

BGSzC Pestszentlőrinci Közgazdasági és Informatikai Szakgimnáziuma

1184 Budapest Hengersor 34.

# Vizsgaremek dokumentáció *S-port*

A csoport tagjai:

Naszvetter Béla

Vámos Attila

Király Kristóf

Készítette:

Naszvetter Béla

Vámos Attila

Király Kristóf

## Tartalom

1	Bevezetés	3
1.1	Feladat leírás	3
1.2	A felhasznált ismeretek	3
1.3	A felhasznált szoftverek	4
2	Felhasználói dokumentáció	5
2.1	A program általános specifikációja	5
2.2	Rendszerkövetelmények	5
2.2.1	Hardver és szoftver követelmények	5
2.3	A program telepítése	6
2.4	A program használatának a részletes leírása	7
3	Fejlesztői dokumentáció	11
3.1	Az alkalmazott fejlesztői eszközök	11
3.2	Adatmodell leírása	11
3.3	Részletes feladat-specifikáció, algoritmusok	15
3.4	Front-end fejlesztése, specifikációi	20
3.5	Tesztelési dokumentáció	23
4	Összefoglalás	25
4.1	Önértékelés	25
4.2	Továbbfejlesztési lehetőségek	25
5	Felhasznált irodalom	26

# 1 Bevezetés

## 1.1 Feladat leírás

Feladatként egy sport alkalmazást elkészítését választottuk, amelynek segítségével a sportolni szerető emberek könnyebben tudnak kapcsolatot teremteni egymással, egy “kikötőként” megtudják osztani egymással azokat a sportolási alkalmakat amelyekhez bármely regisztrált tag szabadon tud csatlakozni.

Azért ezt a témát választottuk, mert mindhárman szeretünk sportolni, és többször találkoztunk olyan helyzettel amikor szívesen kimozdultunk volna otthonról, de azt nem egyedül, hanem egy csoportban szerettük volna megtenni. Erre a problémára megoldás az általunk elkészített alkalmazást.

## 1.2 A felhasznált ismeretek

Webtechnológiák:

HTML: leíró nyelv amelynek segítségével weboldalak lehet létrehozni

CSS: stílusleíró nyelv, amely a megjelenítéshez és kinézetéhez szükséges

Bootstrap 5: CSS keretrendszer amelynek a segítségével reszponzív oldalakat lehet létrehozni, és megkönnyíti a fejlesztést

PHP: Hypertext Preprocessor- szerver oldali szkriptnyelv dinamikus weboldalak készítésére. Segítségével biztosított a weboldal és az adatbázis közötti kapcsolat.

Javascript: kliens oldali programozási nyelv, segítségével interaktívvá tehető egy weboldal

JQUERY: népszerű könyvtára a javascripteknek, egyszerűbbé teszi a HTML elemekre való hivatkozást

Adatbázis kezelés

MYSQL: SQL alapú relációs adatbázis kezelő szerver

C# : általános célú programnyelv, a .NET keretrendszer legfontosabb programnyelve

### **1.3 A felhasznált szoftverek**

Visual Studio Code

Visual Studio

XAMPP

phpMyAdmin

GITHUB

Chrome

Azure

Project Libre

Workbench

Paint

## 2 Felhasználói dokumentáció

### 2.1 A program általános specifikációja

A program használói azonos jogosultságokkal rendelkeznek. A használathoz regisztráció szükséges, és a belépést követően van lehetősége a felhasználónak keresni a sportesemények között, illetve új események rögzítésére.

Az adminisztrátori feladatok az elkészített asztali alkalmazás segítségével kerülnek megvalósításra. Az adminisztrátornak van lehetősége egy régebbi eseményt törölni, illetve felhasználót is rögzíthet/törölhet.

### 2.2 Rendszerkövetelmények

#### 2.2.1 Hardver és szoftver követelmények

##### **Minimum követelmények:**

XAMPP valamint Google Chrome telepítésére és futtatására alkalmas számítógép: 2 GB RAM és 20 GB ROM.

Operációs rendszer: Windows 8, Windows 8.1.

Képernyő: 800x600

##### **Ajánlott követelmények:**

Processzor: 2 GHz, RAM: 4 GB, 40 GB ROM és Windows 10, Windows 11 valamint Google Chrome használata.

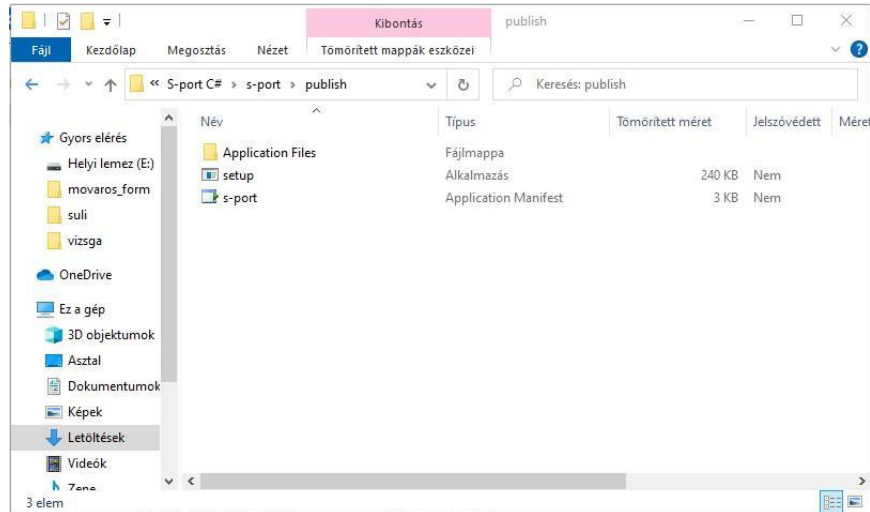
Képernyő: 1920x1080

## 2.3 A program telepítése

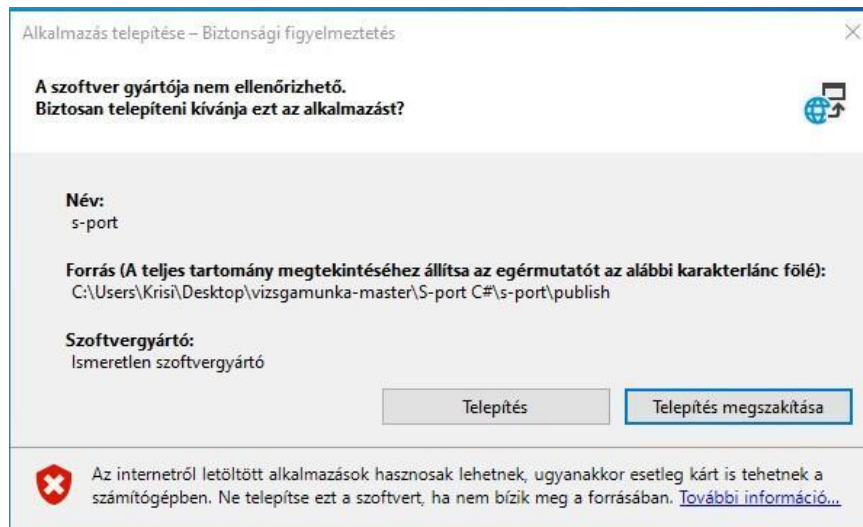
Készítette: Király Kristóf

Képekkel illusztrált, részletes leírás a program telepítésének a menetéről.

