1. **Тема:** „Уеб приложение за приемане и предлагане на услуги (Freelancer)“
2. **Автор** – Сечкин Салим Рахим, 0452168740, с. Живково ул. Калоян №43 , общ. Хитрино, обл. Шумен, 0895073711, [seckins191@gmail.com](mailto:seckins191@gmail.com), ПГМЕТТ „Христо Ботев“, 12-а клас
3. **Ръководител:** инж. Недялка Маринова Йорданова, 0888930603, [yordanowa@gmail.com](mailto:yordanowa@gmail.com), старши учител по практическо обучение
4. **Резюме**
   1. **Цели:**

Целта на уеб приложението е да помага на хора, които имат определени знания и умения, желаят да работят на свободна практика, но не знаят от къде и как да започнат. Сайтът позволява качване и приемане на обяви, създаване на специална заявка, комуникация между доставчика и получателя на услугата, комуникация с администратора. Получателят на услугата може да види напредъка на доставчика, но само ако доставчика го отбележи в самата обява. Отбелязването става в проценти. Получателят на услугата има право да избере допълнителни услуги от сравнителните таблици, според цената. Доставчикът на услугата избира колко пъти да изпрати проекта на получателя за съгласуване и корекции.

**4.2 Основни етапи в реализирането на проекта:**

1. Идеен проект (Декември 2022)

2. Работа по дизайна (Декември, 2022)

3. Проектиране на Базата данни (Декември, 2022)

4. Свързване на Уеб приложението с База данни (Декември, 2022)

5. Тестване (Януари, 2023)

6. Публикуване в интернет (Февруари, 2023)

**4.3 Ниво на сложност на проекта:**

Основните проблеми при реализирането на проекта са рейтингите и забраняването на акаунти от администраторския панел.

**4.4 Логическо и функционално описание на решението:**

Уеб приложението „Freelancer“ е изградено чрез платформата .NET по модел MVC. Съставено е от три свързани компонента:

● Модели (Models) - слоят с бази данни – използвам MS SQL Server и ORM технологията Entity Framework Core;

● Изгледи (Views) - презентационният слой, изграден на основата на HTML5, CSS3, JavaScript, Bootstrap и Razor View;

● Контролери (Controllers) - междинният слой или слоят за обработка на данни е изграден чрез програмният език C# .

