Vizsgaremek dokumentáció

Tartalomjegyzék

Bevezetés 1

Kommunikáció 2

Trello 2

Github 2

Általunk használt szoftverek és technológiák 3

Xampp 4

React 5

Wpf 6

C# 7

Web-api 9

Források 10

**Bevezetés**

Kezdetben projektünk témájának kitalálásával volt problémáink. Mivel szerettünk volna egy olyan programot tervezni ami viszonylag egyedülálló és könnyen megvalósítható. Így az elején már megegyeztünk ,hogy a legjobb választás egy ismeretterjesztő weboldal és fórum lenne.

Mivel így rengeteg ember akinek egy adott témában nincs sok ismeretre viszont szeretne megismerkedni vele segítséget tudnánk nyújtani. Valamint az oldalon lévő felhasználók közötti kommunikáció létesítésével a már témában jártas személyek vagy idővel az oldal veterán felhasználói tudjanak segíteni az új felhasználóknak .

Sok eszmecsere után a retróautók ra esett a választásunk mivel csapatunk tagjai a történelmet és az autók terén jártasak és így mind hozzá tudunk tenni a témához. Viszont felmerült az a probléma ,hogy már létezik ilyen oldal és csak egy újabb unalmas oldal lennénk a sok közül.

Éppen ezért kutatásokat végeztünk ,hogy utána járjunk van-e igény ilyen oldalra vagy már van-e ilyen ,ha igen akkor ,hogyan tudnánk jobbat csinálni. Viszont kutátsunk arra jutott ,hogy retró autókkal csak árusító oldalak foglalkoznak és a megismertetésükkel egyik sem. Így nyugodt szívvel kezdhettünk bele projektünkbe.

1.

**Kommunikáció**

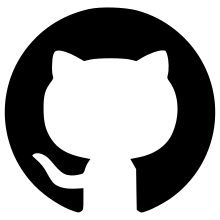
Mielőtt neki kezdtünk volna a projektünk fejlesztésébe el kellet döntenünk ,hogy hol kommunikáljunk online és osszuk meg egymással a project haladását úgy ,hogy visszakövethető legyen. Volt szó discordról és messengerről ,majd végül a Trellot választottuk.

**Trello:**

A felhasználók különböző oszlopokat tartalmazó feladattáblákat hozhatnak létre, és mozgathatják közöttük a feladatokat. Az oszlopok jellemzően olyan feladatállapotokat tartalmaznak, mint a Teendő, Folyamatban, Kész. Az eszköz használható személyes és üzleti célokra, beleértve az ingatlankezelést, szoftverprojektmenedzsmentet, iskolai hirdetőtáblákat, óratervezést, könyvelést, webtervezést, játékokat és ügyvédi irodai ügykezelést.

**Github:**

AGitHub egy fejlesztői platform, amely lehetővé teszi a fejlesztők számára kód létrehozását, tárolását, kezelését és megosztását. Git szoftvert használ, amely minden projekthez elosztott verzióvezérlést biztosít a hozzáférés-vezérléshez , a hibakövetéshez , szoftverfunkciókérésekhez , a feladatkezeléshez , a folyamatos integrációhoz és a wikikhez. Általában nyílt forráskódú szoftverfejlesztési projektek fogadására használják .



2

**Az általunk használt szoftverek és technológiák**

Miután megvolt a téma és hogyan kommunikáljunk valamint megbizonyosodtunk arról ,hogy a miénk az egyetlen ilyen oldal kezdődhetett a tervezés és a programozás. Először kiválasztottuk milyen adatokat fogunk használni és hány adatbázis táblát használjunk, milyen programozási nyelven fogunk programozni és ,hogy milyen programozói környezetben dolgozunk.

