2. Fejezet

Felhasznált technológiák és fejlesztői környezetek

2.1 Backend

A backend a webfejlesztés szerver oldala amit nem lát a felhasználó. Itt szokott történni az adatbázissal való kommunikálás. Itt történnek az authentikációk(hitelesítések) és az authorizációk(engedélyezések). Nagyon sok féle backend technológia létezik mint például: PHP, Python, Node.js, Java. Mint már említettem én a Javat fogom használni a fejlesztések során.

2.1.1 JAVA

A Java egy objektum orientált programozási nyelv, amit James Gosling fejlesztett a 90’-es években Sun Microsystemsnél. Eredetileg televíziózásra készült, de azokban az időkben túlságosan fejlett volt hozzá.

Java célja a hordozhatóság ami azt jelenti, hogy a Java-ban írt programoknak hasonlóan kell futnia bármely operációs rendszeren. Java nyelvi kódot elősször bájtkódra fordítják ami hasonló a gépi kódhoz, de virtuális gép általi végrahajtásra készülnek. Vannak mikrovezérlők mik képesek a Java bájtkódját hardverben futtatni.

Java a Szintaxisát a C és C++ nyelvből örökölte, még a C++ egyesíti a strukturált, általános és objektumorientált programozás szintaxisát addig a Java csak Objektumorientált nyelvnek keszült. Java nem támogatja az operátok túlterhelést vagy a többszörös öröklődést (interfészek esetében lehetséges).

Java jelenlegi tulajdonosa az Oracle Corporation 2010-óta hogy felvásárolta a Sun Microsystems-t

Java az egyik legelterjettebb programozási nyelv, mivel mindegyik operációs rendszerre tudunk vele alkalmazást fejleszteni, és könnyen tanulható.Nagyon felhasználó barát.

2.1.3 Spring Boot

A Spring Boot Spring-keretrendszerre épülő bővítmény a Spring pedig egy Java alapú webalkalmazás-keretrendszer ami nyílt forráskódú. A Spring Testreszabott Webalkalmazások létrehozásához tökéletes ami teljesen konfigurálható az előre elkészített kódterekkel és kódtárakkal. Spring Bottal különálló Spring alkalmazásokat hozhatunk létre amit azonnal futtathatunk.

Spring Boot használata nagyon egyszerű, jó minőségű alkalmazásokat lehet benne fejleszteni kevesebb fejlesztési idő alatt. Beépített http-kiszolgálókat tartalmaz mint a Tomcat és Jerry.

Spring Boot tartalmaz egy Maven-hez tartozó (POM.XML) fájlt amiben amiben spring-boot-dependencies-eket tudunk megadni, mint például:

* spring-boot-starter-web
* spring-boot-starter-data-jpa
* spring-boot-starter-test

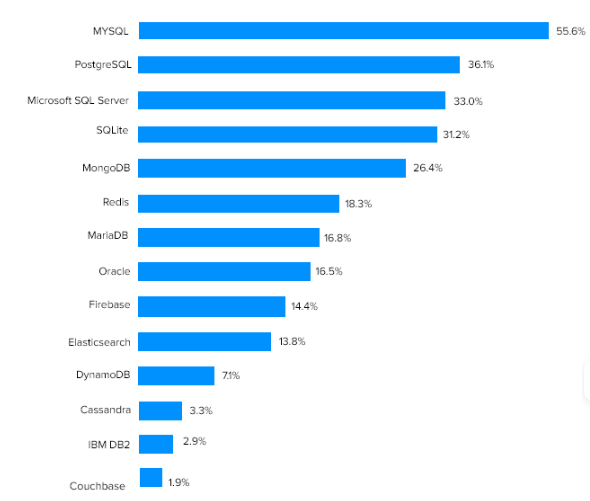
2.2. Adatbázis

Az adatbázis nem más mint elektronikusan tárolt adatok amihez hozzá lehet férni.

2.2.1 PostgreSQL

Nyílt forráskódú relációs adatbázis kezelő rendszer ami az egyik legrégebbi relációs adatbázis nem mellesleg ingyenes. Relációs adatbázisok relációs modellen alapulnak. A relációs modell az adatokat egy vagy több sorból és oszlopból álló táblázatba(relációba) rendezi és minden sorhoz rendel egy egyedi kulcsot.

A PostgreSQL Kaliforniai egyetem Ingres projektjéből fejlődött. Az összes operációs rendszerrel kompatibilis. Támogatja a relációs és a nem relációs lekérdezéseket is így JSON vagy SQL alapú útvonal-kifejezésekkel is elérhetőek az adatok. Mind ezek miatt az egyik leghasználtabb Adatbázis-motor.



