

ALKALMAZÁSOK FEJLESZTÉSE

PROJEKT LABOR II.

PROJEKT TERV

Orosz Kristóf

Radvánszky Ádám János

2026. február 24.

Tartalomjegyzék

Bevezető	3
Programozási nyelvek, fejlesztő eszközök	3
Fejlesztési terv	4
Funkciólista	4
Oldalstruktúra ábrája	4
Csoport tagjai és struktúrája	5
Dokumentációs követelmények	6
Tesztelési követelmények	7

Bevezető

A fél éves beadandó keretében egy modern, felhasználóbarát szállásfoglaló webalkalmazás megvalósítását és a hozzá tartozó projekt dokumentáció elkészítését tervezzük elvégezni. A rendszer célja, hogy a felhasználók regisztrációt követően megtekinthessék a rendelkezésre álló szállások részletes információkat kapjanak róluk, valamint foglalásokat hozhassanak létre. A felhasználók saját foglalásaikat egyszerűen tudják majd kezelni, szükség esetén módosítani vagy törölni tudják.

Az alkalmazás több rétegű felépítésben tervezzük elkészíteni. A frontend megvalósításához HTML, CSS és JavaScript technológiákat fogjuk alkalmazni, amelyek biztosítják a felhasználói felület megjelenítését és az interaktív működést. A backend fejlesztése Java nyelven fog történni, amely a kliens és az adatbázis közötti kommunikációért lesz felelős. A rendszer REST alapú kommunikációt használ a frontend és a backend között. Az adatok tárolására az XAMPP szerver MariaDB relációs adatbázis-kezelő rendszert tervezzük használni amelyben külön táblákban kerülnek tárolásra a felhasználók, szállások és foglalások adatai.

A dokumentumkészítés során a projekt teljes fejlesztési folyamatát szeretnénk bemutatni. Ennek megfelelően a dokumentumok tartalmazni fogják többek között a projekt célját, követelményeit, rendszertervét, architektúra leírását, adatbázis-tervét, UML diagramokat, a megvalósítás bemutatását, valamint a tesztelési és felhasználói dokumentációt is.

Programozási nyelvek, fejlesztő eszközök

TECHNOLÓGIA	LEÍRÁS
HTML	A weboldal szerkezetének kialakításához fogjuk használni, amely meghatározza az oldalak felépítését és az egyes felületi elemek elrendezését.
CSS	A weboldal vizuális megjelenésének és reszponzív kialakításának biztosításához fogjuk alkalmazni, hogy egységes és felhasználóbarát felületet hozzunk létre.
JavaScript	A kliensoldali dinamikus funkcionalitások megvalósításához fogjuk használni, például adatlekérdezésekhez, eseménykezeléshez és interaktív műveletekhez.
Java	A szerveroldali működés és az üzleti logika megvalósításához fogjuk alkalmazni, amely biztosítja a frontend és az adatbázis közötti kommunikációt.
REST API	A frontend és a backend közötti kommunikáció megvalósítására fogjuk használni, hogy az adatok szabványos módon kerüljenek továbbításra.
MariaDB	Az adatok tárolására és kezelésére fogjuk használni, ahol a felhasználók, szállások és foglalások adatai relációs adatbázisban kerülnek tárolásra.
Fejlesztő eszközök	Az Eclipse környezetet fogjuk használni a backend fejlesztéshez, a Visual Studio Code-ot a frontendhez és XAMPP Maria DB-t az adatbázishoz, valamint Git verziókezelőt a projekt változásainak követésére.