**Xampp:**

A XAMPP egy szabad és nyílt forrású platformfüggetlen webszerver-szoftvercsomag, amelynek legfőbb alkotóelemei az Apache webszerver, a MariaDB (korábban a MySQL) adatbázis-kezelő, valamint a PHP és a Perl programozási nyelvek értelmezői (végrehajtó rendszerei). Ez a szoftvercsomag egy integrált rendszert alkot, amely webes alkalmazások készítését, tesztelését és futtatását célozza, és ehhez egy csomagban minden szükséges összetevőt tartalmaz. A rendszer egyik nagy előnye az összehangolt elemek könnyű telepíthetősége.

Használata:

Hivatalosan a XAMPP tervezői az eszközt egy fejlesztőrendszernek szánták, amellyel a web-tervezők és programozók internetes kapcsolat nélkül fejleszthetik és tesztelhetik alkalmazásaikat. Ennek érdekében több fontos biztonsági funkció alapértelmezésben ki van kapcsolva a csomagban, ennek ellenére a XAMPP szoftvert valódi webes szolgáltatóként is használják. A csomag egy külön eszközt tartalmaz a legfontosabb részek jelszavas védelmének beállítására.

3.

A XAMPP többféle adatbázis-kezelő használatát is támogatja, ilyenek pl. a MySQL és az SQLite és mások.

A XAMPP telepítése után a helyi gép (a localhost) hálózati gépként is hozzáférhetővé válik, pl. FTP kliensprogrammal elérhető. Alkalmazható

pl. FileZilla fájlkezelő, telepíthetők tartalomkezelő rendszerek, mint

a WordPress vagy a Joomla!. A *localhost* közvetlenül is kezelhető egyes HTML vagy általános szövegszerkesztőkből, az FTP protokoll használatával.

Az alapbeállítás szerinti FTP felhasználónév a „newuser”, jelszava „wampp”. Az alapbeállítás szerinti MySQL felhasználónév a „root”, jelszó nélkül.

****

4.

**React:**

**A React** (más néven **React.js** vagy **ReactJS** ) egy ingyenes és nyílt forráskódú elülső JavaScript-könyvtár  , amelynek célja, hogy a komponenseken alapuló felhasználói felületek felépítését "zökkenőmentesebbé tegye". A Meta (korábban Facebook) és az egyes fejlesztőkből és cégekből álló közösség tartja karban .

A React használható egyoldalas , mobil vagy szerver által megjelenített alkalmazások fejlesztésére olyan keretrendszerekkel, mint a Next.js . Mivel a React csak a felhasználói felülettel és a DOM- hoz való megjelenítési összetevőkkel foglalkozik , a React alkalmazások gyakran támaszkodnak könyvtárakra az útválasztáshoz és egyéb kliensoldali funkciókhoz. A React legfontosabb előnye, hogy csak az oldal megváltozott részeit jeleníti meg újra, elkerülve a változatlan DOM-elemek szükségtelen újramegjelenítését.

JSX:

A JSX vagy JavaScript XML a JavaScript nyelv szintaxisának kiterjesztése. A HTML-hez hasonló megjelenésű, A JSX módot ad a komponensek megjelenítésének strukturálására a sok fejlesztő számára ismert szintaxis segítségével . A React komponenseket jellemzően JSX használatával írják, bár nem kötelező (az összetevők tiszta JavaScript-ben is írhatók). A JSX hasonló egy másik kiterjesztési szintaxishoz, amelyet a Facebook készített PHP- hez, az XHP- t .

Vírtuális DOM:

Egy másik figyelemre méltó funkció a virtuális dokumentumobjektum-modell vagy a virtuális DOM használata . A React létrehoz egy memórián belüli adatstruktúra gyorsítótárat, kiszámítja az ebből eredő különbségeket, majd hatékonyan frissíti a böngészőben megjelenített DOM-ot.Ezt a folyamatot egyeztetésnek nevezik . Ez lehetővé teszi a programozó számára, hogy kódot írjon úgy, mintha a teljes oldalt minden változtatáskor renderelné, míg a React csak a ténylegesen megváltozott összetevőket jeleníti meg. Ez a szelektív megjelenítés jelentős teljesítménynövekedést biztosít.

