** **

**ZÁRÓDOLGOZAT**

Készítették:

Fazekas Réka

Dömötör Gyula

Kolyankovszky Gusztáv

Konzulens:

Farkas Zoltán

Miskolc

2023.

Miskolci SZC Kandó Kálmán Informatikai Technikum

Miskolci Szakképzési Centrum

SZOFTVERFEJLESZTŐ- ÉS TESZTELŐ SZAK

ZÁRÓDOLGOZAT

Fadoko

szolgáltató ipari program

Fazekas Réka

Dömötör Gyula

Kolyankovszky Gusztáv

2022-2023

Tartalomjegyzék

[Miért ezt a programot választottuk (Gyula írása) 5](#_Toc133434818)

[Adatbázisról (Gyula írása) 6](#_Toc133434819)

[Adatbázisról (Gyula írása) 7](#_Toc133434820)

[Adatbázisról (Gyula írása) 8](#_Toc133434821)

[Backend bemutatása (Guszti írása) 9](#_Toc133434822)

[Backend bemutatása (Guszti írása) 10](#_Toc133434823)

[Backend bemutatása (Guszti írása) 11](#_Toc133434824)

[Frontend bemutatása (Réka írása) 12](#_Toc133434825)

[Frontend bemutatása (Réka írása) 13](#_Toc133434826)

[Frontend bemutatása (Réka írása) 14](#_Toc133434827)

[Irodalomjegyzék 15](#_Toc133434828)

[A vizsgaremek bemutatója (1 oldal) 17](#_Toc133434829)

[A vizsgaremek bemutatója (2 oldal) 18](#_Toc133434830)

[Mellékletek 19](#_Toc133434831)

[Index.js: 20](#_Toc133434832)

[index.html: 21](#_Toc133434833)

[Irodalomjegyzék 22](#_Toc133434834)

Záródolgozat tartalmi követelmények

A záródolgozatnak tartalmaznia kell:

* A szoftver célját, komponenseinek technikai leírását, működésének műszaki feltételeit és használatának rövid bemutatását tartalmazó dokumentáció is része a csomagnak.
* A dokumentációnak tartalmaznia kell a fenti 2 oldalt: konzulensként nappalin és estin az IKT tantárgyat oktatót kell megjelölni
* A dokumentáció 3.oldalától kezdve tartalomjegyzék, majd a dokumentáció végén irodalomjegyzék szerepel. A dokumentáció minimum 20 oldal, maximum 30 lehet. A forrásfájlok szerepelhetnek a dolgozat mellékletében, de a kötelező oldalakba nem számíthatnak bele! A dolgozat melléklete az irodalomjegyzék előtt kell szerepeltetni, melléklet felirattal, esetlegesen külön oldalszámozással!
* A záródolgozatnak a tartalomjegyzék után kezdődik a számozása, az oldalak alján, tetszőleges stílusban, elsősorban az olvashatóságra kell törekedni!
* Tartalomjegyzéket csak úgy lehet készíteni, ha a tartalomban a szövegeket tagolva és címsoros megkülönböztetés szerint címkézzük fel, maximum 2 mélységben (Címsor 1 és címsor 2), így a tartalomjegyzék is áttekinthetőbb lesz. (A tartalomjegyzéket a dokumentum elkészítésének végén célszerű készíteni! Ha nem így történik, folyamatosan frissítsük azt!) Az oldalszámozást szakasztörések beiktatásával lehet elkészíteni, hogy ne számozza pl. az első és a második oldalakat (borítók), jelen jegyzet úgy lett elkészítve, ahogy elvárt, mintaként szolgál!
* Képek beszúrása lehetséges, de nem foglalhatja el csak egy oldal felét maximum, a képek alá kép vagy ábra számozást kell írni, a kép rövid címét. Ha nem saját gyűjteményből van, akkor a kép számozásához csatolni kell lábjegyzetben a kép webcímét és a letöltés dátumát! A dokumentációban utaljunk a képekre a számukkal!
* A záródolgozat elmaradása automatikusan a követelmények nem teljesítését eredményezi és a vizsgát a következő vizsgaidőszakban a vizsgázó megismételheti. Irodalomjegyzék elmaradása, valamint képek nélküli dolgozat, levon az értékéből!
* Határidő: április 24. leadás Classroom-on keresztül (IKT)! Mindenki adja le a vizsgaremekét github-on keresztül (linkként publikus láthatósággal Classroom-on)!
* Hosszabbítási kérelmek leadása: április 24-ig , Classroom-on keresztül (IKT)! A hosszabbítás április 30-ig érvényes!