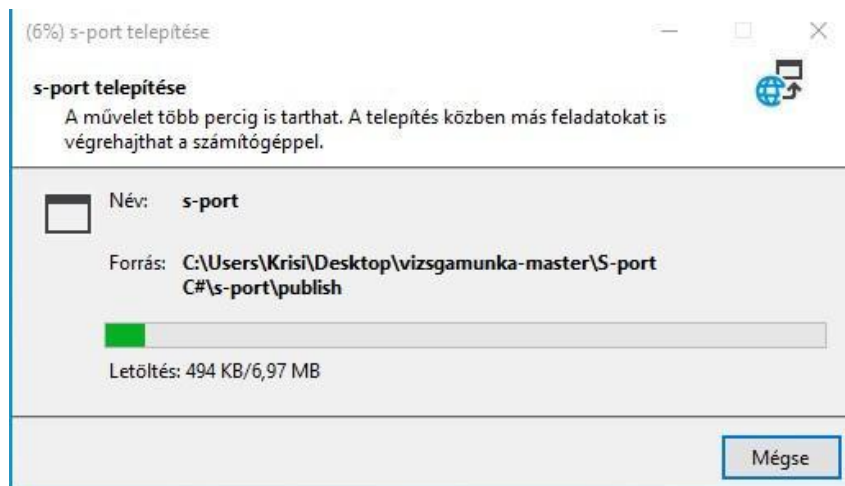
- Ki kell választani a setup.exe alkalmazást amit el kell indítani



- Ha elindítottuk az alkalmazást rá kell kattintani a telepítés gombra



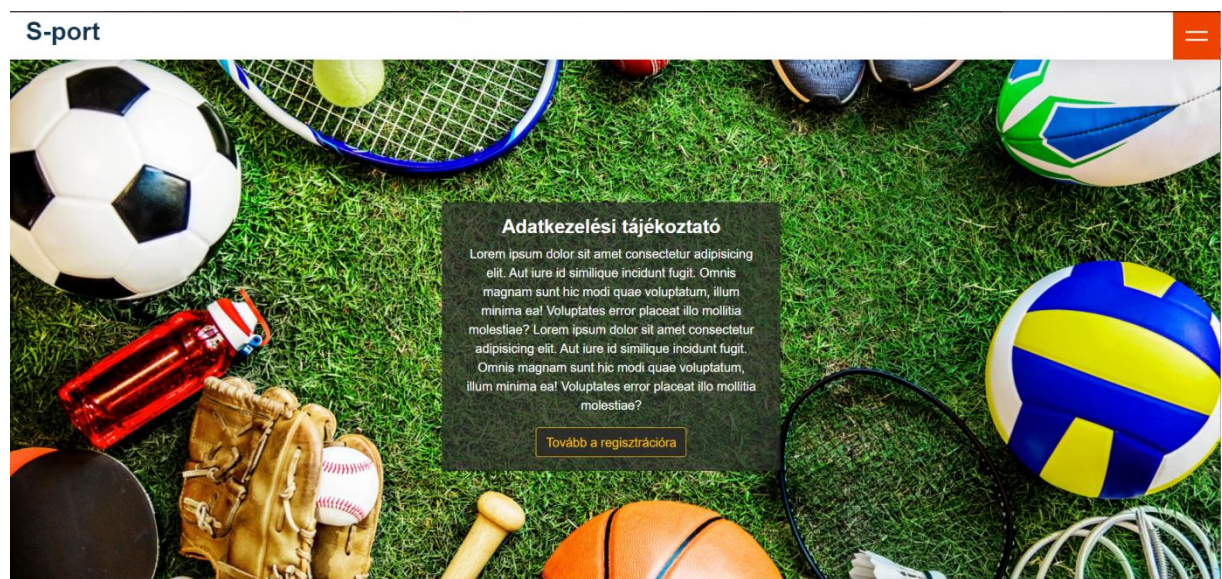
- Ha elindul a telepítés meg kell várni amíg végez a betöltés



- Amint végzett a telepítés egyből elindul a szoftver

## 2.4 A program használatának a részletes leírása

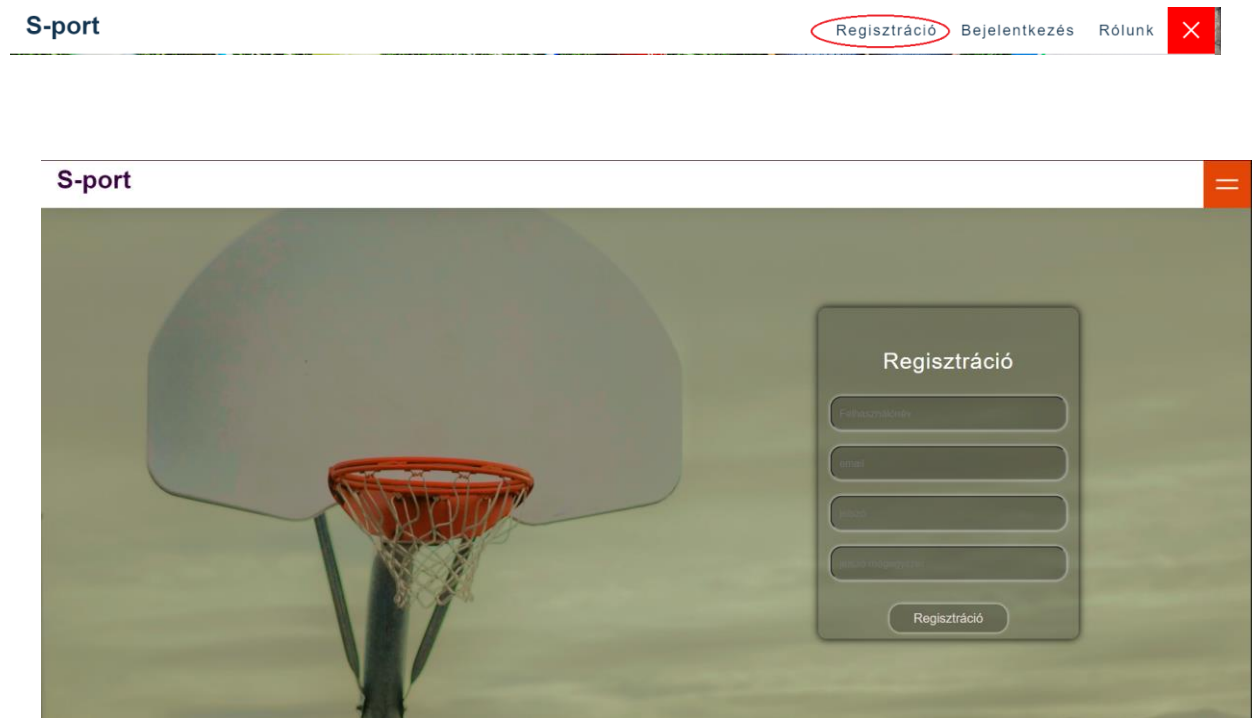
Készítette: Vámos Attila



A fenti képen az adatkezelési tájékoztató oldal látható. Erre az oldalra a regisztrációs oldalon rákattintva a S-port logóra vagy a menüből kiválasztva az Adatkezelés menüpontot lehet eljutni.

