

**Gymnázium, Praha 6, Arabská 14**

Programování

## **ROČNÍKOVÁ PRÁCE**



2024/2025

**Gymnázium, Praha 6, Arabská 14**

Arabská 14, Praha 6, 160 00

## **ROČNÍKOVÁ PRÁCE**

**Předmět:** Programování

**Téma:** Stats Score

**Autor:** Vojtěch Kadlec

**Třída:** 4. E

**Školní rok:** 2024/2025

**Vedoucí práce:** Mgr. Jan Lána

**Třídní učitel:** Mgr. Blanka Hniličková

# Prohlášení

Prohlašuji, že jsem jediným autorem tohoto projektu, všechny citace jsou řádně označené a všechna použitá literatura a další zdroje jsou v práci uvedené. Tímto dle zákona 121/2000 Sb. (tzv. Autorský zákon) ve znění pozdějších předpisů uděluji bezúplatně škole Gymnázium, Praha 6, Arabská 14 oprávnění k výkonu práva na rozmnožování díla (§ 13) a práva na sdělování díla veřejnosti (§ 18) na dobu časově neomezenou a bez omezení územního rozsahu.

V ..... dne .....

.....

Kadlec Vojtěch

## **Anotace**

Toto je dokumentace mé ročníkové práce. Zadáním ročníkového projektu bylo vytvořit webovou aplikaci, která bude využívána pro sledování a zapisování statistik sportovních utkání. Podstatou této aplikace je, že si uživatel otevře webovou stránku a bude si moci zobrazit sportovní utkání, událost a sledovat její výsledky. Vždy bude mít na výběr sport, soutěž, zápasu. Po otevření zvoleného utkání se objeví statistiky. Díky této webové aplikaci může uživatel sledovat průběh utkání a zároveň zpětně prohlížet statistiky týmů a hráčů. V aplikaci je dále možné využít funkce administrátora, který se stará o zakládání utkání a zapisování statistik do systému. Cílem tohoto ročníkového projektu bylo vyvinout aplikaci tak, aby fanoušci daného sportu mohli sledovat sportovní událost bez navštívení zápasu a současně zjednodušit zapisovatelům život tím, že nebudou muset zdlouhavě vyplňovat papírové verze zápisů o utkáních, ale budou mít vše potřebné přehledně na jednom místě.

## **Annotation**

This is documentation of my year's work. The assignment of the year project was to create a web application that will be used to track and record statistics of sports matches. The idea of this application is that the user will open a web page and be able to view a sports match, event and track the results. There will always be a choice of sport, competition, match. After opening the selected match, the statistics will appear. With this web application, the user can follow the progress of the match and also view the statistics of the teams and players backwards. The application also provides an administrator function that takes care of setting up matches and entering statistics into the system. The aim of this year's project was to develop the app so that fans of a given sport can follow a sporting event without attending a match and at the same time simplify the life of the recorders by not having to tediously fill out paper versions of match records, but having everything they need clearly in one place.

## **Zadání**

Cílem této ročníkové práce je vytvořit webovou aplikaci, která umožní administrátorovi spravovat sportovní utkání pro vybrané sporty (hokej, házená, fotbal). Aplikace poskytne možnost zaznamenávání výkonů týmů a jednotlivých hráčů (góly, tresty, vyloučení apod.), a bude současně obsahovat funkci časomíry pro sledování průběhu utkání. Webová aplikace bude sloužit pro administrátory k jednoduššímu zaznamenávání statistik v utkání. Zároveň bude sloužit pro fanoušky a samotné hráče, aby mohli sledovat průběh zápasu a výsledky.

