**Ceglédi SZC Közgazdasági és Informatikai Technikum Szoftver Fejlesztő és -tesztelő**

**5 0613 1203**

**Dokumentáció a „MathSolve” weboldalhoz**

**Bujdosó Bálint László | Mózes Bence | Pintea Roland Dániel**

**Cegléd, 2025**A képen kör, szöveg, Betűtípus, embléma látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

**Math Solve**

Általános dokumentáció

# Bevezetés

## Projekt bemutatása

|  |  |
| --- | --- |
| Mi a projekt célja? | Ez a webalkalmazás lehetőséget biztosít a felhasználóknak, hogy fejlesszék matematikai készségeiket gyakorlófeladatok és versenyek segítségével. Célja, hogy interaktív és motiváló környezetet teremtsen a tanulás és fejlődés érdekében. |
| Miért készült a projekt? | Az oldal oktatási és gyakorlási célokra egyaránt használható, emellett lehetőséget biztosít a versenyzésre és a közösségi élmény megélésére is. Az alkalmazás nyitott minden matematikát kedvelő felhasználó számára, és folyamatos fejlesztés alatt áll, hogy minél több igényt kielégíthessen. |
| Kik a célfelhasználók? | Az alkalmazás bárki számára elérhető, aki szeretné fejleszteni matematikai készségeit, legyen szó diákokról, tanárokról, versenyzőkről vagy egyszerűen csak a matematika iránt érdeklődő felhasználókról. |

## Főbb funkciók összefoglalása

* **Felhasználói fiók kezelése** – Regisztráció, bejelentkezés és profilkezelés
* **Matematikai feladatok megoldása és létrehozása** – Feladatok gyakorlása, saját feladatok feltöltése
* **Fejlődési statisztikák** – Teljesítmény és előrehaladás nyomon követése
* **Versenyeken való részvétel** – Kihívások és matematikai versenyek megoldása
* **Közösségi interakciók** – Kapcsolódás más felhasználókhoz, tapasztalatok megosztása, segítségnyújtás és támogatás kérése

## Pár használt technológia

* **Frontend:**
  + **Vue 3** – A projekt alapja, a felhasználói felület építéséhez használt modern JavaScript framework.
  + **Vuetify** – A felhasználói felület gyors fejlesztését segítő material design komponens könyvtár**.**
  + **Vitest** –A tesztelési keretrendszer, amely biztosítja az alkalmazás egységtesztelését.
  + **Concurrently** –A frontend és backend egyszerre történő futtatásához szükséges eszköz.
  + **Pinia** –Állapotkezelésre szolgáló könyvtár a Vue 3-ban, amely biztosítja a globális állapot kezelését az alkalmazásban.
* **API:**
  + **Axios** –HTTP kérés küldésére és válaszok kezelésére használt könyvtár.
  + **TanStack Query** – Aszinkron adatlekérések és szerver oldali állapotkezelés kezelésére szolgáló könyvtár.
* **Backend:**
  + **Express** – A backend szerver keretrendszer, amely az alkalmazás logikáját és API-jait kezeli.
  + **Sequelize** – ORM (Object-Relational Mapping) könyvtár, amely lehetővé teszi az adatbázis kezelést JavaScript-ben.
  + **Dotenv** – Környezeti változók kezelésére használt könyvtár.
  + **Swagger** – API dokumentáció generálására és tesztelésére használt eszköz.
  + **Multer** – A fájlok feltöltéséhez használt middleware.
* **Database:**
  + **Mysql** – Adatok tárolására való környezet.

### Csapattagok szerepkörei és hozzájárulásai

A projekt megvalósítása során fontos szempont volt számunkra, hogy minden csapattag teljes körű tapasztalatot szerezzen a fejlesztési folyamat minden területén. Ezért úgy döntöttünk, hogy nem szigorúan backend és frontend szerepkörökre osztjuk a munkát, hanem inkább oldalak és komponensek szerint haladunk. Ez a módszer lehetővé tette, hogy mindenki dolgozzon az adatbáziskezelésen (MySQL), a backend fejlesztésén (Express.js, Sequelize ORM), az API integráción (Axios, TanStack Query), valamint a frontend megvalósításán (Vue.js).

Ez a megközelítés segítette a csapattagokat abban, hogy átfogóbb képet kapjanak a teljes rendszer működéséről, és megkönnyítette az egyes funkciók összekapcsolását. A munkamegosztás során mindenki egy-egy oldalt vagy funkcionális egységet kapott, amelyért teljes mértékben felelősséget vállalt. Ezzel biztosítottuk, hogy mindenki aktívan hozzájáruljon a projekt sikeréhez, miközben fejlesztette a különböző technológiákhoz kapcsolódó készségeit.

