

# Digitális Fogásnapló

fejlesztési dokumentáció

# TARTALOMJEGYZÉK

Általános ismertető .....	3
Cél .....	3
Fejlesztői környezet .....	3
BACKEND.....	4
Authentikáció.....	4
Jogosultságok meghatározás .....	4
Identity.....	5
Adatszerkezetek.....	7
Adatmodellek .....	7
Osztályok.....	8
FRONTEND.....	9
Bootstrap, HTML5, CSS .....	9
Tesztelés.....	10
Fejlesztési lehetőségek.....	11

# **Általános ismertető**

## **Cél:**

Magyarországon hetente rendeznek kisebb és nagyobb egyéni vagy csapat horgászversenyeket, amelyeken a fogások rögzítése manuálisan történik. A gyorsabb, pontosabb és „real time” eredmény elérést szeretnénk a versenyek szervezőinek biztosítani a tervezett web applikációval.

## **Fejlesztői környezet:**

A fejlesztéshez a Microsoft Visual Studio 2019 alkalmazást használtuk. A web applikáció alapja ASP .NET Core Web App (Model-View-Controller), user interface dizájnjának kialakításához pedig bootstrappet használtunk. A kódolás folyamatát a git segítségével végeztük. A projekt felépítését a klasszikus frontend és backend vonalokon osztottuk fel és külön külön hajtottok végre a szekciók programozását, majd ezután teszteltük egymás munkáját. A feladatmegosztást és a folyamatok nyomon követéséhez a JIRA task managerét használtuk.

# **BACKEND**

## **Authentikáció**

### **Jogosultságok meghatározása:**

A web applikáció linkje nem publikus, csak az adott verseny szervezője által megadott személyek részére kerülne megküldésre az elérési út, amivel kizárnánk a „fake” regisztrációkat.

A versenyeken résztvevő személyek titulusa és feladat rendkívül sokrétű lehet (horgász, segítő, szervező, ellenőr, csapat koordinátor). A jogosultságokat igyekeztünk úgy meghatározni, hogy a különböző típusú versenyek miatt ne keljen folyamatosan a programon változtatni, hanem az integrált jogosultságokkal biztosítani tudjuk a felmerülő igényeket. Ennek érdekében a következő jogosultsági szinteket alakítottuk ki:

- Látogató (segítő, ellenőr, szervező, csapat koordinátor)  
Bejelentkezés nélkül tudja megtekinteni a rögzített fogásokat és az eredményeket.
- User (horgász, segítő, csapat koordinátor)  
Bejelentkezés után a felhasználó tudja rögzíteni a fogásait és lehetősége van a látogatói jogosultságok alkalmazására.
- ADMIN (ellenőr, szervező)  
Az ADMIN jogosultsággal lehetőség van a rögzített fogások szerkesztésére, törlésére.
- SUPERADMIN (program üzemeltetője)  
Teljes hozzáférés és jog a web applikációhoz.

## Identity:

### */Areas/Identity*

A programhoz a beépített Identity funkciót vettük alapul és azt szerkesztettük, egészítettük ki. A látogatók nem rendelkeznek autentikációval így az ő jogaik a web applikáció linkjének birtoklásával generálódnak.

A Userek jogosultságait a */Views/Fogasok/Index.cshtml* fájlban az alábbi módon szabályoztuk:

```
@{
    if (User.Identity.IsAuthenticated)
    {
        <div class="list-group list-group-horizontal">
            <p>
                <a class="nav-link-1" asp-action="Create">Új fogás rögzítése</a>
            </p>
        </div>
    }
    else
    {
        <p>Új fogás rögzítéséhez be kell jelentkeznie!<br/>
        <a class="nav-link-1" asp-area="Identity" asp-
page="/Account/Login">Bejelentkezés</a>
        </p>
    }
}
```

Az ADMIN és SUPERADMIN fiókokat hardkódolással hoztuk létre a *Startup.cs* fájlban:

```
private async Task JogosultsagokBeallitasa(IServiceProvider serviceProvider)
{
    var RoleManager = serviceProvider.GetRequiredService<RoleManager<IdentityRole>>();
    var vanUser = await RoleManager.RoleExistsAsync("User");
    if (!vanUser)
    {
        IdentityRole szerepkor = new IdentityRole("User");
        await RoleManager.CreateAsync(szerepkor);
    }

    var vanAdmin = await RoleManager.RoleExistsAsync("Admin");
    if (!vanAdmin)
    {
        IdentityRole szerepkor = new IdentityRole("Admin");
        await RoleManager.CreateAsync(szerepkor);
        UserManager<IdentityUser> userManager =
            serviceProvider.GetRequiredService<UserManager<IdentityUser>>();
        IdentityUser Sadmin =
            await userManager.FindByIdAsync("admin@admin.com");
        if (Sadmin == null)
        {
            var felhasznalo = new IdentityUser
            {
                UserName = "admin@admin.com", Email = "admin@admin.com",
                EmailConfirmed = true, LockoutEnabled = false };
            var letre = await userManager.CreateAsync(felhasznalo, "Almafa12;");
            if (letre.Succeeded)
            {
                await userManager.AddToRoleAsync(felhasznalo, "Admin");
            }
        }
    }
}
```

```

    }
}

var vanSuperAdmin = await RoleManager.RoleExistsAsync("SuperAdmin");
if (!vanSuperAdmin)
{
    IdentityRole szerepko = new IdentityRole("SuperAdmin");
    await RoleManager.CreateAsync(szerepko);

    UserManager<IdentityUser> userManager =
    serviceProvider.GetRequiredService<UserManager<IdentityUser>>();
    IdentityUser Sadmin = await
    userManager.FindByIdAsync("superadmin@admin.com");
    if (Sadmin == null)
    {
        var felhasznalo = new IdentityUser { UserName = "superadmin@admin.com", Email
        = "superadmin@admin.com", EmailConfirmed = true, LockoutEnabled = false };
        var letre = await userManager.CreateAsync(felhasznalo, "Almafa13");
        if (letre.Succeeded)
        {
            await userManager.AddToRoleAsync(felhasznalo, "SuperAdmin");
        }
    }
}
}
}