5.

React fejlődése:

A projekt állapota nyomon követhető az alapcsapat vitafórumán keresztül. A React főbb változásai azonban a Future of React adattárral kapcsolatos problémákon és lekéréseken mennek keresztül . Ez lehetővé teszi a React közösség számára, hogy visszajelzést adjon az új potenciális funkciókról, kísérleti API-król és JavaScript szintaktikai fejlesztésekről.

**WPF:**

A Windows Presentation Foundation ( WPF ) egy ingyenes és nyílt forráskódú felhasználói felület keretrendszer Windows -alapú asztali alkalmazásokhoz. A WPF-alkalmazások .NET -en alapulnak , és elsősorban C# és XAML használatával készülnek .

Az eredetileg a Microsoft által kifejlesztett WPF-et 2006-ban a .NET Framework 3.0 részeként adták ki. 2018-ban a Microsoft kiadta a WPF-et nyílt forráskódúként az MIT-licenc alatt . A WPF dizájnját és elrendezési nyelvét, az XAML-t számos más UI-keretrendszer is átvette, mint például az UWP , a .NET MAUI és az Avalonia .

A WPF támogatja a legtöbb modern médiatípust, beleértve a vektoros és raszteres képeket, hangot, videót, és támogatja a 2D és 3D renderelést is. Támogatja továbbá a grafikus effektusok megvalósítását a vizuális elemeken, például a HLSL pixel shadereket és a beépített effektusokat, például az elmosódásokat és a vetett árnyékokat .

A WPF főként vektorgrafikára támaszkodik, amely lehetővé teszi a legtöbb vezérlőelem és elem átméretezését minőségromlás vagy pixelezés nélkül .

6.

**C#:**

A C# egy általános célú, magas szintű programozási nyelv, amely több paradigmát is támogat . A C# magában foglalja a statikus gépelést, erős gépelést , lexikális hatókörű , felszólító , deklaratív , funkcionális , általános , objektum orientált ( osztály -alapú) és komponens-orientált programozási tudományágat.

A C# támogatja az erősen, implicit módon begépelt változódeklarációkat a var, kulcsszóval , valamint az implicit módon beírt tömböket a kulcsszóval, new[]amelyet egy gyűjteményinicializáló követ.

Típusrendszere két családra oszlik: Értéktípusokra, mint például a beépített numerikus típusokra és a felhasználó által definiált struktúrákra, amelyek automatikusan másolatként kerülnek átadásra, amikor paraméterként használják őket, valamint hivatkozási típusokra, beleértve a tömböket, osztálypéldányokat és karakterláncokat. , amelyek csak egy mutatót adnak át az adott objektumra. Az egyenlőség operátor speciális kezelésének köszönhetően a karakterláncok ennek ellenére úgy viselkednek, mintha értékek lennének, minden gyakorlati célból. Akár tokcímkeként is használhatja őket . Ahol szükséges, az értéktípusok automatikusan dobozba kerülnek .

A C# szigorú logikai adattípust támogat , bool. Az olyan utasításokhoz, amelyek feltételeket vesznek fel (például whileés if), olyan típusú kifejezésre van szükség, amely megvalósítja az trueoperátort, például a logikai típust. Míg a C++-nak van logikai típusa is, szabadon konvertálható egész számokká és egész számokból, és az olyan kifejezésekhez, mint például az if (a)szükséges, csak ez akonvertálható bool-ba, lehetővé téve, ahogy int vagy mutató legyen. A C# nem engedélyezi ezt az „igaz vagy hamis jelentésű egész” megközelítést, azon az alapon, hogy a programozók arra kényszerítésével, hogy pontosan visszatérő kifejezéseket használjanak, boolmegelőzhetők bizonyos típusú programozási hibák, mint például if (a = b)(hozzárendelés használata =egyenlőség helyett ==).