#### Az alábbiakban összefoglaljuk, hogy ki melyik oldalon vagy komponensen dolgozott:

* **Bujdosó Bálint László**
  + Fő oldal | Feladat létrehozása | Feladat ellenőrzése | Profil oldal | Áruház oldal
* **Mózes Bence**
  + Verseny feladatok oldal | Verseny oldal | Rólunk oldal
* **Pintea Roland Dániel**
  + Közösségi oldal | Bejelentkezés / Regisztráció oldal | Felhasználó beállítások | Admin panel

# Telepítési Útmutató

A projekt telepítése és futtatása a következő lépésekben történik:

1. Előfeltételek
   * Telepítsd a [Node.js-t](https://nodejs.org/), ha még nincs telepítve.
   * Telepítsd a [Git-t](https://git-scm.com/), ha még nincs telepítve.
   * Telepítsd a [XAMPP](https://www.apachefriends.org/)-ot, ha még nincs telepítve.
   * Telepítsd a Python-t (ajánlott a Microsoft áruházból a 3.12-es verziót), ha még nincs telepítve.
   * Bizonyosodj meg arról hogy az adatbázis MySQL-ben létre lett hozva elindítás után. (adatbázis neve: [math\_solve\_vizsgaremek](http://localhost/phpmyadmin/index.php?route=/database/structure&db=math_solve_vizsgaremek))
2. A projekt klónozása
   * Klónozd a projektet a következő parancs futtatásával: git clone <https://github.com/MozesBence/matekos_leetcode.git>
3. Projekt futtatása

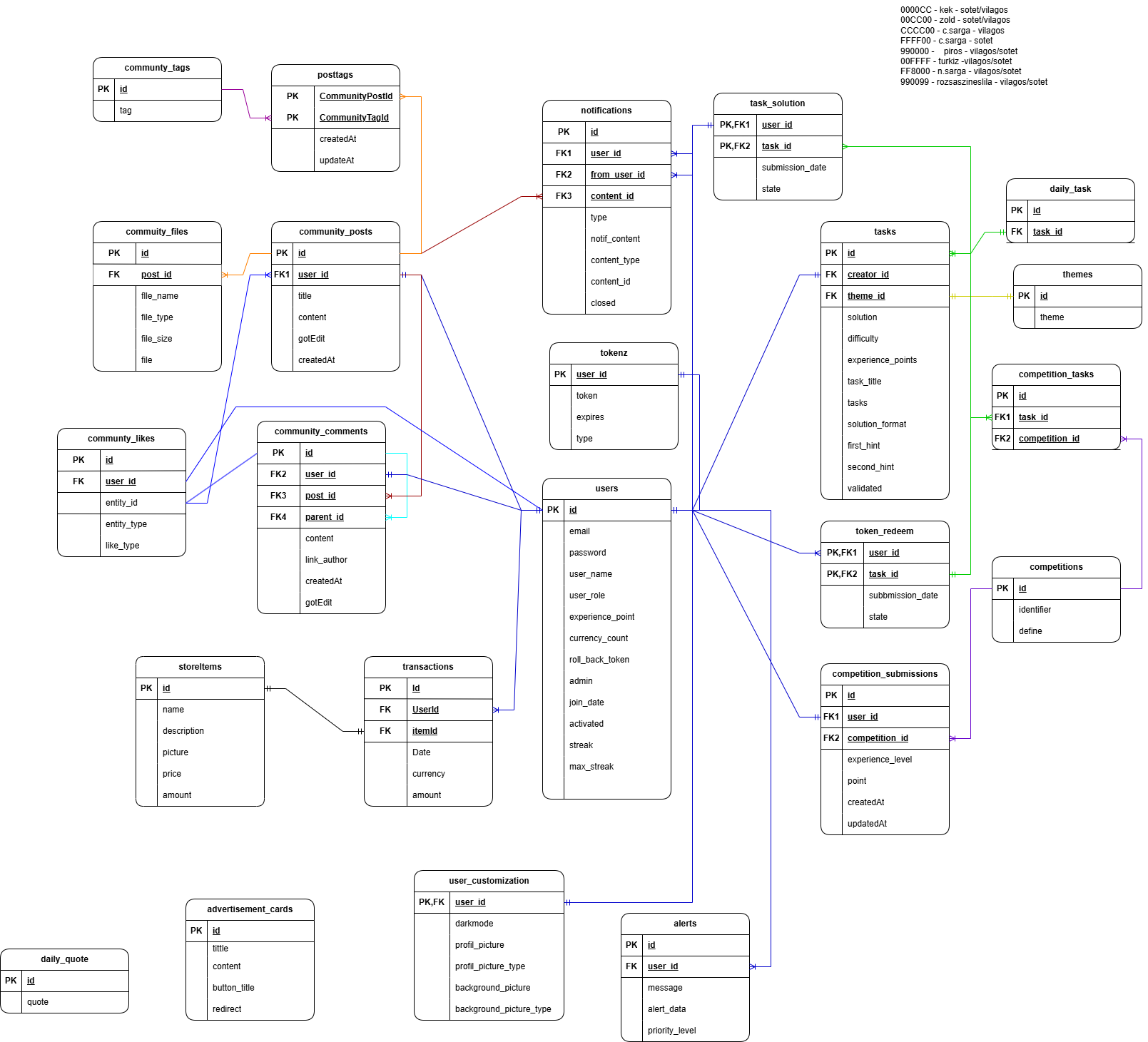
* Elsőnek indítsd el a start.bat fájlt ami a projekt mappájában találsz.

