на технологии и моделиране на системата.
3. **Практическа част** – проектиране, имплементация, тестване и внедряване на уеб приложението.
4. **Ръководство за потребителя** – обяснение на функциите и процесите в приложението, съчетано с инструкции за ползване.

В заключение настоящият увод очертава рамката на проучването и разработката, като задава ясни цели и задачи за постигане на ефективно и надеждно решение за създаване на уеб приложение за кандидатстване за работа.

**Първа Глава: Проучвателна част**

В тази глава ще разгледаме някои от най-разпространените уеб платформи за публикуване и търсене на работни позиции, ще идентифицираме основните им силни и слаби страни, и ще формулираме целите и задачите на нашия проект въз основа на направения анализ.

**1. Преглед на съществуващи решения**

* 1. **Jobs -** български сайт за обяви за работа с фокус върху местния пазар.
  2. **Основни функционалности:**

Филтриране по локация, сфера и вид заетост

Възможност за кандидатстване директно през платформата

Профили на компании със списък на всички техни обяви

* 1. **Плюсове:**

Добра локализация и SEO за български работодатели

Лесен за използване интерфейс

* 1. **Минуси:**

Ограничени възможности за персонализация на кандидатската анкета

Липса на разширени аналитични инструменти за работодателите

* 1. **Workable** - SaaS решение за управление на подборните процеси на големи и средни компании.
  2. **Основни функционалности:**

Цялостна ATS (Applicant Tracking System) – публикуване, скрийнинг, интервюта

Колаборация в реално време – бележки, оценки на кандидатите от различни рекрутъри

Вградени аналитични табла и отчети за ефективността на кампанията

* 1. **Плюсове:**

Пълна интеграция на всеки етап от подбора

Интуитивен back-office за HR специалисти

* 1. **Минуси:**

Висока цена за малки фирми

Липса на публично достъпни профили за кандидати – всичко е затворено в системата

* 1. **LinkedIn Jobs -** част от професионалната социална мрежа LinkedIn, предлагаща публикуване на обяви и таргетиране на кандидати.

**2. Основни функционалности:**

Публикуване на „sponsored jobs“ с насочване по умения, география и ниво

Директни съобщения („InMail“) до потенциални кандидати

Преглед на профила на кандидата с пълна история на професионалното развитие

**2.1 Плюсове:**

Моментален достъп до мрежа от над 700 млн. Специалисти

Богати данни за всеки кандидат (умения, препоръки, проекти)

* 1. **Минуси:**

Ценово не е достъпна за по-малки фирми без значителен бюджет

Зависимост от активността на кандидатите в LinkedIn

**2. Анализ на проблемите и пропуските**

* 1. **Липса на персонализация**

JobsBg предлагат базови филтри, но не дават възможност за създаване на комплексни анкети според нуждите на компанията.

* 1. **Съвместна работа на екипи**

Повечето публични платформи нямат интегрирани инструменти за съвместно оценяване на кандидати (освен професионалните ATS решения като Workable, но те са скъпи).

* 1. **Ниво на достъпност -** Международните агрегатори често пренасочват към други сайтове, което затруднява потребителското изживяване и прави проследяването на кандидатурите по-трудно.
  2. **Ценообразуване -** Платформи като LinkedIn Jobs и Workable са нерентабилни за малки и средни предприятия.

**3. Цел и задачи на проекта**

* 1. **Цел -** Да се разработи уеб приложение за публикуване и управление на обяви за работа, което обединява предимствата на локалните и международните платформи, но е достъпно, гъвкаво и ориентирано към потребителското изживяване на малки и средни компании, HR екипи и кандидати.
  2. **Задачи**
  3. **Персонализирани формуляри за кандидатстване** - Позволяване на работодателите да създават собствени въпроси и избори при кандидатстване.
  4. **Екипна колаборация -** Вграждане на коментари и оценки на кандидати директно в интерфейса на приложението.
  5. **Интуитивен интерфейс без пренасочване -** Кандидатите могат да подават документи и да следят процеса само в рамките на нашата платформа.
  6. **Гъвкаво ценообразуване** - Безплатен базов план
  7. **Аналитични табла за работодатели -** Преглед на брой кандидатури, източници на трафик и ефективност на обявите.