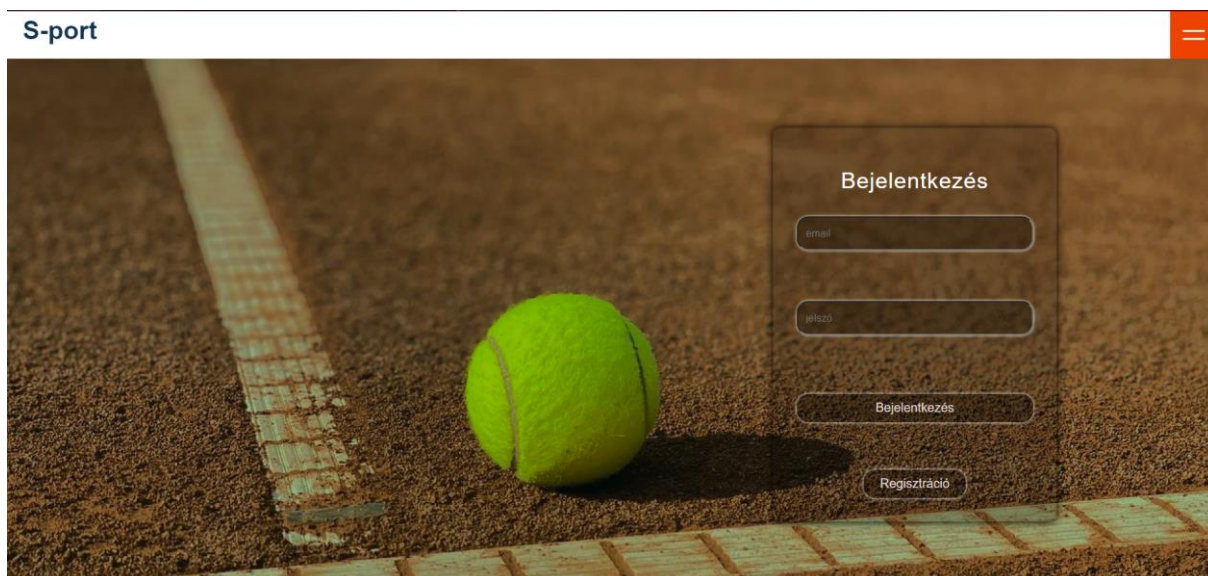


Azért így alkottuk ezt meg, hogy a felhasználó még a regisztráció előtt tájékozódni tudjon az adatkezeléssel kapcsolatosan és eldönthesse, hogy szeretne e regisztrálni. A regisztrációs oldalra a menüben található „Regisztráció” gomba segítségével lehet visszajutni.

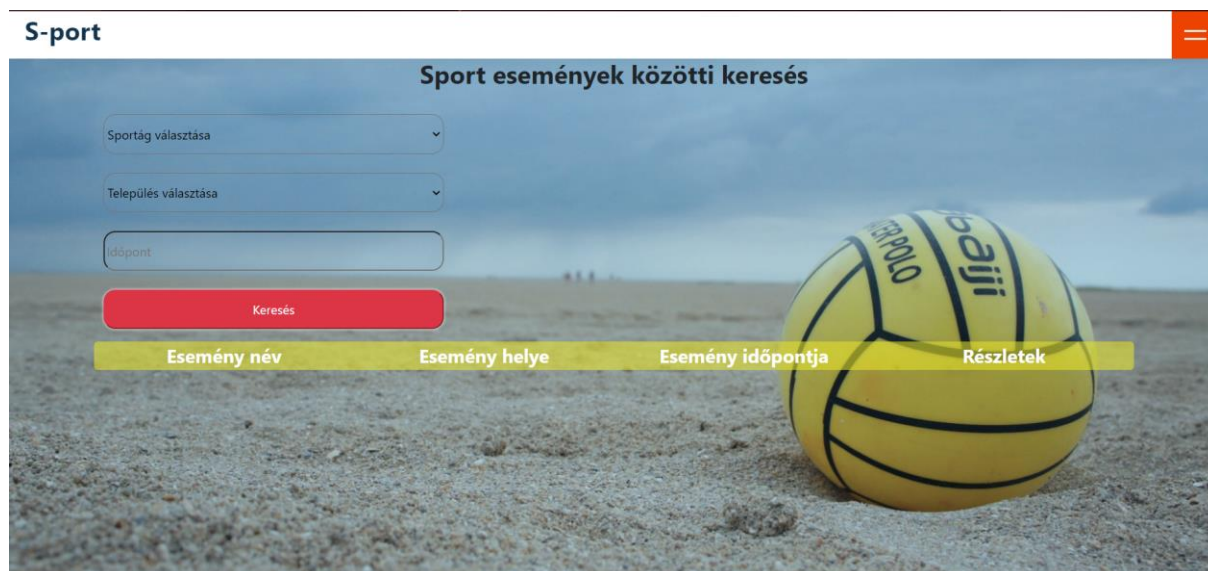


Első lépésként regisztrálni kell a weboldalra. A felhasználó egyedi felhasználónevet tud választani magának, majd egy e-mail címet is meg kell adnia. Fontos, hogy az e-mail formátumnak stimmelnie kell. Végül jelszót kell beírni. Ezek után a regisztráció gombra kattintva egyrésztől tárolódnak a megadott adatok az adatbázisban másrésztől átkerül a bejelentkezés oldalra, ahol a fentiekben leírt felhasználónév és jelszó párossal már be tud jelentkezni az oldalra.





