Pannon Egyetem

Műszaki Informatikai Kar

Projektlabor

Bojsza Réka Laura

Hajtó Kornél

Fekete Gergő

Témavezető: Horváth Ádám

Jelölésjegyzék

API: alkalmazásprogramozási felület; a kliens és a szerver közti hívások és szerződések összessége.

UI: felhasználói felület; képernyők, komponensek és interakciók együttese.

JWT: JSON Web Token; aláírt, tömör token felhasználói hitelesítéshez és munkamenethez.

Token: rövid, aláírt adatcsomag (pl. hozzáférési vagy frissítési) a hitelesítéshez.

Controller: (vezérlő) a backend erőforrás-végpontjait megvalósító osztály.

Feed: hírfolyam; időrendi/kártyás lista a tartalmakról.

AI: mesterséges intelligencia; jegyzetek automatikus feldolgozását segítő funkciók.

PDF: hordozható dokumentumformátum; feltöltött vagy beágyazott jegyzetfájl.

Q&A: kérdezz–felelek jellegű támogatás/szolgáltatás.

HTTP: a webes kliens–szerver kommunikáció protokollja (kérések/válaszok).

Hot reload: fejlesztés közbeni azonnali UI-frissítés (Flutter).

Plugin: bővítmény/csomag, amely extra funkciókat ad a projekthez.

Dio: HTTP klienskönyvtár Flutterhez.

Riverpod: állapotkezelő könyvtár Flutterhez.

Provider: állapotkezelő könyvtár Flutterhez.

LINQ: lekérdezési nyelv és műveletkészlet C#-ban gyűjtemények/adatok kezelésére.

Git: elosztott verziókezelő rendszer.

GitHub: Git-alapú tárhely és együttműködési platform.

Visual Studio: integrált fejlesztőkörnyezet (.NET/C# fejlesztéshez).

Paywall: előfizetéses/korlátozó hozzáférési „fal”.

For-You Feed: személyre szabott ajánlás-alapú hírfolyam.

PoC: Proof of Concept; működőképességet bemutató, minimális megvalósítás

SRS: Spaced Repetition System; ismétlésritkításos tanulási módszer.

1. Projekt munkaterv

1.1. Cél és fő funkciók:

A projekt célja egy modern, jegyzet alapú közösségi média platform létrehozása, amely diákok és tanárok számára egyaránt készült. A rendszer lehetővé teszi a felhasználók számára, hogy oktatási jegyzeteket rendszerezzenek, osszanak meg és böngésszenek egy együttműködő digitális környezetben.

A felhasználók feltölthetik saját jegyzeteiket, menthetik mások által megosztott anyagokat, és kapcsolatba léphetnek hasonló tanulmányi érdeklődésű emberekkel. A platform célja egy támogató, tudás vezérelt közösség létrehozása, ahol a tanulás társas élménnyé válik, nem pedig egyéni erőfeszítéssé.

1.2. Modulok

1.2.1. Hitelesítés (Bejelentkezés, Regisztráció, Beállítások)

A hitelesítési modul biztonságos felhasználói fiók létrehozását és kezelését teszi lehetővé. A regisztráció során a felhasználók megadhatják iskolai végzettségüket, érdeklődési körüket és tanulási preferenciáikat, hogy személyre szabott tartalomajánlásokat kapjanak.

1.2.2. Jegyzet feltöltése

A felhasználók különböző formátumokban (szöveg, PDF, képek stb.) tölthetnek fel jegyzeteket. Minden jegyzet tartalmazhat metaadatokat, például címet, leírást és címkéket, ami megkönnyíti a rendszerezést és a keresést. A feltöltött jegyzetek lehetnek nyilvánosak vagy privátak, a felhasználói beállításoktól függően.

1.2.3. Keresés

A keresőmodul támogatja a gyors kulcsszóalapú keresést és a téma, szerző, dátum vagy címkék szerinti speciális szűrést. Ez hatékony navigációt biztosít a nagy mennyiségű megosztott tartalomban.

1.2.4. Gyűjtemények

A felhasználók tematikus gyűjteményeket hozhatnak létre, amelyek csoportosítják a kapcsolódó jegyzeteket, akár saját maguk hozták létre, akár másoktól mentették el őket. A gyűjtemények segítenek az anyagok téma, kurzus vagy vizsgatéma szerint strukturálásában, hatékonyan személyre szabott tudástárként működve.

1.2.5. Jegyzetkérések

Ez a modul lehetővé teszi a felhasználók számára, hogy jegyzetigényeket tegyenek közzé, segítséget kérve a közösségtől. Más felhasználók válaszolhatnak, magyarázatokat oszthatnak meg, vagy további forrásokat biztosíthatnak. Ez a funkció ösztönzi a társak közötti tanulást és együttműködést.

1.2.6. Hírfolyam / Interakciók

A fő hírfolyam a követett felhasználók frissítéseit jeleníti meg, beleértve az újonnan feltöltött jegyzeteket, hozzászólásokat és ajánlásokat. A felhasználók kedvelhetik, kommentelhetik, megoszthatják vagy menthetik a bejegyzéseket, elősegítve az elköteleződést és a közösségi részvételt.

