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

**ДИПЛОМЕН ПРОЕКТ**

На Боян Маденов Александров

ученик/ученичка от XII клас

професия-код: **481030, „Прилоцен програмист“**

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

**Тема: Разработване на чат-бот под формата на правен асистент „IBM Watson Chat Bot“**

**Ръководител-консултант: Кирил Поповски**

Сесия: май-юни 2024г.

Дата:29.04.2024

**СЪДЪРЖАНИЕ**

[1. Увод 3](#_Toc165211399)

[1.1. Описание 3](#_Toc165211400)

[1.2. Мотивация 3](#_Toc165211401)

[1.3. Основна цел 5](#_Toc165211402)

[1.4. Очаквания 6](#_Toc165211403)

[1.5. Предизвикателства 7](#_Toc165211404)

[2. Изложение 9](#_Toc165211405)

[2.1. Описание на използвани софтуерни технологии 9](#_Toc165211406)

[2.1.1. **Уеб Приложение** 9](#_Toc165211407)

[2.1.2. **C#** 10](#_Toc165211408)

[2.1.3. **HTML** 12](#_Toc165211409)

[2.1.4. **CSS** 13](#_Toc165211410)

[2.1.5. **JavaScript** 14](#_Toc165211411)

[2.1.6. **Bootstrap** 15](#_Toc165211412)

[2.1.7. **ASP .NET** 16](#_Toc165211413)

[2.1.8. **ASP .NET MVC** 17](#_Toc165211414)

[2.1.9. **MySQL** 20](#_Toc165211415)

[2.1.10. **ASP.NET Core MVC Auth SDK** 23](#_Toc165211416)

[2.1.11. **IBM Watson** 24](#_Toc165211417)

[3. Заключение 26](#_Toc165211418)

[4. Информационни източници 27](#_Toc165211419)

[5. Приложения 28](#_Toc165211420)

# **Увод**

## **Описание**

Живеем във времена на бързо сменящи се тенденции, бързо развиващи се технологии и бърз начин на живот. Всичко това води до грешки, обърквания и злополуки, с които рано или късно всеки един от нас се среща. Нарастващ проблем при новите, както и при старите поколения, е че с бързите промени идват и съмненията, не знаем какво можем и какво не можем да направим в случай на инцидент. Mr. Lawyer е проект, подпомагащ за бърз достъп до информация при несигурност.

Проектът предоставя лесно за ползване уеб приложение, включващо интригуващ потребителски интерфейс (UI), архитектура на данни и най-важното защитата им. Mr. Lawyer e виртуален правен помощник, революциониращ начина, по който хората взаимодействат с правната информация онлайн. Уеб приложението, съчетава в себе си последните технологии с експертното знание на правните професионалисти, предоставяйки лесно точна и достъпна правна информация. Приложението е разработено с помощта на ASP.NET MVC framework и захранван от IBM Watson AI.

В основата си Mr. Lawyer е чатбот, безпроблемно интегриран в удобен за потребителя интерфейс. Независимо дали се ориентирате в сложна правна терминология или търсите насоки по конкретен правен въпрос, Mr. Lawyer е тук, за да Ви помогне 24/7. Със своя интуитивен дизайн и възможности за обработка на естествен език, взаимодействието с Mr. Lawyer се чувства като разговор с опитен правен експерт.

## **Мотивация**

Проектът Mr. Lawyer се ръководи от дълбоко вкоренена мотивация за иновации в правната индустрия, насочени към няколко критични фактора, влияещи върху достъпността и ефективността на правните услуги.

Първо, достъпността е сериозен проблем, тъй като традиционните правни услуги могат да бъдат финансово и логистично недостъпни за много лица и малки предприятия. Mr. Lawyer има за цел да разруши тези бариери, като предлага безплатна и лесно достъпна платформа, където потребителите могат да получат надеждни правни насоки без ограничения.

Второ, сложността на юридическия език представлява значително предизвикателство за тези без юридическо обучение. Mr. Lawyer се стреми да опрости тази сложност, като предоставя удобен за потребителя интерфейс и използва възможностите за обработка на естествен език. Това позволява на потребителите да задават въпроси на обикновен език и да получават ясни, разбираеми отговори, което прави правната информация по-разбираема и достъпна за по-широка аудитория.

Освен това ограничената наличност на традиционните правни услуги през определени работни часове може да възпрепятства способността на хората да потърсят незабавна помощ, когато е необходимо. Mr. Lawyer се справя с това ограничение, като предоставя денонощен достъп до правна информация, като гарантира, че потребителите могат да получат навременни насоки, когато имат нужда от тях, повишавайки удобството и достъпността.

Бързият напредък на AI технологията предоставя уникална възможност за трансформиране на предоставянето на правни услуги. Използвайки мощта на IBM Watson AI, Mr. Lawyer може да анализира огромно количество правни данни и да предоставя точни отговори в реално време. Това не само подобрява ефикасността на правната помощ, но също така гарантира, че потребителите получават актуална и подходяща информация.

Освен това Mr. Lawyer има за цел да даде възможност на физическите лица и бизнеса, като ги оборудва със знанията, от които се нуждаят, за да вземат информирани решения и да се справят уверено с правни предизвикателства. Информираното вземане на решения е от решаващо значение по правни въпроси и Mr. Lawyer се стреми да предостави на потребителите необходимите инструменти и ресурси, за да разберат напълно своите права и задължения.

Един от най-важните аспекти на една добре развиваща се страна е нейното младо поколение и именно затова е от съществено значение е дасе инвестира в образованието и осведомеността му. Аз като част от него всеки ден наблюдавам околните и забелязвам, че ние сме недообразовани в сферата на личните ни права. Независимо от ситуацията, ние трябва да бъдем запознати с основните аспекти на закона.В противен случай, рискуваме да бъдем използвани.

Вземайки за пример пътната проверка, като нов шофьор от личен опит мога да кажа, че повечето от нас не знаем правата си, когато ни спре представител на полицията, несигурността относно какво можем да правим или какъв е нашият статут, често ни прави безпомощни и ни излага на възможността от злоупотреба. Това колкото ме притеснява, толкова ме и мотивира да разработя този проект, който с една проста концепция може да реши един голям проблем в нашето общество.Mr. Lawyer е колкото проект, толкова и инициатива да се даде гласност на това, че всеки от нас има статут.

## **Основна цел**

Основната цел на проекта Mr. Lawyer е да демократизира достъпа до правна информация и експертиза. Чрез използване на технологии като ASP.NET MVC и IBM Watson AI, проектът се стреми да предостави безплатна, достъпна и надеждна платформа, на която физически лица и бизнеси могат да получат точно правно ръководство.