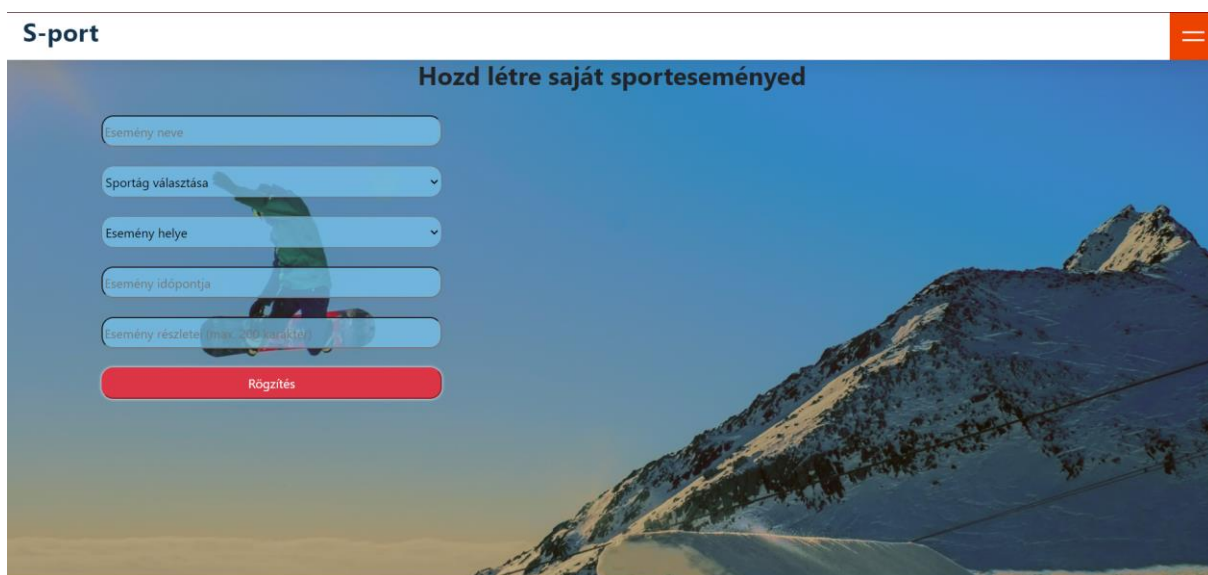
Bejelentkezés után az események oldalra kerül, ahol már keresni tud az események között, egyszerűen csak sportágot, települést és időpontot kell választania majd a “Keresés” gombra kattintani. Természetesen a keresést úgy alkottuk meg, hogy már egy megadott keresési feltételre is adjon találatot, ha azt tartalmazza az adatbázis.



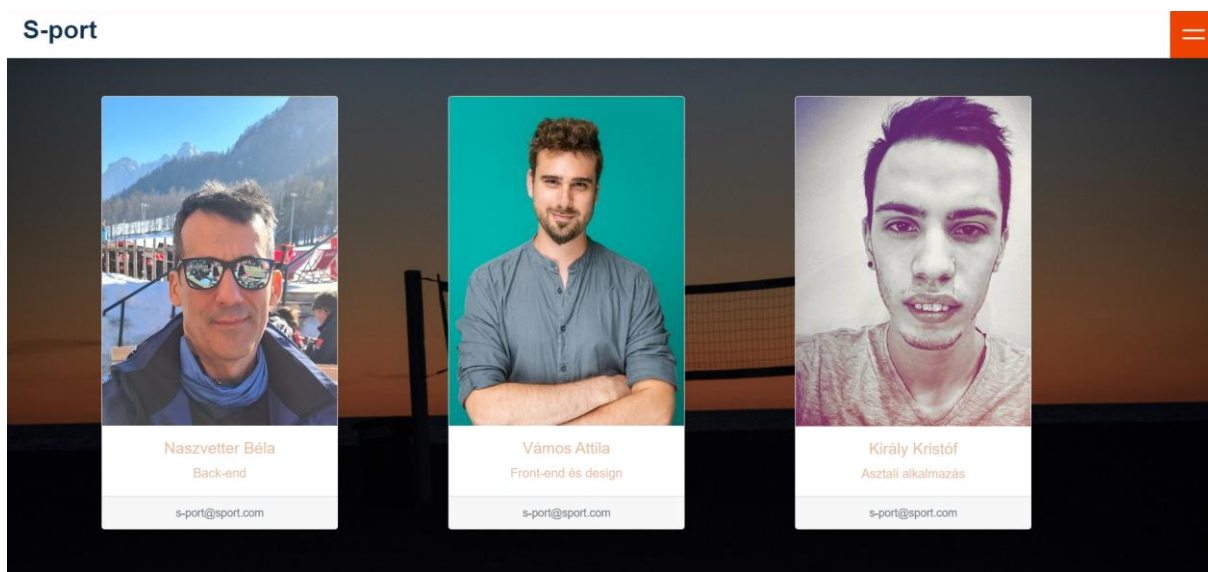
Innen a navigációs menüre vagy angol nevén “Navbar”-ra kattintva tudunk átmenni az új események oldalra, ahol a felhasználó eseményeket tud létrehozni és felvinni az adatbázisba.



Az új esemény oldalon a felhasználó elnevezheti saját eseményét, majd ezt a sportágak kategóriában tudja pontosítani. Ezek után meg kell adnia az esemény helyszínét, amit az “Esemény helye” nevű legördülő listából tud kiválasztani. Ha ezzel meg van, időpontot is kell választania egy legördülő naptár segítségével. Itt az évet, hónapot és napot tudja megadni. Az “Esemény részletei” szöveg dobozba pedig pontos leírást tud hozzáadni.



A “Rögzítés” gombra kattintva tudja létrehozni az új eseményt mely az Események oldalon egyből kereshetővé válik. Természetesen a S-port alkalmazás fejlesztőiről is elérhetőek információk, azt a látogatók és tagok ezen az oldalon tekinthetik meg.



## 3 Fejlesztői dokumentáció

### 3.1 Az alkalmazott fejlesztői eszközök

A program fejlesztéséhez használt eszközök:

