**Tartalomjegyzék**

**Tartalomjegyzék…………………….…………………………………..1**

**Lakásfoglaló oldal……………..………………………………………2**

**HTML………………………………………………………………………..4**

**CSS………………………………………………………………………….5**

**Javascript……………………………………………………………….…7**

**React………………………………………..………………………………8**

**React működés………………………..………………………………..9**

**Bootstrap…………….……………………………………………………10**

**Trello………………………………………………………………………..11**

**Github………………………………………………………………………13**

**Alaprajz......…………………………………………………………….…14**

**Tesztelés………………………..…………………………………………14**

**Források……………………………………………………………………18**

**Mellékletek………………………………………………………….……19**

**Lakásfoglaló oldal**

Egy **lakásfoglaló oldal** célja az, hogy megkönnyítse a lakások és ingatlanok bérlését vagy megvásárlását keresők és az ingatlanokat kínálók számára. Ezek az oldalak egyfajta online platformként működnek, ahol a kereslet és a kínálat találkozik. A következőkben részletesen bemutatom, hogy mire jó egy lakásfoglaló oldal, milyen funkciókat kínál, és miért vált elengedhetetlenné a modern ingatlanpiacon.

**Miért választottuk ezt a feladatot?**

Azért választottuk a feladatot,hogy egy lakásfoglaló oldalon keresztül biztosítsuk azt,hogy az emberek akik bérelni,venni akarnak lakást,azok tudjanak egy oldalon keresztül az ajánlókkal kapcsolatba kerülni és biztonságos körülmények között.Mivel tapasztalataink szerint az átlag vásárlónak nagyon nehéz helyzete van a mai ingatlanpiacon és ha talál is ingatlant,akkor is sok megtévesztésnek esik áldozatául.Mi ezekre a problémákra hozzuk el a weboldalunkat,ahol a vásárlás sokkal kényelmesebb,hatékonyabb,olcsóbb és biztonságosabb mint a normál körülmények közötti ingatlankereséskor.

**Az alapvető célunk: kereslet és kínálat összekötése**

A lakásfoglaló oldalak alapvető funkciója, hogy kapcsolatot teremtsünk a lakást keresők (bérlők vagy vásárlók) és az ingatlant kínálók (tulajdonosok, bérbeadók, ingatlanirodák) között. Ez egy egyszerű, gyors és hatékony megoldást biztosít, amely időt takarít meg mindkét fél számára, és minimalizálja a kommunikációs akadályokat.

**Mire jó egy lakásfoglaló oldal?**

**Széles választékot kínál az ingatlanokból**

Egy lakásfoglaló oldal lehetőséget ad arra, hogy a felhasználók egyszerre több száz vagy akár több ezer ingatlan között böngésszenek. Ezek az oldalak szűrőkkel és kategóriákkal könnyítik meg a keresést, például:

**Elhelyezkedés:** Város, kerület, utca vagy akár földrajzi régió.

**Árkategória:** Min/max ár megadása, napi/heti/havi bérleti díj szerint.

**Lakás jellemzői:** Méret, szobák száma, erkély, garázs, bútorozottság, stb.

**Ingatlan típusa:** Kiadó vagy eladó lakás, családi ház, nyaraló, garzon, stb.

**Kényelmes és gyors foglalási folyamat**

A lakásfoglaló oldalak egyik legnagyobb előnye, hogy a teljes folyamat – a lakás kiválasztásától kezdve a foglalásig – online lebonyolítható. A felhasználó:

Böngészhet a lakások között.

Megtekintheti az elérhető időpontokat.

Közvetlenül foglalhat időpontot a lakás megtekintésére vagy bérlésére.

Előlegfizetést vagy foglalót küldhet online, ami biztosítja a tranzakció gyors lezárását.

**Átlátható információk nyújtása**

Egy jól működő lakásfoglaló oldal részletes információt nyújt az ingatlanokról. Ide tartoznak:

Fényképek és videók az ingatlan állapotáról.

Pontos leírás az ingatlan jellemzőiről.

Elérhető közszolgáltatások (pl. közlekedés, boltok, iskolák).

Átlátható árak és extra költségek (pl. közös költség, rezsi).