7.

A C# típusbiztonságosabb, mint a C++. Az egyetlen implicit konverzió alapértelmezés szerint azok, amelyek biztonságosnak tekinthetők, például az egész számok kiszélesítése. Ez fordításkor, JIT alatt és bizonyos esetekben futási időben érvényesül. Nem történik implicit konverzió a logikai értékek és az egész számok között, sem a felsorolási tagok és az egész számok között (kivéve a literál 0-t, amely implicit módon konvertálható bármilyen felsorolt ​​típusra). Minden felhasználó által definiált konverziót kifejezetten explicitként vagy implicitként kell megjelölni, ellentétben a C++ másoláskonstruktorokkal és konverziós operátorokkal, amelyek alapértelmezés szerint mindkettő implicit.

A C# kifejezetten támogatja a kovarianciát és a kontravarianciát az általános típusokban, ellentétben a C++-szal, amely bizonyos mértékben támogatja az ellentmondást egyszerűen a virtuális metódusok visszatérési típusainak szemantikája révén.

A felsorolási tagok saját hatáskörükbe kerülnek .

A C# nyelv nem engedélyez globális változókat vagy függvényeket. Minden metódust és tagot az osztályokon belül deklarálni kell. A nyilvános osztályok statikus tagjai helyettesíthetik a globális változókat és függvényeket.

A helyi változók nem árnyékolhatják a befoglaló blokk változóit, ellentétben a C-vel és a C++-val.

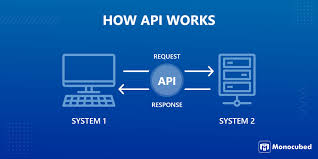
8.

**Web-api:**

A webes API egy alkalmazásprogramozási felület (API) webszerverhez vagy webböngészőhöz . Webfejlesztési koncepcióként kapcsolódhat egy webalkalmazás ügyféloldalához (beleértve a használt webes keretrendszereket is ). A szerveroldali webes API egy vagy több nyilvánosan elérhető végpontból áll egy meghatározott kérelem-válasz üzenetrendszerhez, amelyet általában JSON-ban vagy XML- ben fejeznek ki egy HTTP -alapú webszerver segítségével . A szerver API (SAPI) nem tekinthető szerveroldali webes API-nak, kivéve, ha nyilvánosan elérhető egy távoli webalkalmazás által.

A végpontok fontos szempontok a szerveroldali webes API-kkal való interakció során, mivel meghatározzák, hogy hol vannak olyan erőforrások, amelyekhez harmadik féltől származó szoftverek hozzáférhetnek. A hozzáférés általában egy URI-n keresztül történik, amelyre a HTTP-kéréseket elküldik, és amelytől a válasz így várható. A webes API-k lehetnek nyilvánosak vagy privátak, amelyek közül az utóbbihoz *hozzáférési jogkivonat* szükséges .

A végpontoknak statikusnak kell lenniük, különben nem garantálható a velük kölcsönhatásba lépő szoftver megfelelő működése. Ha egy erőforrás helye (és vele együtt a végpont) megváltozik, akkor a korábban írt szoftverek tönkremennek, mivel a szükséges erőforrás már nem található ugyanazon a helyen. Mivel az API-szolgáltatók továbbra is frissíteni szeretnék webes API-jaikat, sokan bevezettek egy olyan verziókezelő rendszert az URI-ban, amely egy végpontra mutat.



9.

Források:

<https://hu.wikipedia.org/wiki/Trello>

<https://hu.wikipedia.org/wiki/XAMPP>

<https://en.wikipedia.org/wiki/React_(software)>

<https://en.wikipedia.org/wiki/Windows_Presentation_Foundation>

<https://en.wikipedia.org/wiki/C_Sharp_(programming_language)>

<https://en.wikipedia.org/wiki/Web_API>