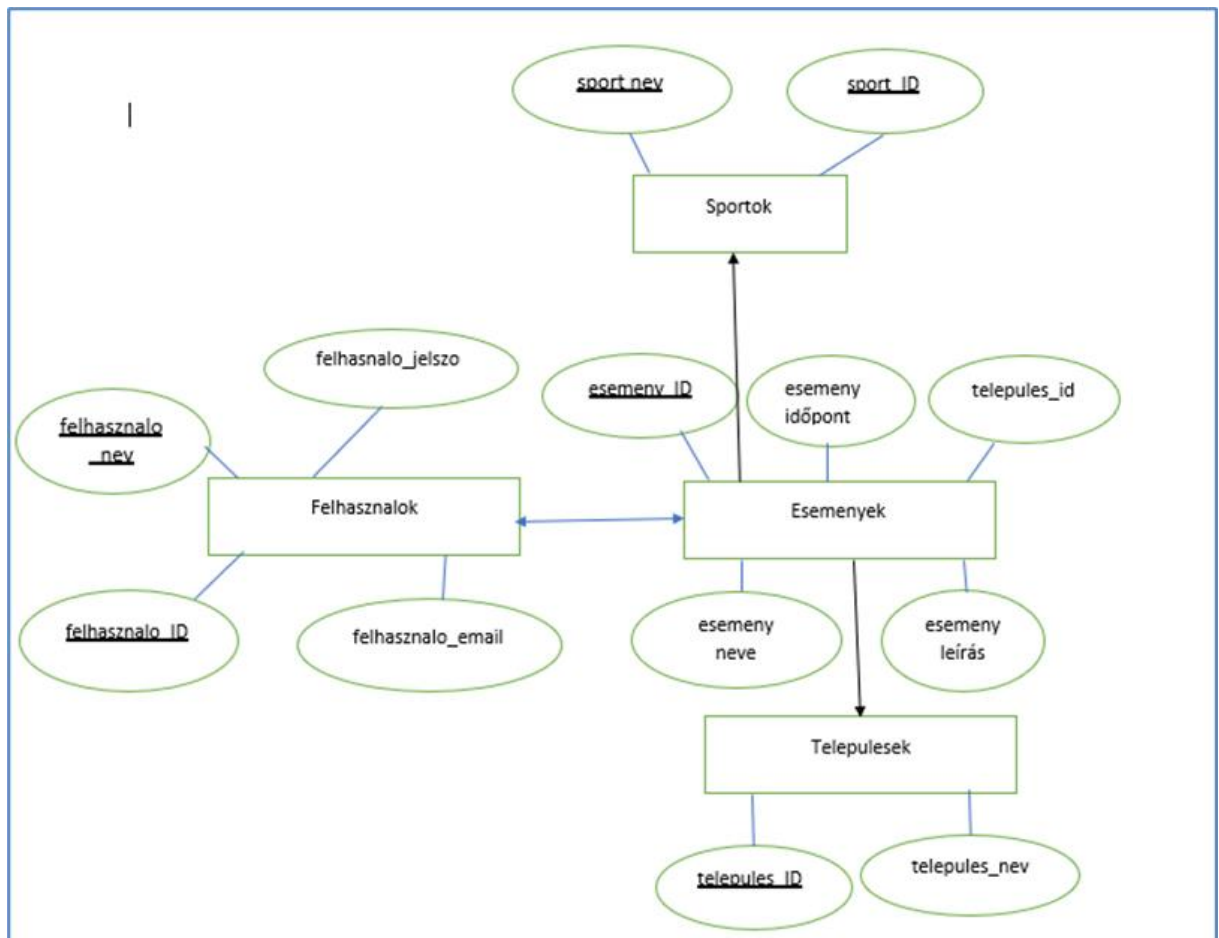
- Programozási leíró, és lekérő nyelvek:
  - HTML
  - CSS, Bootstrap5
  - Javascript
  - JQUERY
  - PHP
  - MYSQL
- Adatbázis kezelő rendszer:
  - XAMPP
- Fejlesztői környezet:
  - Visual Studio
- Adatbázis modell kialakításához használt program:
  - Workbench

### 3.2 Adatmodell leírása

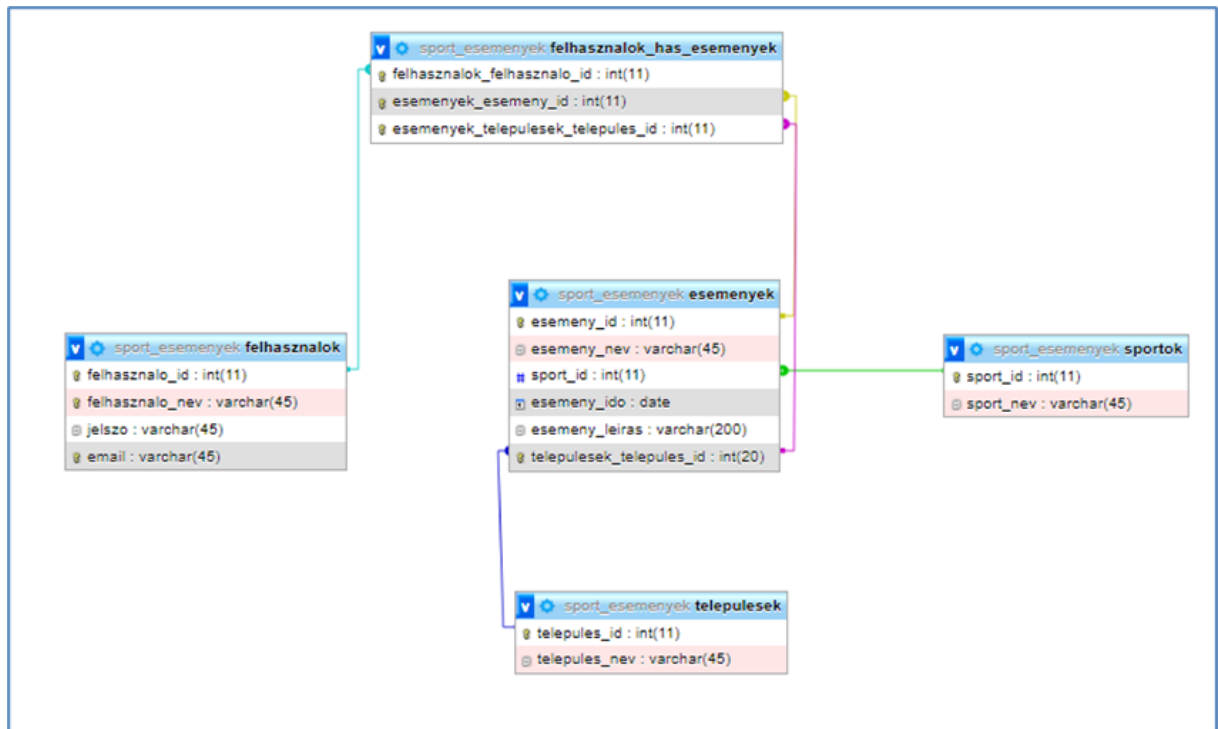
Készítette: Naszvetter Béla

Az adatbázis célja, hogy tárolja a regisztrált felhasználókat, a rögzített sport eseményeket, településeket, és sportágakat.

S-port adatbázisának egyed diagramja:






S-port adatbázisának diagramja:



Az adatbázis táblái:

#### 1. felhasználók tábla:

Ennek a táblának a feladata a regisztrációs adatok tárolása, azoknak a felhasználói adatoknak a tárolása amelyek a bejelentkezéshez szükségesek.

#	Név	Típus
1	felhasznalo_id 	int(11)
2	felhasznalo_nev 	varchar(45)
3	jelszo	varchar(45)
4	email 	varchar(45)

- A felhasznalo-id mező a felhasználók tábla elsődleges kulcsa ami egyértelműen azonosítja az egyedet.

-a felhasznalo\_nev és az email címeknek egyedieknek kell lenniük.

## 2. telepulesek tábla:


Ennek a táblának a feladata, hogy tartalmazza a magyarországi települések neveit, amelyekre szükségünk lesz az egyes események helyének meghatározásakor

#	Név	Típus
1	telepules_id 	int(11)
2	telepules_nev	varchar(45)

-A telepules\_id mező a telepulesek tábla elsődleges kulcsa ami egyértelműen azonosítja az egyedeket.

## 3. sportok tábla:

Ennek a táblának a feladata, hogy tartalmazza a sportágak neveit amelyek az egyes események rögzítéséhez szükséges.

#	Név	Típus
1	sport_id 	int(11)
2	sport_nev	varchar(45)

-A sport\_id mező a sportok tábla elsődleges kulcsa ami egyértelműen azonosítja az egyedeket.

## 3. esemenyek tábla:

Ennek a táblának a feladata, hogy tartalmazza a rögzített sport eseményeket.

