

Hephaistos

Név

February 21, 2025



Contents

Bevezetés	3
1 Használt Technológiák	3
1.1 Backend	3
1.1.1 ASP.NET Core	3
1.2 Adatbázis	4
1.3 Frontend	4
1.3.1 React	4
1.4 Fejlesztői Eszközök	4
2 Projekt Felépítése	4
2.1 Backend	5
2.2 Frontend	5
2.3 Adatbázis	5
3 Funkcionalitás	5
3.1 Órarend Generálás	5
3.2 Órarend Megtekintése és Módosítása	6
3.3 Értesítések és Emlékeztetők	6
3.4 Statisztikák és Jelentések	6

Bevezetés

A középiskolai projektünk célja egy olyan egyetemi órarendgeneráló alkalmazás kifejlesztése, amely segíti a diákokat az órarendjük megtervezésében, különösen akkor, ha el vannak csúszva az adott félévvel. A felsőoktatásban gyakran előfordul, hogy a hallgatók különböző okok miatt nem tudják időben teljesíteni a tárgyaikat, és ez bonyolult órarend-tervezést eredményezhet. Az alkalmazásunk abban nyújt segítséget, hogy optimalizált órarendet készítsen, figyelembe véve a követelményeket és az előfeltételeket, minimalizálva az ütközéseket és a szabadidő-kieséseket.

A projekt keretében megvizsgáljuk a jelenlegi órarend-készítési módszereket és azok korlátait, majd egy algoritmus alapú megoldást dolgozunk ki, amely hatékonyabbá és rugalmasabbá teszi a tervezési folyamatot. A fejlesztés során kiemelt figyelmet fordítunk a felhasználóbarát felület kialakítására, amely lehetővé teszi a hallgatók számára, hogy egyszerűen hozzáadhassák a kötelező és választható tantárgyakat, valamint figyelembe vehessék a személyes preferenciáikat, például a szabadnapokat vagy a kedvelt időpontokat.

Ez az alkalmazás nemcsak azoknak a diákoknak nyújt megoldást, akik késésben vannak a tanulmányaikkal, hanem azoknak is, akik előre szeretnék látni a lehetséges órarendjeiket, így jobban megtervezhetik a félévüket. Célunk egy olyan eszköz létrehozása, amely megkönnyíti a hallgatók életét, csökkenti a stresszt és növeli a tanulmányi hatékonyságot.

1 Használt Technológiák

A Hephaistos projekt különböző technológiákat használ a backend és a frontend fejlesztéséhez, valamint az adatbázis kezeléséhez.

1.1 Backend

A backend a .NET Core keretrendszert használja, amely egy nyílt forráskódú, platformfüggetlen keretrendszer a modern, felhőalapú, internetkapcsolattal rendelkező alkalmazások fejlesztéséhez. A .NET Core lehetővé teszi a fejlesztők számára, hogy nagy teljesítményű és skálázható alkalmazásokat hozzanak létre.

1.1.1 ASP.NET Core

Az ASP.NET Core a .NET Core keretrendszer része, amelyet webalkalmazások és API-k fejlesztésére használnak. Az ASP.NET Core előnyei közé tartozik a magas teljesítmény, a platformfüggetlenség és a moduláris felépítés. Az ASP.NET Core támogatja a modern webfejlesztési szabványokat és eszközöket, mint például a dependency injection, a middleware-ek és a RESTful API-k.

Az ASP.NET Core alkalmazások könnyen telepíthetők és skálázhatók, így ideálisak a felhőalapú környezetekben történő futtatásra. Az ASP.NET Core lehetővé teszi a fejlesztők számára, hogy gyorsan és hatékonyan hozzanak létre biztonságos és megbízható webalkalmazásokat.

1.2 Adatbázis

Az adatok tárolására MySQL adatbázist használunk. A MySQL egy népszerű, nyílt forráskódú relációs adatbázis-kezelő rendszer, amely nagy teljesítményt és megbízhatóságot kínál. A PHPMyAdmin eszközt használjuk az adatbázis kezelésére, amely egy webalapú felületet biztosít az adatbázisok kezeléséhez.

1.3 Frontend

A frontend fejlesztéséhez a React könyvtárat használjuk. A React egy népszerű JavaScript-könyvtár, amelyet a felhasználói felületek egyszerűbb fejlesztésére használnak. A React lehetővé teszi a fejlesztők számára, hogy dinamikus és interaktív felhasználói felületeket hozzanak létre.