**Базата данни е съставена от следните таблици:**

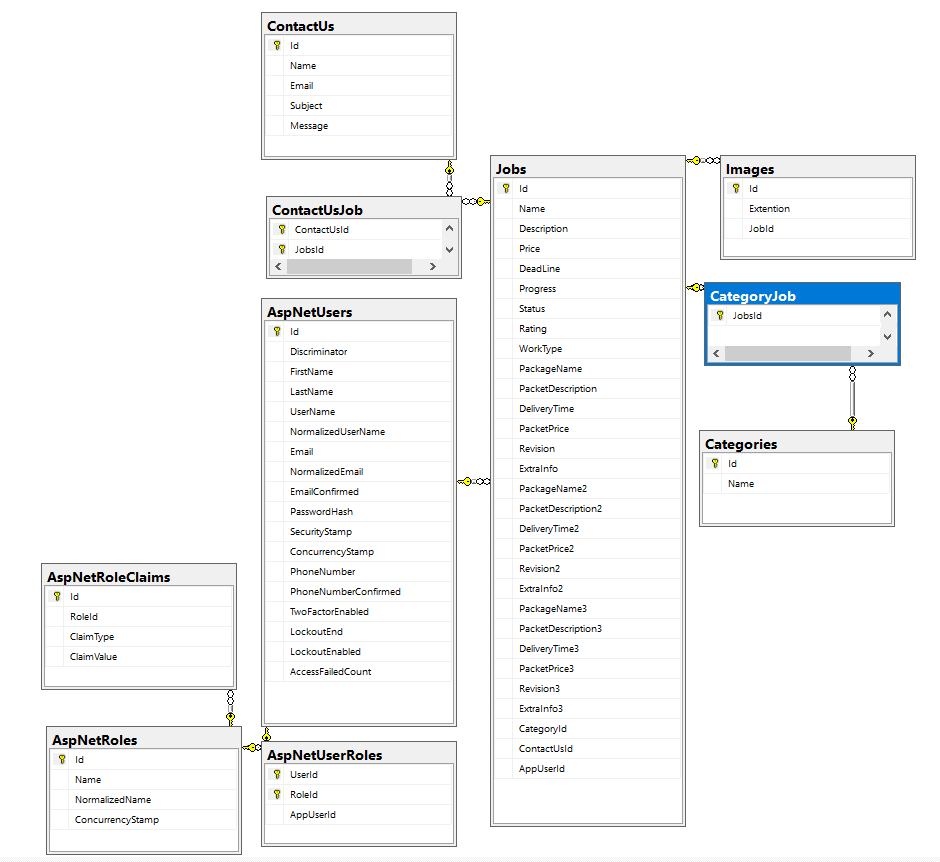
* *Таблица „Job“* – тя е главната таблица, всички други са свързани към нея. В тази таблица се записват данните, които се въвеждат от потребителя при създаване на обява. Включително и данните за статуса, тоест дали обявата е в пауза или не.
* *Таблица „Category“* – тази таблица е свързана към главната таблица с връзка от тип „Много : Много“ (М:М), защото една обява може да участва в много категории като например в категория „Front end“ и „Back end“. Междинната таблица е създадена автоматично с благодарение на .NET 6.
* *Таблица „Images“* – в тази таблица се пази името и разширението на снимката. Връзката е от тип „Едно : Много“ (1:М). Една обява може да има много снимки.
* *Таблица „Contact Us“* – в тази таблица се пазят съобщенията от потребителите към администратора. Включва името, имейла, заглавието на съобщението и самото съобщение на потребителя. Връзката към таблица “Job” е от типа М:М.
* *Таблица “AspNetRoles” -* в тази таблица се пази данните за ролите.
* *Таблица “AspNetUser” -* тази таблица замества таблицата “AppUser”. Тя не се появява при другите таблици. Паролите и имейлите се пазят тук.

**Модели**

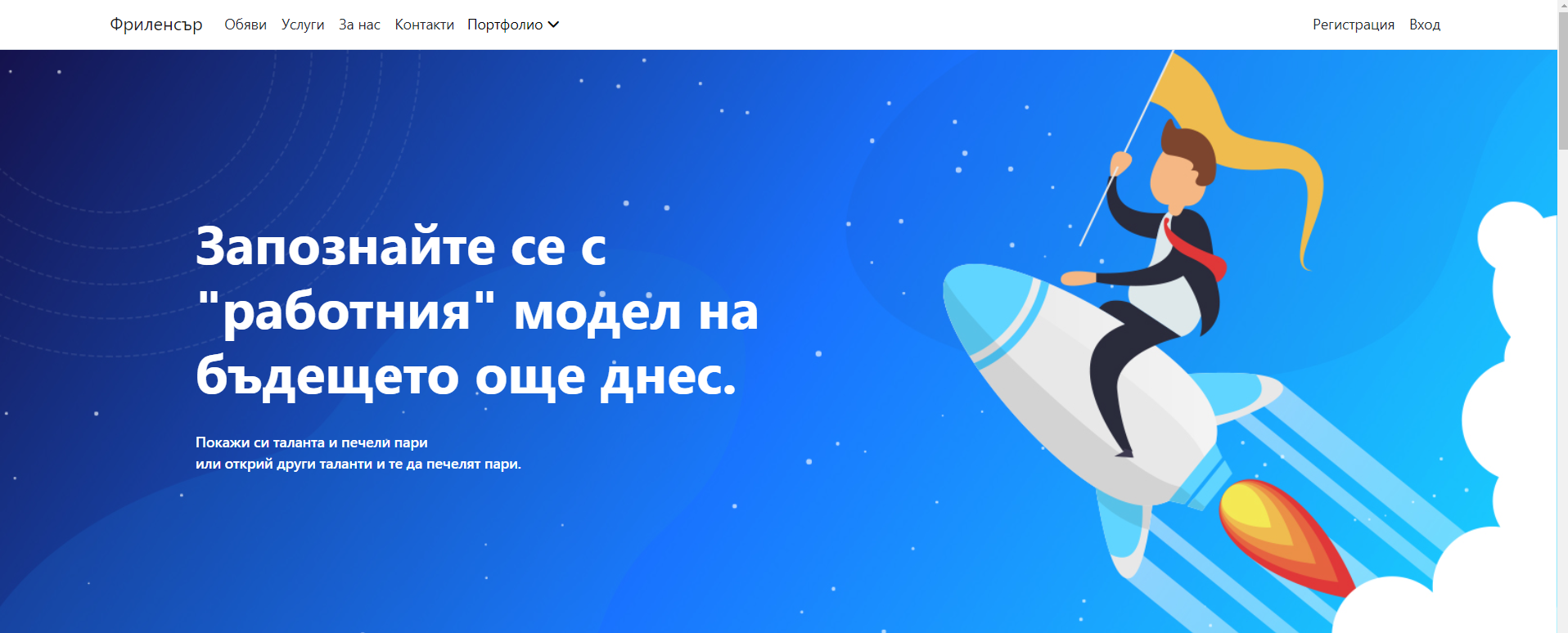
Базата данни с таблиците се генерира чрез модела CodeFirst от класовете с данни (Entity), описани в папка Data/Models на проекта. Използвам NuGet Package Entity Framework Core за връзка между приложението и базата данни.