```

Az általuk elérhető Szerkesztés, Részletek, Törlés funkciók elérhetőségét szintén a */Views/Fogasok/Index.cshtml* fájlban szabályoztuk a következő módon:

```

@{
    if (User.IsInRole("Admin"))
    {
        <td>
            <a class="nav-link-1" asp-action="Edit" asp-route-
            id="@item.Id">Szerkesztés</a> |
            <a class="nav-link-1" asp-action="Details" asp-route-
            id="@item.Id">Részletek</a> |
            <a class="nav-link-1" asp-action="Delete" asp-route-
            id="@item.Id">Törlés</a>
        </td>
    }
    else if (User.IsInRole("SuperAdmin"))
    {
        <td>
            <a class="nav-link-1" asp-action="Edit" asp-route-
            id="@item.Id">Szerkesztés</a> |
            <a class="nav-link-1" asp-action="Details" asp-route-
            id="@item.Id">Részletek</a> |
            <a class="nav-link-1" asp-action="Delete" asp-route-
            id="@item.Id">Törlés</a>
        </td>
    }
    else
    {
    }
}
}

```

# Adatszerkezetek

## Adatmodellek:

Az applikáció két adatmodelt használ, egyet a felhasználók rögzítésére (*dbo.ASPNetUsers*) és egyet a fogások (*dbo.Fogas*) rögzítésére.

## Felhasználók rögzítése:

Az alap adattábla az Identity funkció hozzáadásával jött létre, amelyet a *Users.cs* class hozzáadásával egészítettünk ki. A végeleges adattábla a következő adatokat tárolja:

Id	nvarchar(450)
UserName	nvarchar(256)
NormalizedUserName	nvarchar(256)
Email	nvarchar(256)
NormalizedEmail	nvarchar(256)
EmailConfirmed	bit
PasswordHash	nvarchar(MAX)
SecurityStamp	nvarchar(MAX)
ConcurrencyStamp	nvarchar(MAX)
PhoneNumber	nvarchar(MAX)
PhoneNumberConfirmed	bit
TwoFactorEnabled	bit
LockoutEnd	datetimeoffset(7)
LockoutEnabled	bit
AccessFailedCount	int
Csapatnev	nvarchar(100)
Discriminator	nvarchar(MAX)
HorgaszIgazolvany	nvarchar(100)
KeresztNev	nvarchar(100)
VezetekNev	nvarchar(100)

## Fogások rögzítése:

Az adattábla a következő adatokat tárolja:

Id	int
Nev	nvarchar(60)
CsapatNev	nvarchar(60)
Ido	datetime2(7)
HalFajtak	nvarchar(MAX)
Suly	decimal(18,2)
Allasok	int

## Osztályok:

- *Users.cs*
- *Fogas.cs*
- *FogasokKereses.cs*
- *HomeController.cs*
- *FogasokController.cs*
- *ApplicationDbContext.cs*

### Users.cs:

A szükséges felhasználói adatokat írja az Identity által létrehozott adattáblába és onnan olvassa ki.

### Fogas.cs:

A fogások rögzítése során beírt adatokat írja és olvassa ki a *dbo.Fogas* adattábla használatával.

### FogasokKeresese.cs:

A *dbo.Fogas* adattáblából olvassa ki a szükséges adatok az eredmények megjelenítéséhez.

### HomeController.cs:

A */Views/Home*-ban elérhető frontend felületeket vezérli.

### FogasokController.cs:

A */Views/Fogasok*-ban elérhető frontend felületeket vezérli. A *Fogas.cs* és *FogasokKeresese.cs* classokat vezérli és a *dbo.Fogas* adattáblával áll kapcsolatban.

