**ДИПЛОМЕН ПРОЕКТ**

За придобиване на трета

степен на професионална

квалификация

Тема: Изграждане на комплекс от методи, средства и подходи при разработване на приложение:

**“СИСТЕМА ЗА ОРГАНИЗИРАНЕ НА УЧЕНИЧЕСКИ СЪСТЕЗАНИЯ“**

Ученик: Ръководител:

Николай Миленов Симеонов Мариела Маринова

Професия: „Приложен програмист“

Специалност: „Приложно програмиране“

Съдържание

[**Увод** 2](#_Toc163075683)

[**Използвани технологии** 2](#_Toc163075684)

[Програмни езици: 2](#_Toc163075685)

[Framework-ове 8](#_Toc163075686)

[Конвенции 8](#_Toc163075687)

[Библиотеки 8](#_Toc163075688)

[Среди за разработка и контрол на версиите 8](#_Toc163075689)

# **Увод**

Предмет на дипломната работа е разработване на приложение, целящо да улесни работата на учителите и организаторите на учебни състезания и олимпиади. Системата се ползва от учители и ученици, като учителите създават и организират състезания, а учениците могат да ги преглеждат филтрират по предмет, дата и други фактори и да се записват за тях. Системата има администратор, който управлява потребителите. Функционалностите включват: онагледяване на информацията за отделните състезания, водене на статистика, разделение за състезанията и потребителите по определени критерии и пълен контрол над организацията на учебни състезания.

Уеб системата поддържа три потребителски роли:

* Администратор***-*** Одобрява акаунтите на учители и ученици. Има възможност да преглежда, променя, филтрира и изтрива вече регистрираните потребители, като преди необратими промени бива предупреден, че действието е необратимо.

Нерегистрираните потребители може да бъдат филтрирани, одобрявани и отхвърляни. Той има правото да създава, изтрива, променя състезания както и да вади статистика за учениците записани за дадено състезание.

* Учител***-***  Учителят може да вижда основния прозорец на приложението, в който се визуализират всички активни състезания. Той има право да го преглежда, променя, деактивира, изтрива и да вади статистика за записаните ученици.
* Ученик***-*** Ученикът може да вижда основния прозорец на приложението, в който се визуализират всички активни състезания. Той има право да се записва за тях, като при натискане на бутона записване, състезанието изчезва от основната страница и се премества в страницата „Моите състезания“.

Целта на дипломната работа е показване на знания свързани със изработването на функционално уеб приложение, което включва неговата архитектура, логика и потребителски интерфейс.

# **Използвани технологии**

### Програмни езици:

* **C#:** е обектно-ориентиран програмен език, разработен от Microsoft. Той е изключително популярен за създаване на софтуерни приложения за различни платформи като десктоп, уеб и мобилни устройства. Ето някои от предимствата и ползите на C#:
* \*\*Обектно-ориентиран подход\*\*: C# поддържа основните концепции на обектно-ориентираното програмиране (ООП), като инкапсулация, наследяване и полиморфизъм. Това прави кода по-структуриран, поддържаем и разширяем.
* 2. Мощна платформа за разработка: C# е част от платформата .NET на Microsoft, която предлага богат набор от инструменти и библиотеки за разработка на различни видове приложения.
* 3. „Типобезопасност“: C# е статично типизиран език, което означава, че типовете на данните се проверяват по време на компилацията. Това помага за намаляване на грешките в кода и улеснява поддръжката му.
* 4. Интеграция с други технологии: C# се интегрира добре с различни технологии на Microsoft, като SQL Server, Azure, Windows Forms, WPF, ASP.NET и други. Това позволява лесна разработка на приложения, които се взаимодействат с различни системи и услуги.
* 5. Широка общност и ресурси за обучение: Поради популярността си, C# разполага с голяма общност от разработчици и обширен набор от уроци, ръководства и ресурси за обучение. Това прави изучаването и развиването на умения в C# достъпно и удобно.
* 6. \*\*Крос-платформеност\*\*: С развитието на .NET Core и по-късно на .NET 5 и .NET 6, C# става все по-крос-платформен език, който може да се използва за разработка на приложения за различни операционни системи като Windows, Linux и macOS.
* 7. Висока производителност: C# е известен с високата си производителност, която го прави подходящ за разработка на приложения с високи изисквания към бързодействие и ефективно използване на ресурсите на системата.