# Obsah

<b>1 Úvod</b>	<b>1</b>
<b>2 Využité technologie</b>	<b>2</b>
2.1 Backend	2
2.2 Frontend	2
2.3 Databáze	2
2.4 Další knihovny a nástroje	2
<b>3 Řešení problémů</b>	<b>3</b>
3.1 Návrh databázové modelu	3
3.1.1 Problém	3
3.1.2 Řešení	3
3.2 Implementace real-time sledování zápasů	4
3.2.1 Problém	4
3.2.2 Řešení	4
3.3 Správa trestů a časomíry	5
3.3.1 Problém	5
3.3.2 Řešení	5
3.4 Uživatelská rozhraní a UX design	7
3.4.1 Problém	7
3.4.2 Řešení	7
3.5 Řešení specifík jednotlivých sportů	8
3.5.1 Problém	8
3.5.2 Řešení	8
<b>4 Popis funkcionality</b>	<b>10</b>
4.1 Zobrazení zápasů a soutěží	10
4.1.1 Domovská stránka	10
4.1.2 Seznam soutěží a zápasů	10
4.2 Detail zápasu a statistiky	10
4.2.1 Hlavní informační panel	10
4.2.2 Přehled hráčů a statistik	10
4.2.3 Trestná lavice	10
4.2.4 Souhrnné statistiky	11
4.3 Ovládací panel pro zapisovatele	11
4.3.1 Řízení průběhu zápasu	11
4.3.2 Správa statistik hráčů	11
4.3.3 Správa trestů	11
4.4 Správa týmů a hráčů	12
4.4.1 Správa sportů	12
4.4.2 Správa týmů	12
4.4.3 Správa hráčů	12
4.4.4 Správa zápasů a lig	12
<b>5 Závěr</b>	<b>13</b>

# 1 Úvod

Stats Score je webová aplikace pro správu a sledování sportovních zápasů založená na frameworku Django. Projekt byl vytvořen pro organizaci a monitorování sportovních utkání různých typů (fotbal, házená, hokej) v reálném čase. Aplikace je rozdělena na dvě části - veřejnou část pro diváky a fanoušky a specializované rozhraní pro zapisovatele. Veřejná část umožňuje procházet ligy, zápasy a sledovat statistiky, zatímco rozhraní pro zapisovatele poskytuje nástroje pro řízení průběhu zápasu, zaznamenávání událostí a správu časomíry. Hlavními cíli projektu bylo vytvořit univerzální platformu pro správu zápasů více sportů, implementovat real-time sledování, zajistit přehlednou vizualizaci statistik hráčů a týmů, poskytnout rozhraní pro zapisovatele a implementovat specifické funkce pro jednotlivé sporty. Systém je navržen pro využití sportovními organizacemi, zapisovateli, rozhodčími, trenéry a diváky. Technicky je aplikace rozdělena do několika aplikací (sporty, teamy, zápasy, stats), což umožňuje snadnou údržbu a rozšiřitelnost. Důraz byl kladen zejména na správné zpracování času a pravidla specifická pro jednotlivé sporty, jako jsou tresty, karty nebo vyloučení hráčů.

## 2 Využité technologie

Ročníková práce byla vyvinuta s využitím webových technologií, které zajišťují její spolehlivost a snadnou údržbu. V této kapitole jsou popsány hlavní technologie a frameworky použité při vývoji.

### 2.1 Backend

Jádrem aplikace je Python framework Django, který poskytuje základ pro vývoj webových aplikací. Django byl zvolen z několika důvodů: ORM systém umožňuje efektivní práci s databází bez nutnosti psát SQL dotazy a administrační rozhraní poskytuje předpřipravené nástroje pro správu zápasů. Backend je strukturován do několika Django aplikací, které spolu komunikují: sporty pro správu sportů a jejich specifických pravidel, teamy pro správu týmů a hráčů, zapasy pro správu zápasů a lig, a stats pro ukládání a správu statistik. Využitý byl také Django Template Engine pro renderování HTML stránek s dynamickým obsahem a Django ORM pro práci s databázovými modely.

