**Тема: Уеб приложение за**

**приемане и предлагане на**

**услуги**

(Freelancer)

**Автор**:

**Имена**: Сечкин Салим Рахим

**ЕГН:** 0452……

**Адрес:** с. Живково, ул. Калоян №43,

общ. Хитрино, обл. Шумен

**Телефон:** 0895073…

**Имейл:** [seckins191@gmail.com](mailto:seckins191@gmail.com)

**Училище:** ПГМЕТТ „Христо Ботев“, гр. Шумен

**Клас: 12-а**

**Ръководител:**

**Имена:** Недялка ….. Йорданова

**Телефон: ..........**

**Имейл:**

**Длъжност:**

**Резюме**

**Цели:**

Целта на уеб приложението е да помага на хора, които имат определени знания и искат да печелят пари, но не знаят от къде и как да започнат. Сайтът позволява качване и приемане на обяви, създаване на специална заявка, комуникация между продавача и купувача, комуникация с администратора. Продавачът може да види напредъка на купувача, но само ако купувача го отбележи в самата обява. Отбелязването става в проценти. Купувачът има право да избере допълнителни услуги от сравнителните таблици, според зависи цената. Ревизията, времето за предаване и другите неща зависят от цената.

**Основни етапи:**

1.Идеен проект

2.Работа по дизайна

3.Проектиране на Базата данни

4.Свързване на Уеб приложението с База данни

5.Тестване

6.Публикуване в интернет

**Ниво на сложност на проекта:**

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

**Логическо и функционално описание на решението**

**-----БД, Снимка, описание на навигацията, контролерите и т.н**

**Реализация**

Уеб приложението е лесно за използване с проста навигация и с лесно разбираеми опции. Човек, който може да използва социални медии като Facebook, Instagram и други подобни сайтове, той ще може да управлява и този сайт. Функциите са следните: качване на обява - такава каквато трябва на Вас или на другите. При добавяне, добавящият е задължен да попълва всички полета които имат червена звезда \*. Тези полета са „Какво предлагаш?“, 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 внимания; „Снимка“ – тя трябва да е оригинална и собствена, а не открадната от интернет; „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. Следващата секция е „Услуги“, който включва в себе си категориите и кратка информация под тях за наличните услуги. Секция „За нас“ включва информация за администратора, реклама на социалните му медии и процеса на обработка на самия сайт, използвани езици, технологии и платформи. Следващата публична секция от навигацията е „Контакти“, който включва себе си един формуляр за изпращане на съобщение до администратора със следните полета: Име на изпращащия, И-мейл на изпращащия, заглавие на наличния проблем и поле за въвеждане на съобщение, който няма ограничение на знаци. Секцията „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 им е доста просто. След натискането на бутона „Създай заявка“, се препраща към друг HTML файл който е стилизиран с CSS, Bootstrap, JavaScript и функционализиран със C#. Той включва падащо меню за избиране от наличните категории, инпут поле за 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“.

Приложението е програмиран с езиците HTML, CSS, Bootstrap, JavaScript, MVC .Net Core и C# във Visual Studio 2022.

**Използвани технологии:**

**HTML**-ът е използван за скелета (строежа) на самия сайт. С езика CSS е оформен дизайнът, с JavaScript са направени по-напредналите ефекти и някои прости математически пресмятания, като пресмятане на рождена дата. А със C# са направени операциите, като например Качване на нова обява, редактиране на вече съществуваща обява и изтриване на както съществуваща и редактирана, така и съществуваща но не редактирана обява.

**MVC -** MVC е съкратено от “Model-view-controller”. Представлява архитектурен шаблон при програмния дизайн. Той отговаря за разделянето на бизнес логиката на три взаимосвързани части: “Model”, “View” и “Controler”. Какво представляват те и какви са техните функции?