**Втора Глава: Задание за софтуерна разработка**

**1. Функционални изисквания**

* 1. **Управление на потребители и роли**
* Регистрация и верификация на имейл
* Логин/логоут с ASP .NET Identity
* Профилиране според роля: Admin, HR, Company, Registered
* Администраторски панел за създаване, редакция и блокиране на потребители
  1. **Профили за компании**
* Създаване и редакция на фирмен профил: име, описание, лого, уебсайт, контакти
* Администраторско одобрение (флаг IsVerified)
* Публична страница на компанията с обявите ѝ за работа
  1. **Управление на категории**
* CRUD операции върху категории (Job Category)
* Разпределяне на обяви към категории
  1. **Обяви за работа (Offers)**
     + Създаване/редакция/изтриване на обяви от компании и HR
     + Заглавие, описание, заплащане, категория, дата на публикуване
     + Публичен списък и детайлна страница на всяка обява
     + Филтриране и търсене по ключова дума, категория и локация
  2. **Кандидатстване и кандидатски профили**
     + Потребителите (регистриррани) подават кандидатури към обяви
     + Прикачване на CV, мотивационно писмо, портфолио
     + Статус на кандидатурата: Pending → Approved / Rejected
     + Известия по имейл за промяна на статус (синхронно)
  3. **Управление на кандидатури (ViewApplications)**
     + Company/HR могат да преглеждат всички кандидатури за дадена обява
     + Филтриране по статус
     + Одобряване/отхвърляне с бутон (промяна на статус + пощенско известие)
  4. **Панел „ManageJobs“**
     + Пагиниран изглед (5–10 обяви на страница)
     + Линкове за редакция и изтриване
  5. **Инструменти за колаборация**
     + Коментари и бележки на HR върху кандидатските профили (само за вътрешно ползване)
     + Споделяне на кандидатури между потребители в ролята HR
  6. **Admin и табла**
     + Dashboard за Admin: брой потребители, компании, обяви и категории
     + Dashboard за HR: брой кандидатури по статуси, активни обяви
     + Графики за нови обяви по дни (за Admin)

1. **Нефункционални изисквания**
   1. **Сигурност**
      * Пълно HTTPS криптиране
      * ASP .NET Identity с хеширани пароли и email confirmation
      * Ролеви разрешения на ниво контролер (Policy-based Authorization)
   2. **Производителност**

* Lazy-loading/Rabbit caching за често заявявани справки (категории, обяви)
* Минимум 95% от страниците да зареждат под 1 s при 100 едновременни потребителя
  1. **Надеждност**
* Автоматично почистване на неактивни сесии и изчаквания
* Graceful error pages и централизирана логика за exception handling
  1. **Мащабируемост**
* Разделяне на слоевете: Web → Services → Data
* Приготвен Dockerfile и docker-compose за лесен rollout
  1. **Поддръжка и разширяемост**
* Ясна SOLID архитектура, DI през конструктора
* Unit тестове за критичните методи (Services, Helper класове)
* Покритие ≥ 65% от бизнес-логиката с NUnit/Moq

1. **Избор на технологии**

|  |  |
| --- | --- |
| **Слой** | **Технология** |
| Презентация | ASP .NET Core MVC с Razor Views |
| Автентикация | ASP .NET Core Identity |
| Dependency Injection | Вграден в ASP .NET Core |
| Клиентски UI | Bootstrap 5 + jQuery |
| Емайл | `IEmailSender` с SMTP |
| Хостинг / CI/CD | Docker + GitHu |
| ORM / База данни | Entity Framework Core (Code-First) + MS SQL |

1. **Моделиране на системата**
   1. **Основни Entities**

* User
* CompanyProfile
* Category
* Offer
* JobApplication
* HRTicket
* Profile
  1. **Връзки на User (Потребител)**
* **User ↔ Profile:** Един към един

Всеки потребител има точно един профил (User.Profile)

Всеки профил принадлежи на точно един потребител (Profile.User)

* **User ↔ CompanyProfile:** Един към един

Всеки потребител (ако е компания) има точно един CompanyProfile (User.CompanyProfile)

Всеки CompanyProfile принадлежи на точно един потребител (CompanyProfile.User)

* **User ↔ Offer:** Един към много

Потребител (като компания) може да публикува много обяви (User.Offers)

Всяка обява принадлежи на една компания (Offer.Company)

* + **User ↔ JobApplication**: Един към много

Потребител може да кандидатства за много работни позиции (User.JobApplications)