1.2.7. Értékelés / Tesztelés

A platform támogatja a jegyzetek és anyagok értékelését. A felhasználók visszajelzést adhatnak, értékelhetik mások jegyzeteit, és opcionálisan kvízeket vagy rövid önértékelő teszteket csatolhatnak a jegyzetekhez. Ez a funkció javítja a tanulási eredményeket, és elősegíti a tartalommal való aktív interakciót.

1.3. Követelmény-dokumentáció

A rendszer első iterációjának célja egy működő alapfolyamat biztosítása: regisztráció és bejelentkezés, jegyzet feltöltése metaadatokkal, hírfolyam-listázás és alap keresés. A funkciók az 1. fejezetben rögzített modulokra épülnek (Hitelesítés; Jegyzet feltöltése; Keresés; Hírfolyam) és ezek PoC-szintű, összekapcsolt működését demonstrálják.

1.3.1. Funkcionális követelmények:

- Regisztráció / Bejelentkezés: email + jelszó, valamint harmadikfeles bejelentkezés opció megjelenítése; sikeres hitelesítés után JWT alapú munkamenet.
- Jegyzet feltöltése: cím, leírás, címkék megadása; fájl csatolása; láthatóság beállítás (privát/nyilvános). Elfogadási kritérium: feltöltés után a jegyzet megjelenik a hírfolyamban a tulajdonos számára.
- Keresés / Szűrés: kulcsszó alapú keresés címben és leírásban; címke szerinti szűrés. Elfogadási kritérium: csak a szűrési feltételeknek megfelelő jegyzetek listázódnak.
- Hírfolyam: kártyás lista nézet; alap metaadatok (cím, kivonat, címkék). Elfogadási kritérium: új feltöltés után a lista frissíthető és megjeleníti az elemet.

1.3.2. Nem-funkcionális követelmények:

- Használhatóság: konzisztens gomb- és lépéssorrend; értesítések siker/hiba állapotokról.
- Biztonság: védett végpontok JWT-vel, szerepkör-előkészítés a későbbi bővítéshez.
- Bővíthetőség: moduláris szerkezet a további 1.2.x modulokhoz (Gyűjtemények, Értékelés stb.).

1.4. Feladat-nyilvántartó és csapatmunka

A csapat GitHub Projects kanban táblát használ („To do / In progress / Review / Done”), az egyes kártyákhoz issue és branch tartozik; merge csak jóváhagyott PR-rel történhet. Ez illeszkedik a dokumentumban rögzített front–back együttműködéshez és az inkrementális fejlesztéshez.

1.5. Git kipróbálása, ágak, egyesítés

- Branch-stratégia: main (kiadások), develop (integráció), feature/<szám>-<leírás>.
- PR-szabály: legalább 1 review; „build & format” ellenőrzés; konfliktusmentes merge.
- Konvenciók: commit üzenet „type: rövid leírás”, issue hivatkozás (#id).

A fenti folyamat biztosítja a visszakövethetőséget és a párhuzamos fejlesztést.

1.6. Egyszerű GUI (PoC képernyők)

- Menü / navigáció: alsó navigációs sáv (Feed, Keresés, Feltöltés, Profil).
- Vezérlők: első iterációban gombok, panel/kártya, jelölőnégyzet (címkeszűrő), lista „grid” elrendezés helyett egyszerű listanézet.
- Stílus: modern, kék hangsúlyok, konzisztens gomb-elhelyezés.

Ez a „legkisebb még értelmes” UI, amely már demonstrálja a fő folyamatokat.

2. Munka felosztása

A platform fejlesztési folyamata két fő csapat – a Front-End és a Back-End – között oszlik meg, amelyek mindegyike a rendszer különböző összetevőiért felelős. Ez a strukturált elosztás biztosítja a párhuzamos fejlesztést, a hatékony integrációt és a feladatok egyértelmű tulajdonjogát a projekt életciklusa során. A projekt becsült időtartama körülbelül tizenegy hét volt, amely több fázisra oszlott, amely során fokozatosan kiépítettük és integráltuk a rendszer funkcionalitását.

2.1. Front-end fejlesztés

A front-end fejlesztés a felhasználói felületre, a felhasználói élményre és a platform általános interakciós folyamatára összpontosít. Csapatunkból Hajtó Kornél felelős a vizuális és interaktív komponensek tervezéséért és megvalósításáért, biztosítva, hogy a felület intuitív, hozzáférhető és reszponzív legyen a különböző eszközökön.

2.1.1. Fejlesztési időszak első fele

Felhasználói felület tervezése és kivitelezése A kezdeti fázisban az elsődleges cél egy egységes és felhasználóbarát felület megtervezése. Ez magában foglalja az elrendezés, a színséma, a tipográfia és a navigációs struktúra meghatározását. Drótvázakat és mockupokat készítettünk a felhasználói folyamatok és a felületelemek, például a hírfolyam, a profiloldalak, a jegyzetfeltöltési űrlapok és a keresési nézetek vizualizálására. A megvalósítás megkezdése előtt a korai felhasználói visszajelzések is összegyűjthetők a terv finomítása érdekében.