Стреми се да създаде лесно за ползване и достъпно уеб приложение, което да позволява на ползващите го да достъпят информация относно закона за кратко време в случай на ситуация развиваща се на момента. Ако си представим един закъсняващ полет, при търсене на компенсация, ще можем лесно и бързо да проверим какво ни се полага като обезщетение с Mr. Lawyer. С цел пазене на историята на чатовете, проектът поддържа идентификация (акаунти), като други основни цели са лесен за окото дизайн, с лесно достъпни функционалности.

Чрез потребителски интерфейс, който е лесен за използване, и възможностите за обработка на естествен език, Mr. Lawyer цели да опрости сложните правни концепции и термини, правейки правната информация по-разбираема и достъпна за по-широка аудитория. Най-високата цел на проекта е да даде сила на потребителите, като ги запознае със знанията, от които се нуждаят, за да вземат информирани решения и да се справят с правните предизвикателства с увереност.

A blue robot with a phone

Description automatically generated with medium confidence

## **Очаквания**

Проектът Mr. Lawyer очаква да постигне няколко ключови резултата. На първо място, той предвижда значително увеличение на достъпа до правна информация и експертиза за физически лица и бизнеси, като премахва бариерите пред правната помощ. Този достъп се очаква да бъде насочен към разнообразни групи потребители, включително лица, търсещи правни съвети, малки бизнеси, които се справят с правни сложности, и правни специалисти, търсещи бързи справки.

Освен това, проектът очаква широко приемане, като цели да стане основен източник на информация в правния домейн. Той предвижда предоставяне на безпроблемно и интуитивно потребителско изживяване, като отдава приоритет на удовлетворението на потребителите чрез удобна платформа и точни отговори. Непрекъснатото развитие е също ключова точка на очакванията, с ангажимент за изграждане на платформата въз основа на обратна връзка от потребителите и итеративни актуализации, за да се подобрят възможностите му с времето.

Доверието и надеждността са от основни очаквания, като се фокусира върху предоставянето на точна, своевременна и поверителна правна помощ. Чрез отдаване на приоритет на защитата на данните и поверителността, Mr. Lawyer цели да внуши доверие в потребителите, гарантирайки, че те се чувстват уверени да търсят съвет по чувствителни правни въпроси.

В крайна сметка, проектът цели да оказва значително влияние върху правния пейзаж, като демократизира достъпа до правна информация и осигурява на физически лица и бизнеси възможността да се справят с правни предизвикателства с увереност. Чрез използване на технологии и иновации, Mr. Lawyer цели да допринесе за по-справедлив и информиран правен екосистем, изпълнявайки своята мисия за революционизиране на предоставянето на правни услуги.

## **Предизвикателства**

При разработката на проекта Mr. Lawyer се сблъсках с няколко предизвикателства, които обхващаха технически, логистични и концептуални аспекти.

Първоначално, интегрирането на IBM Watson AI предизвика някои трудности, които включваха разбиране на възможностите на платформата за изкуствен интелект, безпроблемно интегриране с ASP.NET MVC рамката и точно настройване на алгоритмите за обработка на естествен език, за да се осигури точност и релевантност на отговорите.

Второ, гарантирането на поверителността и сигурността на данните беше от основно значение. Беше необходимо внедряване на здрави протоколи за криптиране, контроли на достъпа и техники за анонимизиране на данните, за да се защитят чувствителните правни информации и да се запази доверието на потребителите.

Поддържането на точността на правните отговори се оказа постоянно предизвикателство. Изискваха се обширни изследвания, валидация на правни източници и доработка на алгоритмите за изкуствен интелект, за да се осигури надеждно ръководство за различни правни теми.

Проектирането на интуитивен потребителски интерфейс предизвика собствен набор от предизвикателства, изискващи баланс между простота и функционалност, оптимизация за различни устройства и включване на обратна връзка от потребителите за непрекъснато подобряване.

Разработването на система за обработка на естествен език, способна да разбира точно потребителските запитвания, изискваше обучение на моделите на изкуствения интелект върху големи правни текстове, подобряване на разбирането на езика и адресиране на двусмислеността в потребителските входове.

Тестовете и процесите за осигуряване на качество бяха от съществено значение за гарантирането на надеждност и производителност под различни сценарии, включително крайни случаи, мащабируемост и съвместимост с различни платформи.

Спазването на правните и регулаторни изисквания, като законите за защита на данните и регулациите, уреждащи предоставянето на правна информация, добавяше допълнителна сложност. Спазването на съответните рамки, като се поддържат целите на проекта, беше от съществено значение.

Въпреки тези предизвикателства, сътрудничеството, иновациите и итеративното развитие позволиха на проекта Mr. Lawyer да преодолее трудностите. Чрез справяне с техническите сложности, доразвиване на потребителското изживяване и отдаване на приоритет на защитата на данните и правната точност, проектът успешно достави ценен ресурс за достъп до правна информация и помощ.