#	Név	Típus
1	esemeny_id 	int(11)
2	esemeny_nev	varchar(45)
3	sport_id 	int(11)
4	esemeny_ido	date
5	esemeny_leiras	varchar(200)
6	telepulesek_telepules_id 	int(20)

-Az esemenyek\_id mező az esemenyek tábla elsődleges kulcsa ami egyértelműen azonosítja az egyedeket.



- A `sport_id`, és a `telepulesek_telepules_id` mezők idegen kulcsok, amelyek segítségével biztosított a kapcsolat a sportok, és a telepulesek táblák között.

### 3.3 Részletes feladatspecifikáció, algoritmusok

Készítette: Naszvetter Béla

#### Kapcsolódás az adatbázishoz

##### · Ismertetés:

Az adatbázishoz való kapcsolódást a `connect.php` fájlban valósítottam meg. Minden alkalommal amikor szükséges az adatbázis elérése ez a fájl kerül meghívásra.

Ennek előnye, hogy ha a későbbiekben módosítani kell az adatbázis nevét, vagy elérést, akkor azt csak egy helyen kell megtenni.

#### Kódrészlet(ek) és magyarázat

```
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "sport_esemenyek";

$connect = new mysqli($servername, $username, $password, $dbname);
if ($connect->connect_error) {
    die("Sikertelen kapcsolódás: " . $connect -> connection_error);
} else {
    echo "";
}
```

A MySQLi modult használtam, az „\$connect” változó tartalmazza az adatbázissal való kapcsolatot, és külön változóba tettem a server nevét, felhasználó nevét, jelszót, és az adatbázis nevét.

## Regisztráció

### Ismertetés:

A regisztrációhoz a felhasználótól csak egyedi felhasználó nevet, email címet, és jelszót kérünk. A nagyobb biztonság érdekében a felhasználótól kapott adatokat nem közvetlenül az SQL utasításba építettem be, hanem paraméterezve kapja meg az értékeit, majd végrehajtódik az SQL lekérdezés. A következő kódrészlet a index\_registration.php fájlban található

### Kódrészlet(ek) és magyarázat

```
//regisztráció
$felhasznalo_nev=mysqli_real_escape_string($connect, $_POST['felhasznalo_nev']);
$email=mysqli_real_escape_string($connect, $_POST['email']);
$jelszo=mysqli_real_escape_string($connect, $_POST['jelszo']);
$jelszo=md5($jelszo);

$sql="INSERT INTO felhasznalok(felhasznalo_nev, jelszo, email) VALUES
('$felhasznalo_nev','$jelszo','$email')";
$connect-> query($sql);
session_start();
$_SESSION['email']=$email;
header('location:index_login.php');
```

mysqli\_real\_escape\_string(): ez a függvény paraméterként egy kapcsolatváltozót és egy stringet vár, amit átalakít úgy, hogy a „kártékony” SQL kódokat kiveszi belőle és enélkül adja vissza a stringet. Ez a függvény véd az SQL injection támadás ellen.

a „\$sql” változó tartalmazza az SQL INSERT INTO utasítást amellyel rögzítésre kerül a felhasználó név, email cím, és a jelszó.

A megadott jelszó MD5 (message digest algorithm 5) 128 bites egyirányú kódolású titkosítással kerül mentésre az adatbázisba.

## Bejelentkezés

### Ismertetés:

A nagyobb biztonság érdekében a felhasználótól kapott adatokat nem közvetlenül az SQL utasításba építettem be, hanem paraméterezve kapja meg az értékeit,

majd végrehajtódik az SQL lekérdezés. A következő kódrészlet az index\_login.php fájlban található:



```
$email=mysqli_real_escape_string($connect, $_POST['email']);
$jelszo=mysqli_real_escape_string($connect, $_POST['jelszo']);
$jelszo=md5($jelszo);

$sql="SELECT * FROM felhasznalok WHERE email = '$email' AND jelszo='$jelszo'";
$query = $connect -> query ($sql);
if (mysqli_num_rows($query) == 1)
{
    session_start();
    $_SESSION['email'] = $email;
    header('location: esemenyek.php');
}
```

mysqli\_real\_escape\_string(): ez a függvény paraméterként egy kapcsolatváltozót és egy stringet vár, amit átalakít úgy, hogy a „kártékony” SQL kódokat kiveszi belőle és enélkül adja vissza a stringet. Ez a függvény véd az SQL injection támadás ellen

## Sport események oldal

### Ismertetés:

Az oldalon három paraméter megadásával van lehetőség az események közötti keresésre.

-sportág neve (a kiválasztható sportágak az adatbázis sportok táblázatából töltődnek be)

-az esemény helye (a kiválasztható települések az adatbázis települések táblázatából töltődnek be)

-az esemény ideje: az időpont megadásánál a jQuery javascript keretrendszer datepicker modulját használtam

April 2022						
Su	Mo	Tu	We	Th	Fr	Sa
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

A következő kódrészlet az esemenyek.php fájlban található:

## Kódrészlet(ek) és magyarázat

```
include_once 'include/connect.php';
if(isset($_POST['submit']))
{
    $telepulesek=$_POST['esemeny_hely'];
    $sportok=$_POST['sportag'];
    $fromdate=$_POST['esemeny_ido'];
    $fdate= strtotime($fromdate);
    $fdate= date("Y/m/d", $fdate);

    if ($telepulesek != "" || $fromdate != "" || $sportok!="")
    {
        $query= ("SELECT esemenyek.esemeny_nev , esemenyek.sport_id , esemenyek.
        telepulesek_telepules_id , esemenyek.esemeny_ido , esemenyek.esemeny_leiras FROM (esemenyek
        INNER JOIN telepulesek ON esemenyek.telepulesek_telepules_id=telepulesek.telepules_id)
        INNER JOIN sportok ON esemenyek.sport_id=sportok.sport_id WHERE esemeny_ido='$fdate' OR
        telepulesek.telepules_nev ='$telepulesek' OR sportok.sport_nev='$sportok'");
        $data= mysqli_query($connect, $query) OR die('error');
```

if utasítással vizsgáljuk meg, hogy a kereséshez megadható három paraméter közül van-e legalább egy olyan amelyikhez tartozik érték.

Amennyiben igen, akkor a \$query változóban meghatározott SQL lekérdezés megtörténik. A három adattábla van használva (esemenyek, telepulesek, sportok) amelyeket INNER JOIN-nal kötöttem össze.