### 2.2 Frontend

Uživatelské rozhraní aplikace je postaveno na kombinaci následujících technologií: HTML5 pro strukturování obsahu webových stránek, CSS3 pro stylování a responzivní design, Bootstrap 5.3 jako framework poskytující předpřipravené komponenty a responzivní grid systém, a JavaScript pro interaktivní prvky aplikace. Pro dynamickou funkcionalitu na straně klienta byly vytvořeny specializované JavaScript moduly: core.js zajišťuje základní funkcionalitu a komunikaci se serverem, timer.js implementuje časomíru pro zápasy, penalty.js spravuje tresty a odpočítávání trestných minut, detail.js poskytuje funkcionalitu detailního zobrazení zápasu a control.js nabízí ovládací prvky pro zapisovatele. Pro komunikaci mezi klientem a serverem je využit AJAX (Asynchronous JavaScript and XML), který umožňuje aktualizovat části stránky bez nutnosti jejího opětovného načtení. Tato technika je klíčová pro real-time sledování zápasů.

### 2.3 Databáze

Aplikace využívá databázový systém SQLite, který je integrován přímo v Django frameworku. Databázová struktura zahrnuje několik klíčových modelů: Sport pro podporované sportovní disciplíny, Team pro týmy, Hrac pro hráče, Zapas pro zápasy, Liga pro soutěže a Stats pro statistiky hráčů. Relace mezi modely jsou implementovány pomocí cizích klíčů a vazeb many-to-many, což umožňuje efektivní dotazování a zobrazení souvisejících dat.

### 2.4 Další knihovny a nástroje

Pro vývoj a správu projektu byly využity následující nástroje: Git jako verzovací systém pro správu zdrojového kódu, Virtual Environment (venv) jako izolované prostředí pro Python, Django Debug Toolbar pro ladění a optimalizaci aplikace během vývoje a Django Template Tags jako vlastní template tagy pro zpracování a zobrazení dat. Pro uživatelské rozhraní byly využity také Bootstrap Icons jako sada ikon pro vizuální prvky a CSS vlastní styly pro přizpůsobení vzhledu aplikace.



## 3 Řešení problémů

Při vývoji aplikace Stats Score jsem musel řešit několik problémů, které vyžadovaly efektivní řešení. V této kapitole popisují klíčové problémy a následně způsoby, kterými byla implementována jejich řešení.

### 3.1 Návrh databázové modelu

Jedním z prvních a nejdůležitějších kroků bylo navrhnout databázový model, který by byl dostatečně flexibilní pro různé typy sportů a zároveň efektivní z hlediska dotazování a práce s daty.

#### 3.1.1 Problém

Různé sporty mají odlišná pravidla, rozdílné statistiky hráčů (žluté karty ve fotbale, trestné minuty v hokeji a vyloučení v házené). Bylo potřeba navrhnout model, který by umožnil reprezentovat různé sporty s jejich specifiky, zajistil konzistentní způsob ukládání statistik a poskytl efektivní přístup k datům pro zobrazení na frontendu.

#### 3.1.2 Řešení

Zvolil jsem strukturu založenou na několika klíčových modelech:

1. **Sport** - základní model definující typ sportu (fotbal, házená, hokej) s identifikací specifických pravidel

```
class Sport(models.Model):
    | jmeno=models.CharField(max_length=20, choices=[("fotbal", "Fotbal"), ("hazena", "Házená"), ("hokej", "Hokej")], unique=True)
```

Obrázek 3.1: model sportu

2. **Team a Hrac** - modely pro správu týmů a hráčů s vazbou na konkrétní sport
3. **Zapas** - model pro správu zápasů s informacemi o čase, stavu a hráčích, včetně důležitých časových atributů pro sledování průběhu (start\_time, pause\_time, elapsed\_time)
4. **Stats** - univerzální model pro statistiky hráčů zahrnující jak obecné statistiky (góly), tak specifické podle sportu (žluté karty, trestné minuty)

Klíčovým aspektem bylo vytvoření vazby mezi modely pomocí ForeignKey a ManyToMany relací a zajištění unikátnosti záznamu statistik pro každou kombinaci hráče a zápasu pomocí unique\_together. Tento návrh umožnil flexibilně pracovat s různými typy sportů a jednoduše přidávat další sporty bez změny databázové struktury.