Всички тези фактори правят C# изключително полезен език за разработка на различни видове софтуерни приложения, което го прави предпочитан избор за много програмисти и организации.

* **HTML 5: HTML** е съкращение от "**HyperText Markup Language**" (Хипертекстов език за означаване), който е стандартен език за описване и структуриране на уеб страници. HTML използва различни елементи, наречени тагове, за да определи структурата и съдържанието на уеб документите.

Основната цел на **HTML** е да позволи на уеб разработчиците да създават документи, които могат да бъдат изобразени в уеб браузъри. Тези документи могат да включват текст, изображения, връзки към други уеб страници, мултимедийни елементи и форми за въвеждане на данни от потребителя.HTML се използва за създаване на различни видове уеб съдържание, включително:

Уеб страници: **HTML** се използва за създаване на уеб страници, които могат да съдържат текст, изображения, линкове, таблици и други елементи.

Уеб приложения: **HTML** се използва за създаване на интерактивни уеб приложения, които могат да предлагат различни функционалности като форми за въвеждане на данни, анимации, мултимедия и други.

Електронни бюлетини и имейл шаблони: **HTML** се използва за създаване на електронни бюлетини и имейл шаблони, които могат да се изпращат на потребителите по електронна поща.

Мобилни приложения: **HTML** се използва за създаване на мобилни приложения с помощта на технологии като **Apache** **Cordova** или **PhoneGap**.

HTML работи заедно с други технологии като CSS (**Cascading** **Style** **Sheets**) за стилизиране на уеб страници и **JavaScript** за добавяне на интерактивност и динамично съдържание.

Съвременните уеб страници и приложения обикновено използват комбинация от **HTML**, **CSS** и **JavaScript** за да предложат богато и атрактивно потребителско изживяване.

**HTML 5** е петата версия на стандарта за маркиране на хипертекстови документи (**HTML**), който се използва за създаване на уеб страници.

**HTML 5** включва множество нови функции и подобрения спрямо предишните версии, които го правят по-мощен и гъвкав инструмент за уеб разработка. Ето някои от основните тагове и атрибути в **HTML 5**:

* <**html**>: Определя началото на **HTML** документа и съдържа целия **HTML** код.
* <**head**>: Съдържа метаданни за документа, като заглавие, метатагове, връзки към външни файлове за стилове и скриптове и други.
* <**title**>: Задава заглавието на документа, което се показва в заглавната част на прозореца на браузъра или във вкладката на страницата.
* <**body**>: Съдържа съдържанието на уеб страницата, като текст, изображения, линкове и други елементи.
* <**div**>: Използва се за създаване на контейнери за групиране на други елементи и за прилагане на стилове чрез **CSS**.
* <**span**>: Подобно на <**div**>, но обикновено се използва за групиране на текстови елементи.
* <**p**>: Определя параграф, който съдържа текст.
* <**a**>: Създава хипервръзка към други уеб страници, документи, изображения и други ресурси.
* <**img**>: Вмъква изображение в документа.
* <**ul**> и <**ol**>: Създават неномериран и номериран списък, съответно. <**li**>: Определя елемент в списък.
* <**table**>, <**tr**>, <**td**>: Създават таблица, ред и клетка в таблица.
* <**form**>: Създава формуляр за въвеждане на данни от потребителя.
* <**input**>: Използва се за създаване на поле за въвеждане на данни във формуляр.
* <**textarea**>: Създава текстово поле с множество редове във формуляр. Toва са само някои от основните тагове и атрибути в **HTML 5**. В **HTML 5** има още много други тагове и възможности за създаване на богато и интерактивно съдържание за уеб.
  + **CSS:**

**CSS** е съкращение от "**Cascading** **Style** **Sheets**", което е език за стилизиране и оформление на уеб страници. **CSS** позволява на уеб разработчиците да контролират външния вид на уеб страници, като определят цветове, шрифтове, размери, позициониране и други стилови елементи.

Основната цел на **CSS** е да раздели съдържанието от представянето му, като позволява на разработчиците да променят външния вид на множество елементи на уеб страницата едновременно, като променят само една линия код.