A projektet a következő parancs futtatásával indíthatod el:

* + XAMPP elindítása
    - MySQL elindítása
    - Apache elindítása
  + Felhasználó környezet indításához írd be az 1-es számot. [[Felhasználói dokumentáció](#_Felhasználó_környezet_beállítása)]
  + Fejlesztői környezet indításához írd be a 2-es számot. [[Fejlesztői dokumentáció](#_Fejlesztői_környezet_beállítása)]

## Adatbázis

A projekt során az adatok hatékony és biztonságos tárolása érdekében a MySQL adatbázis-kezelőt választottuk. A MySQL egy megbízható, skálázható és széles körben elterjedt relációs adatbázis-kezelő rendszer, amely lehetővé teszi az adatok strukturált és optimalizált tárolását.

Az adatbázis kialakításakor törekedtünk a megfelelő normalizálásra, hogy minimalizáljuk az adatredundanciát és biztosítsuk a gyors lekérdezéseket. Az adatbázis szerkezetét az alkalmazás igényeihez igazítottuk, beleértve a felhasználói adatok, jogosultságok és egyéb fontos entitások kezelését. Az adatbázis műveleteket a Sequelize ORM segítségével valósítottuk meg, amely megkönnyíti a MySQL adatbázissal való kommunikációt, biztosítja a tranzakciókezelést, valamint elősegíti a biztonságos és hatékony adatkezelést.

Felhasználói dokumentáció

# Bevezetés

A képen szimbólum, embléma, Grafika, Betűtípus látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.Weboldalunk célja, hogy matematikai feladatok megoldásával fejleszd tudásodat, versenyezz másokkal, és élvezd a közösségi interakciókat. A felhasználók számára lehetőség van feladatokat beküldeni, kihívásokban részt venni, valamint statisztikákat követni a fejlődésükről. Az oldal felhasználóbarát és lehetővé teszi a testre szabást, mint például a sötét mód funkció.

# Felhasználó környezet beállítása

## Előfeltételek:

* Telepítsd a [Node.js-t](https://nodejs.org/), ha még nincs telepítve.
* Telepítsd a [Git-t](https://git-scm.com/), ha még nincs telepítve.
* Telepítsd a [XAMPP](https://www.apachefriends.org/)-ot, ha még nincs telepítve.
* Telepítsd a Python-t (ajánlott a Microsoft áruházból a 3.12-es verziót), ha még nincs telepítve.
* Bizonyosodj meg arról hogy az adatbázis MySQL-ben létre lett hozva elindítás után. (adatbázis neve: [math\_solve\_vizsgaremek](http://localhost/phpmyadmin/index.php?route=/database/structure&db=math_solve_vizsgaremek))

## Projekt indítása

* Indítsd el a start.bat fájlt ami a projekt mappájában találsz
* Írd be az 1-es számot

**Regisztráció és Bejelentkezés**

**Regisztráció**

* A regisztrációhoz az alábbi adatok szükségesek:
  + **E-mail cím**
  + **Felhasználónév**
  + **Jelszó**
* A regisztráció megerősítéséhez a megadott e-mail címre egy megerősítő linket küldünk. A regisztráció csak a megerősítés után válik véglegessé. A képen szöveg, Betűtípus, képernyőkép, fehér látható

  Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

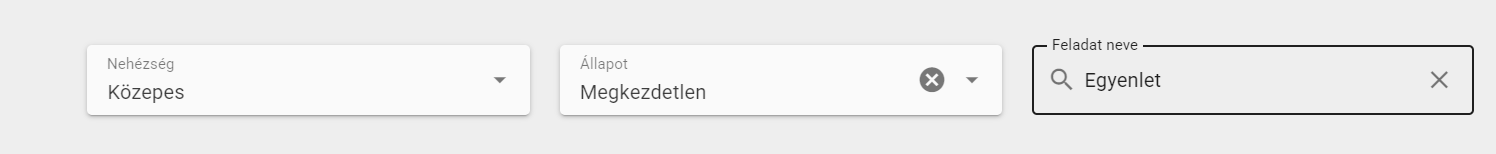
**Bejelentkezés**

* A bejelentkezéshez add meg a regisztrált **e-mail címed** és **jelszavad**.

**Funkciók és Navigáció**

**Feladatok**

* A főoldalon matematikai feladatok találhatóak, amiket három szempont alapján szűrhetsz:
  1. **Téma** (pl. algebra, geometria, stb.)
  2. **Nehézség** (könnyű, közepes, nehéz)
  3. **Állapot** (kész, függőben lévő, megkezdett)
* **Keresés**: A felhasználók saját gondolatuk alapján is kereshetnek feladatokat.



* A képen képernyőkép, zöld, Grafika, szimbólum látható

  Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.Ha nem tudod, melyik feladatot oldd meg, kattints a **"Random feladat"** gombra, és egy véletlenszerű feladatot kapsz.
* **Napi feladatok**: A napi feladatok teljesítése után **aranyat** gyűjthetsz, amit az **áruházban** különböző termékekre válthatsz be.

A képen szöveg, képernyőkép, szoftver, Számítógépes ikon látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

A képen szöveg, képernyőkép, szoftver, Számítógépes ikon látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

**Statisztikák**

* Az oldal alján található egy statisztikai panel, amely a feladatok megoldását mutatja, csoportosítva a nehézségi szintek szerint, illetve a profil fülön is látható egy statisztikai panel.

A képen szöveg, képernyőkép, diagram, sor látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

**Napi Idézet és Szint**

* Minden nap másik **napi idézet** jelenik meg, hogy inspiráljon.
* A felhasználók szintje folyamatosan frissül a feladatok megoldása alapján.

**Feladat Beküldése**

* Ha saját feladatot szeretnél megosztani, lehetőséged van beküldeni egy új feladatot. Az adminisztrátorok elfogadják, és csak ezt követően válik láthatóvá.

A képen szöveg, képernyőkép, szoftver, Számítógépes ikon látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

A képen szöveg, képernyőkép, szoftver, Weblap látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

**Versenyek és Kihívások**

* **Heti versenyek**: Minden hétfőn új heti kihívás indul. Ha megoldod, jutalmat kapsz, amely lehetővé teszi, hogy felkerülj a ranglistára.
* **Havi versenyek**: Minden hónap első napján új havi kihívások indulnak.
* Az előző kihívások megtekinthetők, és azokat is megoldhatod.

A képen szöveg, Téglalap, Betűtípus, fehér látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

**Jutalmak**

* A kihívások teljesítése után a felhasználók **jutalmakat** szerezhetnek, melyeket **MathSolve merch** termékekre válthatnak be.

A képen szöveg, képernyőkép, Márka, szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

**Közösségi Fórum**

* A közösségi rész lehetőséget ad a felhasználóknak arra, hogy megosszák gondolataikat, segítséget kérjenek, vagy egyszerűen beszélgessenek másokkal.

A képen szöveg, Betűtípus, képernyőkép látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

* A fórumon keresni tudsz posztokra:
  + **Cím** alapján
  + **Feltöltési dátum** alapján
  + **Népszerűség** alapján
  + A képen szöveg, képernyőkép, Betűtípus, szám látható

    Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.**Tagek** alapján

**Felhasználói Adatok Módosítása**

* A felhasználói adatok módosítása lehetséges egy külön ablakban.
* Itt tudod frissíteni **személyes adataidat**, és láthatod az **értesítéseket** is.

A képen képernyőkép, rajzfilm, Számítógépes játék, Animáció látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

A képen szöveg, képernyőkép, Betűtípus, szám látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

**Rólunk**

* Az oldal tartalmaz egy **Rólunk** oldalt, ahol megismerheted a fejlesztőket és az oldal céljait.