* „**Model“ (Модел)** – това представлява „ядрото“ на приложението ви. В него се записват всички данни, върху които искаме да работим; ядрото на приложението, предопределено от областта, за която се разработва; обикновено това са данните от реалния свят, които се моделират и над които се работи – въвеждане, промяна, показване и т.н. Трябва да се прави разлика между реалния обкръжаващ свят и въображаемият абстрактен моделен свят, който е продукт на разума, който се възприема като твърдения, формули, математическа символика, схеми и други помощни средства.
* **„View“(Изглед)** – най-общо казано това е изходна част от софтуера, която визуализира наличните, обработени данни; тази част от изходния код на приложението, отговорна за показването на данните от модела. Например изгледът може да се състои от PHP шаблонни класове, JSP страници, Asp страници. Зависи от това какъв графичен инерфейс се прави и каква платформа се използва;
* **„Controller“(Контролер)** – това е тази част от библиотеката (library) или самия код, която извиква допълнителни методи върху събраните в модела данни и по този начин те могат да бъдат обработвани. Тази част от сорс кода (клас или библиотека), която взима данните от модела или извиква допълнителни методи върху модела, предварително обработва данните, и чак след това ги дава на изгледа.

**CSS** (**Cascading Style Sheets**) е език за описание на стилове (език за стилови файлове, style sheet language) – използва се основно за описание на онлайн представянето на уеббазиран документ, който написан на език за маркиране. Най-често се използва допълнително към чистия HTML, но се прилага и върху XML уебстраници и документи

**Bootstrap** - Основната полезност на използването на bootstrap е да се намали пристрастието в рамките на анализа или, с други думи, да се сближи дисперсията чрез извършване на произволно преизбиране на първоначалната извадка, а не на популацията. По този начин изграждането на статистически модели се улеснява чрез създаване на доверителни интервали и тестове за хипотези.

**JavaScript** - позволява създаването на интерактивни и адаптивни интерфейси с много добър UI дизайн и динамични функционалности, изцяло с насоченост към крайния потребител. Функциите на JavaScript са се разширили значително през годините.

**ASP.NET Core** - модуларна софтуерна рамка, която може да върви както на пълната .NET рамка, така и на крос-платформената .NET Core. Въпреки, че е нова софтуерна рамка, изградена върху нов web stack, тя има висока степен на съвместимост с ASP.NET MVC.

**C#** - **обектно-ориентиран език**, т.е. той изгражда системи посредством обекти, които взаимодействат помежду си. Обектно-ориентираното програмиране се отличава от традиционното програмиране, при което програмата е просто списък от инструкции, които компютърът изпълнява.

***Подход CodeFirst***

При изработването на уеб приложението е използван подходът CodeFirst, при който ние пишем кода, а таблицата (базата данни) се генерира автоматично.

**Какво е база данни?**

Представлява колекция от логически свързани данни в конкретна предметна област, които са структурирани по определен начин. В първоначалния смисъл на понятието, използван в компютърната индустрия, базата от данни се състои от записи, подредени систематично, така че компютърна програма да може да извлича информация по зададени критерии. Например БД може да се използват в моделирането на хотелските системи, за да се проверява дали има налични свободни стаи в даден хотел.

**CRUD операции**

В проекта CRUD операциите са използвани при обявите за качване, прочитане, редактиране и изтриване на записи. CRUD парадигмата е обичайно явление при създаването на уеб приложения, защото осигурява начин, който да напомни на разработчиците как да конструират пълноценни и полезни използваеми модели. В този контекст, CRUD абревиатурата означава **Create, Read, Update, Delete**. Това са четирите основни функции, свързани с т.нар. **persistent storage,** т.е. съхранение на данни, при което данните се запазват и след като захранването към устройството за съхранение бъде прекъснато. Повечето приложения разполагат с някаква форма на CRUD функционалности и на практика всеки програмист работи с такива в даден момент. Такива приложения имат **три основни съставки** – API, база данни, която съхранява информацията и позволява на потребителите да я достъпят, и UI, чрез който да го направят. А в следващите редове ще разгледаме и самите операции една по една.