2.1.2. Fejlesztési időszak második fele

Miután a felhasználói felület terve véglegesült, a fejlesztés az interaktív komponensek megvalósítására és a back-end API-kkal való integrálására irányul. A feladatok közé tartozik az újrafelhasználható komponensek fejlesztése, a zökkenőmentes állapotkezelés biztosítása és az aszinkron adatkommunikáció kezelése a szerverrel. Kornél szorosan együttműködik a backend csapattal az integráció tesztelésében és finomításában, garantálva, hogy minden felhasználó számára elérhető funkció – például a regisztráció, a jegyzetek feltöltése, a keresés és a hírfolyamok frissítése – zökkenőmentesen működjön. A fázis végére a frontend egy

kifinomult és intuitív élményt nyújt, támogatva az összes kulcsfontosságú platformfunkciót, és lehetővé téve a valós idejű interakciót a rendszer adataival.

2.2. Back-end fejlesztés

Csapatunkból Bojsza Réka és Fekete Gergő felelősek az alkalmazás backend infrastruktúrájáért, adatbázis-kezeléséért és alapvető logikájáért. Munkájuk biztosítja, hogy a platform megbízhatóan működjön, biztonságosan kezelje az adatokat, és jól strukturált API-kat biztosítson a frontend integrációhoz. A backend fejlesztése több inkrementális szakaszra oszlik, amelyek mindegyike kritikus funkciókat vezet be a rendszerbe.

2.2.1. Fejlesztési időszak első negyede

Kezdeti beállítás és alapvető funkciók Az első szakasz a szerverkörnyezet, az adatbázis-séma és a nélkülözhetetlen hitelesítési mechanizmusok létrehozására összpontosít. Ezek az alapvető elemek teremtik meg az alapot a felhasználói interakcióhoz és a tartalommegosztáshoz a platformon keresztül. A legfontosabb eredmények a következők:

- Felhasználói hitelesítés: Bejelentkezés, regisztráció és felhasználói beállítások kezelésének megvalósítása.
- Jegyzetkezelés: Lehetővé teszi a felhasználók számára, hogy jegyzeteket töltsenek fel és tároljanak biztonságosan az adatbázisban.
- Felhasználókövető rendszer: Lehetővé teszi a felhasználók számára, hogy másokat kövessenek, és megtekinthessék a megosztott tartalmakat a hírfolyamukban.
- Hírfolyammodul: Dinamikus hírfolyam létrehozása, amely összesíti és megjeleníti a követett felhasználók frissítéseit.

2.2.2. A fejlesztési időszak második negyede

Speciális funkciók és közösségi funkciók Ebben a fázisban a csapat kibővíti a platform képességeit a mélyebb interakció és együttműködés támogatása érdekében. Ez a szakasz gazdagítja a felhasználói élményt a felfedezhetőség, a visszajelzés és a strukturált tanulás elősegítésével. A legfontosabb fejlesztések a következők:

- Keresési modul: Alapvető kulcsszó- és speciális szűrőalapú keresés megvalósítása a felhasználók számára a releváns jegyzetek könnyű megtalálásához.
- Értékelő rendszer: Lehetővé teszi a felhasználók számára a jegyzetek értékelését, áttekintését és visszajelzését.
- Gyűjtemények: Lehetővé teszi a felhasználók számára, hogy saját és mások jegyzeteit tematikus gyűjteményekbe csoportosítsák és rendszerezzék a könnyebb hivatkozás érdekében.

2.2.3. A fejlesztési időszak harmadik negyede

Személyre szabás és felhasználói adatkezelés Ezen a ponton a hangsúly a személyre szabás és a felhasználói kontroll javítására helyeződik át. Ezek a funkciók erősítik a felhasználók adatvédelmét, és tulajdonjogot biztosítanak a tartalom és az interakciók felett.A legfontosabb eredmények a következők:

- Előzmények megtekintése: Lehetővé teszi a felhasználók számára, hogy nyomon kövessék korábbi tevékenységeiket, például a megtekintett vagy letöltött jegyzeteket.
- Privát profil: Minden felhasználó számára testreszabható profilt biztosít, ahol a láthatósági beállítások és a személyes adatok kezelhetők.

2.2.4. A fejlesztési időszak utolsó negyede

Együttműködés, megosztás és ajánlások A fejlesztés utolsó fázisa olyan együttműködési és közösségi funkciókat vezet be, amelyek ösztönzik az elköteleződést és a tartalomcserét. A fázis végére a platform érett és funkciókban teljes állapotot ér el, egyetlen összefüggő rendszerbe integrálva a közösségi, együttműködési és oktatási eszközöket. A legfontosabb fejlesztések a következők:

- Jegyzetkérés modul: A felhasználók konkrét jegyzeteket vagy témákat kérhetnek másoktól.
- Kérdéskészlet/vizsgaanyag modul: Lehetővé teszi strukturált teszt- vagy ismétlőanyagok létrehozását és megosztását.
- Külső megosztás: Lehetővé teszi a tartalom külső alkalmazásokon keresztüli megosztását a jobb hozzáférhetőség érdekében.