**CSS** се използва за следните цели:

* ***Стилизиране на текст***: **CSS** позволява да се променят различни аспекти на текста като шрифт, размер, цвят, подчертаване и т.н.
* ***Стилизиране на елементи на страницата***: **CSS** позволява да се променят визуалните атрибути на различни елементи на уеб страницата като фонови цветове, граница, размери, позиции и други.
* ***Работа със специфични устройства***: **CSS** позволява да се създадат различни стилове, които да се прилагат в зависимост от устройството, на което се визуализира уеб страницата, като например мобилни устройства, таблети или десктоп компютри.
* ***Анимации и преходи***: **CSS** позволява да се създават анимации и преходи, които да подобрят визуалното изживяване на потребителите, като например анимирани менюта, промени на цветове и други.
* ***Поддръжка на медийни елементи***: **CSS** позволява да се стилизират медийни елементи като изображения, видео и аудио файлове, като се контролират техните размери, позиции и други аспекти.

**CSS** обикновено се използва заедно с **HTML** и **JavaScript**, за да се създаде уеб страница със структура, стил и интерактивност. Стандартите на **CSS** се поддържат от всички модерни уеб браузъри, което гарантира, че стиловете, които се дефинират с помощта на **CSS**, ще бъдат правилно интерпретирани и показани на различните устройства и браузъри.

* **href**: Атрибут за <**a**>, който определя адреса на дестинацията на хипервръзката.
* **src**: Атрибут за <**img**>, който указва пътя към изображението.
* **alt**: Алтернативен текст за <**img**>, който се показва, ако изображението не може да бъде заредено.
* **id**: Уникален идентификатор за елемента, което е полезно за стилизиране или манипулиране с **JavaScript**.
* **class**: Клас на елемента, който се използва за групиране на елементи със сходни стилове или за прилагане на JavaScript функционалност.
* **style**: Атрибут, който позволява директно дефиниране на **CSS** стилове за даден елемент.
* **title**: Показва заглавието на елемента, което се показва при *ховър* или при фокусиране върху него.

Това са някои от основните **HTML** тагове и атрибути, които се използват за създаване на уеб страници. С тях можете да създадете разнообразни и интересни уеб преживявания за вашите потребители.

* + **JavaScript:**

Предимствата и ползите на **JavaScript**:

* ***Интерактивност на уеб страниците***: **JavaScript** позволява създаването на интерактивни елементи и приложения в уеб страниците, като например форми за въвеждане на данни, анимации, интерактивни менюта и други. Това прави уеб страниците по-привлекателни и полезни за потребителите.
* ***Клиентска страна****:* ***JavaScript*** се изпълнява в браузъра на потребителя, което го прави идеален за разработка на функционалности, които се изискват в реално време, без да се налага презареждане на страницата. Това включва валидация на форми, динамично обновяване на съдържанието и други.
* **Широка поддържка**: **JavaScript** е поддържан от всички модерни уеб браузъри, което означава, че кодът, който създадете, ще работи в различни браузъри и операционни системи без проблеми.
* **Лесно изучаване и употреба**: **JavaScript** е език с прост синтаксис, който е лесен за научаване за начинаещи и за разработчици, които идват от други програмни езици.
* ***Богат набор от библиотеки и фреймуърк-ове***: Съществуват множество библиотеки и фреймуърки като **jQuery**, **React.js**, **Angular.js**, **Vue.js** и други, които улесняват разработката на сложни уеб приложения и повишават производителността на разработчиците.
* ***Разширяемост****:* ***JavaScript*** позволява внедряване на допълнителна функционалност чрез използване на външни библиотеки и плъгини, които са лесни за интегриране във вашите проекти.
* ***Поддръжка на мобилни устройства***: **JavaScript** се използва за разработка на мобилни приложения с помощта на фреймуърк-ове като **React** **Native** или **Ionic**, което позволява създаването на крос-платформени приложения, които работят на различни мобилни устройства.

Тези са само някои от предимствата и ползите на **JavaScript**, които го правят един от най-широко използваните езици за уеб разработка в момента.