A képen szöveg, képernyőkép, szoftver, Operációs rendszer látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

Fejlesztői dokumentáció

# Bevezetés

Ez a dokumentáció a projekt fejlesztésében közvetlenül részt vevő csapattagoknak, valamint az újonnan érkező fejlesztőknek szól. A dokumentum segít abban, hogy mindenki pontosan megértse a rendszer felépítését, a használt eszközöket, valamint a fejlesztési és tesztelési folyamatokat.

## Dokumentáció célja:

* Részletes architekturális áttekintést nyújtani.
* Leírni a fejlesztői környezet és eszközök beállítását.
* Ismertetni a kódszerkezetet, a mappastruktúrát és a moduláris felépítést.
* Bemutatni a tesztelési stratégiát és az automatizált tesztelési folyamatokat.

# Projekt architektúra és mappastruktúra

## Technológiai Stack:

A projekt több rétegből épül fel:

* **Frontend:**
  + **Vue 3** – A projekt alapja, a felhasználói felület építéséhez használt modern JavaScript framework.
  + **Vuetify** – A felhasználói felület gyors fejlesztését segítő material design komponens könyvtár**.**
  + **Vitest** –A tesztelési keretrendszer, amely biztosítja az alkalmazás egységtesztelését.
  + **Concurrently** –A frontend és backend egyszerre történő futtatásához szükséges eszköz.
  + **Pinia** –Állapotkezelésre szolgáló könyvtár a Vue 3-ban, amely biztosítja a globális állapot kezelését az alkalmazásban.
* **API:**
  + **Axios** –HTTP kérés küldésére és válaszok kezelésére használt könyvtár.
  + **TanStack Query** – Aszinkron adatlekérések és szerver oldali állapotkezelés kezelésére szolgáló könyvtár.
* **Backend:**
  + **Express** – A backend szerver keretrendszer, amely az alkalmazás logikáját és API-jait kezeli.
  + **Sequelize** – ORM (Object-Relational Mapping) könyvtár, amely lehetővé teszi az adatbázis kezelést JavaScript-ben.
  + **Dotenv** – Környezeti változók kezelésére használt könyvtár.
  + **Swagger** – API dokumentáció generálására és tesztelésére használt eszköz.
  + **Multer** – A fájlok feltöltéséhez használt middleware.
* **Database:**
  + **Mysql** – Adatok tárolására való környezet.

## Mappastruktúra:

Ezen struktúra részletes leírása segíti a fejlesztőket abban, hogy gyorsan megtalálják a releváns kódrészeket.

[[mappa struktúrák](structure.txt)]

# Fejlesztői környezet beállítása

## Előfeltételek:

* Telepítsd a [Node.js-t](https://nodejs.org/), ha még nincs telepítve.
* Telepítsd a [Git-t](https://git-scm.com/), ha még nincs telepítve.
* Telepítsd a [XAMPP](https://www.apachefriends.org/)-ot, ha még nincs telepítve.
* Telepítsd a Python-t (ajánlott a Microsoft áruházból a 3.12-es verziót), ha még nincs telepítve.
* Bizonyosodj meg arról hogy az adatbázis MySQL-ben létre lett hozva elindítás után. (adatbázis neve: [math\_solve\_vizsgaremek](http://localhost/phpmyadmin/index.php?route=/database/structure&db=math_solve_vizsgaremek))

## Projekt indítása

* Indítsd el a start.bat fájlt ami a projekt mappájában találsz
* Írd be a 2-es számot

## Admin felhasználó:

* Fiók név: pinteadani88@gmail.com
* Password: admin12345

# Kódszerkezet és moduláris felépítés

**Frontend:**

* A komponensek egyértelműen elkülönülnek, minden komponens saját mappában található.
* Az állapotkezelés a Pinia segítségével történik, így a globális állapot könnyen kezelhető.
* A Vue komponensekben a logika és a megjelenítés elkülönítése segíti az új funkciók gyors integrálását.

**Backend:**

* Az Express.js szerver felelős az API végpontok kezeléséért.
* A middleware-ek és router-ek használata biztosítja a kód moduláris felépítését.
* Az adatbázis műveleteket a Sequelize ORM kezeli, így a modellek és az adatbázis kapcsolatok egy helyen kezelhetők.
* A hibakezelés és logging megfelelően megoldott, hogy könnyen nyomon követhetőek legyenek a problémák a futás során.