Моделите, чрез които се генерират таблиците са следните:

* Job – съдържа нужните полета за добавяне на обява.
* Category – съдържа полетата за добавяне на категории в базата, след което излизат в падащ списък и потребителя избира от там. Но ако няма нужната категория, потребителя може да си го добави чрез знака плюс.
* AppUser – данните на потребителя при създаването на акаунт.
* ContactUs – съдържа свойства, които са нужни при изпращането на писмо(съобщение) до администратора.
* Image – съдържа името и разширението на снимката.



***Презентационният слой*** е структуриран в папка Views и включва файлове, създадени чрез технологията Razor View Engine. Файловете са с разширение .cshtml и съдържат код на HTML и C#. Дизайнът на сайта е чрез CSS, Bootstrap и JavaScript в папка wwwroot.

****

Начален екран на приложението

**Навигацията** е лесно използваема. Състои се от 8 секции:

„Начало“ – за визуализиране на рекламациите;

„Обяви“ - за споделените оферти;

„Услуги“ - за категориите, които са ориентировъчни;

„За нас“ - информацията за администратора;

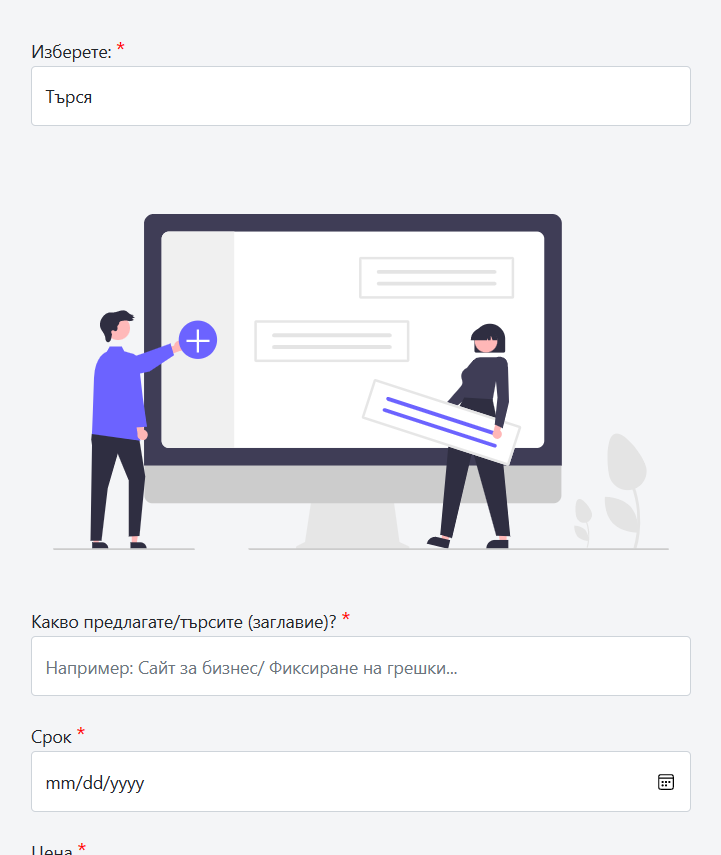
„Контакти“ - формуляр за изпращане на съобщение до администратора;