## 3.2 Implementace real-time sledování zápasů

### 3.2.1 Problém

Bylo potřeba vytvořit systém, který by umožňoval sledovat průběh zápasu téměř v reálném čase bez nutnosti obnovovat celou stránku a zároveň zajistit, aby všichni uživatelé viděli aktuální stav.

### 3.2.2 Řešení

Implementoval jsem aktualizace pomocí AJAX a JavaScript. Základní principy řešení:

1. **Serverová část** - view v Django byl navržen tak, aby mohl vracet buď kompletní HTML při prvním načtení stránky, nebo pouze JSON data při AJAX požadavku

```
def zapas(request, zapas_id):
    zapas = get_object_or_404(Zapas, id=zapas_id)
    stats = Stats.objects.filter(zapas=zapas)

    if request.headers.get('X-Requested-With') == 'XMLHttpRequest':
        player_stats_data = {}
        for player_id, stat in player_stats.items():

            return JsonResponse({
                'status': zapas.status,
                'status_display': zapas.get_status_display(),
                'timer_running': timer_running,
                'elapsed_time': elapsed_time,
                'player_stats': player_stats_data,
                'domaci_stats': domaci_stats,
                'hoste_stats': hoste_stats
            })

    return render(request, 'zapasy/detail.html', {
        'zapas': zapas,
        'stats': stats,
        'player_stats': player_stats,
        'timer_running': timer_running,
        'elapsed_time': elapsed_time,
        'active_penalties': active_penalties
    })
```

Obrázek 3.2: view funkce

2. **Klientská část** - JavaScript kód pravidelně zasílá AJAX požadavky na server a aktualizuje pouze změněné části stránky bez nutnosti jejího znovunačtení
3. **Modularizace frontendu** - JavaScript kód byl rozdělen do celků (core.js, timer.js, detail.js, control.js), což umožnilo přehlednou organizaci

Tento přístup umožnil aktualizovat jen změněné části stránky, minimalizovat zatížení serveru a poskytovat aktuální data všem uživatelům.

## 3.3 Správa trestů a časomíry

### 3.3.1 Problém

Jeden z nejobtížnějších úkolů bylo správně implementovat systém pro sledování času zápasu a odpočítávání trestných minut hráčů, zejména s přihlédnutím k možnosti pozastavení zápasu.

### 3.3.2 Řešení

Vytvořil jsem komplexní systém pro správu času zahrnující:

**Serverovou část** - ukládání času a stavu zápasu pomocí modelových atributů v databázi

**Klientskou časomíru** - JavaScript modul pro přesné měření času

```
function startTimer() {  
    if (!timerInterval) {  
        timerInterval = setInterval(updateTimer, 1000);  
    }  
}  
  
function updateTimer() {  
    elapsedTime += 1;  
    updateTimerDisplay();  
  
    if (matchDuration > 0 && elapsedTime >= matchDuration) {  
        stopTimer();  
        if (matchStatus === 'probihajici') {  
            performAction('pause');  
        }  
    }  
}
```

Obrázek 3.3: timer

**Správu trestů hráčů** - systém pro odpočítávání trestných minut, který vytváří vizuální reprezentaci trestu a zajišťuje automatické odstranění trestu po jeho skončení

**Synchronizaci** - klíčovou funkcionalitou byla implementace mechanismu, který při pozastavení zápasu zastaví všechny běžící odpočty trestů

```
function pauseAllPenaltyTimers() {
  Object.keys(penaltyTimers).forEach(playerId => {
    const penaltyTimer = penaltyTimers[playerId];

    if (penaltyTimer.interval) {
      clearInterval(penaltyTimer.interval);
      penaltyTimer.interval = null;
    }
  });

  if (penaltyTimer.started && penaltyTimer.endTime) {
    const remainingMs = penaltyTimer.endTime - Date.now();
    penaltyTimer.remainingSeconds = Math.max(0, Math.ceil(remainingMs / 1000));
  }
}
```

Obrázek 3.4: penalty

Zvláště důležitá byla implementace logiky pro správu trestů v házené, kde po třech trestech dochází k automatické diskvalifikaci hráče. Systém musel počítat všechny tresty hráče a automaticky provést diskvalifikaci při dosažení limitu. Tento systém zajistil přesné měření času zápasu, správné odpočítávání trestů hráčů, synchronizaci všech časomír při pozastavení zápasu a vizuální indikaci stavu časomíry a trestů.