2.1. ábra. 14 leghasználtabb adatbázis

2.3 Front-end

A frontend a szoftver megjelenítési rétege. A weboldalnak azon réteg amit a felhasználó is lát és interakcióba tud vele lépni. Legfontosabb része a HTML(Hypertext Markup Language) ez adja meg a Weboldal kinézetének a vázát és hozzá csatlakozik a CSS(Cascading Style Sheets) ami segítségével egyedi megjelenést biztosíthatunk a weboldalunknak. Több fajta framework létezik Front-end fejlesztésre mint például:

* React
* Angular
* Vue.js.

2.3.1 Angular

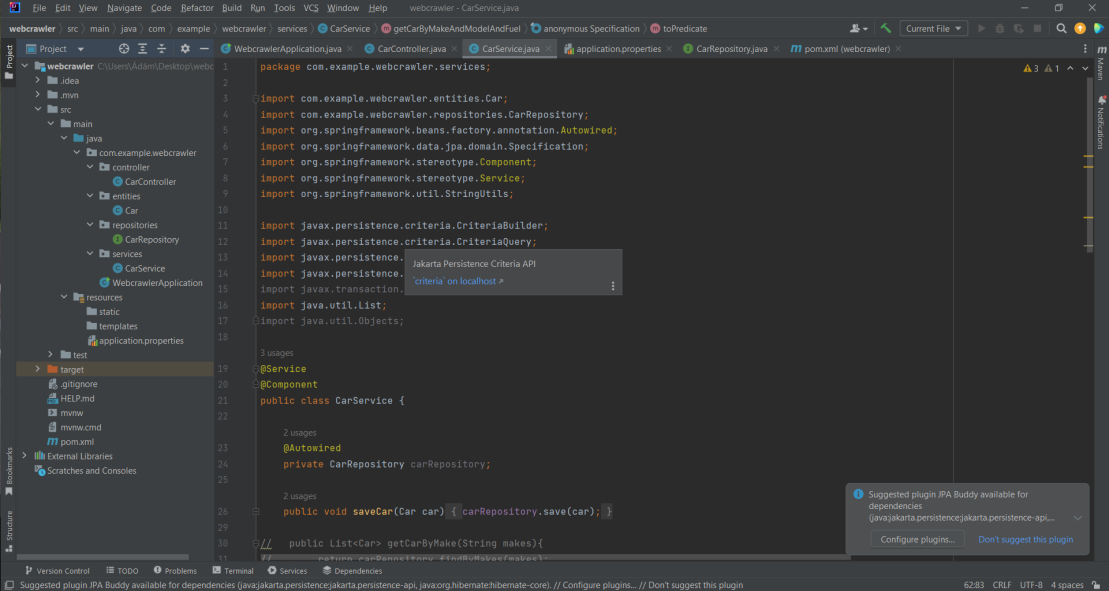
Az Angular egy TypeScript-alapú nyílt forráskódú webalkamazás-keretrendszer. 2016-ban jelent meg az első verzió.

Egyben tartalmazza TypeScript osztályt HTML-sablont stílusokkal. HTML sablon lehetővé teszi dinamikus értékek beszúrását mint például szöveg. Tartalmaz komponenseket ezek NgModulokba vannak rendezve. Minden alkalmazásnak van egy úgynevezett gyökérmodulja aminek a neve általában az AppModule, amely a bootstrap mechanizmust biztosítja, amely elindítja az alkalmazást. Ngmodulok is importálhatnak más Ngmodulokat, például az útválasztó szolgáltatás használatához a Router NgModul-t. Minden Angular alkalmazásnak van egy gyökérkomponense, ami összekapcsolja a komponens hierarchiát. Mindegyik komponenshez tartozik egy HTML-sablon, ami segítségével megjeleníthető a tartalom. Tartalmaz egy services osztályt is ezt akkor használjuk ha van olyan adat vagy logika, amelyek nem kapcsolódnak, de meg szeretnénk osztani a komponensek között.

2.4 Fejlesztői környezetek és fejlesztéshez használt programok

2.4.1 Intelij IDEA

Egy integrált fejlesztői környezet amit a JetBrains fejlesztett ki. Ebben az IDEA-ban Java, Kotlin Groovy és más JVM alapú nyelveken írt szoftvereket lehet fejleszteni. Az integrált fejlesztői környezet(IDE) egy olyan  szoftveralkamazás, amely átfogó lehetőségeket biztosít a számítógép programozóknak a szofverfejlesztéshez . Az IDE általában legalább egy forráskód-szerkesztőből ,építési autómatizálási eszközökből és egy hibaleresőből áll . Egyes IDE-k, például a NetBeans és az Eclipse tartalmazzák a szükséges fordított , értelmezőt vagy mindkettőt; mások, például a SharpDevelop és a Lazuras nem.[Link].