- „For-You Feed”: Egy ajánlási rendszer, amely a felhasználók érdeklődési köre és a követett témák alapján válogatja össze a jegyzeteket és a tartalmakat.
- Jegyzet letöltése: Offline hozzáférést biztosít a jegyzetekhez a biztonságos letöltési funkció engedélyezésével.

3. Hasonló szoftverek

3.1. StuDocu

1. Ábra: StuDocu logója

A StuDocu kurzus- és egyetem-szintű jegyzetmegosztásra épít: hallgatók kitudják választani az egyetemüket, töltenek fel jegyzeteket, vázlatokat, diasorokat, melyek címkézve és kurzusokhoz kötve böngészhetők. A hozzáférés vegyes modellű: „feltöltésért feloldás” és prémium előfizetés is van. Az elmúlt időszakban AI-eszközökkel bővült (pl. jegyzetekből összefoglalók, kvíz-/kártya-generálás, előadás-rögzítésből automatikus transzkript és kivonat), ami a sima fájlfeltöltést aktív tanulási tartalomra alakítja. Előnye a hatalmas, kurzus-szintű kínálat és az AI-kiegészítések; hátránya a vegyes minőség és a paywall.

A mi projektünk és a StuDocu között a legnagyobb különbség a felhasználói élmény és a közösségi funkciók mélységében rejlik. Míg a StuDocu inkább egy passzív jegyzetmegosztó piactér, ahol a fő cél a fájlok elérése, addig a mi programunk aktív interakciót ösztönöz a felhasználók között, például értékeléssel, jegyzetkéréssel és hírfolyammal. A mi rendszerünk nem korlátozza a hozzáférést paywall mögé, így nyíltabb tudásmegosztást tesz lehetővé. Ugyanakkor hasonlóság, hogy mindkettő támogatja a jegyzetek feltöltését, rendszerezését és címkézését, valamint a kurzusokhoz kötött struktúrát.

3.2. Course Hero

2. Ábra: Course Hero logója

A Course Hero szintén dokumentum- és kurzusalapú könyvtárat kínál (jegyzetek, feladatok, megoldások), melyet 24/7 tutoros Q&A és AI-funkciók egészítenek ki (pl. „AI Chat with PDF”, személyre szabott magyarázatok). A hozzáférési modell itt is kombinált: feltöltéssel „unlock” kreditet szerezhetsz, emellett előfizetés és szakértői segítség adott. Erősség a széles tananyag-kínálat és a többféle tanulási mód; kockázat a fizetős korlátok és a közösségi tartalom minőségének szórása. A kurzus-oldalak és a „feltöltésért feloldás” mechanika jó mintát ad az ösztönző rendszerhez.

A Course Hero-hoz képest a mi programunk közösségibb megközelítést alkalmaz, és nem a kreditrendszerre vagy előfizetésre épít, hanem a közös tudásmegosztást helyezi előtérbe. Míg a Course Hero főként dokumentumtár és szakértői válaszokra épülő szolgáltatás, addig a mi projektünk célja a tanulók és oktatók közötti kapcsolat erősítése valós idejű interakciókkal és jegyzetkérésekkel. Mindkettő támogatja a fájlmegosztást és a kurzusalapú keresést, azonban a mi rendszerünk átláthatóbb, személyre szabottabb hírfolyamot biztosít, és nem alkalmaz fizetős korlátokat.

3.3. StudySmarter

A StudySmarter (Vaia) „all-in-one” tanulóapp: jegyzetelés, megosztás, flashcardok, kvízek, tanulási tervező és erős AI-támogatás. Kiemelt funkció, hogy feltöltött anyagokból kártyák és rövid kivonatok készülnek, a tanulást pedig SRS és statisztikák segítik. A közösségi kártyapaklik megosztása és böngészése gyors tudásépítést ad; a fókusz itt kevésbé nyílt „jegyzetpiactér”, inkább személyes tanulófelület, megosztható csomagokkal. Előnye a hatékony tanulásszervezés és a mobil+web jelenlét; kompromisszum, hogy a dokumentum-centrikus kurzus-katalógus kevésbé hangsúlyos.

3. Ábra: StudySmarter logója

A StudySmarterhez hasonlóan a mi projektünk is törekszik a tanulási folyamat támogatására és a jegyzetek rendszerezésére, de a megközelítés eltér. Míg a StudySmarter inkább egy egyéni tanulási platform (kártyákkal, statisztikákkal, tanulási tervekkel), a mi programunk a közösségi jegyzetmegosztást és az együttműködést helyezi középpontba. Azonos vonás, hogy mindkettő lehetővé teszi a tartalom strukturált kezelését és a személyre szabott szűrést, azonban a mi rendszerünk nagyobb hangsúlyt fektet a felhasználók közötti kommunikációra és a közös tudásépítésre.

