

**Mobil programozási alapok**  
**Fegyvernyilvántartó mobilapp**

Név: Urbán Olivér

Szak: Mérnökinformatikus

Neptun kód: HEPMIU

## **1.Feladat leírása**

Egy fegyver nyilvántartó alkalmazást készítettem el egy crossplatformos mobil app keretein belül.

Az appban a user listázhatja az elemeket, valamint módosíthat és törölhet, továbbá létre is hozhat új fegyvereket.

Az alkalmazást C# MAUI keretrendszerben készítettem el, ennek egyik oka az hogy a szakdolgozatom is ebben a keretrendszerben készül.

A következő oldalakon bemutatom a különböző komponenseket, és a mobil app működését.

A mobil app két nagy részre bontható egy backend és egy frontend projektre.

## 2.Backend

Az API-ban megtalálható egy modell mappa amelyben maga az entity leképzését jelentő osztályt, valamint egy DTO-t tartalmaz amely a módosítás művelethez szükséges a későbbiekben.

```
using System.ComponentModel.DataAnnotations.Schema;

namespace WeaponMobileAPI.Models
{
    [Table("Weapons", Schema = "dwh")]
    5 references
    public class Weapon
    {
        1 reference
        public int Id { get; set; }
        1 reference
        public string? Name { get; set; }
        1 reference
        public string? Manufacturer { get; set; }
        1 reference
        public int Price { get; set; }
        1 reference
        public int Quantity { get; set; }
    }
}
```

```
1 reference
public class UpdateWeaponDTO
{
    2 references
    public string? Name { get; set; }
    2 references
    public string? Manufacturer { get; set; }
    2 references
    public int Price { get; set; }
    2 references
    public int Quantity { get; set; }
}
```

A backend tartalmaz továbbá egy kontrollert is amely a Http metódusokat kezeli:

```
namespace ...MobiWeaponsAPI.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    1 reference
    public class WeaponsController : ControllerBase
    {
        private readonly AppDbContext _context;

        0 references
        public WeaponsController(AppDbContext context)
        {
            _context = context;
        }

        [HttpGet]
        0 references
        public async Task<ActionResult<List<Weapon>>> GetWeapons()
        {
            var weapons = await _context.Weapons.ToListAsync();

            return weapons;
        }

        [HttpGet("{Id}")]
        1 reference
        public async Task<ActionResult<Weapon>> GetWeapon(int Id)
        {
            var weapon = await _context.Weapons.FindAsync(Id);
            if (weapon == null) return NotFound();
            return weapon;
        }
    }
}
```

```

[HttpPost]
0 references
public async Task<ActionResult<Weapon>> CreateWeapon([FromBody] Weapon weapon)
{
    _context.Weapons.Add(weapon);
    await _context.SaveChangesAsync();
    return CreatedAtAction(nameof(GetWeapon), new { Id = weapon.Id }, weapon);
}

[HttpDelete("{Id}")]
0 references
public async Task<ActionResult> DeleteWeapon(int Id)
{
    var weapon = await _context.Weapons.FindAsync(Id);
    if (weapon == null) return NotFound();

    _context.Weapons.Remove(weapon);
    await _context.SaveChangesAsync();
    return NoContent();
}

[HttpPatch("{Id}")]
0 references
public async Task<IActionResult> UpdateWeapon(int Id, UpdateWeaponDTO updateWeaponDto)
{
    var weapon = await _context.Weapons.FindAsync(Id);
    if (weapon == null)
    {
        return NotFound();
    }
}

```

A Db-hez való csatlakozást a következőképpen valósítottam meg:

```

"ConnectionStrings": {
  "DefaultConnection": "Host=localhost;Port=5433;Database=mobile_app_weapon;Username=postgres;Password=admin;Search Path=dwh"
}

```

DbContext:

```

0 references
public AppDbContext(DbContextOptions<AppDbContext> options) : base(options) { }

6 references
public DbSet<Weapon> Weapons { get; set; }

0 references
protected override void OnModelCreating(ModelBuilder modelBuilder)
{
    base.OnModelCreating(modelBuilder);

    modelBuilder.HasDefaultSchema("dwh");
}