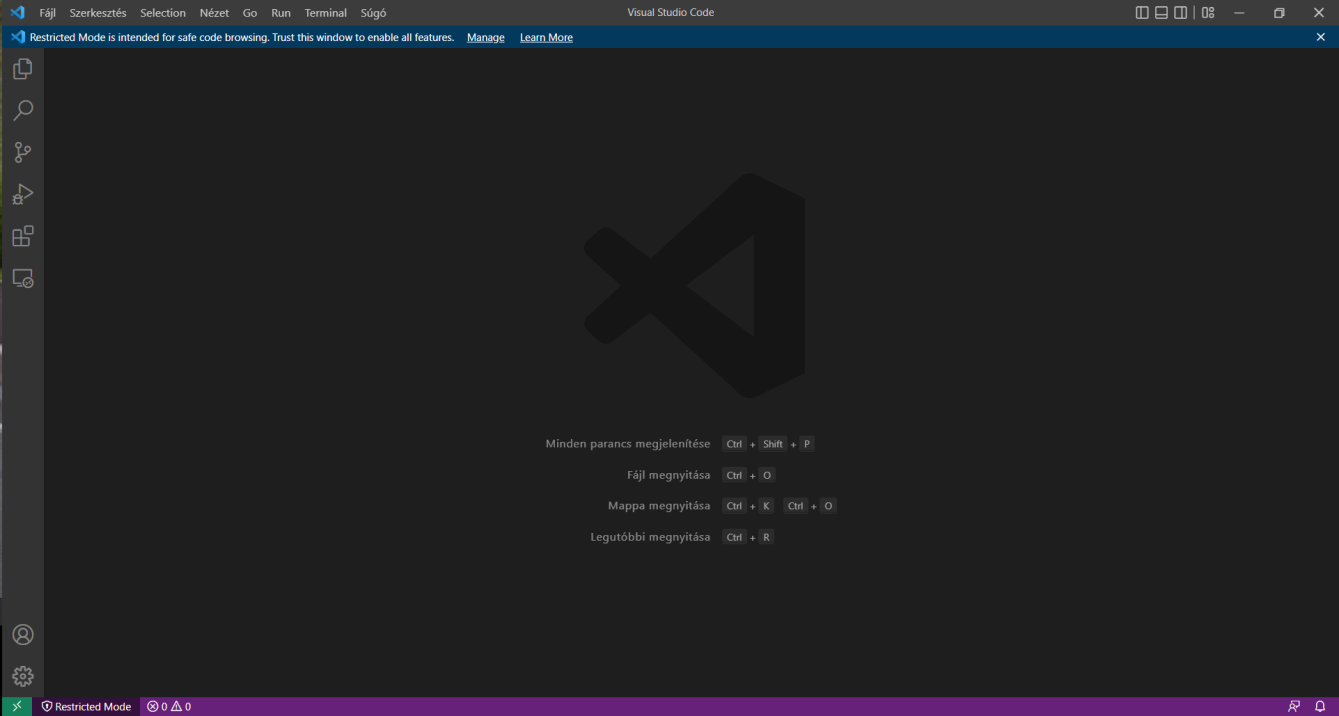
2.4.1 ábra Intelij IDEA

2.4.2 Visual Studio Code

VS Code egy forráskód szerkesztő amit a Microsoft fejlesztett ki 2015-ben.Csak Windows Linux és MacOS operációs rendszereken elérhető. Nagyon sok programozási nyelvvel lehet használni mint például:

* JavaScript
* Go
* Node.js
* C++
* Python

Rengeteg kiegészítővel lehet bővíteni a fejlesztő környezetet ami könnyebbé és átláthatóbbá teszi fejlesztést, ezért esett nekem a választás erre a környezetre a frontend fejlesztéséhez.