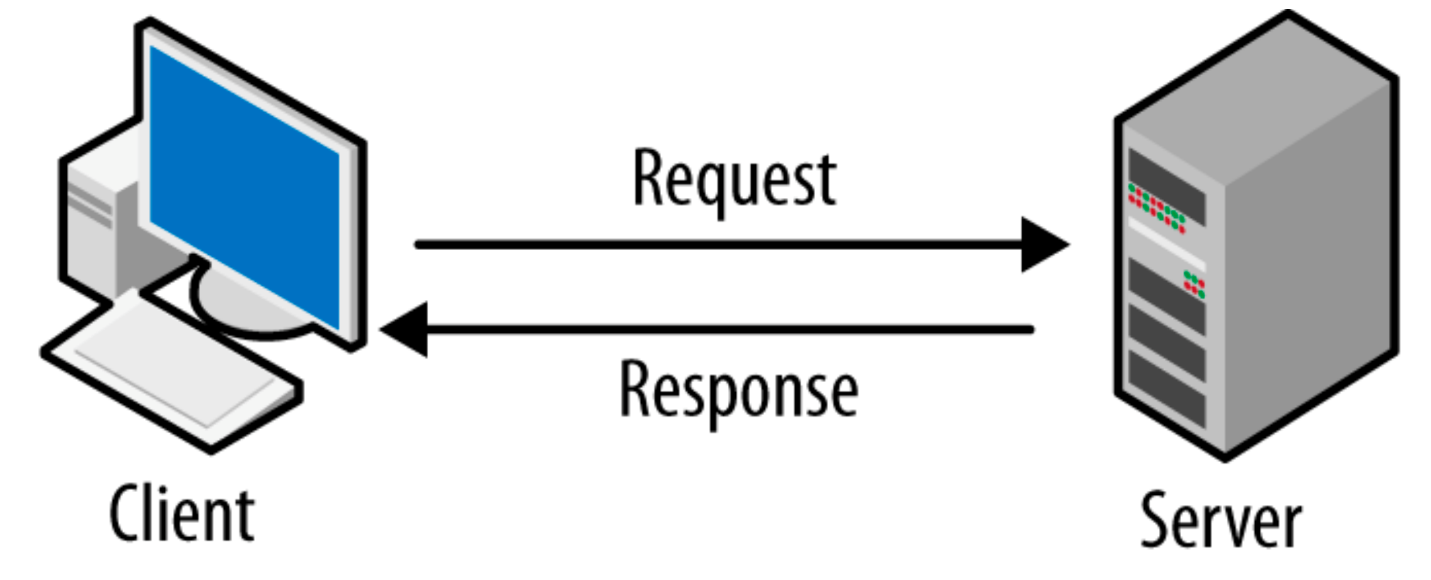
# **Изложение**

## **Описание на използвани софтуерни технологии**

### **Уеб Приложение**

Уеб приложение, е софтуерно приложение, което е достъпно и управлявано изцяло чрез уеб браузър или по-точно казано през интернет. За разлика от традиционните софтуерни приложения, които изискват инсталиране на устройството на потребителя, уеб приложенията се хостват на отдалечени сървъри и се осъществяват достъп чрез уеб браузър, което ги прави независими от платформата и достъпни от всяко устройство с интернет връзка.

Уеб приложенията следват модел на архитектура клиент-сървър, където логиката и данните на приложението се намират на отдалечен сървър (backend), а потребителят взаимодейства с приложението чрез уеб браузър (клиент). Тази архитектура позволява безпроблемна комуникация между клиента и сървъра, като клиентът изпраща заявки до сървъра и получава отговори, съдържащи исканите данни или функционалност. Потребителският интерфейс на уеб приложение се представя на потребителя чрез уеб браузър.



Състои се от HTML (Hypertext Markup Language), CSS (Cascading Style Sheets) и JavaScript, които заедно определят структурата, стила и интерактивността на интерфейса на приложението. Съвременните уеб приложения често включват усъвършенствани UI framework-ове и библиотеки като в нашия случай ASP .NET MVC за създаване на динамични и отзивчиви потребителски интерфейси.

Бекендът на уеб приложение се състои от код от страна на сървъра, отговорен за обработката на клиентски заявки, изпълнението на бизнес логиката и взаимодействието с бази данни или външни услуги. Обичайните технологии, използвани в backend разработката, включват езици за програмиране като JavaScript (Node.js) и framework-ове като Express.js, Django, Flask, Rails и Firebase. Бекендът отговаря за задачи като удостоверяване на потребителя, валидиране на данни и операции с бази данни. Уеб приложенията често изискват постоянно съхранение на данни, за да управляват потребителска информация, данни за приложения и други ресурси. Базите данни се използват за ефективно съхраняване и извличане на данни.

Популярните избори за бази данни за уеб приложения включват релационни бази данни като в Mr. Lawyer и бази данни от типа NoSQL. Уеб приложенията разчитат на стандартни комуникационни протоколи като HTTP (Hypertext Transfer Protocol) и HTTPS (HTTP Secure), за да улеснят комуникацията между клиента и сървъра. HTTP се използва за предаване на данни между браузъра и уеб сървъра, докато HTTPS добавя слой на криптиране, за да защити предаването на данни, като гарантира поверителността и целостта на потребителските данни.

### **C#**

C# е основен елемент в съвременните програмни езици, известен със своята универсалност, здравина и мащабируемост. Разработен от Microsoft, C# въплъщава принципите на обектно-ориентираното програмиране (ООП), улеснявайки създаването на модулни и повторно използваеми структури от код. Неговата подкрепа за концепции като капсулация, наследяване и полиморфизъм позволява на разработчиците да проектират сложни софтуерни системи с лекота, насърчавайки организацията и поддръжката на кода.

Една от определящите характеристики на C# е акцентът върху типовата безопасност. Като статично типизиран език, C# изисква променливите да бъдат декларирани с техните типове данни по време на компилацията. Този подход не само подобрява надеждността на кода, като открива грешки в ранен етап от разработката, но и помага за четливостта и поддръжката на кода.

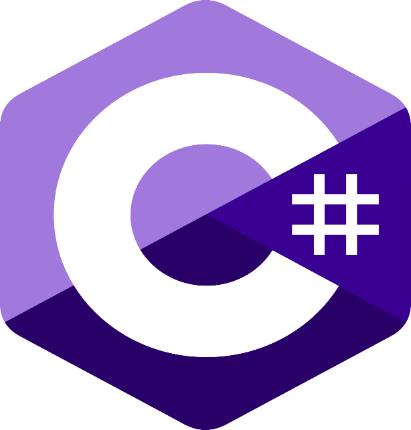
Управлението на паметта е още еднa област, в която C# блести. Чрез автоматичното управление на паметта, основано на механизма за събиране на отпадъци (garbage collection), C# управлява динамично разпределението и освобождаването на памет, намалявайки риска от утечки на паметта и други свързани с нея проблеми. Тази функционалност олекотява бремето на разработчиците, като им позволява да се съсредоточат върху създаването на надеждни приложения, без да се тревожат за ръчното управление на паметта.

C# разполага с богат набор от езикови функции, които допринасят за неговата изразителност и продуктивност. Генеричните типове позволяват създаването на многократно използуеми структури от данни и алгоритми, докато ламбда изразите улесняват създаването на кратки и гъвкави кодови конструкции. LINQ (Language Integrated Query) предоставя единен синтаксис за заявки към данни от различни източници, опростявайки манипулацията на данни.

Освен това C# се отличава с асинхронното програмиране, което е особено полезно за създаването на реактивни и мащабируеми приложения. С функциите async/await, разработчиците могат да пишат асинхронен код по по-четлив и лесно поддържаем начин.

В последните години C# е направил големи промени, за да подкрепи разработката на приложения за различни платформи. С появата на .NET Core (сега .NET), C# разшири своето влияние извън екосистемата на Windows, позволявайки на разработчиците да създават приложения, които работят без проблеми на Windows, macOS и Linux. Тази експанзия отвори нови възможности за разработчиците на C#, като им предостави възможност да създават разнообразни софтуерни решения за различни платформи.