# Miért ezt a programot választottuk (Gyula írása)

# Adatbázisról (Gyula írása)

# Adatbázisról (Gyula írása)

# Adatbázisról (Gyula írása)

# Backend bemutatása

Backend fejlesztéshez is számtalan eszköz áll rendelkezésre. Döntés előtt érdemes felmérni a használathoz szükséges környezetet, bár ezen a területen nagy az átjárhatóság.

Mire van szükség a backend fejlesztéshez? Fejlesztőeszköz, fejlesztő program, tesztelő program és mind három a feladatnak megfelelő ismerete.

Fejlesztő eszköz a Microsoft Visual Studio 2022

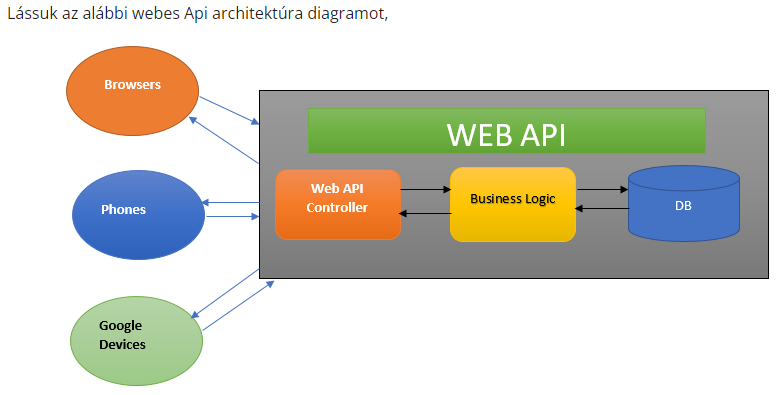
A Microsoft 1997-ben adta ki a több programozási nyelvet tartalmazó fejlesztő eszközt, a neve Microsoft Visual Studio 97. A mai napig 13 kiadás és 17 verzió készült el. Az évek során programozási nyelvekkel bővült. 2002-ben került fel a programozási nyelvek közé a C#. Szintén ekkor mutatták be a .NET keretrendszert. A C# a .NET keretrendszer részeként kifejlesztett objektum központú programozási nyelv, a fejlesztést a Microsoft végezte. A nyelv alapjait a C++ és a JAVA nyelvek adták. Fontos megjegyezni, hogy a .NET jelenleg több mint 40 programozási nyelvet támogat. A .NET keretrendszert platform független alkalmazásfejlesztésre szánták, az elképzelésnek célja volt a gyors fejlesztés lehetősége.

A program fejlesztéséhez használt alkalmazás:

ASP.NET Core Web API

Tartalma: C#, Linux, macOS, Windows, Cloud Service Web, WebAPI

Az elnevezés több rövidítést és szóképet is tartalmaz. A rövid névben igyekeznek mindent felsorolni, amit tartalamzhat a program. A lecsót hasonló módon próbálnám leírni: PapParHagySzalKolOlSo, hm lássuk be ezen a területen nem hétköznapi emberek dolgoznak.



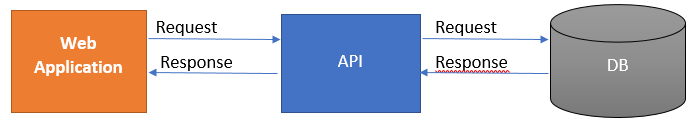
*Forrás: https://www.c-sharpcorner.com/article/asp-net-core-5-0-web-api/ időpont 2023.04.25.*

Kibontva az ASP.NET Core (ASP = Active Server Pages, .NET = Network Enabled Technologies) egy klasszikus technológia az ASP.NET utódja web programozáshoz. Egy nyílt forráskódú, moduláris szerver oldali alkalmazás keretrendszere. Teljesíti a .NET elvárásokat, programozási nyelv független és .NET osztálykönyvtárral rendelkezik, több platformon működik. Alkalmas: Windows, Linux, macOS rendszerekhez.