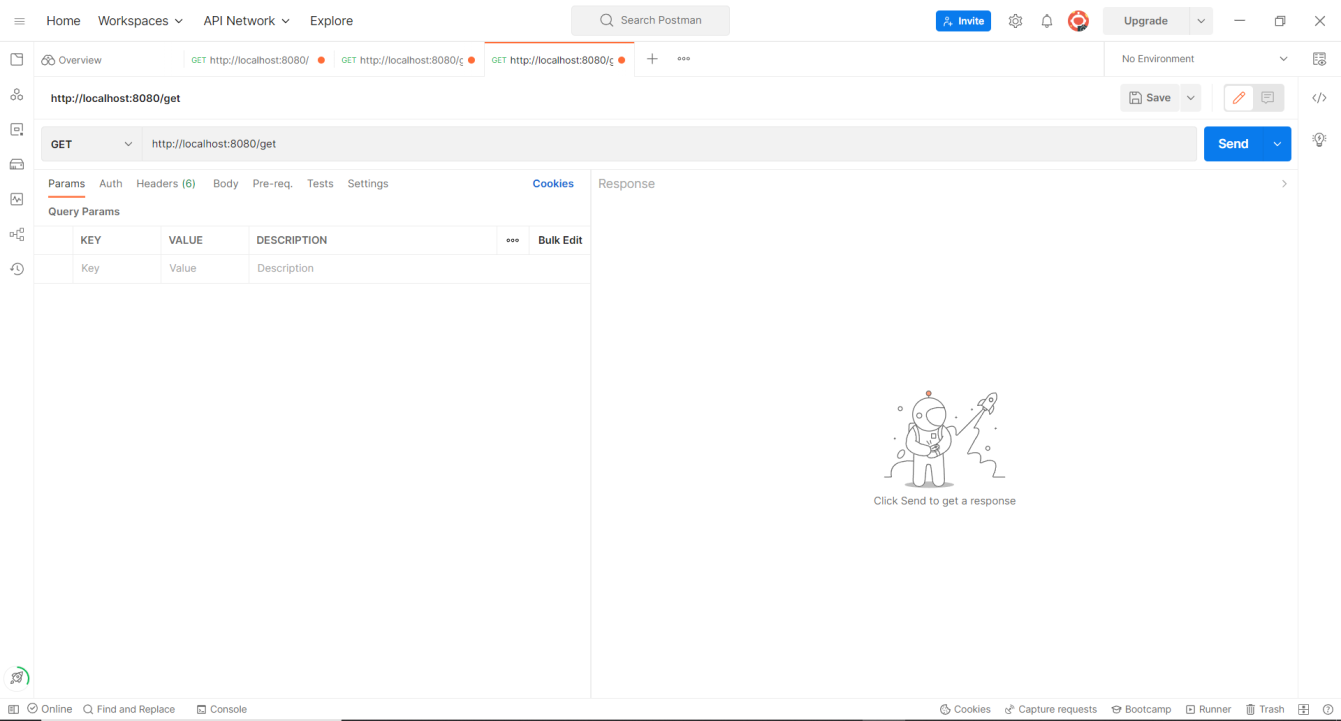


2.4.2 ábra Visual Studio Code

2.4.3 Postman

A Postman API-k létrehozására és használatára, tesztelésére létrehozott plattform. Használatával egyszerűbben és gyorsabban hozhatunk létre jobb minőségű API-kat. Rengetek eszközkészletet tartalmaz amely tovább gyorsítja az API létrehozását a tervezéstől egyenesen a tesztelésig. Ilyen eszközök:

* API kliens: Lehetővé teszi API-k tesztelését hibakeresését és van lehetőség HTTP, REST, SOAP és GraphQL kéréseket is.
* API tervezés: OpenAPI, RAML, GraphQL vagy SOAP formátumban tervezhetjük meg az API-kat. A Postman Schema szerkesztője megkönnyíti a különböző méretű fájlokkal való munkát.
* API dokumentáció: Postman automatikusan géppel olvasható dokumentációt hoz létre amit OpenAPI-fájlokon keresztül dokumentál. Tartalmazza a kérések részleteit, mintakódokkal.
* API tesztelés: Lehetőség van használni funkcionális teszteket, integrációs teszteket, regressziós teszteket. A Postman egy Node.js alapú futtatókörnyezet ami támogatja gyakori mintákat és könyvtárakat ami segíti a gyors teszt készítését.
* Monitorozás: Naprakészek lehetünk az API állapotát és teljesítményét. A monitorok a Postman felhőjében vannak tárolva és ennek köszönhetően gyorsan beállíthatjuk őket.
* Mock szerverek: Más néven „Áll szerverek” aminek segítségével láthatjuk hogyan fog futni az API-nk mielőtt kihelyeznénk az éles környezetbe. A Postman felhő üzemelteti és így bárhonnan elérhetőek.