**HTML**

A HTML a weboldalunk alapját képzi.A projektünk a HTML alapelveire épült.A **HTML** (angolul: *HyperText Markup Language*, „hiperszöveges jelölőnyelv”) egy HYPERLINK nyelv, melyet [weboldalak](https://hu.wikipedia.org/wiki/Weblap) készítéséhez fejlesztettek ki, és mára már internetes szabvánnyá vált a [W3C](https://hu.wikipedia.org/wiki/World_Wide_Web_Consortium) (World Wide Web Consortium) támogatásával. Az aktuális változata az 5, mely az [SGML](https://hu.wikipedia.org/wiki/Standard_Generalized_Markup_Language) általános [jelöl HYPERLINK](https://hu.wikipedia.org/wiki/Jel%C3%B6l%C5%91nyelv)  egy konkrét alkalmazása (azaz minden 5-ös HTML dokumentum egyben az SGML dokumentumszabványnak is meg kell hogy feleljen). Ezt tervek szerint lassan kiszorította volna az [XHTML](https://hu.wikipedia.org/wiki/HTML), amely a szintén SGML alapú [XML](https://hu.wikipedia.org/wiki/XML) leíró nyelven alapul, de még csak kevés weboldal használja.

**Használata**

**HTML** általában szöveges állományokban található meg olyan [számítógépeken](https://hu.wikipedia.org/wiki/Sz%C3%A1m%C3%ADt%C3%B3g%C3%A9p), melyek az internethez kapcsolódnak. Ezek az állományok tartalmazzák azokat a szimbólumokat, amelyek a megjelenítő programnak leírják, hogyan is kell megjeleníteni illetve feldolgozni az adott állomány tartalmát. Megjelenítő program lehet egy (angolul: web browser), aural böngésző (olyan, amelyik a felhasználónak felolvassa a megjelenítendő szöveget), [braille](https://hu.wikipedia.org/wiki/Braille-%C3%ADr%C3%A1s) olvasó, amely konvertálja a szöveget braille „formátumba.

**Szimbólumok, tartalom**

Négyfajta szimbólum ([leíró elem](https://hu.wikipedia.org/wiki/HTML-elem)) található meg a **HTML**-ben:

*strukturális* elemek, amelyek leírják az adott szöveg "célját" például <h1>A legnagyobb szöveg</h1> mint első szintű címsor (alcím).

*prezentációs* szimbólumok, amelyek leírják, hogy az adott szöveg hogyan nézzen ki: például <b>Vastag szöveg</b> **vastag** kinézetet eredményez. (Ez a forma azonban ma már *elavultnak* számít, helyette a [CSS](https://hu.wikipedia.org/wiki/CSS) használata javasolt, ugyanis a legújabb irányelv szerint szét kell választani a tartalmat (amit a HTML kódol) és a formát (amit [CSS](https://hu.wikipedia.org/wiki/CSS)-ben szokás kódolni), és ezáltal mindenki számára elérhetővé tehetőek a weblapok: felolvasó gépeknek, amit a gyengén látók használnak, szöveges böngészőknek, és nyomtatóbarát verziót is könnyebb készíteni.)

[*hiperszöveg*](https://hu.wikipedia.org/wiki/Hipersz%C3%B6veg)*(hypertext)* elemek, amelyek segítségével kapcsolat létesíthető a dokumentum egyes elemei és más dokumentumok között (például a <a href="https://hu.wikipedia.org/">Wikipedia</a> a [Wikipedia](https://hu.wikipedia.org/) szót mint egy kapcsolatot (angol szóval: link) a megadott [URL](https://hu.wikipedia.org/wiki/URL)-hez jeleníti meg.

**CSS**

A CCS a projektünkben a különböző kínézetéért felelős.Például a háttér a betűstílus,méret stb.A **CSS** (*Cascading Style Sheets*, magyarul: „lépcsőzetes stíluslapok”) a számítástechnikában egy [stílusleíró nyelv](https://hu.wikipedia.org/w/index.php?title=St%C3%ADlusle%C3%ADr%C3%B3_nyelv&action=edit&redlink=1), mely a [HTML](https://hu.wikipedia.org/wiki/HTML) vagy [XHTML](https://hu.wikipedia.org/wiki/XHTML) típusú strukturált dokumentumok megjelenését írja le. Ezenkívül használható bármilyen [XML](https://hu.wikipedia.org/wiki/XML) alapú dokumentum stílusának leírására is, mint például az [SVG](https://hu.wikipedia.org/wiki/SVG), [XUL](https://hu.wikipedia.org/wiki/XUL) stb.