“Портфолио” – падащо меню с две подсекции “Моите заявки” и “Моите поръчки”, появяват се след логването на потребителя.

„Вход“ и „Регистрация“ са за създаване на акаунт и логване.

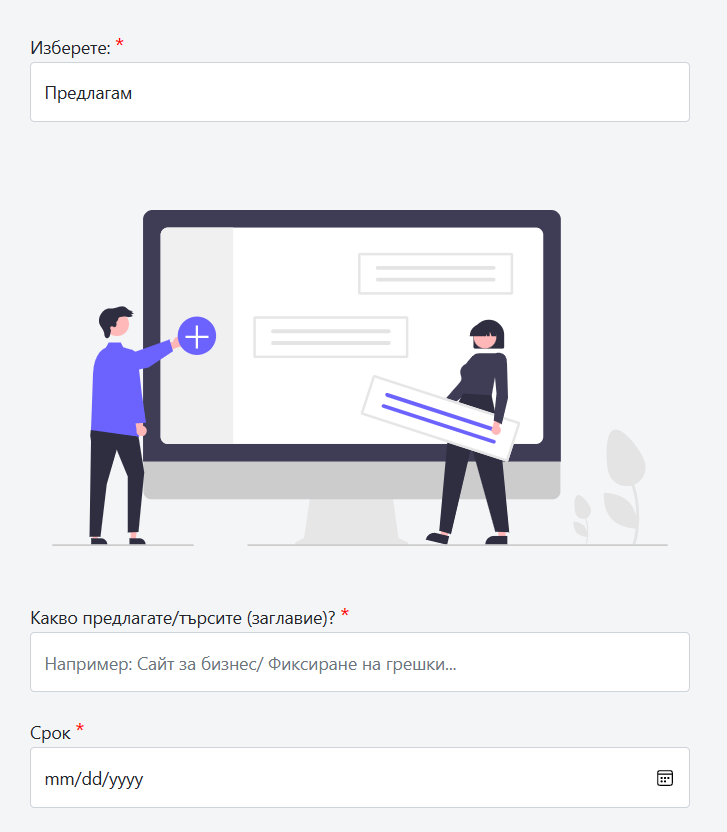
**Най-долната част на приложението (футър)** съдържа една малка навигация. Там се намира и полето за бюлетина, от който потребителят може да изпраща съобщение до администратора. Има и линкове на социалните медии на администратора.

**Изглед „Търся услуга“ /Job/Add.cshtml/**



Първо се избира от падащия списък вида на услугата, който в този случай е “Търся”. Задължителните полета за попълване имат червена звезда. В полето “Какво предлагате” се пише името на обявата. Задължително се добавя и крайната дата за предаване на работата. Цената е в българска парична единица, лева. Описанието трябва да е възможно дълго и ясно. Най-важното поле е за категорията. С помощта на категорията, търсещите работа хора, лесно ще могат да се ориентират.