A program fejlesztéséhez használt alkalmazás (folytatás az előző oldalról):

Web API jelentése, Webes Application Programming Interface.

Feladata a felhasználó oldali webes applikációt összekötni például egy web serverrel az adatokhoz történő hozzáférés miatt. A Web API-t elsősorban adatbázis műveletekhez alkalmazzák. Adatbázis művelet az adat létrehozása CREAT, adat kiolvasása READ, adat szerkesztése EDIT, adat törlése DELETE. A kezdőbetűket összeolvasva CRED műveleteknek hívják őket. A kommunikációhoz http protokolt használ, például create = HTTP POST, read = HTTP GET, edit = HTTP PUT, delete = HTTP DELETE, a kérésekre a válasz JSON formátumban érkezik.



*Forrás: https://www.c-sharpcorner.com/article/asp-net-core-5-0-web-api/ időpont 2023.04.25.*

Az alkalmazást két keretrendszer alkotja a Microsoft AspNetCore.App és a Microsoft.NETCore.App.

Emellett feladattól függően szükség van kiegészítő csomagokra, amit a jelenleg feladathoz használtam, ezeket felsorolom, részletekbe nem megyek bele:

Microsoft.EntityFrameworkCore(5.0.17)

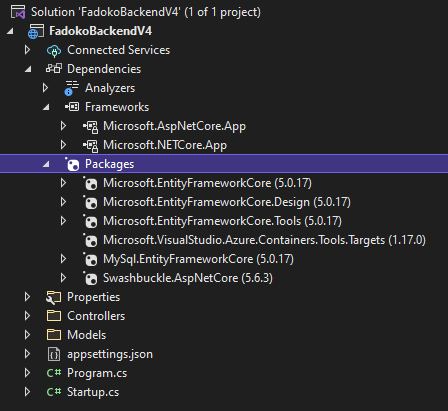
Microsoft.EntityFrameworkCore.Design(5.0.17)

Microsoft.EntityFrameworkCore.Tools(5.0.17)

Microsoft.VisualStudio.Azure.Containers.Tools.Targets(1.17.0)

MySql.EntityFrameworkCore(5.0.17)

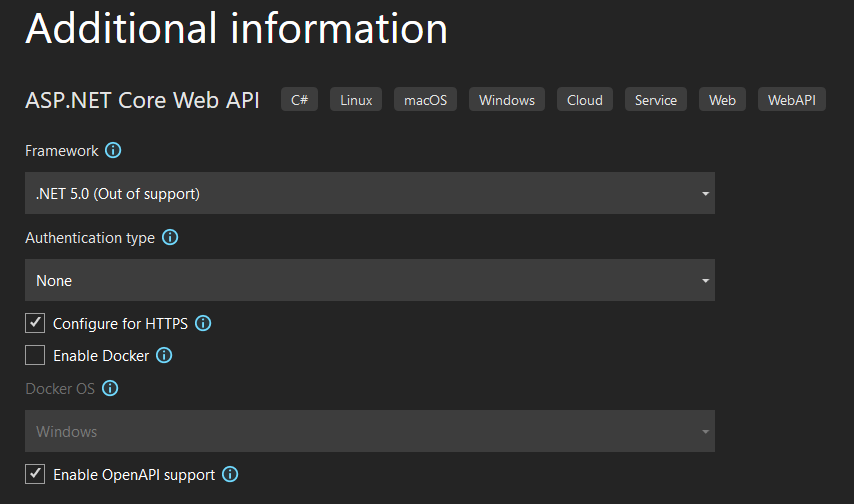
Swashbuckle.AspNetCore(5.6.3)



*Saját képernyőkép*

A keretrendszerek verziói követik a haladás ütemét. Fejlesztés alkalmával érdemes a hosszú távon támogatott verziót választani, bár ez azt is jelenti, hogy az elején új dolgokat kell megismernünk és régi megszokásokat elfelejtenünk. Jelen esetben egy ismert verziót választottam, ami megfelel a feladat megvalósításához.