**Áttekintés**

A CSS-t a [weblapok](https://hu.wikipedia.org/wiki/Weblap) szerkesztői és olvasói egyaránt használhatják, hogy átállítsák vele a lapok színét, betűtípusait, elrendezését, és más megjelenéshez kapcsolódó elemeit. A tervezése során a legfontosabb szempont az volt, hogy elkülönítsék a dokumentumok struktúráját (melyet HTML vagy egy hasonló leíró nyelvben lehet megadni) a dokumentum megjelenésétől (melyet CSS-sel lehet megadni). Az ilyen elkülönítésnek több haszna is van, egyrészt növeli a weblapok [használhatóságát](https://hu.wikipedia.org/wiki/Weblap-haszn%C3%A1lhat%C3%B3s%C3%A1g), rugalmasságát és a megjelenés kezelhetőségét, másrészt csökkenti a dokumentum tartalmi struktúrájának komplexitását. A CSS ugyancsak alkalmas arra, hogy a dokumentum stílusát a megjelenítési módszer függvényében adja meg, így elkülöníthető a dokumentum formája a képernyőn, nyomtatási lapon, hangos segítségével olvassa fel a weblapok szövegét), vagy [Braille](https://hu.wikipedia.org/wiki/Braille-%C3%ADr%C3%A1s)-készüléken megjelenítve.

A CSS használható XML fájlok megjelenítésére is, így a strukturált dokumentumokhoz teljes körű stílust lehet megadni, befolyásolva az elrendezését, színét, betűtípusait az erre alkalmas kliensekben.

A CSS használatának legfontosabb előnyei:

Több lap vagy akár egy teljes webhely stílusait egy helyen lehet tárolni, így gyorsan és könnyen frissíthető

Különböző felhasználókhoz különböző stílusokat lehet rendelni: például könnyen olvasható, nyomtatható stílusok

A dokumentum mérete és komplexitása csökken, mivel nem tartalmaz információkat a megjelenítéshez

A CSS egyszerű [szintaxissal](https://hu.wikipedia.org/wiki/Szintaxis_(programoz%C3%A1si_nyelvek)) rendelkezik, csak néhány angol nyelvű kulcsszót használ a stílusok tulajdonságaihoz. A stíluslap maga a stílust leíró **szabályok** sora. Minden szabályhoz tartozik egy **szelektor** és egy **deklarációs szakasz**. Ez utóbbi kapcsos zárójelek között pontosvesszővel elválasztott **deklarációkat** tartalmaz. A deklarációk

formája a következő: a **tulajdonság** neve, egy kettőspont, majd az adott tulajdonság **értéke**.

**Kaszkád, öröklődés és specifikusság**

A **kaszkád logikája** nem csupán azt jelenti, hogy több forrásból érkező stílusok összeolvadnak, hanem egy hierarchikus rendszerben – a böngésző alapértelmezett stílusaitól az inline deklarációkig – a stílusok egymásra hatását és érvényességi sorrendjét szabályozza.

**Specifikussági pontszámok:** Minden szelektorhoz rendelünk egy pontszámot, ami meghatározza, hogy melyik szabály "nyeri el" az adott elem stílusának meghatározását.

**Öröklődés:** Bizonyos tulajdonságok automatikusan öröklődnek az ős elemekből, míg másokat explicit módon kell definiálni.

**!important direktíva:** Ez az eszköz lehetővé teszi, hogy bizonyos stílusok felülírják a természetes kaszkád sorrendet, de ennek használata gondos mérlegelést igényel, mert a karbantarthatóság rovására mehet.

**A böngésző renderelő motorja és a CSSOM**

Amikor a böngésző betölti az oldalt, először a HTML-t egy **DOM (Document Object Model)** struktúrává alakítja, majd a CSS-t a **CSSOM (CSS Object Model)** formájában értelmezi.