Освен това, C# е дълбоко интегриран с .NET framework, който предоставя богат набор от библиотеки и инструменти за създаване на разнообразни приложения, включително десктоп софтуер, уеб приложения, мобилни приложения и други. Тази интеграция ускорява процесите на разработка и оптимизира общите задачи, позволявайки на разработчиците да предлагат висококачествени софтуерни продукти ефективно.



### **HTML**

HTML (HyperText Markup Language) служи като гръбнакът на световната мрежа, осигурявайки структурата и съдържанието за почти всяка уеб страница, която срещнете онлайн. Това е език за маркиране, който дефинира структурата на уеб документи чрез използване на система от тагове и атрибути за описание на елементите в една страница. В основата си HTML се състои от набор от таг-ове за маркиране, които определят структурата и съдържанието на уеб страница.

Чрез тях се форматира, графично оформя текста и неговите отделните части в рамките на една еб страница, като например заглавия, цитати, текстови раздели, хипертекстови препратки и т.н. Тези тагове са затворени в ъглови скоби (< >) и обикновено идват по двойки - отварящ таг и затварящ таг. Отварящият таг показва началото на елемент, докато затварящият таг обозначава края на този елемент. Някои елементи обаче са самозатварящи се и не изискват затварящ таг. HTML елементите могат да бъдат категоризирани в няколко типа: семантични елементи, елементи на текстово съдържание, мултимедийни елементи, елементи на формуляри и елементи на структурата на документа.

HTML документите обикновено са структурирани йерархично, с вложени елементи, образуващи дървовидна структура. Тази йерархична структура е от съществено значение за организиране и стилизиране на съдържанието на уеб страница с помощта на CSS (Cascading Style Sheets) и за добавяне на интерактивност и функционалност с помощта на JavaScript.



### **CSS**

CSS или Cascading Style Sheets е основна технология в уеб разработката, която допълва HTML чрез контролиране на представянето и оформлението на уеб документите.

Той позволява на разработчиците да стилизират и проектират уеб страници, осигурявайки последователност, естетика и приятно потребителско изживяване на различни устройства и размери на екрана. В основата си CSS се състои от набори от правила, които определят как HTML елементите трябва да се показват в браузъра.

Тези набори от правила се състоят от селектори, свойства и стойности. Селекторите са насочени към конкретни HTML елементи, докато свойствата определят визуалните характеристики на тези елементи, като цвят, размер, шрифт, разстояние и позициониране.

Стойностите определят специфичните настройки за всяко свойство. CSS предоставя широк набор от възможности за стилизиране на уеб съдържание, включително селектори, модел на кутия, типография, цветове и фонове, оформление, преходи и анимации и медийни заявки.

В обобщение, CSS е критична технология в уеб разработката, която позволява на разработчиците да стилизират и проектират уеб страници, да контролират оформлението и типографията, да прилагат цветове и фонове, да създават адаптивни оформления, да добавят преходи и анимации и да оптимизират потребителското изживяване на различни устройства и размери на екрана.

Използвайки силата на CSS, разработчиците могат да създават визуално привлекателни и ангажиращи уеб приложения, които завладяват потребителите и ефективно придават съдържание и идентичност на сайта.



### **JavaScript**

JavaScript е многофункционален език за програмиране, който обикновено се използва за създаване на интерактивно и динамично уеб съдържание. Като основен компонент на уеб разработката, JavaScript позволява на разработчиците да добавят функционалност, да манипулират данни и да реагират на потребителските взаимодействия в уеб приложенията.

JavaScript се използва предимно за скриптове от страна на клиента, което означава, че работи в уеб браузъра на потребителя, а не на отдалечен сървър. Това позволява на JavaScript да взаимодейства с Document Object Model (DOM), който представлява структурата на HTML и XML документи.

Чрез достъп до и модифициране на DOM, JavaScript може динамично да актуализира съдържанието и поведението на уеб страниците в отговор на потребителски действия, като кликвания, събития за въвеждане и събития за зареждане на страници.

Ключовите характеристики и възможности на JavaScript включват променливи и типове данни, оператори и изрази, оператори на потока на управление, функции, обекти и обектно-ориентирано програмиране (ООП), обработка на събития, асинхронно програмиране, обработка на грешки, модули и модуларизация и API на браузъра.

В обобщение, JavaScript е мощен език за програмиране, който позволява на разработчиците да създават интерактивни и динамични уеб приложения чрез манипулиране на DOM, обработка на потребителски взаимодействия, извършване на асинхронни задачи и достъп до API на браузъра. Със своята гъвкавост и широк набор от възможности, JavaScript играе централна роля в модерното уеб развитие, като дава възможност на разработчиците да изграждат ангажиращи и отзивчиви уеб изживявания.



### **Bootstrap**



Bootstrap е популярен фронтенд фреймуърк, предоставящ набор от инструменти за разработка на уеб приложения с модерен и отзивчив дизайн. Създаден от екипа на Twitter, Bootstrap е базиран на HTML, CSS и JavaScript и предлага готови компоненти, сетове от стилове и JavaScript плъгини, които значително улесняват процеса на създаване на уеб сайтове и уеб приложения.

Bootstrap предлага голям набор от готови компоненти като бутони, форми, навигационни менюта, карусели, модални прозорци и други. Тези компоненти могат лесно да бъдат интегрирани в уеб приложенията, без да се налага да се създават от нулата.

Освен това, Bootstrap е изцяло отзивчив, което гарантира, че уеб сайтовете и приложенията, създадени с него, автоматично се приспособяват към различни устройства и екрани. Това осигурява оптимално потребителско изживяване както на десктоп компютри, така и на мобилни устройства.

Също така, Bootstrap е създаден с модулна архитектура, която позволява лесно да се променят и допълват неговите компоненти и стилове. Разработчиците могат да използват само тези части от Bootstrap, които са им необходими, и да ги променят според конкретните си изисквания.

Bootstrap е съвместим с всички основни браузъри и гарантира, че уеб приложенията, създадени с него, ще работят коректно на различните браузъри на потребителите.

В контекста на ASP.NET MVC приложения, Bootstrap може да бъде използван за създаване на стилизиран потребителски интерфейс. Разработчиците могат лесно да интегрират Bootstrap във своите проекти, като добавят необходимите CSS и JavaScript файлове и използват готовите компоненти и стилове за стилизиране на различни части на приложението. Това позволява бързо и ефективно създаване на уеб приложения с модерен и привлекателен дизайн, който е отзивчив и съвместим с различни устройства и екрани.