## 3.4 Uživatelská rozhraní a UX design

### 3.4.1 Problém

Bylo potřeba vytvořit intuitivní rozhraní, které by umožnilo snadné sledování zápasu divákům a zároveň poskytlo efektivní nástroje pro zapisovatele.

### 3.4.2 Řešení

Navrhl jsem dvě hlavní rozhraní s odlišnými cíli:

1. **Veřejné rozhraní pro diváky** - zaměřené na přehlednou vizualizaci dat s důrazem na skóre, stav zápasu a přehled hráčů

```
<div class="card-header bg-primary text-white">
  <div class="row align-items-center">
    <div class="col-md-4 text-md-start text-center">
      <h3 class="mb-0">{{ zaspas.domaci.jmeno }}</h3>
    </div>
    <div class="col-md-4 text-center">
      <div class="display-4 fw-bold">
        {% with domaci_score=stats|sum_goals:zaspas.domaci, hoste_score=stats|sum_goals:zaspas.hoste %}
        <span class="text-light">{{ domaci_score }} - {{ hoste_score }}</span>
        {% endwith %}
      </div>
      <span id="match-status" class="badge
        {% if zaspas.status == 'probihajici' %}bg-danger{% elif zaspas.status == 'paused' %}bg-warning{% elif zaspas.status == 'ukoncene' %}bg-success{% else %}bg-info
        {% endif %} mb-2">
        {{ zaspas.get_status_display }}
      </span>
    </div>
    <div class="col-md-4 text-md-end text-center">
      <h3 class="mb-0">{{ zaspas.hoste.jmeno }}</h3>
    </div>
  </div>
</div>
```

Obrázek 3.5: rozhraní pro diváky

2. **Rozhraní pro zapisovatele** - s nástroji pro snadnou úpravu statistik, které využívá modální okna pro rychlé zadávání změn bez nutnosti opouštět hlavní obrazovku

Pro vizualizaci statistik jsem implementoval grafy pro porovnání výkonu týmů, barevné kódování pro stav zápasu a trestů, přehledné tabulky hráčů s jejich statistikami a responzivní design fungující na různých zařízeních. Klíčové prvky UX designu zahrnují konzistentní barevné schéma podle typu týmu (domáci/hosté), přehledné zobrazení nejdůležitějších informací (skóre, čas, stav), intuitivní ovládací prvky pro zapisovatele (tlačítka +/- pro změnu statistik) a modální okna pro rychlou úpravu dat bez opuštění hlavní obrazovky.

## 3.5 Řešení specifik jednotlivých sportů

### 3.5.1 Problém

Každý sport má svá specifická pravidla a statistiky, které bylo potřeba v aplikaci zohlednit a implementovat.

### 3.5.2 Řešení

Vytvořil jsem flexibilní systém, který detekuje typ sportu a podle něj upravuje zobrazení a funkcionalitu:

1. **Dynamické šablony** - využití Django šablonovacího systému pro podmíněné zobrazení relevantních prvků

```
{% if zapas.sport.jmeno == 'fotbal' %}
<div class="mb-3">
  <label class="form-label fw-bold">Yellow Cards</label>
  <div class="input-group">
    <button type="button" class="btn btn-outline-danger" onclick="updateCounter('yellow-cards-{{ player.id }}', -1)">-</button>
    <input type="number" class="form-control text-center" id="yellow-cards-{{ player.id }}" name="zlute_karty" value="{{ stats.zlute_karty }}" readonly>
    <button type="button" class="btn btn-outline-warning" onclick="updateCounter('yellow-cards-{{ player.id }}', 1)">+</button>
  </div>
</div>
{% elif zapas.sport.jmeno == 'hokej' %}
<div class="mb-3">
  <label class="form-label fw-bold">Penalty Minutes</label>
  <div class="input-group">
    <button type="button" class="btn btn-outline-danger" onclick="updateCounter('penalty-{{ player.id }}', -2)">-2m</button>
    <input type="number" class="form-control text-center" id="penalty-{{ player.id }}" name="trestne_minuty" value="{{ stats.trestne_minuty }}" readonly>
    <button type="button" class="btn btn-outline-warning" onclick="updateCounter('penalty-{{ player.id }}', 2)">+2m</button>
    <button type="button" class="btn btn-outline-danger" onclick="updateCounter('penalty-{{ player.id }}', 5)">+5m</button>
  </div>
</div>
```