Всяка кандидатура е на един потребител (JobApplication.User)

* + **User ↔ HRTicket:** Един към много

Потребител може да създаде много HR билети (HRTicket.CreatedBy)

Всеки билет е създаден от един потребител

* 1. **Връзки на Offer (Обява)**
* **Offer ↔ Category:** Много към един

Обява принадлежи към една категория (Offer.Category)

В една категория може да има много обяви (Category.Offers)

* **Offer ↔ JobApplication:** Един към много

Една обява може да има много кандидатури (Offer.JobApplications)

Всяка кандидатура е за една обява (JobApplication.Offer)

* 1. **Връзки на JobApplication (Кандидатура)**
* **Кандидатурата свързва:**

Потребител (JobApplication.User)

Обява (JobApplication.Offer)

* 1. **Връзки на CompanyProfile (Профил на компания)**
* Връзка с потребител (като компания) и неговите обяви.
  1. **Връзки на Category (Категория):**
* Категорията групира множество обяви (Category.Offers).

1. **Views (Razor)**
2. **Admin**

* ChangeRole.cshtml
* CreateCategory.cshtml
* Dashboard.cshtml
* EditCategory.cshtml
* EditCompanyProfile.cshtml
* EditOffer.cshtml
* ManageCategories.cshtml
* ManageCompanies.cshtml
* ManageUsers.cshtml
* ViewCompany.cshtml
* ViewCompanyProfile.cshtml
* ViewOffers.cshtml
* ViewTickets.cshtml

1. **Company**

* AddJob.cshtml
* ManageJobs.cshtml
* ManageProfile.cshtml
* Profile.cshtml
* ViewApplications.cshtml

1. **Home**

* About.cshtml
* Index.cshtml

1. **HR**

* CreatedTicket.cshtml
* CreateTicket.cshtml
* Dashboard.cshtml
* EditJob.cshtml
* ViewApplications.cshtml

1. **Job**

* Browse.cshtml
* Details.cshtml
* Index.cshtml

1. **LoggedUser**

* ManageProfile.cshtml
* MyApplications.cshtml
* PublicProfile.cshtml

1. **Shared**

* \_Layout.cshtml
* \_LoginPartial.cshtml
* \_ValidationScriptPartial.cshtml
* AccessDenied.cshtml
* Error.cshtml

1. **ViewModels**
   1. **Admin**

* ChangeRoleViewModel.cs
* UserViewModel.cs
  1. **Company**
* CompanyProfileFormViewModel.cs
* JobApplicationsViewModel.cs
  1. **DataSeeder**
* DataSeeder.cs
  1. **Home**
* HomeViewModel.cs
  1. **HR**
* CreateTicketViewModel.cs
* HRDashboardViewModel.cs
  1. **JobListing**
* BrowseViewModel.cs
* JobFormViewModel.cs
* OfferCardViewModel.cs
  1. **LoggedUsers**
* LoggedUserProfileViewModel.cs
* MyApplicationsViewModel.cs
* PublicProfileViewModel.cs
  1. **Mail**
* SmtpEmailSender.cs

**6.7 ErrorViewModel.cs**

**7. Controllers**

* AdminController.cs
* CompanyController.cs
* HomeController.cs
* HRController.cs
* JobController.cs
* LoggedUserController.cs

**Трета Глава. Проектиране, имплементация, тестване и внедряване**

* 1. **Проектиране на системата**

**При проектирането на приложението за Path Finder се отдели специално внимание на три основни аспекта:**

* Разделяне на отговорностите – всяка функционална област (управление на потребители, фирмени профили, обяви, кандидатури, административен панел) е капсулирана в отделни модули с изчистени интерфейси.
* Разширяемост – добавянето на нови роли, типове обяви или формати за файлове се предвижда чрез ясни абстракции и DI (Dependency Injection).
* Тестваемост – слоят за бизнес логика е отделен в Services, така че да може да бъде мокван и изолиран в unit тестовете, без нужда от реална база данни.
* Тестваемост – слоят за бизнес логика е отделен в Services, така че да може да бъде мокван и изолиран в unit тестовете, без нужда от реална база данни.
  1. **Архитектура (MVC) Приложението използва класическата Model–View–Controller архитектура:**
* **Models**

Домейн ентитита (User, Offer, JobApplication, Category, CompanyProfile, HRTicket), които отразяват таблиците в базата данни.