Framework .NET 5.0 (Out of support)

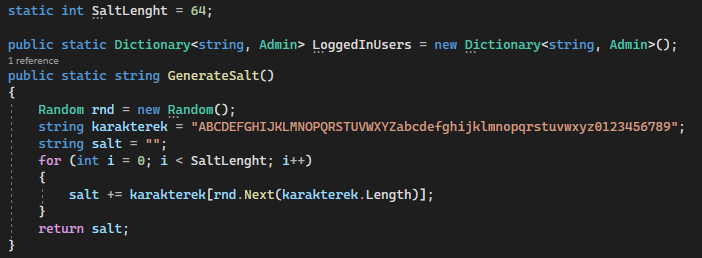


*Saját képernyőkép*

A fejlesztő eszköz alap szerkezetet generál számunkra, amit célunknak megfelelően módosíthatunk, kibővíthetünk. Jelen esetben két plusz könyvtár Models, ide kerülnek az osztályok és a Controllers, itt kerülnek kialakításra a végpontok. Az egyik fájlban történt változtatást kiemelem. A Program.cs fájlba kerül két biztonsági függvény, melyek a jelszavas azonosításhoz szükségesek.

GenerateSalt

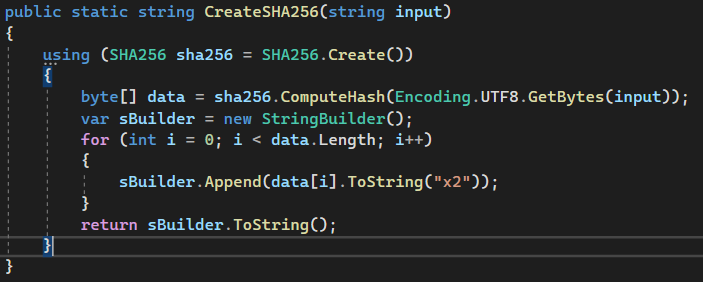
A függvény meghatározott karakter készletből véletlenszerűen generál egy kriptográiailag erős karakter sorozatot. A visszatérő érték 64 hosszú karakterlánc.



*Saját képernyőkép*

CreateSHA256

A függvény SHA-256 hash hexadecimális értéket ad vissza a karakterlánchoz.