Fejlesztési terv

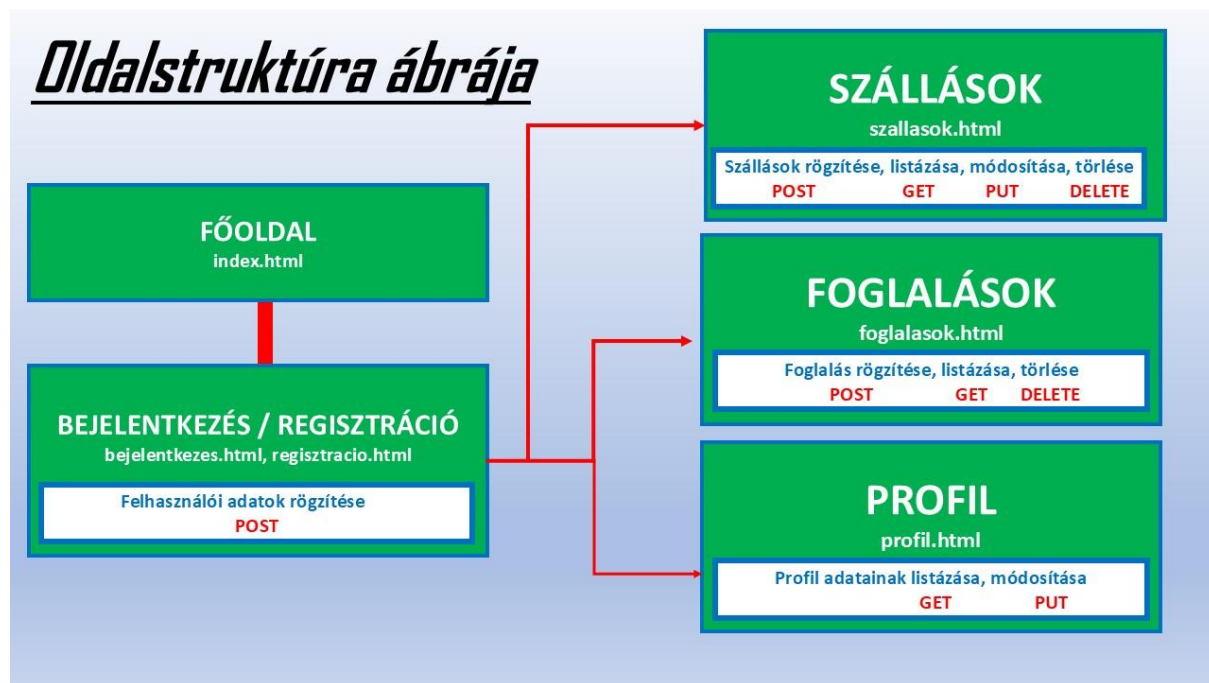
Funkciólista

FUNKCIÓ	HTTP METÓDUS	LEÍRÁS
Szállások listázása	GET	A rendszer megjeleníti az elérhető szálláshelyek listáját és leírásukat.
Foglalások listázása	GET	A felhasználók megtekinthetik a korábban létrehozott foglalásaikat.
Profil adatainak listázása	GET	A rendszer lehetőséget biztosít a felhasználói adatok megtekintésére.
Szállások rögzítése	POST	Lehetővé válik új szálláshelyek adatainak felvétele a rendszerbe.
Felhasználói adatok rögzítése	POST	A rendszer lehetőséget biztosít a felhasználói adatok rögzítésére. Bejelentkezési és regisztrációs adatok megadása.
Foglalás rögzítése	POST	A felhasználók kiválaszthatják a kívánt szállást és létrehozhatják a foglalásukat. A foglalás során a szükséges adatok, például dátumok és vendégszám rögzítésre kerülnek.
Szállások törlése	DELETE	A rendszerből eltávolíthatók a már nem elérhető szállások.
Foglalás törlése	DELETE	A felhasználók törölhetik a korábban létrehozott foglalásaikat.
Szállások módosítása	PUT	A meglévő szállások adatai módosíthatók szükség esetén.
Profil adatainak módosítása	PUT	A felhasználó profiladatai szükség esetén módosíthatók a rendszerben.

Oldalstruktúra ábrája

A rendszer kiindulópontja a főoldal, ahonnan a felhasználók elérhetik a bejelentkezési és regisztrációs felületet, amely biztosítja a rendszer használatához szükséges adatok megadását és a felhasználók azonosítását. A regisztráció során a felhasználók megadják alapadataikat, amelyek az adatbázisban kerülnek tárolásra, majd a bejelentkezést követően hozzáférést kapnak az alkalmazás további funkcióihoz. A sikeres belépést követően a felhasználók a szállások oldalra navigálhatnak, ahol megtekinthetik az elérhető szálláshelyek listáját, részletes információkat olvashatnak az egyes szállásokról, valamint szükség esetén új szállásokat is rögzíthetnek, illetve módosíthatják vagy törölhetik a meglévő adatokat.