```

# Az endpointokat Swaggerrel teszteltem le:

WeaponMobileAPI1.0OAS3

https://localhost:44331/swagger/v1/swagger.json

Weapons

GET/api/Weapons

POST/api/Weapons

GET/api/Weapons/{Id}

DELETE/api/Weapons/{Id}

PATCH/api/Weapons/{Id}

Schemas

UpdateWeaponDTO

Weapon

```
curl -X 'GET' \
'https://localhost:44331/api/Weapons' \
-H 'accept: text/plain'
```

Request URL

https://localhost:44331/api/Weapons

Server response

Code

Details

200

Response body

```
[
  {
    "id": 6,
    "name": "FN SCAR",
    "manufacturer": "FN Herstal",
    "price": 2000000,
    "quantity": 25
  },
  {
    "id": 7,
    "name": "Springfield Armory XD-S",
    "manufacturer": "Springfield Armory",
    "price": 600000,
    "quantity": 40
  },
  {
    "id": 2,
    "name": "M16A",
    "manufacturer": "Colt",
    "price": 1200000,
    "quantity": 75
  },
  {
    "id": 1,
    "name": "AK-74",
    "manufacturer": "Kalashnikov",
    "price": 1500000,
    "quantity": 50
  }
]
```

Response headers

```
content-type: application/json; charset=utf-8
date: Tue, 14 Jan 2025 09:54:16 GMT
server: Microsoft-IIS/10.0
x-powered-by: ASP.NET
```

### 3.Frontend

A WeaponService kezeli a hívásokat a kliens által:

```
8 references
public async Task<List<Weapon>> GetWeaponsAsync()
{
    var weapons = new List<Weapon>();
    Uri uri = new Uri(androidURI);
    //Uri uri = new Uri(windowsURI);
    try
    {
        HttpResponseMessage response = await _httpClient.GetAsync(uri);
        if (response.IsSuccessStatusCode)
        {
            string content = await response.Content.ReadAsStringAsync();
            weapons = JsonSerializer.Deserialize<List<Weapon>>(content, _serializerOptions);
        }
    }
    catch (Exception ex)
    {
        Debug.WriteLine(@"\tERROR {0}", ex.Message);
    }

    return weapons;
}
```

Többféle URI-t kell megadni a különböző platformokon(Android, Windows), ugyanis ezek emulátorai különböző címkészlettel dolgoznak.

Ez a metódus a fegyverek listázásáért felel, először kérést intézünk a szerver felé majd ha sikeres akkor a response után a JSON formátumú adatokat stringgé alakítjuk.

A következő metódusok a mentés, módosítás és törlés műveletekhez szükségesek:

8 references  
public async Task SaveWeaponAsync(UpdateWeaponDTO weapon, List<Weapon> existingWeapons, int? Id = null)

```
{
    if (weapon == null)
    {
        throw new ArgumentNullException(nameof(weapon), "Weapon data cannot be null.");
    }

    Uri uri;
    string json = JsonSerializer.Serialize(weapon, _serializerOptions);
    HttpContent content = new StringContent(json, Encoding.UTF8, "application/json");

    try
    {
        bool isExistingWeapon = Id.HasValue && existingWeapons.Any(w => w.Id == Id.Value);

        if (isExistingWeapon)
        {
            uri = new Uri($"{windowsURI}{Id}");
            Debug.WriteLine($"Updating existing weapon. Request URL: {uri}, Payload: {json}");

            HttpResponseMessage response = await _httpClient.PatchAsync(uri, content);
            string responseContent = await response.Content.ReadAsStringAsync();

            Debug.WriteLine($"Response Status Code: {response.StatusCode}");
            Debug.WriteLine($"Response Content: {responseContent}");

            if (!response.IsSuccessStatusCode)
            {
                throw new HttpRequestException($"Failed to update weapon. Status Code: {response.StatusCode}, Response: {responseContent}");
            }

            Debug.WriteLine("Weapon successfully updated via PATCH.");
        }
    }
```

```
    else
    {
        uri = new Uri(windowsURI);
        Debug.WriteLine($"Creating new weapon. Request URL: {uri}, Payload: {json}");

        HttpResponseMessage response = await _httpClient.PostAsync(uri, content);
        string responseContent = await response.Content.ReadAsStringAsync();

        Debug.WriteLine($"Response Status Code: {response.StatusCode}");
        Debug.WriteLine($"Response Content: {responseContent}");

        if (!response.IsSuccessStatusCode)
        {
            throw new HttpRequestException($"Failed to create weapon. Status Code: {response.StatusCode}, Response: {responseContent}");
        }


        Debug.WriteLine("Weapon successfully created via POST.");
    }
}

catch (Exception ex)
{
    Debug.WriteLine($"Error occurred while saving weapon: {ex.Message}");
    throw;
}
```