**Reflow és repaint:** Az elemek méretének, pozíciójának vagy láthatóságának változásakor a böngésző újraszámolja a layoutot (reflow), majd újrarajzolja az érintett területeket (repaint). Ezek az algoritmusok optimalizálva vannak, de a túlzott stílusmódosítások, animációk vagy komplex szelektorok lelassíthatják az oldalt.

**Renderelési pipeline:** A CSSOM és a DOM összeolvadása után jön a kompozíciós fázis, ahol a böngésző rétegekre bontja az oldalt, és ezekből építi fel a végső megjelenést.

**JavaScript**

A **JavaScript** [programozási nyelv](https://hu.wikipedia.org/wiki/Programoz%C3%A1si_nyelv) egy [objektumorientált](https://hu.wikipedia.org/wiki/Objektumorient%C3%A1lt_programoz%C3%A1s), prototípus-alapú [szkriptnyelv](https://hu.wikipedia.org/wiki/Interpreteres_nyelvek), amelyet [weboldalakon](https://hu.wikipedia.org/wiki/Weblap) elterjedten használnak. Ebből fejlődött ki a [TypeScript](https://hu.wikipedia.org/w/index.php?title=TypeScript&action=edit&redlink=1), ami a JavaScript típusos változatának tekinthető.

**Története**

Eredetileg *Brendan Eich*, a Netscape Communications mérnöke fejlesztette ki; neve először *Mocha*, majd *LiveScript* volt, később „JavaScript” nevet kapott, és szintaxisa közelebb került a Sun Microsystems [Java](https://hu.wikipedia.org/wiki/Java_(programoz%C3%A1si_nyelv)) programozási nyelvéhez. A JavaScriptet először 1997–99 között szabványosította az [ECMA](https://hu.wikipedia.org/wiki/Ecma_International) „[ECMAScript](https://hu.wikipedia.org/wiki/ECMAScript)” néven. A jelenleg is érvényes szabvány az *ECMA-262 Edition 3* (1999. december), ami a *JavaScript 1.5*-nek felel meg. Ez a szabvány egyben [ISO](https://hu.wikipedia.org/wiki/Nemzetk%C3%B6zi_Szabv%C3%A1ny%C3%BCgyi_Szervezet) szabvány is.

**A Microsoft általi átvétele**

A [Microsoft](https://hu.wikipedia.org/wiki/Microsoft) 1995-ben kifejlesztette az [Internet Explorert](https://hu.wikipedia.org/wiki/Internet_Explorer), ami a Netscape-pel való böngészőháborúhoz vezetett. A Microsoft a [Netscape Navigator](https://hu.wikipedia.org/wiki/Netscape_Navigator) JavaScript-feldolgozójának forráskódja segítségével létrehozta a sajátját, a [JScriptet](https://hu.wikipedia.org/w/index.php?title=JScript&action=edit&redlink=1).

A JScript először 1996-ban jelent meg a CSS kezdeti támogatása és a HTML néhány kiegészítése mellett. E megvalósítások merőben különböztek a Navigatorban alkalmazottaktól, ami megnehezítette a fejlesztőknek, hogy a weblapjaik mindkét webböngészőben jól működjenek, és ami a „Netscape-en működik legjobban” és „Internet Exploreren működik legjobban” széles körű használatához vezetett sok éven át.

**Szerkesztés, előállítás**

A JavaScript kód vagy a [HTML](https://hu.wikipedia.org/wiki/HTML) fájlban, vagy külön (jellemzően .js kiterjesztésű) szövegfájlban van. Ezek a fájlok tetszőleges  programmal szerkeszthetőek.

**Futási környezete**

A JavaScript esetében a futási környezet jellemzően egy web, illetve annak [JavaScript-motorja](https://hu.wikipedia.org/wiki/JavaScript-motor).

JavaScript programok [Windows](https://hu.wikipedia.org/wiki/Windows) alapú környezetben futtathatók továbbá a [wscript.exe](https://hu.wikipedia.org/w/index.php?title=Wscript.exe&action=edit&redlink=1) és a [cscript.exe](https://hu.wikipedia.org/w/index.php?title=Cscript.exe&action=edit&redlink=1) segítségével is.