*Saját képernyőkép*

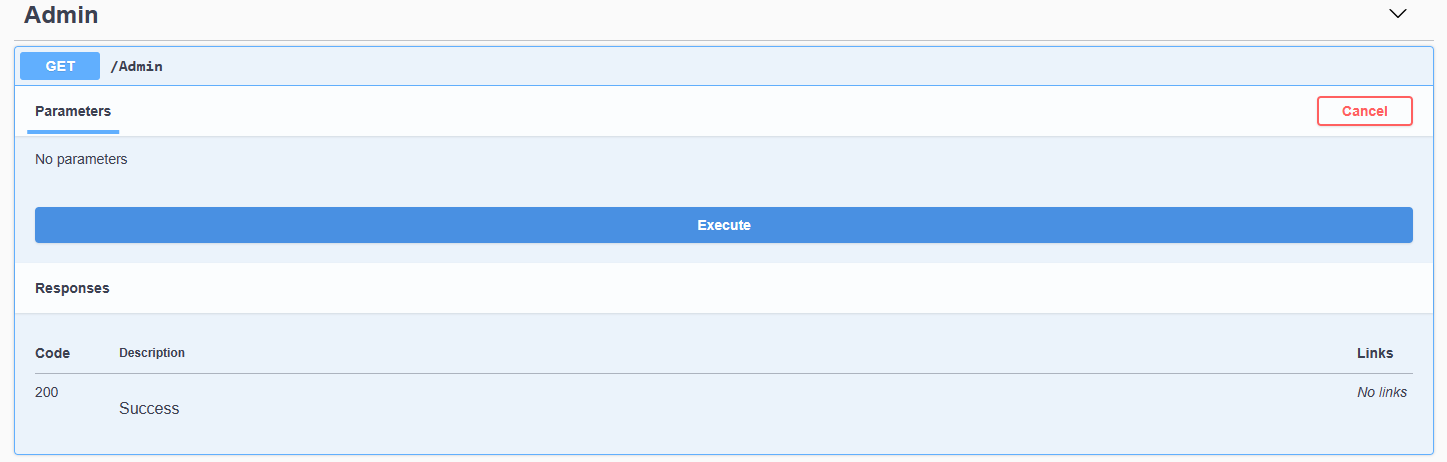
Tesztelő program

A backend fejlesztés folyamata, hogy az elkészített részek működését teszteljük. Mint tudjuk a backend nem a megjelenésért felel, a frontendről érkező kérések fogadása és továbbítása az adatbázis felé, ahonnan megkapja a kért adatokat és megfelelő formában vissza adja a frontend résznek. A frontend résznek nincs szüksége emberi szemmel és értelemmel olvasható adatokra, hanem JSON formátumban vár válaszokat. Ez is olvasható gyakorlott szemmel, nézzünk erre egy példát a kiindulási alapra:



*Saját képernyőkép*

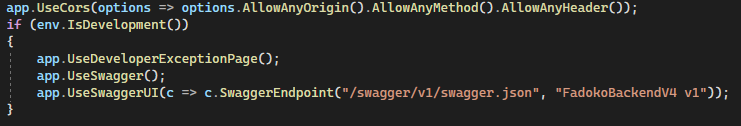
Nézzük meg Swagger programmal, interaktív állapotban, kattintásra készen:



*Saját képernyőkép*

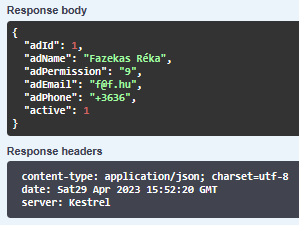
Swegger/OpenAPI program a kimenet megjelenítésére szolgál, ez egy nyílt forráskódú szoftver API fejlesztők részére. RESTful Web API tervezés, készítés, dokumentálás feladatokra.

Teszteléshez meg van szólítva a Startup.cs fájlban.



*Saját képernyőkép*

Amikor elindítjuk a programot a fejlesztő eszközön, akkor a Swagger segítségével tudjuk emulálni a program működését. A három komponensből most elég az adatbázis és a backend, a frontend a megfelelő adatokat várja. Szeretném megtudni, hogy ki az első Admin az adatbázisban, feltételezem, az azonosító száma az 1-es. Nézzük az eredményt:



*Saját képernyőkép*

# Frontend bemutatása (Réka írása)

# Frontend bemutatása (Réka írása)

# Frontend bemutatása (Réka írása)

# Irodalomjegyzék

A ténylegesen felhasznált források száma emeli a dolgozat értékét. Néhány szempontot azonban érdemes figyelembe venni a források kiválasztásánál:

* A hivatkozott irodalomjegyzéket számozzuk, a hivatkozás megjelenése szerint a dokumentáció végén
* Legalább 5, de maximum 10 Internetes, vagy hagyományos, pl. könyv hivatkozást tegyünk a záródolgozatba. A könyv hivatkozást a következőképpen tegyük: Cím, szerző, kiadás éve, kiadó neve, ISBN szám
* A dolgozat szakirodalmi áttekintésének megírása során nem használható fel és nem hivatkozható olyan forrás, amely nem szakemberek által készített, szakmailag nem hiteles, nem megbízható tartalmú (például Wikipédia).
* Internetes források hivatkozási a következő formátumot vegyék fel (a téma megjelölés elhagyható):  
  Téma: a weboldalak böngészéséről, Webcím: www.google.com , letöltés dátuma: 2023.04.30.
* Az irodalomjegyzék és melléklet bemutatása ezen dokumentáció végén található!

Vizsgaremek benyújtása

A vizsgaremek benyújtásának módja: A kész csomagot a vizsga előtt minimum 14 nappal kell a vizsgabizottsághoz benyújtani GitHub vagy más hasonló szolgáltatás segítségével megosztva. A megosztott anyagnak tartalmaznia kell az alábbiakat:

• A szoftver forráskódja.

• Natív asztali alkalmazások esetén a program telepítőkészlete.

• Az adatbázis adatbázismodell-diagramja.

• Az adatbázis export fájlja (dump).

• A szoftveralkalmazás dokumentációja.

• A tesztekhez végzett kód, valamint a teszteredmények dokumentációja.