3.4. GoConqr

A GoConqr nem csak jegyzetmegosztó, hanem komplett tananyagszerkesztő ökoszisztéma: jegyzetek, gondolattérképek, kártyák, kvízek és tanulási tervek készíthetők és

csoportokban megoszthatók. A vizuális tanulást erősíti (pl. mind map → jegyzet konverzió), a megosztás pedig rugalmas (publikus, csoport, beágyazás). Ez inkább „eszköz-csomag” közösségi tudásépítéshez, nem klasszikus egyetemi kurzus-katalógus, de remek referencia a kollekciók, jogosultsági szintek és beágyazhatóság tervezéséhez a saját platformotokban.

4. Ábra: GoConqr logója

A GoConqr és a mi projektünk közötti hasonlóság főként az interaktív és vizuális tanulást segítő elemekben mutatkozik meg. Mindkét rendszer támogatja a gondolattérképeket, kvízeket és tananyagok rendszerezését, de a GoConqr inkább tananyagszerkesztő eszközként működik, míg a mi programunk közösségi platformot épít, ahol a jegyzetmegosztás és a hírfolyam a központi elem. A mi rendszerünk célja nemcsak a tanulás támogatása, hanem a diákok és oktatók közti aktív kapcsolat kialakítása is, így az interakciók szintje magasabb és személyesebb.

3.5. Microsoft OneNote Class Notebook

Az OneNote Class Notebook oktatási környezetre szabott megosztás: minden diáknak személyes szekció, a tanárnak tartalomkönyvtár és közös kollaborációs tér. Multimodális jegyzetelés (szöveg, tinta, kép, fájl), kiosztás/gyűjtés, értékelés és Teams-integráció teszi gördülékennyé a tanórai munkafolyamatot. Bár nem nyílt „piactér”, a kurzus-struktúra, hozzáférés-kezelés és a diák-tanár visszacsatolás mintái közvetlenül hasznosíthatók egy jegyzetmegosztó platform felhasználói jogosultság- és munkafolyamat-tervében.

5. Ábra: OneNote logója

A Microsoft OneNote Class Notebook kifejezetten oktatási környezetekre szabott, tanár-diák struktúrában működik, míg a mi projektünk ennél nyitottabb, közösségi alapokra épít. Mindkét rendszer támogatja a multimodális jegyzetelést, a fájlok csatolását és a tananyagok megosztását, de a mi megoldásunk a szélesebb felhasználói közösségre nem pedig csak az osztályszintű használatra fókuszál. A mi programunknál a hangsúly a jegyzetek értékelésén, a visszajelzéseken és a személyre szabott hírfolyamon van, így a tanulás élménye kevésbé formális, de sokkal inkább közösségi jellegű.

3.6. Notion

6. Ábra: Notion logója

A Notion egy moduláris, adatbázis-alapú munkaterület: oldalakat és kapcsolt adatbázisokat hozhatsz létre (táblázat/board/lista/naptár), sablonokkal és relációkkal rendszerezve, valós idejű együttműködéssel, kommentekkel és finomhangolható jogosultságokkal (vendégek, csak-nézet/szerkesztés, publikus link). Oktatási csomag és „School/Class" sablonok segítik a kurzus- és jegyzetstruktúrák gyors felépítését, a tartalmak pedig egy kattintással megoszthatók vagy akár weboldalként publikálhatók.

A mi projektünkkel a hasonlóság ott erős, hogy a Notionban könnyen kialakítható kurzus- vagy tantárgyadatbázis címkékkel (egyetem, félév, téma), szűrhető és személyre szabható nézetekkel („for you" jelleg), fájlcsatolással és beágyazásokkal (PDF, diák). A szerepkörök és megosztási szintek jó mintát adnak a hozzáférés- és moderációs modellhez, a kommentelés és verziókövetés pedig a közösségi tanulást támogatja. Piactér ugyan nem, de az architektúra és a megosztási logika közvetlenül adaptálható a mi rendszerünkben.

4. Felhasznált technológiák

4.1. Flutter

A Flutter egy nyílt forráskódú UI-fejlesztési keretrendszer, amelyet a Google fejlesztett ki, és amely lehetővé teszi natív alkalmazások létrehozását mobil, web és asztali platformokra egyetlen közös kódbázisból. A keretrendszer a Dart programnyelvet használja, amely szintén a Google fejlesztése, és amelyet kifejezetten a gyors fejlesztés és a kiváló teljesítmény elérésére terveztek.

A projekt során a Fluttert elsősorban a platform független, egységes felhasználói élmény biztosítása céljából választottuk. Mivel az alkalmazás várhatóan különböző eszközökön (pl. Android, iOS, esetleg webes környezet) is elérhető lesz, kiemelten fontos szempont volt a reszponzív és konzisztens megjelenés, amelyet a Flutter beépített widget-rendszere és adaptív felületkezelése lehetővé tesz.

A fejlesztési folyamatban a Flutter előnyei közé tartozott a „hot reload” funkció, amely jelentősen gyorsította a tesztelést és a felhasználói felület iteratív finomítását. A deklaratív felületleírási modell lehetővé tette a UI-komponensek strukturált, moduláris fejlesztését, ami megkönnyítette az alkalmazás komplex elemeinek – például a hírfolyam, keresőfelület vagy regisztrációs folyamat – kialakítását.