Основните понятия, конструкции и методи в **JavaScript**:

Понятия:

1. **Променливи**: Контейнери за съхранение на данни.
2. **Типове данни**: Включват числа, стрингове, булеви стойности, масиви, обекти и други.
3. **Функции**: Блокове код, които могат да се изпълняват при нужда, използвайки параметри и връщайки стойност.
4. **Условни оператори**: Като if, else if и else, които позволяват на програмата да взема решения в зависимост от определени условия.
5. **Цикли**: Като for, while и do-while, които позволяват на програмата да изпълнява определени действия повторно.
6. **Масиви**: Структури от данни, които могат да съдържат набор от елементи, индексирани с числови стойности.
7. **Обекти**: Съдържат ключове и стойности, които позволяват организиране на данните по различни категории.
8. **Събития**: Действия, които могат да бъдат предизвикани от потребителя или от системата, като кликване на мишката или зареждане на страницата.

Конструкции и методи:

1. **console.log()**: Метод за извеждане на информация в конзолата на браузъра, което е полезно за дебъгване и проверка на данни.
2. **document.querySelector() и document.querySelectorAll()**: Методи за избор на елементи от HTML документа по CSS селектори.
3. **addEventListener()**: Метод за добавяне на събития към HTML елементи, като кликване, навлизане на мишката и други.
4. **Array.push()**: Метод за добавяне на нови елементи към края на масива.
5. **Array.pop()**: Метод за премахване на последния елемент от масива.
6. **String.length**: Свойство, което връща дължината на низа.
7. **Math.random()**: Метод за генериране на случайно число между 0 и 1.
8. **setTimeout() и setInterval()**: Функции, които позволяват на програмата да изчака определено време преди да изпълни определени действия или да повтаря действията периодично.

Това са някои от основните понятия, конструкции и методи в JavaScript, които са от съществено значение за разработката на уеб приложения и интерактивни уеб страници.

* + **SQL:**

Ползите и предимствата на **SQL**:

* ***Лесен за използване***: **SQL** е декларативен език за заявки, който е лесен за научаване и разбиране дори за хора без техническо образование.
* ***Мощен и гъвкав***: **SQL** позволява на потребителите да изпълняват сложни заявки за данни, включително извличане, филтриране, сортиране, групиране и обединяване на данни от различни източници.
* ***Скорост и ефективност***: **SQL** е оптимизиран за бърза обработка на големи обеми от данни, като осигурява ефективно търсене и сортиране на информацията.
* ***Скалируемост***: **SQL** базите данни могат да се мащабират вертикално и хоризонтално, което позволява увеличаване на капацитета и производителността им спрямо нуждите на приложението.
* ***Сигурност***: **SQL** предлага механизми за сигурност и управление на достъпа до данните, като ролево базиран достъп и криптиране на данните, което гарантира защитата на чувствителната информация.
* ***Стандартизиран***: **SQL** е стандартизиран език, който е признат и поддържан от множество вендори и организации, което гарантира съвместимост и преносимост на данните между различни системи и платформи.
* ***Разнообразие от инструменти и ресурси***: Съществуват множество инструменти и ресурси за работа с **SQL**, включително мениджмънт системи за бази данни (**DBMS**) като **MySQL**, **PostgreSQL**, **Microsoft SQL Serve**r и **Oracle**, както и богата документация, онлайн курсове и форуми за поддръжка и обучение.
* ***Поддръжка на транзакции***: **SQL** поддържа транзакции, което позволява на потребителите да извършват група от операции като една атомарна и неделима операция, гарантираща цялостността на данните.

Тези са някои от основните ползи и предимства на **SQL**, които го правят незаменим инструмент за управление и манипулиране на данни в различни бизнес среди.

Основни понятия:

* База данни - Съхранява структурирани данни в таблици.
* Таблица - Съдържа редове и колони, които организират данните по категории.
* Колона - Съдържа специфичен вид данни за всеки ред в таблицата.
* Ред - Съдържа конкретните данни за всеки запис в таблицата.
* Ключ - Уникален идентификатор за реда или комбинация от колони, който позволява бързо търсене и сортиране на данните.