Obrázek 3.6: podmíněné zobrazení

2. **Specifické JavaScript funkce** - implementace specifických pravidel času v JavaScript kódu

```
const container = document.querySelector('[data-sport]');
if (container) {
  const sport = container.getAttribute('data-sport');

  if (sport === 'fotbal') {
    matchDuration = 90 * 60;
  } else if (sport === 'hokej') {
    matchDuration = 60 * 60;
  } else if (sport === 'hazena') {
    matchDuration = 60 * 60;
  }
}
```

Obrázek 3.7: nastavení času

3. **Specifické chování v backendu** - implementace pravidel na serverové straně, například automatická diskvalifikace hráče v házené po třech trestech

Důležitou součástí implementace byla také vizuální stránka - pro každý sport jsou použity specifické ikony a pozadí, což zlepšuje uživatelský zážitek a pomáhá na první pohled identifikovat typ sportu. Tato řešení umožnila zobrazovat relevantní statistiky podle typu sportu, implementovat specifická pravidla pro každý sport a správně měřit čas zápasu podle pravidel sportu. Implementace těchto klíčových řešení zajistila, že aplikace Stats Score poskytuje komplexní a uživatelsky přívětivý nástroj pro správu a sledování sportovních zápasů s ohledem na specifika jednotlivých sportů.

## 4 Popis funkcionality

Tato kapitola detailně popisuje hlavní funkční části aplikace Stats Score a vysvětluje, jak jednotlivé komponenty systému pracují z perspektivy uživatele.

### 4.1 Zobrazení zápasů a soutěží

Základním prvkem aplikace je přehledná prezentace zápasů organizovaných podle sportovních soutěží. Tato funkcionality je klíčová pro orientaci uživatelů v systému.

#### 4.1.1 Domovská stránka

Domovská stránka poskytuje úvodní informace o aplikaci a jejím účelu. Obsahuje uvítací sekci s názvem aplikace, stručný popis hlavních funkcí systému a navigační menu pro přístup k soutěžím jednotlivých sportů.

#### 4.1.2 Seznam soutěží a zápasů

Každý sport má vlastní sekci s přehledem soutěží. Po vybrání konkrétní soutěže se zobrazí seznam všech zápasů. Zápas jsou prezentovány jako karty s základními informacemi. U každého zápasu je vidět stav (nadházející, probíhající, pozastavený, ukončený). Implementován je filtr, který umožňuje zobrazit pouze zápasy v určitém stavu. U každého zápasu jsou k dispozici tlačítka pro přechod na detail nebo ovládací panel. Zvláštní pozornost byla věnována vizuální stránce - každý sport má vlastní tematické pozadí, které pomáhá uživatelům v orientaci a zlepšuje celkový dojem z aplikace.

### 4.2 Detail zápasu a statistiky

Detail zápasu je nejkomplexnější částí veřejného rozhraní aplikace. Poskytuje kompletní přehled o průběhu a statistikách konkrétního utkání.

#### 4.2.1 Hlavní informační panel

Horní část obrazovky obsahuje nejdůležitější informace o zápasu: názvy soupeřících týmů, aktuální skóre, stav zápasu, datum a čas zápasu, typ sportu s ikonou a časomíru zobrazující aktuální čas zápasu.