A foglalások oldal a szállásfoglalási folyamat kezelésére szolgál, ahol a felhasználók új foglalásokat hozhatnak létre, megtekinthetik a korábban rögzített foglalásaikat, valamint lehetőségük van azok törlésére is. Ez az oldal biztosítja a foglalások áttekinthetőségét és kezelésének egyszerűségét. A profil oldal a felhasználói adatok megjelenítésére szolgál, ahol a felhasználó saját adatait tekintheti meg és módosíthatja azokat.



Csoport tagjai és struktúrája

A projekt sikeres megvalósítása érdekében a fejlesztőcsapat agilis, scrum-alapú módszertant követve szerveződik meg. A csapatunk két főből áll, akik között a szerepkörök az alábbi táblázat szerint fog felosztódni.

SZEREPKÖRÖK	SZEMÉLYEK	FELADATA
Projektmenedzser	Orosz Kristóf	Felelős a fejlesztési folyamat zavartalanosságáért, és a határidők betartatásáért.
Backend fejlesztő	Orosz Kristóf	A serveroldali üzleti logika implementálását, az adatbázis-műveleteket és az API végpontok kialakítását végzi el.
Frontend fejlesztő	Radvánszky Ádám János	A felhasználói felület (UI) és a felhasználói élmény (UX) megvalósításáért felelős. Feladata az API-kkal való kommunikáció kiépítése és a responszív megjelenítés biztosítása.

Tesztelő	Radvánszky Ádám János	Fő feladata a tesztelési tervek kidolgozása, a manuális tesztelés elvégzése.
----------	-----------------------	--

Dokumentációs követelmények

A projektdokumentáció több szinten valósul meg, amely során az alábbi dokumentumok fognak elkészülni.

DOKUMENTUMTÍPUSOK	TARTALMA
PROJEKT TERV	A projekt terv tartalmazza a csoporttagok adatait, a fejlesztési ütemtervet és az elérendő eredményeket. Meghatározza a programozási nyelveket, technikai eszközöket, valamint a dokumentáció és tesztelés követelményeit.
VÍZÍÓ	Fő célja a rendszer általános képének megalkotása, a probléma megértése. Ennek érdekében vizsgálja, kiknek az érdekét szolgálja a probléma megoldása, kik lesznek a használói, melyek lesznek a legfontosabb hatásai.
SOFTWARE KÖVETELMÉNY ANALÍZIS	A RUP legelső fázisában elkezdett, és a következő fázisban részletesebben kidolgozott dokumentum, amely a tervezéshez szükséges részletességgel foglalja össze a rendszerrel szemben támasztott funkcionális és nem funkcionális követelményeket.
FELHASZNÁLÓI KÉZIKÖNYV	Segítségnyújtás a webalkalmazás használatához.
ANALÍZIS MODELL	Az analízis modell az alkalmazás követelményeinek és üzleti igényeinek részletes feltérképezésére használjuk. Ez segít abban, hogy minden érintett fél közös megértést kapjon a projekt céljairól és elvárásairól.
RENDSZERTERVEZÉS	A rendszertervezési dokumentumok a technikai architektúra és a megoldási stratégia részletezésére szolgálnak. Ezek az útmutatók biztosítják, hogy a fejlesztőcsapat átlássa a rendszer felépítését és integrációs folyamatait a hatékony implementáció érdekében.
OMT OBJEKTUM TERVEZÉS	Prototípus jellegű implementációs kezdetek és egy nagyvonalú tesztelési terv.

Tesztelési követelmények

A szoftver megbízható működésének és a hibamentes felhasználói élménynek az alapja a többszintű tesztelési stratégia. A tesztelési folyamatunk a fejlesztés szerves részét fogják képezni.

Tesztelési szintek

- **Egységtesztek:** A legkisebb, önállóan futtatható kódrészletek tesztelése.
- **Integrációs tesztek:** Célja annak ellenőrzése, hogy az egyedi komponensek mint például az adatbázis-kapcsolatok, külső API hívások, frontend-backend kommunikáció megfelelően működnek-e együtt.
- **Rendszertesztesztelés:** A teljes alkalmazás tesztelése felhasználói szemszögből, szimulálva a valós felhasználói folyamatokat a bejelentkezéstől egy adott tranzakció lezárásáig.