4 references  
public async Task DeleteWeaponAsync(int Id)

```
{
    Uri uri = new Uri($"{windowsURI}{Id}");

    try
    {
        HttpResponseMessage response = await _httpClient.DeleteAsync(uri);
        if (response.IsSuccessStatusCode)
        {
            Debug.WriteLine(@"\tWeapon successfully deleted.");
        }
        else
        {
            Debug.WriteLine(@"\tFailed to delete weapon. Status Code: {0}", response.StatusCode);
        }
    }
    catch (Exception ex)
    {
        Debug.WriteLine(@"\tERROR: {0}", ex.Message);
    }
}
```

 class System.String  
Represents text as a sequence of UTF-16 code units.



A következő oldalakon a page-eket mutatom be.

Home page:

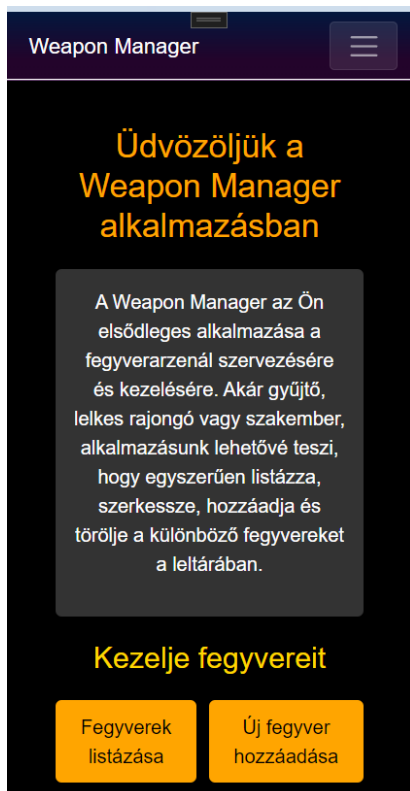
```
@page "/"

<div class="container">
  <header>
    <h1>Üdvözljük a Weapon Manager alkalmazásban</h1>
  </header>

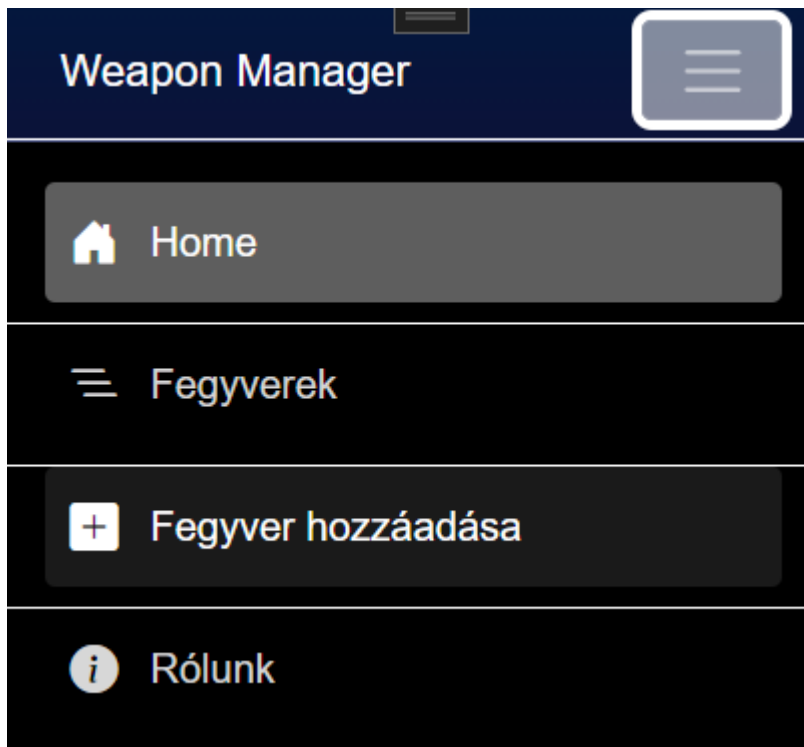
  <section class="intro">
    <p>
      A Weapon Manager az Ön elsődleges alkalmazása a fegyverarzenál szervezésére és kezelésére. Akár gyűjtő, lelkes rajongó vagy szakember, alkalmazásunk lehetővé teszi, hogy egyszerűen listázza, szerkessze, hozzáadja és törölje a különböző fegyvereket a leltárában.
    </p>
  </section>

  <main>
    <h2>Kezelje fegyvereit</h2>
    <div class="actions">
      <a href="/weapon-list" class="button">Fegyverek listázása</a>
      <a href="/add-weapon" class="button">Új fegyver hozzáadása</a>
    </div>
  </main>
</div>
```