**React**

**React általános**

A **React** egy nyílt forráskódú JavaScript-könyvtár, amelyet a **Meta** (korábban Facebook) fejlesztett ki és tart karban. A Reactet főként felhasználói felületek (UI) építésére használják, különösen egyoldalas alkalmazásokhoz (Single Page Applications - SPA), ahol az interaktív elemek gyors és hatékony frissítése kiemelten fontos.

**Története és fejlődése**

**Kezdetek**: A Reactet 2013-ban nyílt forráskódúként adta ki a Facebook. Eredetileg belső használatra fejlesztették, hogy egyszerűsítsék és gyorsítsák az alkalmazásaik UI-fejlesztését.

**Folyamatos fejlődés**: Azóta is aktívan fejlesztik, és számos verziófrissítés történt, amelyek új funkciókat és optimalizációkat hoztak (pl. a **Hooks** bevezetése 2018-ban).

**Cross-platform fejlesztés**

A React nemcsak webalkalmazások fejlesztésére alkalmas, hanem más platformokon is használható:

**React Native**: Mobilalkalmazások készítésére (Android és iOS).

**React 360**: VR alkalmazások fejlesztésére.

**Electron + React**: Asztali alkalmazások létrehozására.

**Nagyvállalati használat**

A Reactet rengeteg nagyvállalat használja, többek között:Facebook (Meta) Instagram Netflix Airbnb Uber Tesla

**React működés**

**Virtual DOM**

A React nem közvetlenül a böngésző **DOM-jával** dolgozik, hanem egy könnyített, memória-alapú **virtuális DOM** másolatot hoz létre.

**Hogyan működik?**

Minden változtatás először a Virtual DOM-ban történik meg.

A React összehasonlítja (diffing) a változtatások előtti és utáni állapotot.

Csak azokat a részeket frissíti a valódi DOM-ban, amelyek ténylegesen változtak.

**Előny**: Gyorsabb frissítés és jobb teljesítmény.

**Egyirányú adatfolyam (Unidirectional Data Flow)**

A Reactben az adatok mindig **felülről lefelé** áramlanak, vagyis a szülő komponens adja át az adatokat a gyermek komponenseknek.

Az adatok átadására a **props** szolgál.

A gyermek komponens nem módosíthatja a kapott adatokat, csak megjelenítheti azokat.

**Előny**: Könnyen követhető adatkezelés, amely csökkenti a hibalehetőségeket.

**Komponens-alapú architektúra**

A React **UI-t** kisebb, független és újrahasználható építőelemekre, azaz **komponensekre** bontja.

**Hogyan működik?**

Minden komponens egy önálló logikával és megjelenítéssel rendelkező JavaScript-függvény vagy osztály.

A komponens **JSX** segítségével írja le, hogyan nézzen ki az adott rész.

**Hooks**

A React egyik forradalmi újítása a **hooks**, amelyek funkcionális komponensekben biztosítanak extra képességeket (pl. állapotkezelést, mellékhatások kezelését).

**Bootstrap**

A **Bootstrap** egy **nyílt forráskódú front-end keretrendszer**, amelyet arra terveztek, hogy egyszerűsítse a reszponzív és modern weboldalak, illetve webes alkalmazások készítését. A Bootstrap HTML, CSS és JavaScript alapú eszközöket biztosít, amelyekkel könnyedén kialakíthatók látványos és funkcionális felületek, anélkül hogy az alapoktól kellene felépíteni a dizájnt.

**Főbb jellemzők**

**Reszponzív design támogatása**

A Bootstrap segítségével az oldalak automatikusan alkalmazkodnak különböző képernyőméretekhez (mobil, tablet, asztali eszközök).

A rendszer az **oszlop-alapú rács (grid system)** elvén működik, amely lehetővé teszi a tartalom egyszerű elrendezését.

**Előre elkészített stílusok**

Tartalmaz előre definiált CSS osztályokat gombokhoz, űrlapokhoz, táblázatokhoz, navigációs menükhöz, tipográfiához stb.

Ez gyorsabbá teszi a fejlesztést, mivel nincs szükség minden elem egyedi stílusának manuális kialakítására.