### **ASP .NET**

ASP.NET (Active Server Pages .NET) е платформа за създаване на динамични уеб приложения, разработена от Microsoft. Тя предлага различни инструменти и технологии за изграждане на разнообразни уеб приложения, включително уеб сайтове, уеб услуги, апликации за социални мрежи и други.

ASP.NET използва компилация на кода преди зареждането на уеб приложението на сървъра, което води до по-бързо и по-ефективно изпълнение. Платформата предлага множество вградени механизми за защита на уеб приложенията от различни видове атаки, като например SQL инжекции, CSRF и XSS.

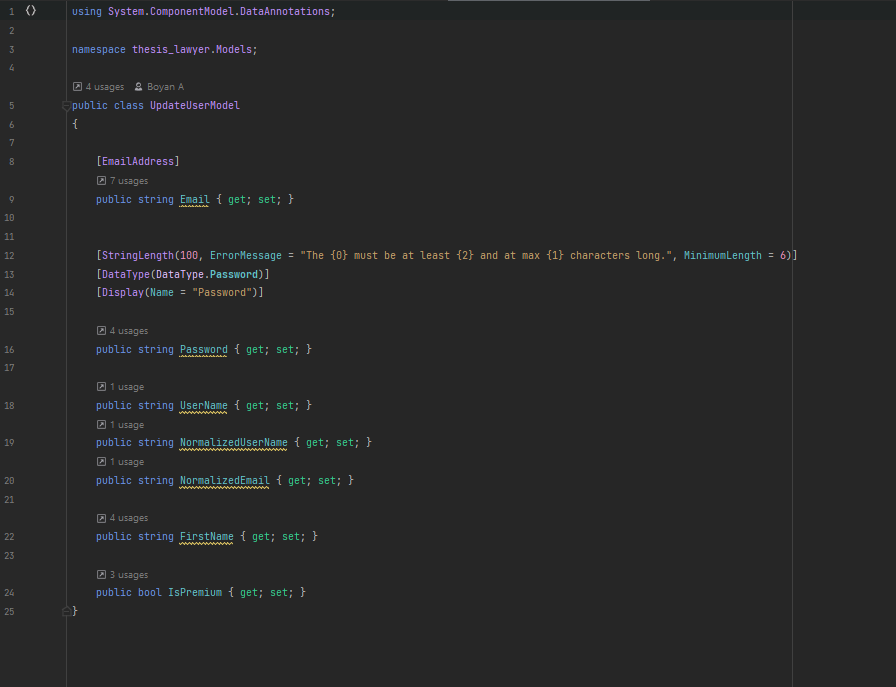
ASP.NET може лесно да се интегрира с други продукти и технологии на Microsoft, като например SQL Server и Azure Cloud Services. Тя поддържа няколко езика за програмиране, включително C#, VB.NET и F#. Платформата предлага богати възможности за създаване на интерактивни и динамични уеб сайтове с помощта на HTML, CSS, JavaScript и AJAX.

ASP.NET предоставя разнообразни инструменти и ресурси за разработчиците, които правят процеса на разработка, разширяване и поддръжка на уеб приложенията по-лесен и ефективен. С ASP.NET разработчиците могат да създадат широка гама от уеб приложения, които са мощни, сигурни и високопроизводителни, като тази платформа продължава да бъде водещ избор за разработка на уеб приложения в корпоративната среда и за стартап компании, които търсят надеждност и мащабируемост.

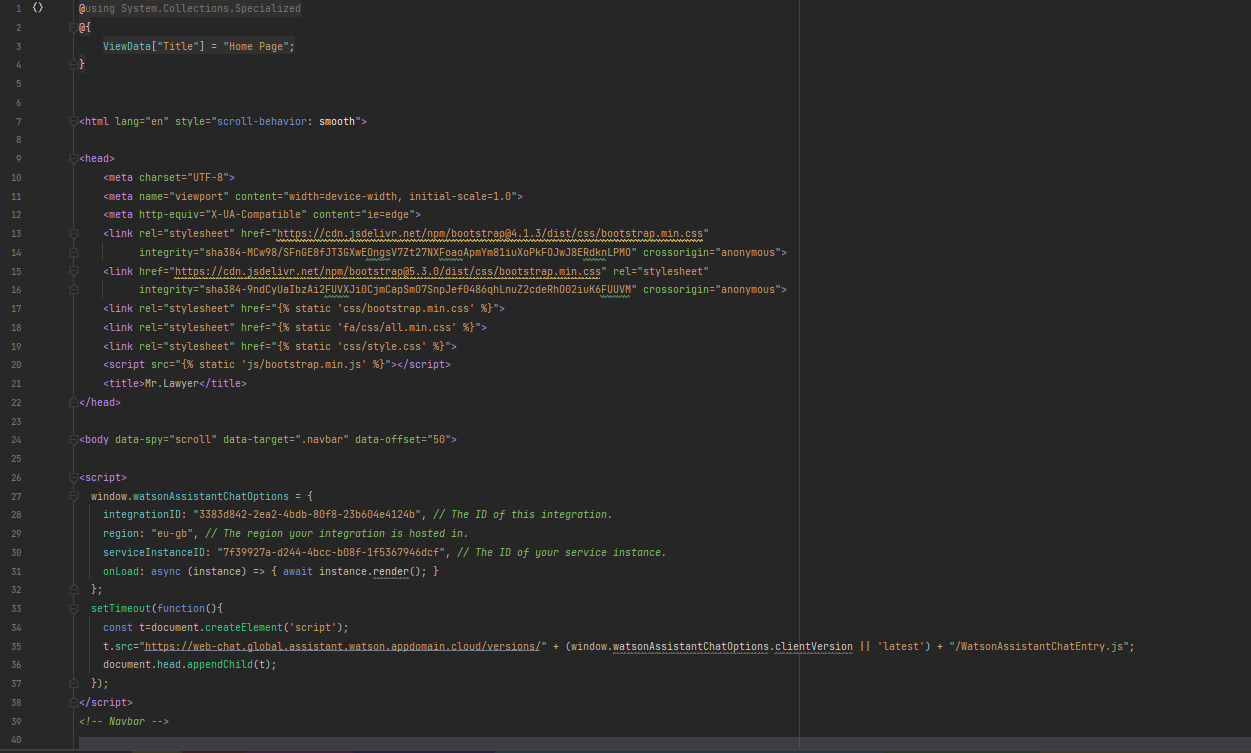
### **ASP .NET MVC**

ASP.NET MVC (Model-View-Controller) е специфичен подход за създаване на уеб приложения в рамките на платформата ASP.NET. Той се базира на архитектурния шаблон MVC, който разделя уеб приложението на три основни компонента:

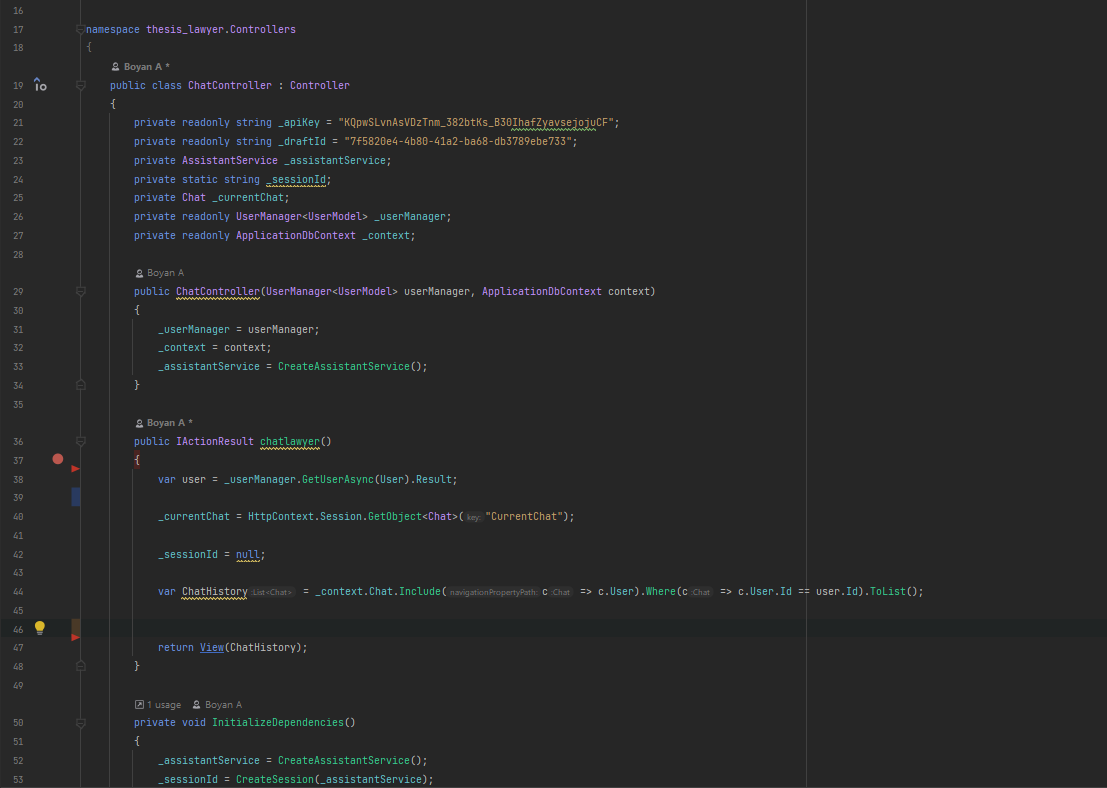
1. **Модел (Model)**: Моделът представлява данните и бизнес логиката на приложението. Той обхваща данните, свързаната с тях логика и правилата за манипулиране на данните. В ASP.NET MVC моделите обикновено се представят чрез класове или обекти, които взаимодействат с базата данни или други източници на данни. Моделите извличат данни от базата данни, ги манипулират, а после ги предават на контролерите за допълнителна обработка или за показване в изгледите. Моделите често съдържат и логика за валидация, за да се гарантира целост и съгласуваност на данните.



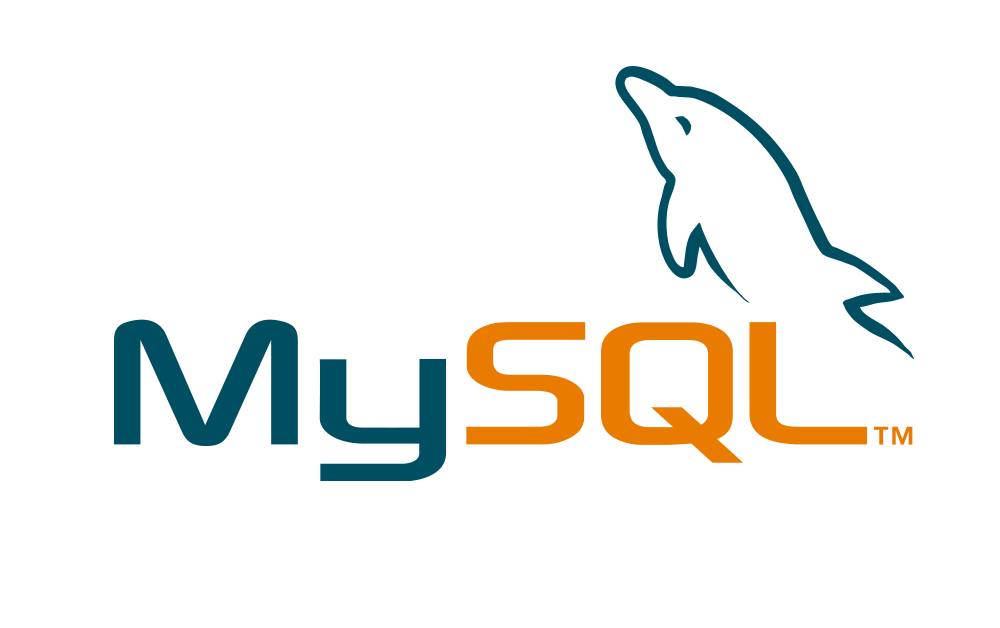
1. **Изглед (View)**: Изгледът е отговорен за представянето на потребителския интерфейс на приложението. Той получава данните от контролерите и ги изобразява в подходящ формат за показване, като HTML, XML или JSON. Изгледите в ASP.NET MVC обикновено се представят чрез HTML шаблони, съдържащи маркери и места за динамичните данни. Изгледите са изградени да бъдат леки, фокусирани само върху логиката за представяне, без да съдържат бизнес или приложна логика. Те са проектирани да бъдат преизползваеми и модулни, позволявайки на разработчиците да създават отделни изгледи за различни части на приложението.