ViewModel-и за обмен на данни между контролерите и изгледите.

* **Views**

Razor страници (.cshtml), организирани по зони: Admin, HR, Company, Job и т.н.

Статични части като layout, навигационни менюта и partial views за логин/регистрация.

* **Controllers**

Отговарят за прием и маршрутизация на заявките: HomeController, JobController, CompanyController, HRController, AdminController.

Делегират бизнес операции към Services, валидират входящи модели и връщат съответните изгледи или JSON.

* **Services**

Слой от класове (в проект JobListingSite.Services), който съдържа логиката за управление на обяви, кандидатури и уведомления по email.

По този начин контролерите остават леки и фокусирани върху HTTP потока.

* **Data**

JobListingDbContext (EF Core) управлява миграции и достъпа до базата, дефинира DbSet<T> за всички ентити.

Този подход осигурява ясна модулност, лесна поддръжка и гъвкаво разгръщане.

* 1. **Моделиране на данните (ER диаграма) в основата на базата данни лежат следните ключови таблици и връзки:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Моделите са означени с атрибути ([Key], [ForeignKey], [Required], [MaxLength]) за автоматично генериране на правилни колони и ограничения чрез миграции.**

* 1. **Навигация и роли**

**Приложението поддържа четири основни роли, всяка с различен достъп и меню:**

|  |  |  |
| --- | --- | --- |
| **Роля** | **Основни права** | **Навигация** |
| **Admin** | CRUD за потребители, роли, категории, фирми, билети; dashboard с графики | Manage Users, Manage Categories, View Companies, View Tickets, Dashboard |
| **Company** | Управление на обяви, преглед и обработка на кандидатури, създаване на HRTickets, собствен dashboard | Dashboard (Jobs, Applications, Tickets), Create Ticket |
| **HR** | CRUD за свои обяви, редакция на фирмен профил, преглед на собствените кандидатури | Manage Jobs, Manage Profile, View Applications |
| **Registered User** | Преглед на обяви, кандидатстване, управление на личен профил | Browse Jobs, My Applications, Manage Profile |

**Менютата в \_Layout.cshtml са условно рендирани чрез @if (User.IsInRole("...")), а авторизацията се налага на ниво контролер с [Authorize(Roles = "...")].**

**По този начин потребителят вижда само релевантните за него секции, а критичните операции са защитени чрез role-based и policy-based authorization.**

**Четвърта Глава: Ръководство за потребителя**

**1. Регистрация, влизане и управление на профил**

1. **Регистрация**

* Отивам на началната страница и кликам върху бутона **„Register“** в горния десен ъгъл.
* Попълвам задължителните полета: *Email*, *Парола* и потвърждавам паролата.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

1. **Влизане и излизане**

* За вход използвам бутона **„Login“**, въвеждам Email и Парола.
* След валидна аутентикация, виждам падащо меню с моето потребителско име.
* За изход избирам **„Logout“** от същото меню.

A screenshot of a computer

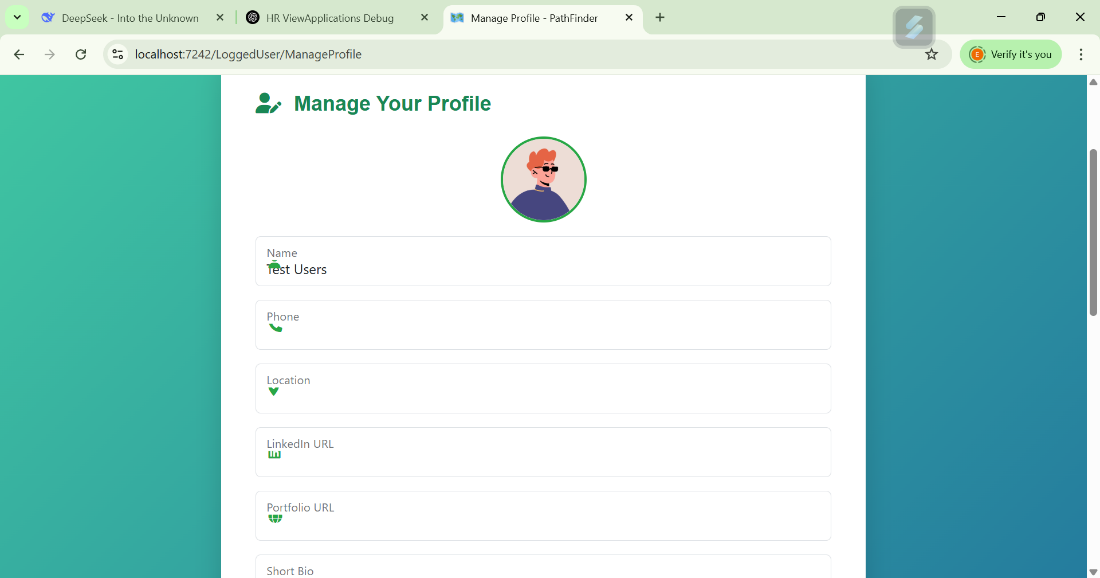
AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