## Új sportesemény rögzítése

### Ismertetés:

Egy új esemény rögzítéséhez öt adatra van szükség:

- sportág neve (a választható sportágak az adatbázis sportok táblázatából töltődnek be)
- az esemény helye (a kiválasztható települések az adatbázis települések táblázatából töltődnek be)
- esemény időpontja
- esemény neve
- esemény leírás

## Kódrészlet(ek) és magyarázat

```
//új esemény rögzítése
$esemenynev=mysqli_real_escape_string($connect, $_POST['esemeny_nev']);

$sportag=mysqli_real_escape_string($connect, $_POST['sportag']);
$query_sportagid= ("SELECT sport_id FROM sportok WHERE sport_nev= '$sportag' ");
$sportagid1= mysqli_query($connect, $query_sportagid) OR die('error');
if(mysqli_num_rows($sportagid1)>0)
{
    while($row=mysqli_fetch_assoc($sportagid1))
    {
        $sportagid= $row['sport_id'];
    }
}
$esemenyhely=mysqli_real_escape_string($connect, $_POST['esemeny_hely']);
$query_telepulesektelepulesid= ("SELECT telepules_id FROM telepulesek WHERE
telepules_nev= '$esemenyhely' ");
$telepulesektelepulesid1= mysqli_query($connect, $query_telepulesektelepulesid) OR
die('error');
if(mysqli_num_rows($telepulesektelepulesid1)>0)
{
    while($row=mysqli_fetch_assoc($telepulesektelepulesid1))
    {
        $telepulesektelepulesid= $row['telepules_id'];
    }
}
```

Mivel az esemény rögzítésekor a település nevét, és az érintett sportágot adja meg a sporteseményt rögzítő, de azok az értékek közvetlenül nem rögzíthetők az események adattáblában, csak az azokhoz tartozó „id”-k.

Ezért a település nevéhez tartozó telepules\_id-t kell megkapni egy SQL SELECT utasítással, amit a \$query\_telepulesektelepulesid1 változóba kell elmenteni, majd azt while ciklussal, és a mysqli\_fetch\_assoc() függvény segítségével a \$telepulesektelepulesid változóba elmenteni.

Majd azt az értéket már az INSERT INTO utasítással már betudjuk írni az események táblázatba:

```
$sql="INSERT INTO esemenyek(esemeny_nev, sport_id,esemeny_ido , esemeny_leiras,
telepulesek_telepules_id) VALUES('$esemenynev','$sportagid','$fdate',
'$esemenyleiras', '$telepulesektelepulesid')";
$connect-> query($sql);
```

### 3.4 Front-end fejlesztése, specifikációi

Készítette: Vámos Attila

Elsőként a regisztrációs és bejelentkezési HTML oldalakat írtam meg. A regisztrációs form kódja itt látható:

```
<form action="index_registration.php" method="POST">
  <div class="container-login">
    <div class="login-form">
      <div class="login">
        <h3>Regisztráció</h3>
      </div>
      <input type="text" name="felhasznalo_nev" placeholder="Felhasználónév">
      <input type="email" name="email" placeholder="email">
      <input type="password" name="jelszo" placeholder="jelszó">
      <input type="password" name="jelszo_ujra" placeholder="jelszó még egyszer">
      <input class="btn" value="Regisztráció" type="submit" name="submit"/>
    </div>
  </div>
</form>
```

Form segítségével kapcsoltuk össze a PHP-val. A HTML kódot containerbe raktam, hogy a bootstrap segítségével reszponzív legyen a felület. Az input-okba a felhasználó név, e-mail, jelszó, jelszó még egyszer és a regisztrációs gomb került. Saját CSS kódot írtam, hogy egyedi megjelenést kapjon az oldal:

```
.container-login{
  position: absolute;
  top: 55%;
  left: 75%;
  transform: translate(-50%, -50%);
}
.login-form{
  height: 420px;
  width: 330px;
  border-radius: 10px;
  display: flex;
  flex-direction: column;
  align-items: center;
  background: rgba(0, 0, 0, 0.2);
  box-shadow: 0 0 10px rgba(0, 0, 0, 1);
  color: #fff;
  padding-top: 50px;
}
.login h3{
  letter-spacing: 1px;
}
```

A reszponzivitást @media queries segítségével oldottam meg, mégpedig úgy, hogy megadtam a oldal töréspontjait és az azokra adott reakciókat:

```
@media screen and (max-width: 900px)
{
  .container-login{
    left:50%;
    z-index: -10;
  }
}

@media screen and (max-height: 500px)
{
  .container-login{
    position: inherit;
    transform: none;
  }
  .login-form{
    margin: 0 auto;
  }
}
```

Ugyan ezen elvek alapján került kódolásra a bejelentkezési form is.

A NAVBAR elkészítéséhez HTML, JavaScript, Bootstrap és CSS-t használtam.

```
<header>
  <a href="esemenyek.php" class="logo">S-port</a>
  <div id="toggle"></div>
  <div id=navbar>
    <ul>
      <li><a href="../vizsgamunka/uj_esemeny.php">Új esemény léterhozása</a></li>
      <li><a href="../vizsgamunka/HTML/about_us.html">Rólunk</a></li>
      <li><a href="index_login.php">Kijelentkezés</a></li>
    </ul>
  </div>
</header>

<!--script kód a NAVbar-hoz-->

<script src="js/navbar.js"></script>
```

A JavaScript kódot külön könyvtárba helyeztem, hiszen a navbar minden oldal alkotóeleme, így kezelhetősége és a kódok átláthatósága is nagymértékben javult.

```
const header = document.getElementById('header');
const toggle = document.getElementById('toggle');
const navbar = document.getElementById('navbar');

document.onclick = function(e){
  if(e.target.id !== 'header' && e.target.id !== 'toggle' && e.target.id !== 'navbar'){
    toggle.classList.remove('active');
    navbar.classList.remove('active');
  }
}

toggle.onclick = function(){
  toggle.classList.toggle('active');
  navbar.classList.toggle('active');
}
```

Az új események oldalon select-ek segítségével kötöttük össze a PHP kódot a HTML-el. A reszponzivitást bootstrapel oldottam meg. Az egész benne van egy container majd egy row és col-ban. A `<div class="col">` -ban lehet megadni, hogy milyen töréspontoknál hogy viselkedjen, jelen esetben a `"col-lg-4"`-et használtam ami azt jelenti, hogy 991 pixelig a mező négy oszlopot foglal el a tizenkettőből.

```
<div class="form-group">
  <label></label>
  <div class="col-lg-4">
    <select name="esemeny_hely" class="radius">
      <option>Esemény helye</option>
      <?php
        $result2=mysqli_query($connect,"SELECT * from telepulesek");
        while($row=mysqli_fetch_array($result2))
        {
          $telepules=$row['telepules_nev'];
          echo "<option> $telepules </option>";
        }
      ?>
    </select>
  </div>
</div>
```

Ehhez tartozó CSS:

```
.btn1{
  box-shadow: 0 0 4px #fff;
}
input[name]:focus {
  color: #fff;
}
input:focus,
.btn1:hover {
  box-shadow: 0 0 10px rgb(0, 0, 0);
}
.radius{
  background: rgba(121, 198, 239, 0.7);
  border-radius: 15px;
  height: 40px;
  width: 100%;
}
```

### 3.5 Tesztelési dokumentáció

Készítette: Naszvetter Béla, Vámos Attila, Király Kristóf

- **Új felhasználó létrehozása amely még biztosan nem létezik**

Várt eredmény: Sikeres regisztráció

Teszt eredménye: Sikeres

- **Új felhasználó regisztrálásakor, ha nem egyezik a két jelszó**

Várt eredmény: Nem történik meg a regisztráció, hibaüzenet- A jelszavak nem egyeznek meg

Teszt eredménye: Sikeres

- **Új felhasználó regisztrálásakor, ha az email cím nem valós**

Várt eredmény: Nem engedi regisztrálni, figyelmezteti a felhasználót, hogy adjon meg valós email címet

Teszt eredménye: Nem történt meg a regisztráció. Hibaüzenet: hiányzik a @ jel  
tesztadat: akarki\_akarki.hu

A mező nincs validálva, csak bootstrap által lekezelve

- **Weboldalak reszponzivitásának tesztelése**

Várt eredmény: A weboldalak reszponzívan jelennek meg a különböző méretű képernyőkön. Ehhez a Google Chrome További eszközök → Fejlesztői eszközök → Toggle device toolbar volt a segítségemre. Itt előre meghatározott méretek alapján lehet tesztelni a reszponzivitást.