1. **Контролер (Controller)**: Контролерът действа като посредник между модела и изгледа. Той получава заявки от потребителите чрез уеб браузъра, обработва тези заявки, взаимодейства с модела за извличане или промяна на данни и избира подходящия изглед за показване на отговора. Контролерите в ASP.NET MVC обикновено се представят чрез класове, които съдържат методи за действия, които съответстват на различни действия или заявки на потребителя. Тези методи за действия обработват входящите заявки, извършват необходимата обработка и връщат съответния изглед или данни за показване на потребителя. Контролерите често съдържат и логика за обработка на потребителски вход, извършване на валидация и координиране на потока на данни между модела и изгледа.



### **MySQL**



MySQL е една от най-популярните релационни бази данни, използвана от милиони уеб приложения и бизнес системи по целия свят. Тя е безплатна и отворен код, което означава, че може да бъде свободно изтеглена, използвана и модифицирана от всеки желаещ.

MySQL е релационна база данни, която организира данните в таблици с редове и колони. Тя поддържа стандартни релационни операции като SELECT, INSERT, UPDATE и DELETE, като предоставя мощно средство за управление на данни.

MySQL е известна със своята бързодействие и ефективност при обработката на големи обеми от данни. Тя е оптимизирана за работа със заявки в реално време и може да поддържа голям брой едновременни връзки към базата данни.

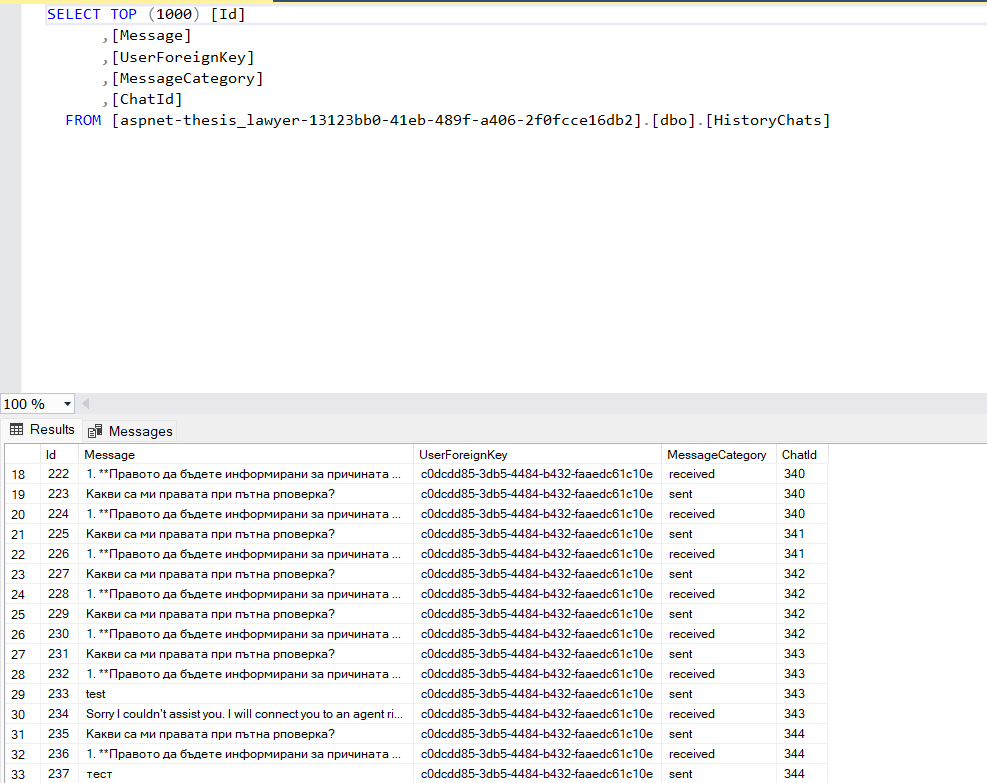
MySQL може лесно да бъде скалирана както вертикално (чрез добавяне на по-мощни хардуерни ресурси), така и хоризонтално (чрез разпределение на данните на няколко сървъра), което позволява на организациите да увеличават капацитета на базата данни по мерка на растежа на техните приложения и нужди.

MySQL предоставя множество механизми за сигурност, включително вградена аутентикация и авторизация, шифроване на данните, контрол на достъпа и други. Тя също така предлага възможности за резервно копиране и възстановяване на данни, гарантирайки защитата на данните от загуба или повреда.

Благодарение на своят отворен код и голямата общност от разработчици и потребители, MySQL има активно обновяване и подобряване на функционалността си. Потребителите могат лесно да намерят помощ, документация и различни добавки и разширения за своите нужди.

MySQL е широко използвана в различни сфери, включително уеб разработка, бизнес приложения, телекомуникации, финанси, медии и др. Нейната надеждност, производителност и гъвкавост я правят една от предпочитаните бази данни за множество приложения и среди разработчиците по целия свят.

В контекста на ASP.NET MVC приложения, MySQL може да бъде използвана като база данни за съхранение на данни за потребителите, продуктите, поръчките и други елементи на приложението. Разработчиците могат лесно да създадат връзка към MySQL базата данни в своите ASP.NET MVC проекти и да използват стандартни SQL заявки за извличане, обновяване и изтриване на данни. Това позволява на разработчиците да създадат мощни и надеждни уеб приложения, които се възползват от функционалността и производителността на MySQL базата данни.



### **ASP.NET Core MVC Auth SDK**

Интегрираният ASP.NET Core MVC Auth SDK, или Комплектът за софтуерна разработка за удостоверяване, е набор от инструменти и библиотеки, предоставени от Microsoft като част от ASP.NET Core рамката. Той предоставя на разработчиците комплексно решение за изпълнение на механизми за удостоверяване и разрешения в ASP.NET Core MVC уеб приложения.

SDK-то включва middleware компоненти, които управляват удостоверяването за различни схеми като бисквитки, JWT, OAuth, OpenID Connect и други. Тези компоненти улесняват процеса на удостоверяване на потребителите и управлението на потребителските сесии в приложенията.

Разработчиците могат да дефинират политики за разрешение, за да контролират достъпа до различни части от приложенията на базата на ролите на потребителите, твърденията или специфични изисквания. Политиките за разрешение помагат за прилагането на правилата за контрол на достъпа и ограничаването на достъпа до чувствителни ресурси.

SDK-то предоставя функционалности за управление на потребителските идентичности и акаунти, включително регистрация на потребители, вход в системата, управление на пароли и профили на потребители. То поддържа и интеграция с външни доставчици на идентичности като Azure Active Directory, IdentityServer и социални доставчици на влизане.