1. **Управление на личен профил**

* В падащото меню на потребителското ми име избирам **„Profile“**.
* В раздела **„Personal Info“** мога да променя име, телефон и био.
* В **„Avatar“** качвам или сменям профилна снимка.
* В **„Email“** и **„Password“** мога да сменя съответно имейл адрес и парола, като въвеждам старите и новите данни.
* В секция **„Delete Account“** мога да изтрия профила си (след въвеждане на парола за сигурност).



**2.** **Добавяне и управление на обяви за работа**

1. **Създаване на нова обява**

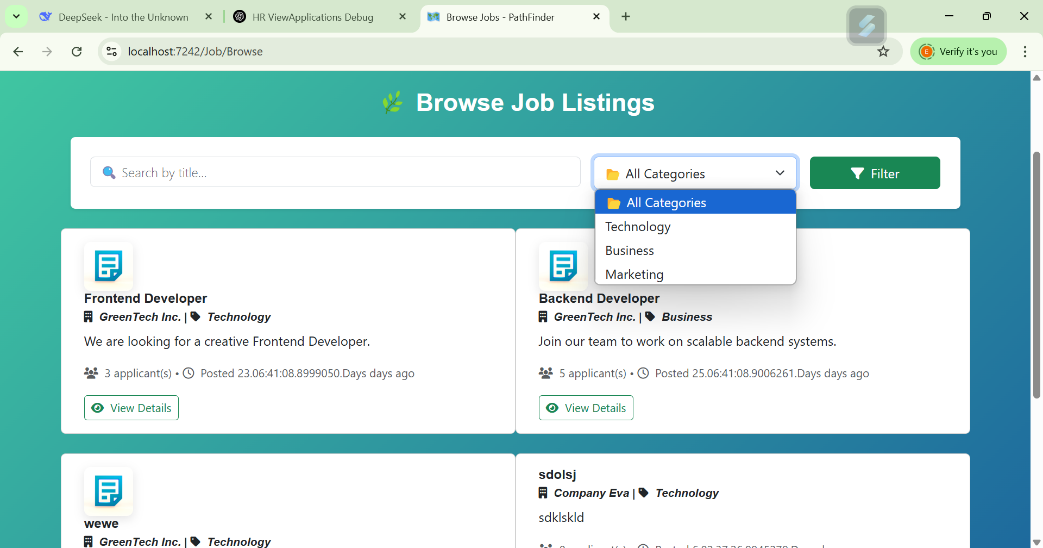
* **„Manage Jobs“**.
* Натискам **„Add Job“**.
* Попълвам полетата: *Title*, *Description*, *Salary*, избирам *Category* и въвеждам *Location*.
* Натискам **„Save“**. Новата обява се появява в списъка ми.

1. **Редакция и изтриване**

* Редакция: в списъка с обяви кликам върху иконата (Edit) до всяка обява, променям данни и записвам.
* Изтриване: кликам (Delete) и потвърждавам, след което обявата се премахва.

1. **Филтриране и търсене**

* В публичния изглед на *Browse* мога да въвеждам ключова дума в полето за търсене.
* По-долу имам падащ списък с *Category* и опция за *Location*, след което филтрирам с бутон **„Search“**.