# API dokumentáció és hibakezelés

A projekt **backend** API-jának dokumentációját a **Swagger** segítségével generáltuk, amely biztosítja a felhasználók és fejlesztők számára az API végpontjainak könnyű elérését és használatát.

### Mi a Swagger?

A **Swagger** (más néven OpenAPI) egy eszközkészlet, amely lehetővé teszi az API dokumentáció automatikus generálását, tesztelését és megértését. A Swagger UI egy vizuális felületet biztosít az API végpontjainak böngészésére, ahol a felhasználók megtekinthetik az összes elérhető végpontot, azok paramétereit, válaszait, és kipróbálhatják a különböző HTTP kéréseket.

### Hogyan érheted el a Swagger dokumentációt?

A Swagger API dokumentáció az alábbi linken érhető el a projekt **backend** részén:

**A helyi fejlesztési környezetben elérhető a Swagger UI az alábbi URL-en, miután a backend szerver fut:**

[**Swagger API Dokumentáció**](http://localhost:3000/api-docs)

### Használat:

1. **API végpontok böngészése** – Az összes elérhető API végpont megtekintése.
2. **API kérések kipróbálása** – A Swagger UI lehetővé teszi a különböző API kérések tesztelését, például GET, POST, PUT, DELETE kéréseket. A felhasználók egyszerűen beírhatják a kívánt paramétereket és végrehajthatják a kérést közvetlenül az UI-ból.
3. **Példák és válaszok** – Minden végponthoz tartozik egy példakérés és a választ is bemutatja a Swagger, így könnyen megérthető, hogy miként működik az adott API végpont.

## Tesztelés

A projekt tesztelése a backend alkalmazásban a [Jest](https://jestjs.io/) és [Supertest](https://www.npmjs.com/package/supertest) segítségével történik, amelyek lehetővé teszik a unit tesztelést és az API végpontok integrált tesztelését.

### Tesztelési keretrendszer

* **Jest**: A Jest egy JavaScript tesztelési keretrendszer, amely a tesztelési logikát és a tesztek futtatását kezeli. Lehetővé teszi az aszinkron műveletek kezelését, és részletes tesztelési eredményeket biztosít.
* **Supertest**: A Supertest egy HTTP aszinkron tesztelő eszköz, amely lehetővé teszi az API végpontok tesztelését. A Supertest segítségével szimulálhatunk HTTP kéréseket és ellenőrizhetjük a válaszokat.

### Tesztelési parancsok

* **A tesztek futtatása**: Elsőnek indítsd el a srart.bat fájlt és írd be a 3-as számot.

### Tesztelési struktúra

A tesztek a backend projektben a *tests* mappában találhatók. Az egyes tesztelési fájlok általában az adott funkciók vagy API végpontok tesztelésére vannak kialakítva.

* **Unit tesztek**: A backend logika egyes részleteit, mint például az adatbázis műveletek vagy egyes segéd funkciók, unit tesztekkel ellenőrizzük.
* **API tesztek**: Az API végpontokat a Supertest segítségével teszteljük. Az API tesztek szimulálják a HTTP kéréseket és ellenőrzik a válaszokat, például a státuszkódokat, a válaszadatokat és a hibakezelést.
* **Adatbázis**: Biztosítani kell hogy az adatbázis MySQL-ben létre van hozva és eléri.

### Tesztelési hibák kezelése

Amennyiben tesztelés közben hibák merülnek fel, ellenőrizni kell:

* **Az adatbázis kapcsolódik-e megfelelően**: Biztosítani kell, hogy a tesztelés előtt az adatbázis csatlakozik és inicializálódik.

Ha a tesztek során egy-egy hiba felmerül, azokat a konzolban részletes üzenetek formájában ellenőrizhetjük, ami segít a problémák gyors azonosításában.

### Teszt eredmények

Az alábbi táblázat a projekt automatikus tesztelésének eredményeit tartalmazza.  
A jelentést a **Jest** tesztfuttató eszköz generálta, amely részletes információt nyújt:

* a tesztlefedettségi arányról,
* a sikeresen és sikertelenül lefutott tesztesetekről,
* valamint a kódban előforduló hibás vagy nem lefedett sorokról.

A riport célja, hogy átlátható képet adjon a jelenlegi tesztelési állapotról, és segítse a fejlesztőket a hibák gyors azonosításában, valamint a tesztlefedettség növelésében.

A képen szöveg, képernyőkép látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

A képen szöveg, képernyőkép, Betűtípus, tervezés látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

A képen szöveg, Betűtípus, képernyőkép látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

A képen szöveg, képernyőkép látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.