



Záródolgozat

Készítették:

Csitári Ádám János – Kristály Adrián

Konzulensek:

Kerényi Róbert Nándor, Németh Bence

Miskolc

2024.

Miskolci SZC Kandó Kálmán Informatikai Technikum

Miskolci Szakképzési Centrum

SZOFTVERFEJLESZTŐ- ÉS TESZTELŐ SZAK

EventWave

Eseménykezelő közösségioldal

Csitári Ádám János – Kristály Adrián

2022-2024

Tartalomjegyzék

Eventwave eseménytár weboldal

1. Miért választottuk ezt a projektet?

-Szükség:

Egy olyan online platformra van szükség, ahol az emberek egyszerűen létrehozhatnak és megtalálhatnak eseményeket a különböző témákban és kategóriákban. Rengeteg ember manapság sajnos nehezen mozdul ki otthonról viszont a mi oldalunknak köszönhetően egy lépéssel előrébb kerülhet az átlagember ahhoz, hogy egy közösség tagja lehessen, sőt, hogy közösséget építhessen.

-Közösségépítés

Az események összehozzák az embereket, lehetőséget teremtve a közösségek építésére és a kapcsolatok kialakítására a közös érdeklődési körök mentén. A közös érdeklődéssel rendelkező emberek így könnyedén találkozhatnak.

-Tájékozódás

A felhasználók szeretnék könnyen és gyorsan megtalálni az érdeklődésüknek megfelelő eseményeket a különböző városokban vagy online.

-Hírdetés: Rengeteg embernek csak egy platformra van szüksége ahhoz, hogy ennek az oldalnak a segítségével indítsa el eseményszervezői karrierjét.

-Rugalmas használat

A weboldal kialakítása intuitív és felhasználóbarát legyen, hogy mindenki könnyen megtalálja az általa keresett információkat, illetve könnyedén tudjon eseményeket létrehozni és kezelni.

Ezen szempontok alapján úgy gondoljuk, hogy az Eventwaves weboldal nagy segítséget nyújthat az embereknek az események felfedezésében és részvételében, valamint elősegítheti az online és offline közösségek kialakulását és erősödését.

2. Technológiák

Adatbázis

- Mint adatbázis kezelő, felváltva használtunk XAMPP-ot illetve Wamp-ot is egyaránt, programozási nyelv mint adatbázisban használtunk az a MySQL.
- Az XAMPP és a WAMP olyan szoftvercsomagok, amelyek teljesen konfigurált fejlesztői környezetet biztosítanak a PHP, a MySQL és az Apache webkiszolgáló számára. Ezek a csomagok könnyen telepíthetőek és konfigurálhatóak, így ideálisak lokális fejlesztéshez és teszteléshez.

Adatbázis-kezelés során változóan használtuk a XAMPP és a WAMP szoftvercsomagokat. Mindkét csomag teljesen konfigurált fejlesztői környezetet biztosít a PHP, MySQL és Apache webkiszolgáló számára. Ezek a csomagok könnyen telepíthetőek és konfigurálhatóak, így ideálisak lokális fejlesztéshez és teszteléshez.

A XAMPP egy szabad és nyílt forrású platformfüggetlen webserverszoftvercsomag, amelynek legfőbb alkotóelemei az Apache webserver, a MariaDB adatbázis-kezelő, valamint a PHP és a Perl programozási nyelvek értelmezői.

A WampServer(WAMP) a Microsoft Windows operációs rendszerhez készült megoldáscsomagra utal, amelyet Romain Bourdon hozott létre, és amely az Apache webserverből, az SSL-támogatáshoz szükséges OpenSSL-ből, a MySQL adatbázisból és a PHP programozási nyelvből áll.

A XAMPP és a WAMP tartalmazza az Apache HTTP szervert, a MySQL adatbázis-kezelő rendszert és a PHP-t, valamint más alkalmazásokat és szolgáltatásokat is, mint például a phpMyAdmin adatbázis-kezelő felület.