**3. Кандидатстване и управление на кандидатури**

1. **Подаване на кандидатура**

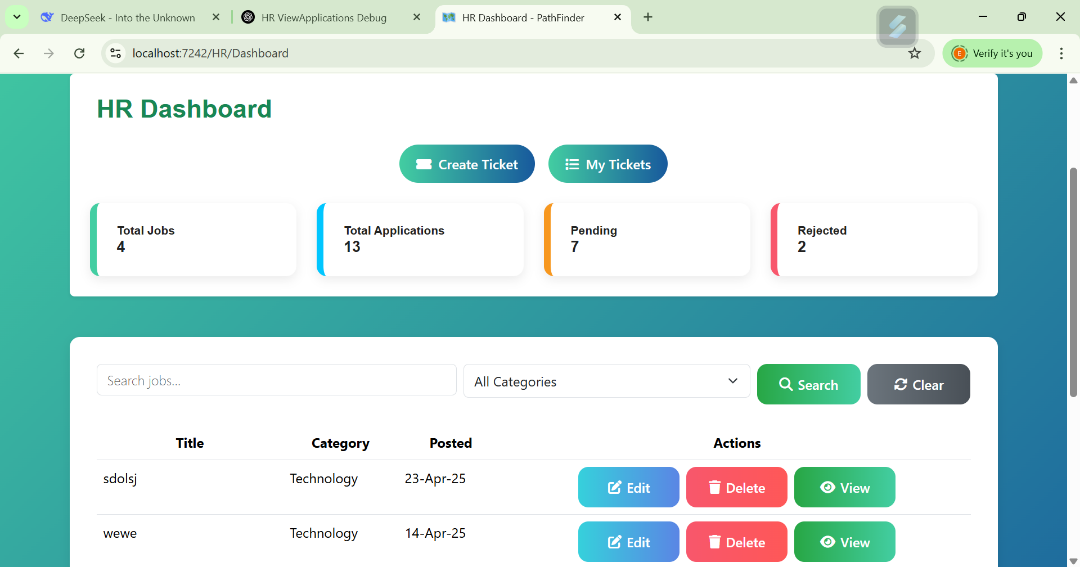
* В публичната страница на обявата натискам **„Apply“** (само ако съм регистриран).
* Попълвам кратко мотивационно писмо и прикачвам *CV* (PDF или DOCX), както и портфолио при нужда.
* Изпращам формата с бутона **„Submit Application“**.
* Получавам потвърждение на екрана и email за успешна регистрация на кандидатурата.

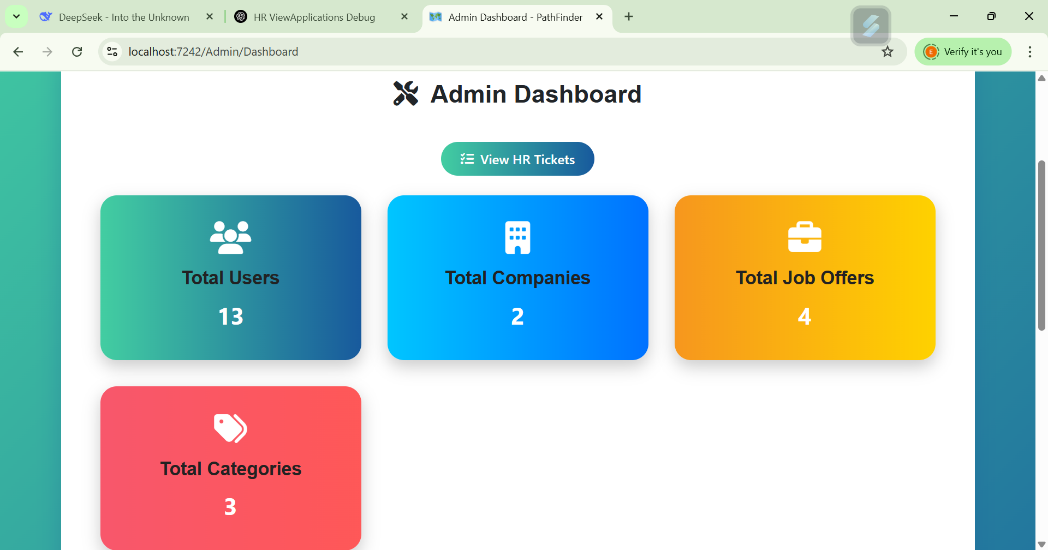
1. **Преглед и филтриране (Company/HR)**

* В секцията **„View Applications“** за избрана обява виждам всички кандидатури.
* Чрез падащото меню за *Status* филтрирам по Pending, Approved или Rejected.
* За всяка кандидатура мога да кликна **„View CV“** или **„Portfolio“** за сваляне.