A technológia további előnye, hogy számos közösségi csomag és plugin áll rendelkezésre hozzá, amelyek megkönnyítik az olyan funkciók integrálását, mint a HTTP kérések (pl. Dio), a fájlkezelés vagy az állapotmenedzsment (pl. Riverpod, Provider). Ezen komponensek révén jelentős fejlesztési idő takarítható meg, miközben a megbízhatóság és karbantarthatóság is magas szinten tartható.

Összességében a Flutter választása lehetővé tette egy olyan modern, skálázható és esztétikailag egységes felület kialakítását, amely megfelel mind a technikai követelményeknek, mind a felhasználói elvárásoknak.

4.2. C#

A C# egy objektumorientált programozási nyelv, amelyet a Microsoft fejlesztett ki a .NET keretrendszer részeként. A nyelv széles körben használható asztali, webes és

mobilalkalmazások fejlesztésére. Erősségei közé tartozik a jól strukturált szintaxis, a típusbiztonság és a fejlett nyelvi funkciók, mint például a LINQ és az aszinkron programozási lehetőségek.

A projektben a C# nyelvet elsősorban a háttérlogika megvalósítására használtuk. A Visual Studio fejlesztőkörnyezet segítségével gyorsan és hatékonyan lehetett dolgozni, a beépített hibakeresési és kódformázási funkciók nagyban támogatták a fejlesztési folyamatot.

4.3. Git

A Git egy nyílt forráskódú verziókezelő rendszer, amely lehetővé teszi a forráskód változásainak nyomon követését. Segítségével több fejlesztő is hatékonyan tud együtt dolgozni egy projekten anélkül, hogy felülírnák egymás munkáját. A Git támogatja az ágazatok (branch-ek) használatát, így biztonságosan lehet párhuzamosan fejleszteni új funkciókat.

A projekt során a GitHub szolgáltatást használtuk a kód tárolására és verziókezelésére. Ez biztosította, hogy a kód mindig visszakereshető, archiválható legyen, illetve támogatást nyújtott a csapatmunka során felmerülő feladatok kezelésére is.

4.4. Docker

A Docker egységes futtatókörnyezetet biztosít az alkalmazás komponenseinek (backend API, relációs adatbázis). Előnye, hogy a fejlesztői és tesztkörnyezet reprodukálható, a beállítások verziózhatók (compose), és gyors a beüzemelés. A PoC-ban egy API- és egy DB-konténer elegendő, perzisztens volume-okkal. Ez támogatja a .NET backend és a Flutter kliens gyors iterációját.

4.5. GitHub (verziókezelés és együttműködés)

A forráskód GitHubon kerül tárolásra; Issues + Projects használatával kanban-alapú követést alkalmazunk, Pull Requestekkel történik az integráció és a kódáttekintés. A Git használatát a projekt során már rögzítettük; itt a GitHub adja hozzá a feladatkezelést és a csapatmunkát támogató folyamatokat.

5. Felhasználói felület

Egy alkalmazás felülete határozza meg, mennyire élvezetes és hatékony a használata. Egy rosszul felépített, zavaros felület frusztrálóva és használhatatlanná teheti a legjobb funkciókat is, ezért alapvető fontosságúnak tartottuk, hogy a NoteSharing felületére nagy figyelmet fordítsunk. Egy ideális felület elősegíti a felhasználót, intuitív, és minimális erőfeszítéssel navigálható.

Az alkalmazás harmóniáját egy letisztult, modern megjelenés és egy egyedi színvilág kiválasztásával kezdtük. A jegyzethez hasonló háttereket és fehér kártyákat éltető design mellett a kék szín több árnyalata lett a hangsúlyos szín. Ezek az elemek érzékeltetik az alkalmazás kreatív és közösségi jellegét/jellegzetességét. A kék szín visszaköszön a legfontosabb gombokon, és elemeken, irányítva a felhasználó figyelmét.

5.1. Regisztráció

Fontos, hogy a felhasználó számára mindig egyértelmű legyen, hol tart a regisztrációs folyamatban, mit kell még tenni a továbblépéshez. Ezt vizuális hierarchiával, konzisztens gomb elrendezéssel és egyértelmű címkékkel értük el.

A NoteSharing regisztrációs folyamata egy intelligens, szerep alapú rendszer, amely a felhasználó típusától függően más-más információkat gyűjt. A cél, hogy minden felhasználó a saját profiljához és igényeihez leginkább illeszkedő tartalmakat kapjon.

Minden felhasználó számára kötelezően megadantó adatok közé tartozik a felhasználónév, teljes név, email cím és jelszó. Azonban vannak adatok amelyeket csak az adott szerepkörnek/szerepköröknek kell megadnia. Ilyenek például, hogy a hallgatóknak meg kell adnia évfolyamot, intézményt, és tanult tárgyakat, míg az oktatóknak intézményt, és oktatott tárgyakat kell megadnia a regisztrációs folyamat során.