**JavaScript funkciók**

A Bootstrap beépített JavaScript bővítményei (pl. modális ablakok, csúszkák, legördülő menük) lehetővé teszik az interaktív elemek könnyű hozzáadását az oldalakhoz.

A Bootstrap 5-től kezdve ezek működése már nem igényel jQuery-t.

**Testreszabhatóság**

A Bootstrap könnyen testreszabható, így az alapértelmezett stílusok módosíthatók a saját projekt igényeinek megfelelően.

SASS-t (Syntactically Awesome Stylesheets) használ, amely lehetővé teszi a változók, mixinek és egyedi CSS könnyű integrálását.

**Könnyen használható**

A dokumentációja jól strukturált és könnyen követhető.

**Trello**

A Trello egy egyszerű, mégis rendkívül hatékony projektmenedzsment eszköz, amelyet arra terveztek, hogy könnyedén kezelhessük a feladatainkat és projektjeinket. Vizualitásra épül, így a Kanban módszerre alapozva segít a feladatok nyomon követésében és szervezésében.

Főbb tulajdonságok és funkciók:

Felület alapjai

Táblák (Boards): Minden projekt vagy feladat egy külön táblán jelenik meg.

Listák (Lists): A táblán belül oszlopokban rendezheted a feladatokat, például „Teendő”, „Folyamatban”, „Kész”.

Kártyák (Cards): A listákon belül helyezkednek el a kártyák, amelyek konkrét feladatokat vagy ötleteket jelentenek.

Kártyák funkciói

Leírás és címkék: A kártyákhoz részletes leírás és színes címkék (tags) adhatók.

Határidők: Beállíthatók határidők a feladatokhoz.

Csatolmányok: Fájlok és dokumentumok csatolhatók közvetlenül a kártyához.

Checklisták: Részfeladatokat listázhatsz a kártyán belül.

Kommentek: Kommunikálhatsz a csapattagokkal a kártyán belüli hozzászólások révén.

**Automatizáció a Butler segítségével**

**ButlerAutomation:**  
A Trello egyik legizgalmasabb haladó funkciója a beépített automatizációs eszköz, a Butler. Ennek segítségével:

**Szabályok létrehozása:** Automatikusan mozgatásra kerülnek a kártyák, ha például egy határidő lejár, vagy ha egy bizonyos címke került hozzáadásra.

**Ismétlődő feladatok automatizálása:** Létrehozhatsz időzített parancsokat, amelyek például minden hétfőn frissítik a boardot, vagy előre beállítják a napi teendőket.

**Egyéni parancsok:** Specifikus műveletek sorozatát is definiálhatod, így a csapatod kevesebb manuális munkával érheti el a kívánt eredmény.

**Csapatmunka és kollaboráció**

**Valós idejű együttműködés:**  
A Trello lehetővé teszi, hogy több felhasználó egyidejűleg dolgozzon ugyanazon a boardon. A kártyákhoz történő kommentelés, mellékletek hozzáadása és tag-ek hozzárendelése segítik az információáramlást és a csapatmunkát.

**Átláthatóság és visszajelzés:**  
A vizuális megjelenítés révén könnyen követhető a feladatok előrehaladása. A boardok rendszeres karbantartása (pl. archiválás, visszajelzések gyűjtése) biztosítja, hogy mindig a releváns információk álljanak rendelkezésre.

**Github**

A **GitHub** egy online platform és verziókezelő rendszer, amelyet elsősorban szoftverfejlesztők használnak a kódok tárolására, verziókövetésére és csapatmunka támogatására. A **Git** verziókezelő rendszerre épül, és lehetőséget nyújt a kódok megosztására, együttműködésre és a fejlesztési folyamat nyomon követésére.

**Főbb jellemzők és funkciók**

**Git alapok**

A **GitHub** a Git verziókezelő rendszerre épül. A Git egy **disztribuált verziókezelő**, amely lehetővé teszi:

**Kód verzióinak követését:** Minden változás menthető, és szükség esetén visszaállítható.

**Csapatmunka támogatását:** Több fejlesztő dolgozhat ugyanazon a projekten párhuzamosan.

**Repository (Tárhely)**

A kódokat **repository-kban** (röviden: repo) tárolják.