### ApplicationDbContext.cs:

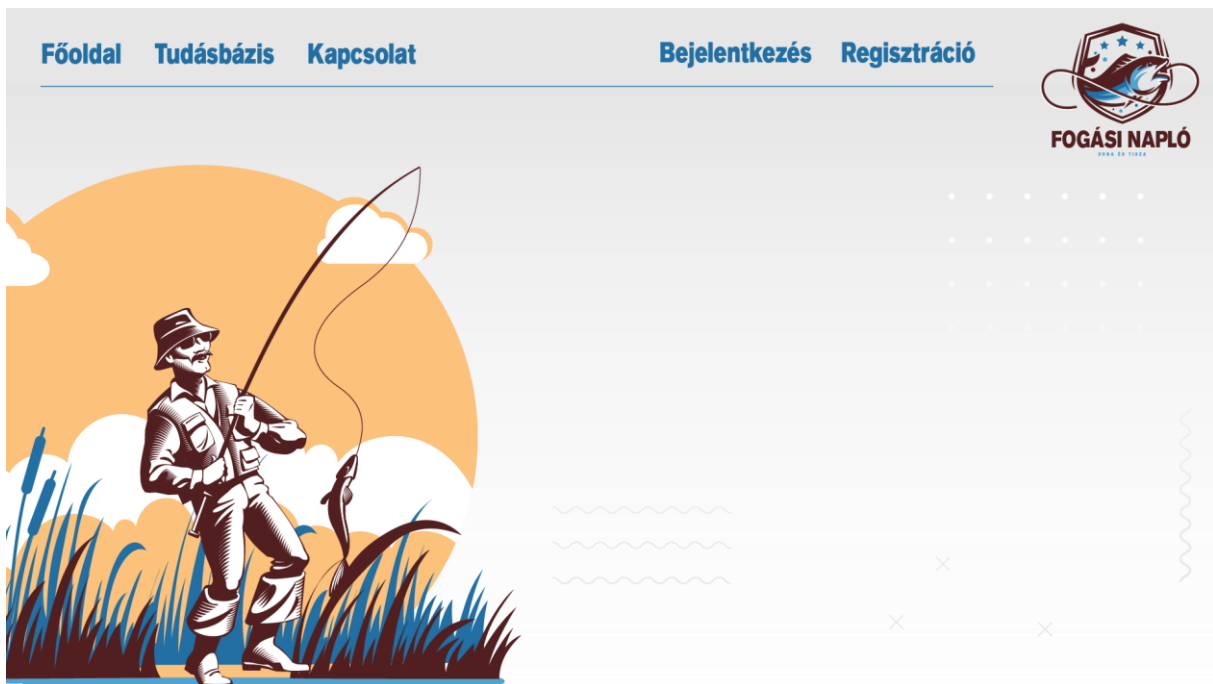
A *Users.cs* és a *Fogasok.cs* classok összeköttetéséért felelős.



# FRONTEND

## Bootstarp, HTML5, CSS

A Bootstrap 4.3.1-as verziója van integrálva a Visual Studio 2019 ASP .NET Core Web App (Model-View-Controller) alapjába. A UI dizájnt erre felépítve a következő módon vizualizáltuk előzetesen.



Az oldal kinézetének kialakításához a *\_Layout.cshtml* és *\_Authlayout.cshtml* fájlokat használtuk és az aloldalakat ezekhez a layout fájlokhoz kapcsoljuk.

A szerkezeti felépítésben a menü a *hederben*, az oldalak a *body* részben a *@RenderBody()* hivatkozással és a segéd dokumentumok a *footerben* vannak elhelyezve.

Kihasználva a bootstrap és a HTML nyújtotta lehetőségeket egy letisztult, minden generáció számára könnyen használható oldalt sikerült alkotni.

A brand irányvonalainak meghatározásához és azok visszaadásához a *css* fájlt használtuk.

# Tesztelés

Az oldal teszteléséhez manuális tesztelést alkalmaztunk elsősorban, melyhez több külsős személyt is bevontunk.

A folyamat során az alábbi hibákat tártuk fel:

- A fogások rögzítésénél a decimális értékek rögzítése nem lehetséges. A hiba esetleges kiküszöböléséhez egy validátor használata lehet megoldás:

```
$.validator.methods.number =  
function (value, element) { return this.optional(element)  
|| /-?(?:\d+|\d{1,3}(?:[\s\.,]\d{3})+)(?:[\s\.,]\d+)?$/ .test(value); }
```

- Az eredmények menüpont alatt, az eredmények rangsorolása előtt nem jelenik meg a sorszám mező.

A hiba javításához egy új oszlop beszúrása szükséges, ahol a sorok sorszámmal történő ellátása valósul meg.

- A bejelentkezés és a regisztráció ablakok középre rendeződése nem valósul meg sikeresen.

# Fejlesztés

A Digitális Fogásnapló fejlesztésének lehetséges lépéseinek megfelelően a következőket határoztuk meg:

- A feltárt hibák és hiányosságok orvoslása.
- A SUPERADMIN jogosultsághoz szükséges elérhetővé tenni egy felületet ahol a felhasználók moderálása lehetséges.
- A lehető legtöbb platformon történő eléréshez szükséges mobil és asztali applikációk fejlesztése.