#### 4.2.2 Přehled hráčů a statistik

Pro oba týmy jsou zobrazeny tabulky se seznamem hráčů a jejich aktuálními statistikami: číslo a jméno hráče, počet vstřelených gólů a specifické statistiky podle typu sportu; fotbal zobrazuje žluté karty, zatímco hokej a házená zobrazují počet a délku trestů.

#### 4.2.3 Trestná lavice

Pro hokej a házenou je implementována speciální sekce "Trestná lavice", která zobrazuje seznam aktuálně potrestaných hráčů, jejich týmovou příslušnost, zbývající čas trestu s automatickým odpočtem a indikací pozastavení při přerušení zápasu.



#### 4.2.4 Souhrnné statistiky

Spodní část obrazovky obsahuje vizualizaci souhrnných statistik zápasu: grafické porovnání počtu gólů mezi týmy, celkový počet gólů každého týmu, počet žlutých karet (fotbal) nebo trestných minut (hokej, házená) a procentuální vyjádření úspěšnosti týmů. Všechny statistiky a informace v detailu zápasu se automaticky aktualizují bez nutnosti obnovení stránky, což zajišťuje plynulé sledování průběhu utkání.

### 4.3 Ovládací panel pro zapisovatele

Ovládací panel je specializované rozhraní určené pro zapisovatele, kteří zadávají statistiky během průběhu zápasu.

#### 4.3.1 Řízení průběhu zápasu

Horní část panelu obsahuje ovládací prvky pro řízení zápasu: tlačítko pro zahájení zápasu (přístupné, když je zápas ve stavu "nadházející" nebo "pozastavený"), tlačítko pro pozastavení zápasu (přístupné, když zápas probíhá), tlačítko pro ukončení zápasu (přístupné, když zápas probíhá nebo je pozastavený) a časomíru zobrazující celkový odehraný čas. Systém automaticky řídí dostupnost těchto tlačítek podle aktuálního stavu zápasu, což zabraňuje nekonzistentním akcím.

#### 4.3.2 Správa statistik hráčů

Pro každého hráče účastnícího se zápasu je k dispozici karta s možností úpravy statistik: zobrazení aktuálních statistik hráče a tlačítko pro otevření modálního okna s detailním zadáváním statistik. Modální okno pro úpravu statistik obsahuje počítadla s tlačítky + a - pro snadné přidávání a odebrání gólů, podle typu sportu specifické prvky (fotbal má počítadlo žlutých karet a tlačítko pro červenou kartu, hokej má tlačítka pro udělení 2 a 5 minutových trestů, házená má tlačítka pro 2-minutové tresty a červenou kartu) a zaškrtnuté pole pro označení diskvalifikovaného hráče. Systém automaticky aplikuje pravidla daného sportu: ve fotbale diskvalifikuje hráče při druhé žluté kartě, v házené počítá celkový počet trestů a po třetím automaticky diskvalifikuje hráče.

#### 4.3.3 Správa trestů

Pro sporty s časovými tresty (hokej, házená) systém poskytuje komplexní nástroje pro správu trestů: automatické odpočítávání zbývajících času trestu, synchronizace odpočtu s časomírou zápasu (pauzování při přerušení zápasu), vizuální indikace stavu trestu včetně automatického odstranění po vypršení a automatická aktualizace databáze po uplynutí trestu.

## **4.4 Správa týmů a hráčů**

Aplikace obsahuje administrační rozhraní pro správu základních funkcí systému.

### **4.4.1 Správa sportů**

Systém podporuje tři typy sportů (fotbal, házená, hokej) s možností rozšíření. Každý sport má definována specifická pravidla, vizuální prvky se přizpůsobují typu sportu a formuláře a statistiky se dynamicky mění podle vybraného sportu.

### **4.4.2 Správa týmů**

Administrátoři mohou spravovat týmy prostřednictvím administračního rozhraní: vytváření nových týmů s přiřazením ke konkrétnímu sportu, přidávání a odebírání hráčů z týmu a vyhledávání týmů podle názvu nebo sportu.

### **4.4.3 Správa hráčů**

Systém umožňuje komplexní správu hráčů: vytváření nových hráčů s přiřazením čísla dresu, přiřazování hráčů k jednomu či více týmům a vyhledávání hráčů podle jména nebo čísla dresu.