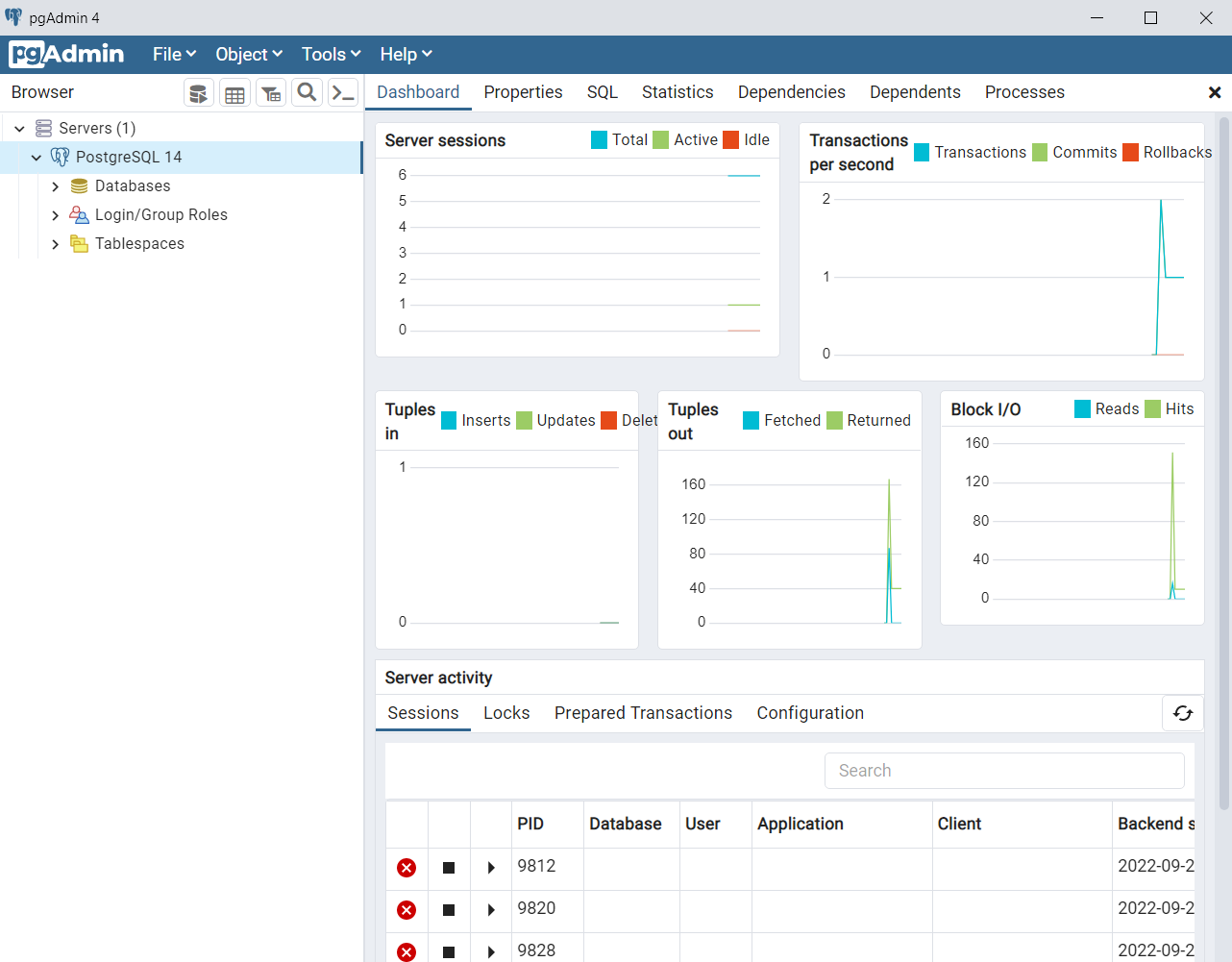
2.4.3 ábra Postman

2.4.4 PgAdmin

A PgAdmin egy grafikus felhasználói felület a PostgreSQL kezelésére ami a lehető legjobb megoldás lehet. Legújabb verziója a PgAdmin 4, jQuerry, JavaScript és Python kombinációjával készült. Előnyei:

* Kompatibilis Windows, linux és Mac operációs rendszerekkel is.
* Bárhova telepíthető ahol PostgreSQL-t használ.
* Van olyan lekérdező eszközei amivel gyorsabb az adatbevitel és a hibakezelés

Rengeteg dokumentáció megtalálható hozzá amivel könnyedén el lehet kezdeni a használatát.



2.4.4 ábra PgAdmin 4