A repo tartalmazhat forráskódot, dokumentációt, konfigurációs fájlokat stb.

A repository lehet:

**Nyilvános:** Bárki láthatja és használhatja.

**Privát:** Csak meghívott személyek férnek hozzá.

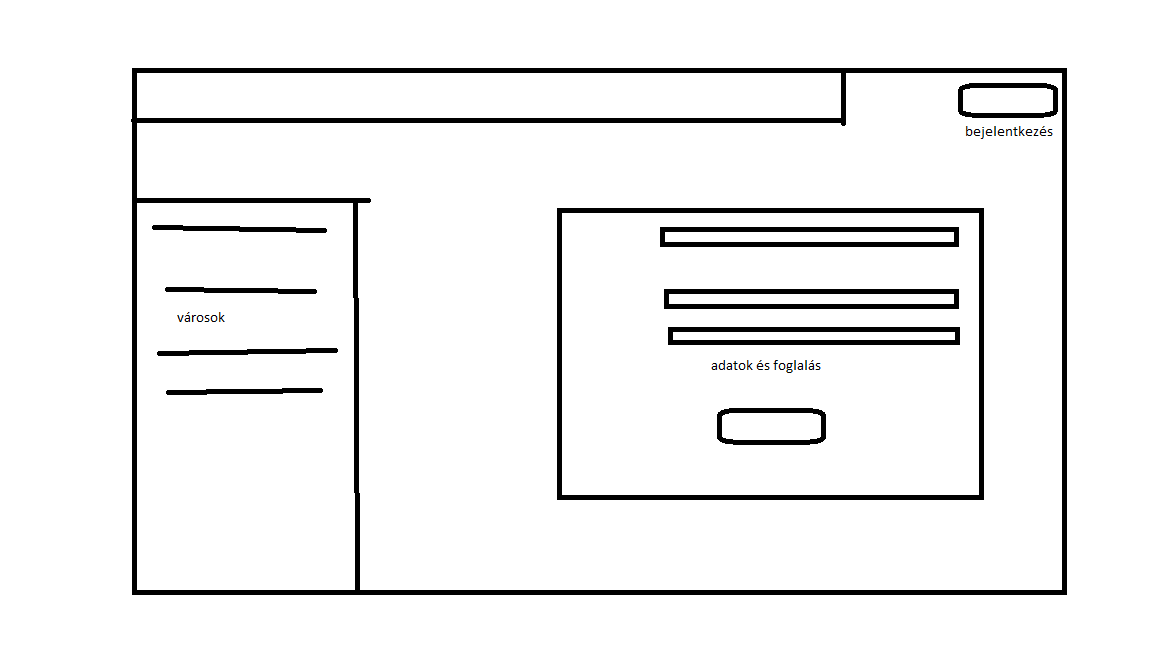
**Branching és Merge (Ágak kezelése)**

**Branch (Ág):** Lehetővé teszi, hogy külön ágakon dolgozz a fő kód bázis módosítása nélkül.

**Merge:** Az elkészült változásokat vissza lehet vonni a fő ágba (általában a „main” vagy „master”).

**Pull Request (PR):** Egy kérés a kódösszevonásra, amit a csapat többi tagja ellenőrizhet, kommentálhat, vagy elfogadhat.

**Alaprajz**



**Tesztelés**

A képen képernyőkép, Multimédiás szoftver, szoftver, Webhely látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.A képen szöveg, képernyőkép, szoftver, Multimédiás szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.A képen szöveg, szoftver, Számítógépes ikon, Weblap 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.

**Források**

HTML:https://hu.wikipedia.org/wiki/HTML letöltés dátuma:2024.12.02.

Javascript:https://hu.wikipedia.org/wiki/JavaScript letöltés dátuma:2024.12.02.

CSS:https://hu.wikipedia.org/wiki/CSS letöltés dátuma:2024.12.02.

React: <https://react.dev/> letöltés dátuma:2024.12.02.