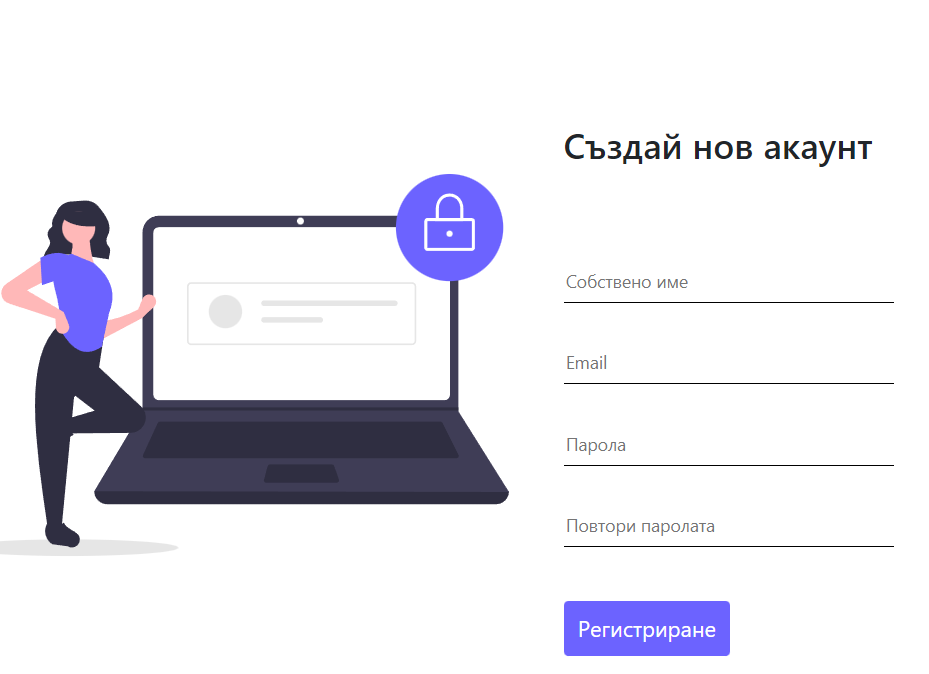
**Изглед „Предлагам услуга“ /Job/Add.cshtml/**



Създаването на заявки става от бутона “Създай заявка”. Тук всички полета са задължителни. Първо се избира вида на услугата, който трябва да е “Предлагам”. После избира се категория, добавя се заглавие, сума, описание, срок и се кликва върху бутона “Създай заявка”.

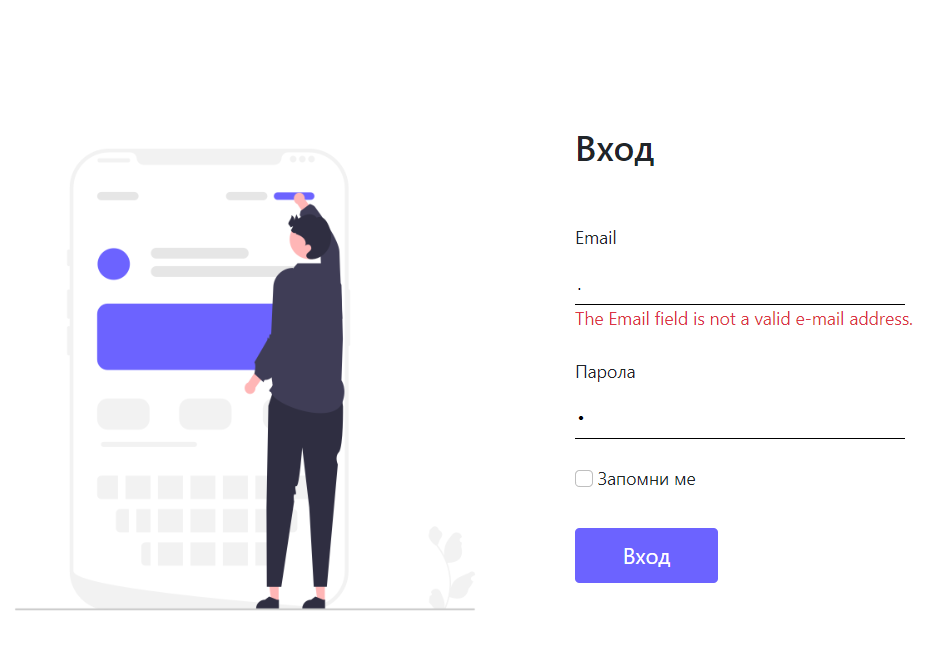
Създадените заявки се появяват при другите обяви, но в подсекцията „Заявки“. Заявките са необходими ако потребителят не може да намери услуги, които му трябват. Единственото важно изискване при създаването на заявки е да не се повтарят, за да не станат нежелани от хората. При такава ситуация потребителят може да изпрати писмо до администратора на сайта, за да блокира акаунта на спамващия потребител. Писмата се проверяват лично от програмиста на сайта, който е и администратора.