### **4.4.4 Správa zápasů a lig**

Administrátoři mohou vytvářet a spravovat zápasy a soutěže: vytváření nových zápasů s určením soupeřících týmů, data a času, organizace zápasů do soutěže podle typu sportu a filtrování a vyhledávání zápasů podle různých kritérií. Všechny tyto administrační funkce jsou přístupné přes integrované Django administrační rozhraní, které poskytuje intuitivní a bezpečný způsob správy dat aplikace.

Celkově aplikace Stats Score nabízí komplexní řešení pro správu sportovních zápasů s důrazem na uživatelskou přívětivost, real-time aktualizace a specifické funkce pro různé typy sportů.

## 5 ZÁVĚR

Ročníkový projekt Stats Score jsem dokončil a vytvořil moderní webovou aplikaci, která splňuje všechny stanovené cíle. Podařilo se mi implementovat komplexní systém pro správu a sledování sportovních zápasů s důrazem na specifika jednotlivých sportovních disciplín.

Vývoj projektu Stats Score pro mě byl velmi cennou zkušeností, která mi umožnila aplikovat a rozšířit mé znalosti v oblastech webového vývoje. Do budoucna by se dala webová stránka vylepšit přidáním dalších sportů, pro které by bylo jednoduché sledování a zapisování statistik z utkání velmi přínosné.

Věřím, že vytvořená aplikace má potenciál pro reálné využití v prostředí amatérských sportovních soutěží a klubů, kde může výrazně zjednodušit správu zápasů a poskytování informací divákům a fanouškům.

## POUŽITÁ LITERATURA

1. Django Software Foundation. (2024, prosinec 12). Django 5.1 documentation. Získáno z <https://docs.djangoproject.com/en/5.1/>
2. Holovaty, A., & Kaplan-Moss, J. (2024). The Definitive Guide to Django: Web Development Done Right. (5. vydání). Apress.
3. MDN Web Docs. (2024, prosinec 18). JavaScript Guide. Získáno z <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide>
4. Web Accessibility Initiative (WAI). (2025, leden 6). Web Content Accessibility Guidelines (WCAG) 3.0 Working Draft. Získáno z <https://www.w3.org/WAI/standards-guidelines/wcag/wcag3-intro/>
5. Richardson, L., & Ruby, S. (2024). RESTful Web Services: Web APIs for the Modern Web. (3. vydání). O'Reilly Media.
6. Mozilla Developer Network. (2025, leden 12). Using the Fetch API. Získáno z [https://developer.mozilla.org/en-US/docs/Web/API/Fetch\\_API/Using\\_Fetch](https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch)
7. Sportovní federace ČR. (2025, leden 20). Pravidla fotbalu 2025. Získáno z <https://www.fotbal.cz/rozhodci/ke-stazeni/p15>
8. Český svaz házené. (2025, leden 28). Pravidla házené 2024-2025. Získáno z <https://www.handball.cz/aktualita/pravidlahazene>
9. Český svaz ledního hokeje. (2025, leden 30). Pravidla ledního hokeje 2024-2025. Získáno z [https://www.ceskyhokej.cz/data/document/20240807/100157\\_d1b0\\_240806-pravidla-2024-25-w.pdf](https://www.ceskyhokej.cz/data/document/20240807/100157_d1b0_240806-pravidla-2024-25-w.pdf)
10. Naik, N. (2025, únor 12). Unleashing Real-time Web Applications with Django Channels. Získáno z <https://naiknotebook.medium.com/unleashing-real-time-web-applications-with-django-channels-7c0194e7c4a6>
11. Autor není znám. (2025, únor 20). Modern JavaScript for Django Developers. Získáno z <https://www.saaspegasus.com/guides/modern-javascript-for-django-developers/>

12. Mozilla Developer Network. (2025, únor 28). Working with the JavaScript Timing Events.

Získáno z <https://developer.mozilla.org/en-US/docs/Web/API/setTimeout>