1.3.1 React

A React egy komponens-alapú JavaScript könyvtár, amelyet a Facebook fejlesztett ki és 2013-ban adtak ki. A React fő célja a felhasználói felületek egyszerűbb és hatékonyabb fejlesztése. A React lehetővé teszi a fejlesztők számára, hogy újrafelhasználható komponenseket hozzanak létre, amelyek könnyen karbantarthatók és bővíthetők.

A React egyik legfontosabb jellemzője a virtuális DOM (Document Object Model), amely javítja az alkalmazás teljesítményét azáltal, hogy minimalizálja a valódi DOM manipulációkat. A React támogatja a JSX (JavaScript XML) szintaxist, amely lehetővé teszi a HTML-szerű kód írását a JavaScript-ben, növelve a kód olvashatóságát és karbantarthatóságát.

A React ökoszisztémája számos kiegészítő könyvtárat és eszközt tartalmaz, mint például a React Router a kliensoldali útvonalkezeléshez és a Redux az állapotkezeléshez. Ezek az eszközök segítenek a fejlesztőknek hatékonyabb és skálázhatóbb alkalmazásokat létrehozni.

1.4 Fejlesztői Eszközök

A fejlesztés során különböző eszközöket használunk a hatékonyság növelése és a hibakeresés megkönnyítése érdekében:

- **Visual Studio Code (VSCode):** Egy népszerű kódszerkesztő, amely számos bővítménnyel rendelkezik, például az ESLint és Prettier segítségével.
- **React Developer Tools:** Egy böngészőbővítmény, amely lehetővé teszi a React komponensek és állapotok ellenőrzését a böngészőben.
- **PHPMyAdmin:** Egy webalapú eszköz a MySQL adatbázisok kezelésére.

2 Projekt Felépítése

A Hephaistos projekt három fő komponensből áll: backend, frontend és adatbázis. Az alábbiakban bemutatjuk ezeknek a komponenseknek a felépítését és funkcióit.

2.1 Backend

A backend a .NET Core keretrendszert használja, és a következő főbb részekből áll:

- **Program.cs:** Az alkalmazás belépési pontja és a szolgáltatások konfigurációja.
- **Controllers:** Az API végpontok kezelése.
- **Models:** Az adatmodellek definiálása.
- **Services:** Az üzleti logika megvalósítása.
- **Repositories:** Az adatbázis műveletek kezelése.

2.2 Frontend

A frontend a React könyvtárat használja, és a következő főbb részekből áll:

- **App.jsx:** Az alkalmazás fő komponense és az útvonalak definiálása.
- **Components:** Az újrafelhasználható komponensek gyűjteménye.
- **Pages:** Az alkalmazás különböző oldalai.
- **Styles:** A stíluslapok és CSS fájlok.

2.3 Adatbázis

Az adatbázis a MySQL rendszert használja, és a következő főbb részekből áll:

- **Tables:** Az adatok tárolására szolgáló táblák.
- **Relationships:** A táblák közötti kapcsolatok definiálása.
- **Indexes:** Az adatok gyors lekérdezését segítő indexek.

3 Funkcionalitás

A Hephaistos projekt számos funkciót kínál a felhasználók számára, hogy megkönnyítse az órarendek tervezését és kezelését.

3.1 Órarend Generálás

Az alkalmazás lehetővé teszi a felhasználók számára, hogy megadják a kötelező és választható tantárgyaikat, valamint a személyes preferenciáikat. Az algoritmus ezután optimalizált órarendet készít, minimalizálva az ütközéseket és a szabadidő-kieséseket.

3.2 Órarend Megtekintése és Módosítása

A felhasználók megtekinthetik és módosíthatják az órarendjüket a felhasználóbarát felületen keresztül. Az alkalmazás lehetővé teszi az órák hozzáadását, eltávolítását és áthelyezését.

3.3 Értesítések és Emlékeztetők

Az alkalmazás értesítéseket és emlékeztetőket küld a felhasználóknak a közelgő óráikról és fontos határidőkről. Ez segít a diákoknak, hogy ne maradjanak le semmiről.

3.4 Statisztikák és Jelentések

Az alkalmazás statisztikákat és jelentéseket készít az órarendekről és a diákok teljesítményéről. Ez lehetővé teszi a felhasználók számára, hogy nyomon kövessék a haladásukat és azonosítsák a problémás területeket.