Основни **SQL** команди:

* **SELECT** - Избира данни от таблица или комбинация от таблици.
* **INSERT INTO** - Добавя нов ред в таблицата със специфични стойности за всяка колона.
* **UPDATE** - Променя данни във вече съществуващи редове.
* **DELETE FROM** - Премахва редове от таблицата.
* **CREATE TABLE** - Създава нова таблица в базата данни със специфични колони и типове данни.
* **ALTER TABLE** - Променя структурата на съществуваща таблица, като добавя, променя или премахва колони.
* **DROP TABLE** - Премахва цялата таблица от базата данни.
* **JOIN** - Обединява данни от две или повече таблиците в един резултатен набор, базиран на общи стойности в определени колони.
* **WHERE** - Филтрира резултатите на **SELECT** заявката базирано на дадени критерии.
* **GROUP BY** - Групира резултатите от **SELECT** заявката базирано на стойности в определени колони.
* **ORDER** **BY** - Сортира резултатите от **SELECT** заявката във възходящ или низходящ ред базирано на стойности в определени колони.
* **HAVING** - Прилага условия върху резултатите от **GROUP BY** операцията.

Това са някои от основните понятия, команди и конструкции в **SQL**, които се използват за създаване и управление на релационни бази данни. **SQL** е език с широко приложение в областта на базите данни и е важен инструмент за разработчиците и администраторите на данни.

### Framework-ове

* + **ASP.Net:**

ASP.NET платформа, разработена от Microsoft, която позволява на програмистите да създават уеб сайтове, уеб услуги и уеб приложения. Тя предоставя мощни инструменти и функционалности за създаване на динамично генерирани уеб страници, използвайки .NET програмни езици като C# или VB.NET.

Ето някои от основните причини за използването на **ASP.NET**:

* ***Бързо разработване на уеб приложения***: **ASP.NET** предлага множество готови компоненти, контролери и шаблони, които ускоряват процеса на разработка на уеб приложенията.
* ***Мощен и сигурен***: Платформата включва вградена система за сигурност, управление на сесии, автентикация и авторизация, което прави уеб приложенията, разработени с **ASP.NET**, мощни и сигурни.
* ***Интеграция с други технологии на Microsoft***: **ASP.NET** се интегрира лесно с други технологии на **Microsoft**, като **SQL Server, Azure, Windows Server** и други, което улеснява разработката и управлението на уеб приложенията.
* ***Мащабируемост и производителност: ASP.NET*** е мащабируема платформа, която може да се справи с големи обеми от трафик и данни. Тя предлага оптимизирани решения за увеличаване на производителността и ефективността на уеб приложенията.
* ***Поддръжка и общност****:* ***ASP.NET*** е широко използвана технология с голяма общност от разработчици, което означава, че има множество ресурси, форуми и обучения за поддръжка и учене.
* ***Крос-платформена поддръжка***: С появата на **.NET Core, ASP.NET** стана крос-платформен, което означава, че уеб приложенията могат да бъдат разработвани и изпълнявани на различни операционни системи.

В общи линии, **ASP.NET** е мощна, сигурна и мащабируема платформа за създаване на уеб приложения, която предоставя инструменти и ресурси за бързо развитие и управление на уеб проекти.

* + **Identity Framework:**

**ASP.NET Identity** е *фреймуърк* за управление на потребителска идентичност в уеб приложенията, разработена от **Microsoft**. Тя предоставя инструменти за регистрация, управление на потребителите, удостоверяване (автентикация) и управление на правата (авторизация).

Ето някои от основните причини за използването на **ASP.NET Identity**:

* ***Гъвкавост и персонализация***: **ASP.NET Identity** позволява на разработчиците да персонализират и адаптират системата за управление на потребителската идентичност според специфичните изисквания на техните уеб приложения.
* ***Безопасност***: Идентичността на потребителите се управлява по сигурен начин, като се включват механизми за *хеширане* на пароли, защита от CSRF атаки, управление на сесии и други.
* ***Интеграция с външни доставчици за идентичност***: **ASP.NET Identity** поддържа интеграция с външни доставчици за идентичност като **Google**, **Facebook**, **Twitter** и други, което позволява на потребителите да се автентицират с техните съществуващи акаунти в социалните мрежи.
* ***Лесна интеграция с ASP.NET приложения***: **ASP.NET Identity** е проектирана за лесна интеграция с уеб приложения, базирани на **ASP.NET MVC, ASP.NET Web Forms, ASP.NET Core** и други.
* ***Поддръжка на различни типове автентикация***: Платформата поддържа различни методи за автентикация, включително паролна, двуфакторна, посредством SMS и други.
* ***Готови за употреба функционалности***: **ASP.NET Identity** предоставя готови функционалности за регистрация, вход, изход, потвърждение на имейл, възстановяване на парола и други, които улесняват разработката на уеб приложения.