Előnézete:



Navbar:



Fegyverek page:

```
else if (weapons == null || weapons.Count == 0)
{
    <p>No weapons available.</p>
}
else
{
    <div class="table-container">
        <table class="weapon-table">
            <thead>
                <tr>
                    <th>Id</th>
                    <th>Név</th>
                    <th class="hide-mobile">Gyártó</th>
                    <th>Ár</th>
                    <th>Mennyiség</th>
                    <th>Műveletek</th>
                </tr>
            </thead>
            <tbody>
                @foreach (var weapon in weapons)
                {
                    <tr>
                        <td data-label="Id">@weapon.Id</td>
                        <td data-label="Név">@weapon.Name</td>
                        <td data-label="Gyártó" class="hide-mobile">@weapon.Manufacturer</td>
                        <td data-label="Ár">@weapon.Price.ToString("C0")</td>
                        <td data-label="Mennyiség">@weapon.Quantity</td>
                        <td data-label="Műveletek" class="actions-cell">
                            <button class="edit-btn" @onclick="() => EditWeapon(weapon.Id)">Szerkesztés</button>
                            <button class="delete-btn" @onclick="() => DeleteWeapon(weapon.Id)">Törlés</button>
                        </td>
                    </tr>
                }
            </tbody>
        </table>
    </div>
}
```

Weapon Manager

Fegyverek Listája

ID	6
NÉV	FN SCAR
GYÁRTÓ	FN Herstal
ÁR	2 000 000 Ft
MENNYISÉG	25
MŰVELETEK	
Szerkesztés	
Törés	

ID	7
NÉV	Springfield Armory XD-S
GYÁRTÓ	Springfield Armory
ÁR	600 000 Ft
MENNYISÉG	40

Fegyver módosítása a Szerkesztés gombra rákattintva:

```

<EditForm Model="@currentWeapon" OnValidSubmit="@HandleValidSubmit">
  <DataAnnotationsValidator />
  <ValidationSummary />

  <div class="form-group">
    <label for="name">Név:</label>
    <InputText id="name" @bind-Value="currentWeapon.Name" class="form-control" />
    <ValidationMessage For="@((C) => currentWeapon.Name)" />
  </div>

  <div class="form-group">
    <label for="manufacturer">Gyártó:</label>
    <InputText id="manufacturer" @bind-Value="currentWeapon.Manufacturer" class="form-control" />
    <ValidationMessage For="@((C) => currentWeapon.Manufacturer)" />
  </div>

  <div class="form-group">
    <label for="price">Ár:</label>
    <InputNumber id="price" @bind-Value="currentWeapon.Price" class="form-control" />
    <ValidationMessage For="@((C) => currentWeapon.Price)" />
  </div>

  <div class="form-group">
    <label for="quantity">Mennyiség:</label>
    <InputNumber id="quantity" @bind-Value="currentWeapon.Quantity" class="form-control" />
    <ValidationMessage For="@((C) => currentWeapon.Quantity)" />
  </div>

  <button type="submit" class="btn btn-primary">Fegyver Mentése</button>

```

```
private async Task HandleValidSubmit()
{
    try
    {
        weapons = await WeaponService.GetWeaponsAsync();
        await WeaponService.SaveWeaponAsync(currentWeapon, weapons, Id);
        Console.WriteLine($"Weapon with ID {Id} updated successfully.");
        NavigationManager.NavigateTo("/weapon-list");
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Error saving weapon: {ex.Message}");
    }
}
```

Weapon Manager

### Fegyver Módosítása

Név:

M16A

Gyártó:

Colt

Ár:

1200000

Mennyiség:

75

Fegyver Mentése

## Fegyver hozzáadása:

```
@code {
    private UpdateWeaponDTO newWeapon = new UpdateWeaponDTO();
    private List<Weapon> weapons;

    private async Task HandleValidSubmit()
    {
        await WeaponService.SaveWeaponAsync(newWeapon, weapons);
        Console.WriteLine($"Adding new weapon: {newWeapon.Name}");
        NavigationManager.NavigateTo("/weapon-list");
    }
}
```