A PHP egy széles körben használt, skriptnyelv alapú programozási nyelv, amely különösen alkalmas webes alkalmazások fejlesztésére. A MySQL pedig egy

ingyenesen elérhető, nyílt forráskódú relációs adatbázis-kezelő rendszer, amely hatékonyan tárolja és kezeli az adatokat.

A XAMPP és a WAMP segítségével könnyedén létrehozhatunk egy lokális fejlesztői környezetet, ahol gyorsan és hatékonyan fejleszthetünk és tesztelhetünk webalkalmazásokat, miközben kihasználjuk a PHP és MySQL által nyújtott lehetőségeket.

Backend

- **Backendbe** az adatok megfelelő áramlását, illetve hibakezelés, adatvédelem egyaránt megjelenik benne:

-**A fejlesztői környezet** backend fejlesztéséhez használtunk egy olyan integrált fejlesztői környezetet (IDE egy olyan szoftveralkalmazás, amely átfogó létesítményeket biztosít a számítógépes programozók számára a szoftverfejlesztéshez. Az IDE általában legalább egy forráskódszerkesztőből, automatizálási eszközökből és hibakeresőből áll.), mint például a Visual Studio, amely lehetővé teszi a könnyű kódírást, hibakeresést és projektmenedzsmentet. A fejlesztői környezet lehetővé teszi a szoftverfejlesztőknek a kódírás és a tesztelés folyamatának optimalizálását. Olyan eszközök, mint például a Visual Studio vagy a Visual Studio Code, számos funkciót és kiterjesztést kínálnak a hatékony fejlesztéshez.

-A **C#** egy erős, típusos programozási nyelv a .NET platformhoz, amelyet gyakran használnak a backend fejlesztéséhez. A C# támogatja az objektumorientált programozást, a modern nyelvi elemeket és erőteljes eszközöket kínál a fejlesztőknek.

A Web API egy alkalmazásprogramozási interfész a webhöz.

A Böngésző API bővítheti a webböngésző funkcionalitását.

A Szerver API bővítheti a webservert funkcionálisitását.

"https://www.w3schools.com/js/js_api_intro.asp"

A Web API-k (Application Programming Interface) lehetővé teszik a backend szolgáltatások elérését HTTP protokollon keresztül. A Web API egy alkalmazás programozási interfésze a weben. A Böngésző API kiterjesztheti a webböngésző funkcionálisitását. A Szerver API kiterjesztheti a webservert funkcionálisitását. A RESTful elveket követve a Web API-k segítségével könnyen kommunikálhatunk a frontend alkalmazásokkal, például AJAX kérések vagy egyéb HTTP kérések révén.

„https://www.w3schools.com/js/js_api_intro.asp”

-Az **Entity Framework** egy ORM (Object-Relational Mapping) keretrendszer a .NET-hez, amely lehetővé teszi az adatbázis-entitások és az objektumok közötti átjárást és az adatbázis-műveletek kezelését C# objektumokként.

Az Entity Framework egy modern objektum-relációs leképező, amely lehetővé teszi egy tiszta, hordozható és magas szintű adatelérési réteg építését .NET környezetben többféle adatbázisra, beleértve az SQL Database-t (helyi és Azure), SQLite-ot, MySQL-t, PostgreSQL-t és Azure Cosmos DB-t is. Támogatja a LINQ lekérdezéseket, az állapotkövetést, a frissítéseket és a séma migrációkat.
"<https://learn.microsoft.com/en-us/ef/>"

Az Object-Relational Mapping (**ORM**) egy olyan technika, amely lehetővé teszi az objektumok és az adatbázisok közötti átjárást, anélkül, hogy a fejlesztőnek közvetlenül SQL-lel kellene dolgoznia. Az ORM segítségével a fejlesztők egyszerűbb és absztrakt módon dolgozhatnak az adatokkal.

- A Cross-Origin Resource Sharing (CORS) egy HTTP fejlécalapú mechanizmus, amely lehetővé teszi a szerver számára, hogy jelezze az eredet (domain, séma vagy port) más forrásokat is, mint a sajátját, amelyekről a böngészőnek engedélyeznie kell az erőforrások betöltését.