**Изглед „Регистрация“ /Register.cshtml/**



Регистрацията в сайта става след създаването на акаунт. Формулярът изисква собствено име, имейл и парола, след което акаунтът се създава. Ако потребителят има създаден акаунт може да си влезе направо в него.

**Изглед „Вход“ /Login.cshtml/**



Страницата за вход се използва, за да позволи на потребителя да влезе в уебсайта и приложението.

Приложенията/уебсайтовете идентифицират потребителите с тяхното потребителско име и парола. Страницата за влизане приема въведените от потребителя данни и предава данните към програмата от страната на сървъра.

Логването в сайта става след написването на имейла и паролата. Ако потребителят веднъж се е логнал, сайтът го помни и се улеснява входът. Ако потребителят не си въведе имейла и паролата, под полетата се появяват червени подсказки.

***Контролери***

***Междинният слой*** включва контролери и екшъни, чрез които се осъществява предаването на данни между презентационния слой и базата данни, както и работата на приложението като цяло.

* *Category Controller* - контролерът за категориите. Съдържа CRUD операциите за създаване, редактиране и изтриване.
* *Job Controller* - главният контролер, който съдържа View-тата за определените секции в сайта и CRUD операциите.
* *Request Controller* - Всички операции за създаване на заявка.

**Екшъни –** отговорни са за изпълнението на заявките и генерирането на отговори към тях. По подразбиране екшънът генерира отговор под формата на IActionResult. Действията обикновено имат съпоставяне едно към едно с потребителските взаимодействия. Някои от екшъните отговарят за CRUD операциите. Операцията CREATE в приложението е реализирана чрез екшъна Add(), UPDATE – чрез екшъна Edit(), който прочита записаните данни в базата, променя ги и ги записва отново в БД. Операцията Delete е за изтриване на записани данни от базата с командата .Remove().

* **Методи на заявките**

HTTP определя 8 различни клиентски метода за заявки:

* **HEAD** – иска изпращане на заглавията отговарящи на посочения с URL ресурс. Отговорът на сървъра е идентичен с този на GET, но е с липсващо тяло. По този начин може да се вземе само метаинформацията, свързана с ресурса, спестявайки трансфера на целия файл.
* **GET** – с него клиентът прави заявка за ресурс, зададен чрез URL. Могат да се изпращат и ограничено количество данни, закодирани директно в самия URL (отделени чрез въпросителен знак).
* **POST** – позволява клиентът да изпрати данни на сървъра. Тази заявка обикновено се генерира при изпращането на уеб формуляр, а данните могат да бъдат: текст, написан от потребителя във формуляра; файл на клиентския компютър и др.
* **DELETE** – изтрива посочения ресурс.

Тези методи се пишат в тага <form method=” POST/GET…”>…</form>.

**4.5. Реализация**

1) Използвана литература:

* ASP.NET MVC - https://dotnet.microsoft.com/en-us/apps/aspnet/mvc
* HTML и CSS – w3schools.com, SoftUni;
* Bootstrap - Bootstrap.com;
* JavaScript - w3schools.com, SoftUni,;
* MSSQL - SoftUni
* Entity Framework Core - https://docs.microsoft.com/en-us/ef/core/

2) Среди за разработка: Microsoft Visual Studio Community, SSMS(Sql Server Management

Studio)

3) Платформи на реализация: MS SQL Server, IIS Express Server

4) Използвани технологии:

**HTML –** за изграждане структурата на сайта.

**CSS –** за стилизиране на сайта и за визуалните ефекти.

**Bootstrap –** съкратен CSS, който е използван при създаването на таблици в сайта и др.

**JavaScript –** прости пресмятания като рождената дата на администратора и по-напредналите визуални ефекти.

**C# -** CRUD операции, създаване на моделите за БД, контролери и екшъни от междинния слой.

**ASP.NET** **MVC – Модел „Model-View-Controller” -** улеснява работата с дизайна и базата данни.

**EntityFrameworkCore –** провайдър на услуги за работа с БД;

**Подход CodeFirst –** използван е при създаването на базата данни.

**Razor View Engine –** Позволява комбинация между C# и HTML. Позволява предаване на данни между изгледа и контролера.

Идеята зад Razor е да предостави оптимизиран синтаксис за генериране на HTML, използвайки фокусиран върху кода шаблонен подход, с минимален преход между HTML и код.