ASP.NET Core MVC Auth SDK използва удостоверяване базирано на твърдения, където информацията за идентичността на потребителя се представя като колекция от твърдения. Тези твърдения се използват за вземане на решения за разрешение и персонализиране на потребителските преживявания.

SDK-то опростява интеграцията с външни доставчици на идентичности, като предоставя предварително изградени компоненти и middleware за управление на потоците за удостоверяване и валидация на токените. Разработчиците могат лесно да интегрират своите приложения с популярни доставчици на идентичности като Microsoft, Google, Facebook и други.

SDK-то е проектирано да бъде разширяемо, позволявайки на разработчиците да персонализират и разширят функционалностите му за да отговарят на специфичните изисквания на техните приложения. Общо казано, Интегрираният ASP.NET Core MVC Auth SDK предоставя надежден и гъвкав фреймуърк за изпълнение на удостоверяване и разрешения в ASP.NET Core MVC приложения, позволявайки на разработчиците да създават сигурни и мащабируеми уеб приложения с лекота.

### **IBM Watson**



IBM Watson е платформа за изкуствен интелект, която се използва за създаване на интелигентни приложения и решения за различни отрасли и бизнес сектори. Тя се базира на напреднали технологии като машинно самообучение, обработка на природни езици (NLP), компютърно зрение и други, които позволяват на приложенията да разбират, анализират и вземат решения въз основа на различни видове данни.

Ето някои от ключовите компоненти и възможности на IBM Watson:

Watson Assistant: Предоставя платформа за създаване на чатботи и виртуални асистенти, които могат да отговарят на въпроси, да изпълняват команди и да предоставят персонализирани услуги на потребителите.

Watson Discovery: Позволява на организациите да откриват, анализират и извличат полезна информация от големи обеми текстови данни като новини, статии, документи и социални медии.

Watson Natural Language Understanding (NLU): Предоставя анализ на текстове и разбиране на естествения език, включително извличане на съдържание, анализ на настроения, категоризация и идентификация на същностите.

Watson Visual Recognition: Позволява разпознаване и анализ на изображения, включително идентификация на обекти, лица, настроения и сцени на снимки и видео материали.

Watson Machine Learning: Предоставя платформа за създаване, обучение и управление на модели за машинно самообучение, които могат да бъдат интегрирани в различни приложения и системи.

Watson IoT: Интегрира се с Интернет на нещата (IoT) устройства и сензори, позволявайки анализ на данни от устройствата в реално време и вземане на интелигентни решения за управление на средствата и процесите.

IBM Watson може да бъде интегриран в ASP.NET MVC приложения чрез използване на Watson Developer Cloud SDK за .NET. Това официално предоставен комплект от инструменти позволява на разработчиците да взаимодействат с различните услуги на Watson чрез програмен интерфейс (API). Посредством този SDK, разработчиците могат да извършват заявки към Watson API-то, за да изпълняват различни операции като анализ на текст, разпознаване на обекти в изображения и други. Резултатите от взаимодействието с Watson могат да бъдат интегрирани във визуалния интерфейс на ASP.NET MVC приложението, като се предоставят на потребителите чрез изграждане на подходящи изгледи и шаблони. Чрез тези стъпки, разработчиците могат да интегрират интелигентни функционалности на IBM Watson в своите ASP.NET MVC приложения, което може да подобри потребителското изживяване, да оптимизира бизнес процесите и да осигури интелигентни решения за своите потребители.

2.2 Ръководство на потребителя

* Влезте в сайта на Mr. Lawyer
* Регистрирате се или в случай, че имате акаунт влизате в него чрез бутоните Register/Login
* За достъп до чат-бота може да използвате синия бутон в горната част на екрана, на който пише “Chat Now” (за достъп е задължително да сте влезли във вашия акаунт)
* За да намерите вашето chat history е необходимо да натиснете синия бутон в горната част на екрана

# **Заключение**

Проектът "Mr. Lawyer" е амбициозна и иновативна инициатива, насочена към развитието на уеб приложение, което използва технологии като ASP.NET MVC и IBM Watson AI с цел предоставяне на автоматизирано и интелигентно обслужване на потребители в областта на правните консултации. Този проект се явява не само техническо предизвикателство, но и реален принос към улесняване на достъпа до правна информация и консултации за широката публика.

Чрез използването на ASP.NET MVC създавам стабилно и сигурно уеб приложение, което предоставя лесен и интуитивен интерфейс за потребителите. Използването на този технологичен стек ми позволява бързо разработване на приложението и гъвкавост в разширяването му с нови функционалности и подобрения.

С добавянето на IBM Watson AI в проекта, "Mr. Lawyer" става по-още по-мощен инструмент за обслужване на потребителите. Технологиите за изкуствен интелект, включени в Watson, като например Natural Language Understanding (NLU) и машинното самообучение, позволяват на приложението да разбира и анализира правните въпроси на потребителите и да им предоставя бързи и точни отговори и съвети.

В заключение, проектът "Mr. Lawyer" представлява иновативно решение за съчетаване на технологиите на уеб разработката и изкуствения интелект с цел предоставяне на висококачествени правни консултации на потребителите. Той има потенциал да промени начина, по който хората получават правна помощ, като го прави по-достъпен, ефективен и удобен. В крайна сметка, "Mr. Lawyer" не само подобрява клиентското обслужване в сферата на правните услуги, но и допринася за по-широкото развитие и приложение на изкуствения интелект в различни области на живота.

# **Информационни източници**

<https://learn.microsoft.com/en-us/dotnet/csharp/>

<https://www.w3schools.com/bootstrap/bootstrap_get_started.asp>

<https://www.techtarget.com/whatis/definition/bootstrap>

<https://developer.mozilla.org/en-US/docs/Web/HTML>

<https://developer.mozilla.org/en-US/docs/Web/CSS>

<https://learn.microsoft.com/en-us/aspnet/core/?view=aspnetcore-8.0>

<https://learn.microsoft.com/en-us/aspnet/mvc/overview/getting-started/introduction/getting-started>

<https://developer.mozilla.org/en-US/docs/Web/JavaScript>

<https://javascript.info/intro>

<https://dataplatform.cloud.ibm.com/docs/content/wsj/getting-started/welcome-main.html?context=wx&audience=wdp>

<https://medium.com/the-power-of-ai/ibm-watsonx-ai-the-interface-and-api-e8e1c7227358>

<https://dev.mysql.com/doc/>

<https://docs.digitalocean.com/products/databases/mysql/>

<https://medium.com/@iammanolov98/what-exactly-is-a-web-application-in-context-of-c-asp-net-9347808c0981>

# **Приложения**