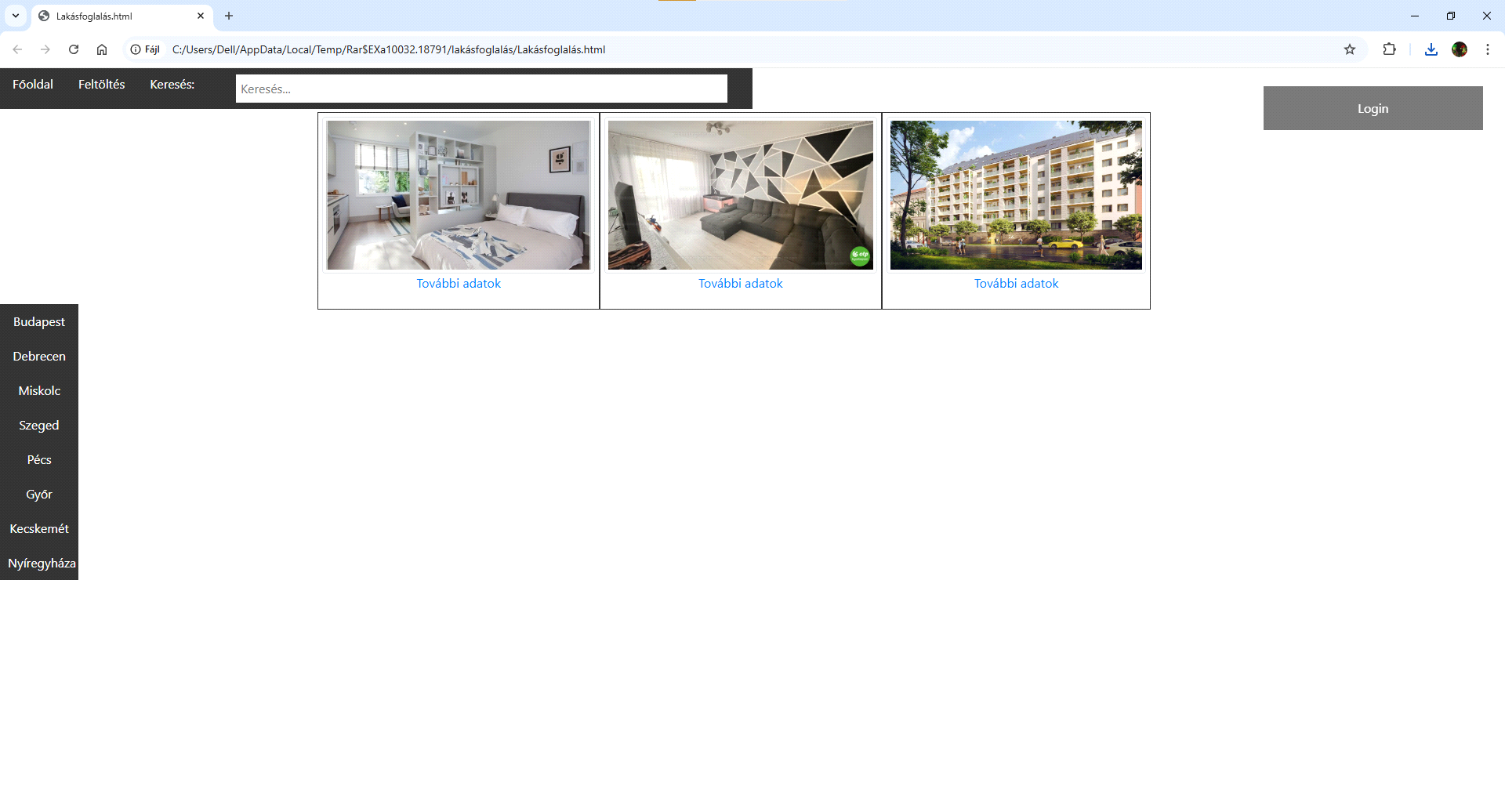
Bootstrap: <https://getbootstrap.com/> letöltés dátuma:2024.12.02.

GitHub: [https://en.wikipedia.org/wiki/GitHub letöltés dátuma 2025.01.10](https://en.wikipedia.org/wiki/GitHub%20letöltés%20dátuma%202025.01.10).

Trello: <https://en.wikipedia.org/wiki/Trello> letöltés dátuma:2025.01.12.

**Mellékletek**

Teszt Főoldal



Teszt Login