Teszt eredménye:

- 4K - 2560 px reszponzív
- LapTop L - 1440 px reszponzív
- LapTop - 1024 px reszponzív
- Tablet - 768 px reszponzív
- Mobil L - 425 px reszponzív
- Mobil M - 375 px reszponzív
- Mobil S - 320 px reszponzív

További reszponzivitás tesztet végeztem el a Google hivatalos teszt oldalán, amit ezen a linken lehet elérni: <https://search.google.com/test/mobile-friendly>

A tesztek várt eredménye, hogy a weboldalak mobil eszközön is könnyedén használhatóak.

A tesztek eredménye: sikeres teszt, az oldalak reszponzív módon jelennek meg a felhasználók számára mobil eszközön is.

- **Weboldal linkek tesztelése**

Várt eredmény: A weboldalakon és a navbar-ban elhelyezett gombok linkelése a helyes oldalakra továbbítja a felhasználót.

Ezt úgy teszteltem, hogy minden linkelést végig próbáltam.

Teszt eredménye: A linkelések a megadott weboldalakra mutatnak.

- **Megadott maximális karakterszámok tesztelése**

Várt eredmény: Az új sportesemény létrehozása oldalon található “Részletek” megadása mezőbe maximálisan 200 darab karaktert lehet írni.

Ezt úgy teszteltem, hogy Lorem Ipsum karakter generátorral létrehoztam 210 karaktert és beillesztettem a mezőbe. Összehasonlítottam a létrehozott és a mezőben lévő utolsó tíz karaktert.

Teszt eredménye: A teszt sikeres volt, a mezőbe maximálisan 200 karaktert lehet beírni.

- **Asztali alkalmazás bejelentkezés**

Várt eredmény: helyes felhasználónév és jelszó esetén sikeres belépés, hiba esetén hibaüzenet.

Teszt eredmény: Sikeres

- **Asztali alkalmazás keresés**

Várt eredmény: keresésnél a lekért táblába kiadja a találatot, ha nincs találat hibaüzenetet küld

Teszt eredmény: Sikeres

- **Asztali alkalmazás adatbázis kapcsolat**

Várt eredmény: kapcsolódik az adatbázishoz

Teszt eredmény: nem kapcsolódott az adatbázishoz a program összeomlott. Javítás: hibaüzenet beiktatása ha nincs adatbázis kapcsolat, a program nem omlik össze



## 4 Összefoglalás

### 4.1 Önértékelés

A kitűzött célunkat sikeresen elértük. Az alkalmazással lehetőség van felhasználóknak regisztrálni, azon sport eseményeket megosztani egymással. A rögzített sport események között lehetőség van a keresésre település, sport, és időpont szerint.

Ahhoz, hogy ezt az alkalmazást el tudjuk készíteni csapatmunkára volt szükségünk. Meg kellett tanulnunk használni a GitHub-ot, hogy meg tudjuk osztani egymással a külön-külön elvégzett munkánkat, majd azokat közösen szinkronizálni kellett, hogy megfelelően működjön.

A fejlesztés során megtanultuk használni a Visual Studio programot a HTML, Javascript, PHP programok megírásához, teszteléséhez. Létre tudtuk hozni az adatbázist különböző táblázatokkal, és azok közötti kapcsolatokkal, és megoldottuk, hogy az adatbázis össze tudjon dolgozni a weblappal, és az asztali alkalmazással amivel az alkalmazás adminisztrátori feladatait tudjuk megvalósítani.

### 4.2 Továbbifejlesztési lehetőségek

- a felhasználók közötti kommunikációra chat megoldás kifejlesztése
- rögzített eseményekre felhasználói visszajelzés: érdekel/jelentkezem
- GPS adatok alapján kereshetnek sportbarátokat azon felhasználók akik ehhez hozzájárultak
- térképes megjelenítés a rögzített sporteseményekhez
- régebbi sport események automatikus törlése, archiválása
- felhasználói regisztráláskor annak a megadása, és adatbázisban tárolása, hogy mely sportágak érdeklik

## 5 Felhasznált irodalom

PHP Manual: <https://www.php.net/manual/en/index.php> , utolsó használat: 2022.04.26

Two linked dropdown list box to select category and subcategory using PHP MySQL & JQuery YouTube: [https://www.youtube.com/watch?v=XDa5LkxoQbQ&ab\\_channel=plus2net](https://www.youtube.com/watch?v=XDa5LkxoQbQ&ab_channel=plus2net) , utolsó használat: 2022.04.26

A Bootstrap használata (v5.1.3) - YouTube: [https://www.youtube.com/watch?v=jbA2dJV679M&t=1378s&ab\\_channel=M%C3%A9rtTam%C3%A1s](https://www.youtube.com/watch?v=jbA2dJV679M&t=1378s&ab_channel=M%C3%A9rtTam%C3%A1s) , utolsó használat: 2022.04.26

Reszponzív táblázatok - YouTube: [https://www.youtube.com/watch?v=V6g8USvRYX0&t=965s&ab\\_channel=M%C3%A9rtTam%C3%A1s](https://www.youtube.com/watch?v=V6g8USvRYX0&t=965s&ab_channel=M%C3%A9rtTam%C3%A1s) , utolsó használat: 2022.04.26

W3Schools Online Web Tutorials: <https://www.w3schools.com/> , utolsó használat: 2022.04.26

Bootstrap · The most popular HTML, CSS, and JS library in the world.: <https://getbootstrap.com/> , utolsó használat: 2022.04.26

Advance Transparent Login Form With Background Video | HTML And CSS - YouTube: [https://www.youtube.com/watch?v=dBJX1T2BQnE&ab\\_channel=CSSCODER](https://www.youtube.com/watch?v=dBJX1T2BQnE&ab_channel=CSSCODER) , utolsó használat: 2022.04.26

The best Favicon Generator (completely free) - favicon.io : <https://favicon.io/> , utolsó használat: 2022.04.26

Stack Overflow - Where Developers Learn, Share, & Build Careers: <https://stackoverflow.com/> utolsó használat 2022.04.26