A regisztráció véglegesítése előtti utolsó lépés a személyes preferenciák megadása. Melyek a következők: saját intézmény előnyben részesítése, oktatói jegyzetek előnyben részesítése, saját jegyzetek privát láthatósága, magasan értékelt jegyzetek előnyben részesítése, követett felhasználók előnyben részesítése

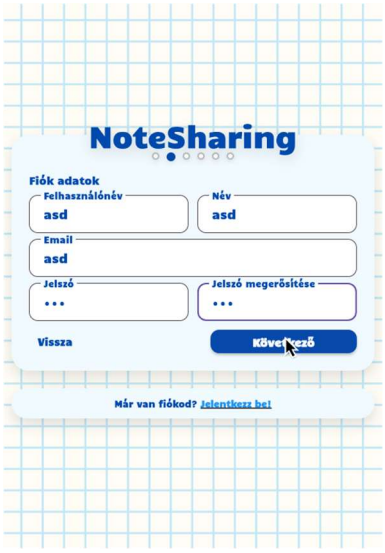
A preferencia beállítások a későbbiekben módosíthatóak lesznek a beállításoknál.

Az alábbi képen a regisztrációs folyamat egyik képernyője látható, amely jól reprezentálja a design alapelveinket.

A design-nak nem csak esztétikai, hanem praktikus feladata is van. A “Vissza” és “Következő” gombok konzisztens elhelyezése minden képernyő alján biztosítja, hogy a felhasználó soha ne vesszen el. Az információkat táblázatos formában, vagy rögzített listákban is megjelenítjük (pl. “Összegzés” képernyő), ami áttekinthetőbbé és gyorsan feldolgozhatóvá teszi az adatokat.

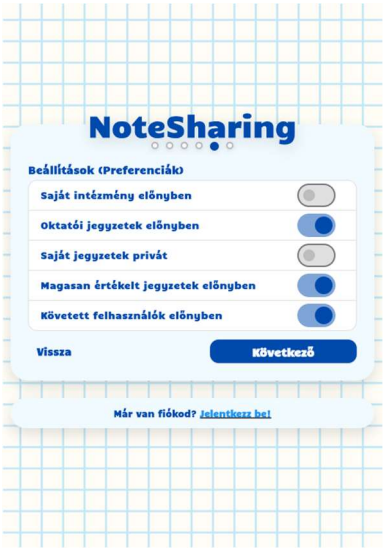
A regisztrációs folyamat lépéseit az alábbi képernyőképeken szemléltetjük:

7. Ábra: Regisztráció első lépése



8. Ábra: Regisztráció második lépése

9. Ábra: A regisztráció harmadik lépése



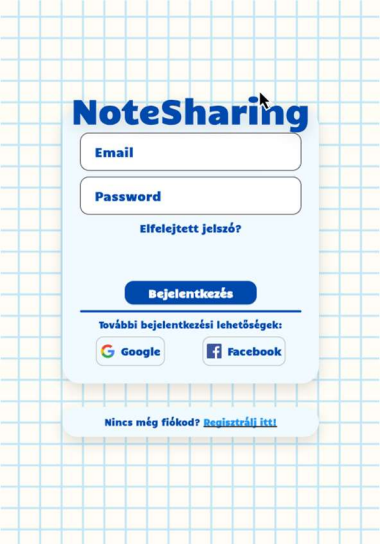
11. Ábra: A regisztráció negyedik lépése

10. Ábra: A regisztráció utolsó lépése

5.2. Bejelentkezés

A bejelentkezési felületet a maximális egyszerűség és biztonság jellemzi. A felhasználók két fajta lehetőségük van, az egyik a hagyományos bejelentkezésre email cím és jelszó megadásával, a másik pedig a gyorsabb, külső szolgáltatásos bejelentkezésre Google vagy Facebook fiókkal.

A harmadik féltől származó bejelentkezési opciók jól látható, ismert brand színekkel és ikonokkal megjelenített gombokon érhetőek el, ami nemcsak felismerhetővé, hanem megbízhatóva is teszi ezek használatát.

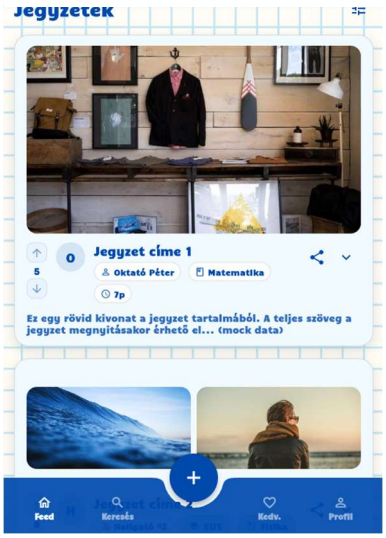


12. Ábra: A bejelentkezési felület

5.3. Feed

Az alkalmazás szíve és központja a jegyzetek idővonala, azaz a Feed kijelző. Itt a feület célja, hogy a lehető leghatékonyabban jelenítse meg a rengeteg információt. Minden jegyzet egy jól elválasztott kártyaként jelenik meg, amely tartalmazza a jegyzet címét és egy rövid kivonat a tartalomról, illetve ikonokkal és címkékkel ellátott meta-információk (pl. oktatói jegyzet, hallgatói jegyzet, tárgy neve, feltöltő neve, stb.)