В общи линии, **ASP.NET Identity** е мощна и гъвкава рамка за управление на потребителската идентичност, която предоставя инструменти и функционалности за сигурно и удобно управление на потребителите в уеб приложенията.

* + **Entity Framework**:

**Entity Framework** е фреймуърк разработен от **Microsoft**, който позволява лесна връзка между базата данни и уеб приложението. С помощта на **Entity Framework**, можем да работим с база данни чрез обекти на езика **C#**, като таблиците в базата данни се представят като класове, а записите в тях като обекти.

**Entity Framework** ни предоставя вградена библиотека за работа с атрибути, като например:

* [Key]: Обозначава основния ключ на таблицата.
* [InverseProperty]: Позволява задаване на връзки между полета в различни класове, когато има връзка "един към много" или "много към много".
* [MaxLength]: Задава максималната дължина на стойностите в дадено поле.
* [MinLength]: Задава минималната дължина на стойностите в дадено поле.
* [Pattern]: Задава шаблон за валидация на стойностите в дадено поле.

**Entity Framework** също така предлага мощни разширения като **EntityFramework.Diagnostics, EntityFramework.Tools** и други, които улесняват работата с базата данни чрез модели и миграции. Миграциите позволяват автоматично генериране на скриптове за промени в базата данни, което улеснява контрола над версиите и автоматизира процеса на актуализация на схемата на базата данни.

Като цяло, **Entity Framework** е мощен инструмент за работа с бази данни в уеб приложенията, който предоставя лесен и удобен начин за взаимодействие с базата данни, като използва обекти и модели на програмния език C#.

* + **Bootstrap**:

**Bootstrap** е популярен *фронт енд фреймуърк*, разработен от **Twitter**, който предоставя готови компоненти, шаблони и стилове за бързо разработване на уеб приложения и уеб сайтове. Някои от основните ползи и причини за използването на **Bootstrap** са:

* ***Бързо разработване***: **Bootstrap** предоставя готови компоненти и стилове, които ускоряват процеса на разработка на уеб приложенията и уеб сайтовете.
* ***Мобилна първа подход***: **Bootstrap** е изцяло отзивчив и мобилен първенец, което означава, че уеб сайтовете, изградени с него, автоматично се адаптират и оптимизират за различни устройства и екрани.
* ***Консистентен дизайн***: **Bootstrap** предоставя единно оформление и дизайн за уеб приложенията, което прави интерфейса по-привлекателен и професионален.
* ***Множество компоненти***: **Bootstrap** включва голямо разнообразие от компоненти като бутони, форми, навигация, карусели, модални прозорци и други, които могат да се използват за бързо добавяне на функционалности в уеб приложенията.
* ***Лесно персонализиране***: **Bootstrap** позволява на разработчиците да персонализират и променят стиловете и компонентите с помощта на собствени **CSS** правила и теми.
* ***Поддръжка на браузъри***: **Bootstrap** е съвместим с всички основни браузъри, което гарантира, че уеб приложенията ще работят правилно за всички потребители.
* ***Активна общност***: **Bootstrap** има голяма и активна общност от разработчици, което означава, че има множество ресурси, документация и обучения за поддръжка и учене.

Това са някои от основните ползи и причини за използването на **Bootstrap**, което го прави предпочитан избор за много разработчици при създаването на модерни и атрактивни уеб приложения и уеб сайтове.

### Конвенции

* + **ASP.Net MVC**:
  + **Service Architecture**
  + **Application Builder Extension**
  + **CamelCase** и **pascalCase**
  + **GUID**

### Библиотеки

* + **LINQ (Language Interpreted Qeury)**
  + **IdentityFramework.Configurations**

### Среди за разработка и контрол на версиите

* + **Visual Studio 2022 Community**
  + **Github** и **Github Desktop**