## **Как работят отделните CRUD операции със SQL?**

Всяка една от буквите на акронима може да бъде свързана с SQL, затова двете вървят ръка за ръка, когато работиш с релационни бази данни. А действието на операциите е сравнително очевидно:

* **CREATE операции** – операцията създава нов запис, например нов акаунт или задача. В SQL база данни, за да създаваш, ще използваш INSERT.
* **READ операции** – чрез тях, използвайки потребителския интерфейс, можеш да достъпиш и видиш създадените записи. В релационни бази данни ще използваш SELECT, за да избереш запис.
* **UPDATE операции** – чрез тях може да променя и обновява съществуващите записи, отново посредством наличния UI. В релационна база данни ще изполни UPDATE.
* **DELETE операции** – чрез тях може да се премахва записи от базата данни.
* **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.
* **PUT** – качва файл, който в бъдеще ще отговаря на посочения URL
* **DELETE** – изтрива посочения ресурс.
* **TRACE** – сървърът връща получената заявка със статус OK. Позволява да се провери в какъв вид пристига заявката при сървъра и дали (и как) е била модифицирана по трасето от междинни прокси сървъри.
* **OPTIONS** – сървърът трябва да отговори с поддържаните от него клиентски методи, съответстващи на зададения URL, или с поддържаните от сървъра методи като цяло, ако е зададено \* вместо URL.
* **CONNECT** – използва се при комуникация през прокси.

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

**Razor View Engine**

Позволява комбинация между C# и HTML. Позволява предаване на данни между изгледа и контролера. Когато се пише код на C# в HTML-а, пред кода се слага кльомба (**@**).

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. Дизайнът намалява броя на знаците и натисканията на клавиши и позволява по-плавен работен процес на кодиране, като не изисква изрично обозначени сървърни блокове в HTML кода. Други предимства:

* Поддържа IntelliSense – поддръжка за завършване на изрази
* Поддържа „оформления“ – алтернатива на концепцията „главна страница“ в класическите уеб формуляри (.aspx)

# **Какво трябва да се избягва в уеб дизайна?**

* **Бъркотия:**

Неорганизираните и хаотични сайтове шашкат хората и ги карат да се чувстват глупаво. Когато един потенциален клиент пристигне на интернет страницата Ви, той търси ценна информация. Потенциалният клиент не търси пъзел, който да разреши. Точно поради тази причина се нуждаете от добре изглеждащ, добре работещ и балансиран уеб дизайн. Това се постига чрез правилното използване на пространствата, показване на най-важната информация - тази, която знаете, че клиентите търсят. Предоставянето на тези материали е от огромно значение, защото на база тях клиентът ще вземе своето решение. Дезорганизираните уеб сайтове изглеждат непрофесионално и могат само да навредят върху репутацията на Вашия бизнес. Добрият сайт е като огледало на целите, които са поставени пред Вашата компания.

* **Купища графични елементи:**

Отличният уеб сайт е този, който е създаден от гледна точка на интерактивност и ползваемост. Уеб сайт, който има труден за схващане дизайн или прекалено сложна навигация, няма да успее да извлече дивиденти. Уеб сайтовете с прекалено много графични елементи или тромави анимации могат значително да забавят времето за зареждане. Друг особен момент е използването на шрифтове, които може да изглеждат много интересни, уникални, но в същото време да са трудни за четене, а също така и да не се отварят през някои браузери.

* **Прекалената креативност и шеги:**

Задачата на един бизнес уеб сайт е да представя компанията Ви в интернет и да продава. Сайтът предоставя, по много удобен за потребителите начин, информация за Вашите услуги или стоки. Понякога е лесно да се отнесеш във вихъра на креативността, в опитите да покажеш чувство за хумор. Но помнете, че това не е универсална формула за всеки бизнес. С подобен креативно-комичен подход може да накарате потенциалните клиенти да се замислят относно Вашата сериозност, отговорност и намерения.

Най-важната част от сайта е началната страница. Началната страница на уеб сайта е МЯСТОТО, превличащо внимание. Точно на него се събират Вашите потенциални клиенти. Това е отправната точка, от където започва всичко и в зависимост от интелигентната му структура - завършва с провал или продажба. Началната страница трябва на мига да предава информация, която да прихване вниманието на потребителя. Трябва да кажете на сърфиращия какво точно предлагате. Това може да стане чрез подходящи изображения, видеа, типография.

**Другите най-важни фактори в един сайт са:**

* Бързото зареждане;
* Сайт печелещ доверието;
* Проста навигация;

**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 базата данни, без да пипаме самата база. Но когато е хостнато, достатъчно е да се имаме само едно устройство, чрез коети 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 лесноизползваема. 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.

**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 едини и други езици.