A CORS továbbá egy mechanizmusra támaszkodik, amelynek segítségével a böngésző "preflight" (egy előzetes kérés, amelyet a böngésző küld a szervernek a tényleges keresés előtt a CORS mechanizmus során.) kérést küld a kereszt eredetű erőforrást hostoló szervernek, annak ellenőrzése érdekében, hogy a szerver engedélyezi-e a tényleges kérést. Ebben a preflight kérésben a böngésző olyan fejléceket küld, amelyek jelzik a tényleges kérésben használt HTTP módot és fejléceket.

<https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS>

-A **CRUD műveletek** (Create, Read, Update, Delete) az alapvető adatműveletek, amelyek lehetővé teszik az adatok kezelését az alkalmazásban. Amikor API-kat építünk, azt szeretnénk, hogy a modellek négy alapvető funkcionalitást nyújtsanak. A modellnek képesnek kell lennie arra, hogy létrehozzon (Create), olvasson (Read), frissítsen (Update) és töröljön (Delete) erőforrásokat. Egy modellnek képesnek kell lennie legalább ezekre a négy műveletre ahhoz, hogy teljes legyen. Ha egy művelet nem írható le egyik ezek közül a négy művelet közül, akkor potenciálisan egy új modell lehet.

A biztonsági szempontok fontosak a backend fejlesztésekor, például az autentikáció, az autorizáció és a JWT tokenek használata révén.

„<https://www.codecademy.com/article/what-is-crud>”

- **JWT token:** A JSON Web Token (JWT) egy kompakt, önálló és biztonságos módon átvihető token formátum, amelyet általában az autentikáció és az autorizáció során használnak. A JWT tokenek lehetővé teszik a biztonságos hitelesítést és az azonosítást a felhasználók számára a backend alkalmazásokban.

-JWT token: A JSON Web Token (JWT) egy kompakt, önálló és biztonságos módon átvihető token formátum, amelyet általában az autentikáció és az autorizáció során használnak. A JWT hitelesítés egy token-alapú állapotmentes hitelesítési mechanizmus. Ez általában ügyféloldali állapotmentes munkamenetként használatos, ami azt jelenti, hogy a szervernek nem kell teljesen támaszkodnia egy adattárolóra vagy adatbázisra a munkamenetinformációk mentéséhez.

A JWT-k titkosíthatók, de általában kódoltak és aláírtak. Az aláírt JWT célja nem a adatok elrejtése, hanem az adatok hitelességének biztosítása. És ezért nagyon ajánlott az HTTPS használata aláírt JWT-kkal.

A JWT három részből áll: fejlécből, adattartalomból és aláírásból.

-Fejléc(Header): A fejléc két részből áll: az algoritmus, amelyet a token aláírásához használnak, például SHA256 vagy RSA, valamint a token típusa, amely ebben az esetben JWT.

-Adattartalom(Payload): Az adattartalom a claim-eket tartalmazza. A claim-ek entitásspecifikusak, például átadhatunk felhasználói azonosítót, felhasználónevet vagy bármit, amit szeretnénk.

-Aláírás(Signature): Az aláírás az encoded header, az encoded payload és egy, csak a szerver által ismert titkos kulcs kombinációjából képződik. Azt szolgálja, hogy ellenőrizze a token integritását és biztosítsa, hogy ne lehessen manipulálni.

„<https://www.miniorange.com/blog/what-is-jwt-json-web-token-how-does-jwt-authentication-work/>”

„<https://medium.com/@ritikkhndelwal/jwt-authentication-in-the-frontend-enhancing-security-in-web-applications-ece418206b4f>”

Frontend

A frontend fejlesztéséhez használtuk a React keretrendszert. A React egy nyílt forráskódú JavaScript könyvtár, amelyet a felhasználói felületek készítésére használnak.