Ez a strukturált megjelenés lehetővé teszi a felhasználók számára, hogy gyorsan áttekintsék és szűrjék a számukra releváns tartalmakat.



13. Ábra: A feed felület

5.4. Értesítések

A felhasználói élmény egyik legfontosabb eleme a folyamatos és egyértelmű visszajelzés. Legyen akár csak arról szó, hogy a bejelentkezés sikeres volt, vagy hogy esetleg nem töltött ki minden kötelező mezőt. Erre épül a NoteSharing értesítési rendszere. A rendszer célja, hogy a felhasználót mindig informáltan

14. Ábra: Az értesítések

tartsa a vele kapcsolatos eseményekről. Az értesítések könnyen felismerhető kék/zöld/piros színekkel és dedikált ikonokkal jelennek meg.

6. Backend

6.1. Authorizáció

Az AuthController kezeli az alkalmazáson belüli összes hitelesítéssel kapcsolatos folyamatot, beleértve a felhasználói regisztrációt, a bejelentkezést, a token frissítését és a hitelesítés tesztelését. Lehetővé teszi az új felhasználók számára a regisztrációt hitelesítő adataik megadásával, egy bejelentkezési végpontot biztosít, amely ellenőrzi a felhasználói hitelesítő adatokat és JWT tokeneket bocsát ki, valamint magában foglalja a hozzáférési tokenek frissítésének funkcióját érvényes frissítési token használatával a munkamenet folytonosságának fenntartása érdekében. Ezenkívül egy védett végpontot kínál, amely a hitelesítést a megadott JWT érvényességének ellenőrzésével és a tokenből kinyert felhasználói adatok visszaadásával teszteli. A hitelesítési teszt kivételével minden végpont nyilvánosan elérhető előzetes hitelesítés nélkül.

6.2. Intézmények

Az InstitutionController kezeli az intézményekkel kapcsolatos összes műveletet, beleértve az intézményi rekordok létrehozását, frissítését, törlését és lekérését. Lehetővé teszi új intézmények létrehozását a releváns adatok adatátviteli objektumon keresztüli fogadásával, támogatja a meglévő intézmények frissítését módosított adatokkal, és funkciót biztosít az intézmények törléséhez inaktívként megjelöléssel a végleges eltávolítás helyett. A vezérlő lehetővé teszi egy adott intézmény egyedi azonosítója alapján történő vagy a rendszerben lévő összes aktív intézmény lekérését is. Minden művelet a munkaminta egységén keresztül kommunikál az intézményi adattárral, biztosítva a konzisztens adatkezelést és a tranzakciók integritását.

6.3. Preferenciák

A PreferenceController kezeli a felhasználói beállítások kezelésével kapcsolatos összes műveletet. Végpontokat biztosít új beállítások létrehozásához, meglévők frissítéséhez, beállítások törléséhez egyedi azonosítóik alapján, valamint adott preferenciarekordok

lekéréséhez. Minden művelet a munkaminta alapján kommunikál a preferenciatárral a hatékony és konzisztens adatkezelés biztosítása érdekében. A vezérlő strukturált válaszokat ad vissza, amelyek a sikert vagy a sikertelenséget jelzik, lehetővé téve a felhasználók vagy a rendszerek számára a preferenciaadatok megbízható és zökkenőmentes kezelését.

6.4. Tantárgyak

A SubjectController kezeli az alkalmazáson belüli összes, a tárgyhoz kapcsolódó műveletet. Funkciókat biztosít új tárgyak létrehozásához, meglévő tárgyrekordok frissítéséhez, tárgyak törléséhez inaktívként való megjelöléssel, valamint adott tárgyak egyedi azonosítóival vagy a rendszerben elérhető összes tárgy lekéréséhez. Minden művelet a munkamintán keresztül kommunikál a tárgyadattárral, hogy biztosítsa a megfelelő adatkonzisztenciát és a tranzakciók megbízhatóságát. A vezérlő strukturált API-válaszokat ad vissza, amelyek jelzik a művelet sikerességét vagy sikertelenségét, támogatva a zökkenőmentes tárgykezelést az alkalmazásban.

6.5. Felhasználók

A UserController kezeli az összes felhasználóval kapcsolatos műveletet, beleértve a felhasználói adatok frissítését, a felhasználók törlését, a felhasználók azonosító vagy felhasználónév szerinti lekérését, valamint a felhasználói kapcsolatok, például más felhasználók követésének kezelését. Lehetővé teszi a felhasználói információk módosítását részleges frissítésekkel, biztosítva, hogy csak a megadott mezők változzanak, és lehetővé teszi a felhasználók törlését az egyedi azonosítójuk használatával. A vezérlő támogatja a felhasználói adatok lekérését egyedi azonosító vagy felhasználónév alapján, és funkciót biztosít két felhasználó közötti egyoldalú követési kapcsolat létrehozásához. Minden művelet a munkaegység mintáján keresztül kommunikál a felhasználói adattárral, biztosítva a hatékony adatkezelést és az alkalmazás konzisztens viselkedését, miközben minden végpont hitelesítés nélkül elérhető.

Ábrajegyzék