Weapon Manager

### Új Fegyver Hozzáadása

Név:

Gyártó:

Ár:

Mennyiség:

Fegyver Hozzáadása

ID	15
NÉV	Példa
GYÁRTÓ	PéldaGyarto
ÁR	15 000 Ft
MENNYISÉG	50
MŰVELETEK	
Szerkesztés	
Törlés	

# Rólunk page:

```
@page "/about"

<div class="container">
  <h1>Rólunk</h1>

  <section class="company-info">
    <h2>MarksmenHub</h2>
    <p>
      A MarksmenHub egy vezető vállalat a fegyverek kereskedelmében, karbantartásában és
      használatában. Több mint 20 éves tapasztalattal rendelkezünk a fegyveriparban, és büszkék
      vagyunk arra, hogy megbízható partnerei vagyunk ügyfeleinknek.
    </p>
  </section>

  <section class="services">
    <h2>Szolgáltatásaink</h2>
    <ul>
      <li>
        <h3>Fegyverkereskedelem</h3>
        <p>
          Széles választékban kínálunk új és használt fegyvereket. Szakértő csapatunk segít
          megtalálni az Ön igényeinek legmegfelelőbb fegyvert.
        </p>
      </li>
      <li>
        <h3>Fegyverkarbantartás</h3>
        <p>
          Professzionális karbantartási szolgáltatásokat nyújtunk, hogy fegyverei mindig
          kifogástalan állapotban legyenek.
        </p>
      </li>
    </ul>
  </section>
</div>
```

Vagyunk arra, hogy megbízható partnerei vagyunk ügyfeleinknek.

## Szolgáltatásaink

- ### Fegyverkereskedelem

Széles választékban kínálunk új és használt fegyvereket. Szakértő csapatunk segít megtalálni az Ön igényeinek legmegfelelőbb fegyvert.
- ### Fegyverkarbantartás

Professzionális karbantartási szolgáltatásokat nyújtunk, hogy fegyverei mindig kifogástalan állapotban legyenek.
- ### Lőtér

Modern, biztonságos lőtérünkön lehetőséget biztosítunk a gyakorlásra és a fegyverek tesztelésére.

## CSS részlet:

```
.container {
  font-family: Arial, sans-serif;
  max-width: 100%;
  margin: 0 auto;
  padding: 20px;
  background-color: black;
}

h1 {
  color: #ffd700;
  text-align: center;
  margin-bottom: 20px;
  font-size: 24px;
}

.table-container {
  overflow-x: auto;
}

.weapon-table {
  width: 100%;
  border-collapse: collapse;
  background-color: #1a1a1a;
  box-shadow: 0 0 20px rgba(0, 0, 0, 0.1);
  border: 1px solid #333;
}

.weapon-table th,
.weapon-table td {
  padding: 12px;
  text-align: left;
```

```
.weapon-table th,
.weapon-table td {
  padding: 12px;
  text-align: left;
  font-size: 16px;
  border: 1px solid #333;
  color: #ffa500;
}

.weapon-table th {
  background-color: #333;
  font-weight: bold;
  text-transform: uppercase;
  color: #ffd700;
}

.weapon-table tr {
  border-bottom: 5px solid #ffa500;
}

.weapon-table tr:nth-child(even) {
  background-color: #2a2a2a;
}

.weapon-table tr:hover {
  background-color: #333;
}
```

Adatbázishoz PostgreSQL-t, adatbáziskezeléshez pgAdmint használtam:

SQL

	<div><div>Id</div><div>[PK] integer</div></div>	<div><div>Name</div><div>character varying (255)</div></div>	<div><div>Manufacturer</div><div>character varying (255)</div></div>	<div><div>Price</div><div>integer</div></div>	<div><div>Quantity</div><div>integer</div></div>
1	6	FN SCAR	FN Herstal	2000000	25
2	7	Springfield Armory XD-S	Springfield Armory	600000	40
3	2	M16A	Colt	1200000	75
4	1	AK-74	Kalashnikov	1500000	50
5	4	Remington 870	Remington	180000	30
6	5	Barrett M82	Barrett	0	47
7	15	Példa	PéldaGyarto	15000	50