1. **Одобряване и отхвърляне**

* **Approve**: бутонът одобрява кандидатурата, изпраща автоматично email с уведомление и сменя статус на *Approved*.
* **Reject**: бутонът отхвърля и изпраща email с мотивация или кратко съобщение.





**4.** **Административен панел (HR/Admin)**

1. **Dashboard (HR)**

* **Total Jobs**: брой публикувани обяви.
* **Applications by Status**: графика и числово разпределение на кандидатурите по Pending, Approved, Rejected.
* Последни 5 обяви с опции за бърза редакция.

1. **Dashboard (Admin)**

* **Total Users**, **Total Companies**, **Total Jobs**, **Total Categories**: агрегирани стойности.
* Графика *New Offers by Day* за последните 7 дни.

1. **Управление на потребители и роли**

* В **„Manage Users“** виждам списък с всички регистрирани (без компании).
* За всеки потребител мога да промотирам или демотирам чрез падащ списък на роли.
* Бутон **Lock/Unlock** блокира или отключва акаунт.

1. **Управление на категории**

* **Manage Categories**: CRUD интерфейс за Job Category.
* Добавям или редактирам име на категория, изтривам ненужни.

1. **Управление на фирмени профили (Admin)**

* **View Companies**: списък с всички CompanyProfile записи.
* Одобрение (Approve) с бутон за флаг *IsVerified*.
* Изтриване на фирма, ако е докладвана или дублирана.

1. **HRTickets (HR/Admin)**

* **Create Ticket** (HR): формуляр за създаване на вътрешен ticket при проблем или въпрос.
* **View Tickets** (Admin/HR): списък с всички тикети, филтриране по статус, Resolve/Delete за приключване.

**Заключение**

В тази дипломна работа беше представено цялостното разработване на уеб приложение за създаване и управление на профили за кандидатстване за работа. **През първата част** обосновахме актуалността на темата и формулирахме целите и задачите на проекта. Втората глава подробно дефинира функционалните и нефункционалните изисквания, изборът на технологии и предварителното моделиране на системата.

**В третата глава** описахме архитектурата на приложението по модела MVC, представихме релационното моделиране на базата данни и ER-диаграмата, разгледахме ключовите модули (профили, обяви, кандидатури и административен панел) и тестването чрез unit и функционални тестове. Накрая описахме разгъщането в Docker среда и CI/CD процеса, осигуряващ автоматизирано билдване и деплоймънт.

В ръководството за потребителя демонстрирахме всички основни сценарии от регистрация до администриране на цялата платформа. Приложението отговаря на всички заложени изисквания по сигурност, производителност и разширяемост.

**Приноси и бъдещи подобрения**

* Изградена е стабилна основа за разширяване с допълнителни модули (напр. нотификации, чат).
* Архитектурата позволява лесно интегриране на външни услуги (e.g. S3, SMS gateway).
* За в бъдеще може да се добавят real-time уведомления, multi-tenant поддръжка и пълна mobile-first адаптация.

В заключение, реализираният продукт изпълнява изискванията на дипломната задача и осигурява пълноценна среда за кандидатстване и управление на фирмени профили, като същевременно предоставя стабилна и сигурна основа за бъдещо развитие.

**Използвани източници**

1. Microsoft Docs – ASP.NET Core MVC Overview.
2. Microsoft Docs – Entity Framework Core (Code First) Guide.
3. Official Selenium Documentation – функционално тестване на уеб приложения.
4. X.PagedList.Mvc.Core GitHub – пагинация в ASP.NET Core.
5. Bootstrap 5 Documentation – респонсив дизайн и компоненти.
6. Docker Documentation – контейнери и docker-compose.
7. Render Documentation – автоматично разгръщане и хостинг на уеб приложения.
8. CycleFish, MotoSocial – анализ на съществуващи платформи за вдъхновение.

**Приложения**

1. Скриптове за инициализация на базата данни
2. DataSeeder.cs (примерни данни за потребители, категории, обяви)
3. SQL скрипт за създаване на схемата
4. Кодови фрагменти (Unit тестове)
5. Примерен тест за HRController.ApproveApplication
6. Тестови сценарии за AdminController.ChangeRole
7. Мокове и настройка на UserManager/RoleManager
8. Съдържание на конфигурационни файлове
9. appsettings.json (ConnectionStrings, SMTP настройки)
10. Dockerfile и docker-compose.yml