# A vizsgaremek bemutatója (1 oldal)

A vizsgafeladat során a vizsgázó gyakorlati bemutatóval összekapcsolt szóbeli előadás formájában mutatja be a

• szoftver célját

• műszaki megvalósítását

• működését

• forráskódját

• a csapaton belüli munkamegosztást, a fejlesztési csapatban betöltött szerepét, a fejlesztés során használt projektszervezési eszközöket.

A fentieken túl maximum 3-5 perces angol nyelven tartott szóbeli előadás formájában összefoglalót ad a szoftver céljáról és működéséről, valamint angolul válaszol a vizsgáztató végfelhasználói szerepben feltett maximum 2-3 kérdésére. Amennyiben a munkacsapat más tagjai is azonos csoportban vizsgáznak, akkor a bemutatót közösen is megtarthatják, de ebben az esetben is biztosítani kell, hogy minden vizsgázó egyenlő arányban vegyen részt a bemutatóban, illetve minden vizsgázónak önállóan kell bemutatnia a saját feladatrészét magyarul és angolul egyaránt. A vizsgaremek elkészítésére rendelkezésre álló idő: A vizsgaremeket a záróvizsga tanévében kell a vizsgázónak elkészítenie. A vizsgaremek bemutatására és megvédésére maximum 30 perc áll a vizsgázó rendelkezésére

Megjegyzés: kérdésekhez fogunk készíteni egy Classroom bejegyzést, ahol le lehet adni a kívánt kérdéseket, amit a csapatok tagjai fogalmaznak meg, (tagonként 2-3 kérdés legyen)!

# A vizsgaremek bemutatója (2 oldal)

# Mellékletek

# Index.js:

const http = require('http');

const fs = require('fs');

const server = http.createServer(function (req, res) {

  switch(true) {

    case req.url === '/' && req.method === 'GET':

      fs.readFile('./index.html', (err, file) => {

        res.setHeader('content-type', 'text/html');

        res.end(file);

      });

      break;

    case req.url === '/script.js' && req.method === 'GET':

      fs.readFile('./public/script.js', (err, file) => {

        res.setHeader('content-type', 'application/javascript');

        res.end(file);

      });

      break;

    case req.url === '/phones' && req.method === 'GET':

      fs.readFile('./phones.json', (err, file) => {

        res.setHeader('content-type', 'application/json');

        res.end(file);

      });

      break;

    case req.url === '/phones' && req.method === 'POST':

      let body = '';

      req.on('data', function (chunk) {

         body += chunk.toString();

      });

      req.on('end', function () {

        const newPhone = JSON.parse(body);

        fs.readFile('./phones.json', (err, data) => {

            const phones = JSON.parse(data);

            phones.push(newPhone);

            fs.writeFile('./phones.json', JSON.stringify(phones), () => {

              res.end(JSON.stringify(newPhone));

            })

        })

      });

      break;

    default:

      res.end('404');

  }

});

server.listen(8080);

# index.html:

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=, initial-scale=1.0" />

    <title>Document</title>

    <link

      href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"

      rel="stylesheet"

      integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh"

      crossorigin="anonymous"

    />

  </head>

  <body>

    <div class="container">

      <div class="mb-2"></div>

      <button id="fetch-phones" class="btn btn-primary">

        Telefonok lekérdezése

      </button>

      <div id="phone-list-component"></div>

      <form id="create-phone">

        <label>

          Név:

          <input type="text" name="name" class="form-control" />

        </label>

        <label>

          Márka:

          <input type="text" name="brand" class="form-control" />

        </label>

        <button type="submit" class="btn btn-primary btn-sm">

          +

        </button>

      </form>

    </div>

    <script src="script.js"></script>

  </body>

</html>

# Irodalomjegyzék

1. Téma: Az állatok farsangja, webcím: [www.allatok.hu](http://www.allatok.hu), letöltés dátuma: 2023. 04. 13.
2. Téma: Feladatok és követelmények, webcím: [www.oktatas.hu](http://www.oktatas.hu), letöltés dátuma: 2023. 04. 13.
3. Téma: A farkasokról, webcím: [www.farkasok.hu](http://www.farkasok.hu), letöltés dátum: 2023. 04. 13.
4. Webcím: [www.sulla.hu](http://www.sulla.hu) , letöltés dátuma: 2023. 04. 13.
5. Webcím: [www.valami.hu](http://www.valami.hu) , letöltés dátum: 2023. 04. 13.