**4.6. Описание на приложението**

Приложението когато не е хостнато се стартира чрез Visual Studio. Трябва да има и инсталиран SSMS. Без тях стартирането на уеб приложението е невъзможно. Преди стартирането трябва да се направи и миграция. Тя разрешава да се правят промени в базата данни, без да пипаме самата база. Но когато е хостнато, достатъчно е да се имаме само едно устройство, чрез което ще се стартира уеб сайта.

Първоначално ще се появи началната страница, която е с цел информиране и рекламиране. Фютърът и навигацията са публично достъпни. Навигацията е проста и лесно използваема. В секция обяви са публикуваните оферти и потребителят има право да си създаде заявки и оферти без ограничения, стига да не са спамове и изискващи изтривания, което води до блокирането на акаунта.

Функциите на приложението са следните:

**„Публикуване на обява“**- При добавяне, потребителят е задължен да попълва всички полета които имат червена звезда \*. Тези полета са:

„Какво предлагате?“, тоест наименованието на обявата;

„Крайна дата за предаване“ или „Валидност на самата обява“;

„ Цена “;

„Описание“, като описанието трябва да е възможно детайлно и подробно без правописни, пунктуационни, лексикални и граматични грешки за да изглежда по-професионално и да извлече повече внимание;

„Снимка“ – тя трябва да е оригинална и собствена;

„Сравнителни пакети“ – те дават възможност на купувачите да си изберат подходяща за тях, цена с плюсове и минуси, които също зависят от цената. Тези плюсове и минуси например са броят на ревизиите, по-бързо предаване, допълнителна информация, която включва в себе си изготвяне на лого и подобни неща.

След попълването на тези полета, се избира бутонът „Създай обява“ и след секунди обявата се появява в секция „Обяви“. В тази секция са всички споделени оферти и обяви, които съдържат кратко представяне на цялата информация. Достъпването на цялата информация става след кликването на бутона „Детайли“, който се намира в най-долната част на описателната картичка на обявата.

„**Услуги**“ - включва в себе си категориите и кратка информация под тях за наличните услуги.

„**За нас**“ - включва информация за администратора, реклама на социалните му медии и процеса на обработка на самия сайт, използвани езици, технологии и платформи. „**Контакти**“- включва себе си един формуляр за изпращане на съобщение до администратора със следните полета: Име на изпращащия, Имейл на изпращащия, заглавие на наличния проблем и поле за въвеждане на съобщение, който няма ограничение на знаци.

„**Портфолио**“ е падащо меню с две опции, които са „Моите поръчки“ и „Моите заявки“. В първата опция са поръчките, в които се отбелязва напредъка, а във втората опция се намират заявките създадени от мен. Създаването е реализирано по следния начин: След натискането на бутона „Създай заявка“, се препраща към друг HTML файл, който е стилизиран с CSS, Bootstrap, JavaScript и функциониращ със C#. Той включва падащо меню за избиране от наличните категории, поле за въвеждане на заглавие, сума, описание и срок. След извършването на всичките нужни функции, заявката се появява в Index-a.

Създаването на акаунт става от бутона “Регистрация„ в публичната навигация и е напълно безплатно, бързо и изключително лесно, а ако имате вече съществуващ акаунт, можете да влезете в профила си от публичния бутон „Вход“.

**4.7. Заключение**

Приложения от този вид са много, но няма налични сайтове, които са на български език и с паричната единица „лева“.

Сайтът може да се подобри като се добавят и други парични единици и други умения от предлагащите работа.