A frontend alkalmazás futtatásához és építéséhez használtuk a node.js-t. A node.js egy olyan JavaScript futtatókörnyezet, amely lehetővé teszi a JavaScript futtatását a szerveroldalon.

A kód szerkesztésére és a projekt menedzselésére a Visual Studio Code-ot használtuk. A Visual Studio Code egy ingyenesen elérhető forráskód szerkesztő és fejlesztői környezet, amelyet a Microsoft fejleszt.

Az autentikáció és az autorizáció megvalósításához használtuk a JWT (JSON Web Token) tokenet. A JWT egy nyílt szabvány, amelyet az információk biztonságos megosztására használnak két fél között – egy kliens és egy szerver között.

WPF

A Windows Presentation Foundation (WPF) egy grafikus felhasználói felületet (GUI) készítő keretrendszer, amelyet a Microsoft fejlesztett. A WPF lehetővé teszi a fejlesztők számára, hogy korszerű, testreszabható és esztétikus felhasználói felületeket hozzanak létre Windows alkalmazásokhoz. A WPF alapvetően XAML (eXtensible Application Markup Language) segítségével épül fel, amely lehetővé teszi a felhasználói felületek deklaratív leírását.

XAML: A WPF felhasználói felületek leírására használt nyelv, amely XML alapú. A XAML segítségével lehetőség van a felületek könnyű tervezésére és testreszabására.

Adatkötés: A WPF erőteljes adatkötési mechanizmust biztosít, amely lehetővé teszi az adatok dinamikus megjelenítését és frissítését a felhasználói felületen.

Stílusok és sablonok: A WPF lehetővé teszi a stílusok és sablonok alkalmazását a felhasználói felület egyes elemeire, így egyszerűen testreszabhatók a megjelenés és viselkedés szempontjából.

Animáció és átmenetek: A WPF lehetőséget biztosít a dinamikus animációk és átmenetek létrehozására, amelyek fokozzák a felhasználói élményt.

Eseménykezelés: A WPF lehetőséget biztosít a különböző események kezelésére és az ezekhez kapcsolódó műveletek végrehajtására.

Képfeldolgozás: A WPF lehetőséget biztosít a különböző képek és grafikák kezelésére, manipulálására és megjelenítésére.

4. Munkafolyamat

Adatbázis

A projekt adatbázisának tervezése során törekedtünk arra, hogy optimalizáljuk a tárolást és minimalizáljuk a redundanciát. Ennek érdekében két fő elvet alkalmaztunk:

3NM (Third Normal Form):

A harmadik normálforma alkalmazása lehetővé teszi számunkra, hogy a tábláinkat úgy tervezzük meg, hogy minimalizáljuk a redundanciát és ezáltal optimalizáljuk a tárolást. Ez azt jelenti, hogy a tábláinkban minden nemkulcs attribútum csak a kulcsoktól függ, és nincsenek tranzitív funkcionális függőségek. Így biztosítjuk, hogy az adatok egyszerűen és hatékonyan tárolhatók legyenek, miközben minimalizáljuk az anomáliákat és a felesleges adatmennyiséget.

Cascade törlés:

A Cascade törlés lehetőséget biztosít számunkra arra, hogy konzisztens állapotban tartsuk az adatbázist. Amikor egy rekordot törölünk az altáblából, automatikusan töröljük a hozzá tartozó rekordokat a fő táblából is. Ez segít elkerülni az adatbázisban a felesleges vagy elavult adatokat, és biztosítja az adatbázis integritását. Fontos azonban megjegyezni, hogy ez a művelet veszélyes lehet, hiszen sokszor nem törlik az adatokat, hanem csak jelzik, hogy törölt rekordként kezeljék. Ezért körültekintően kell eljárni a cascade törlés alkalmazásakor, hogy ne veszítsünk el fontos információkat az adatbázisban.

Az adatbázis tervezése során ezeket az elveket alkalmazva biztosítottuk, hogy az adatok hatékonyan tárolódjanak és könnyen kezelhetőek legyenek, miközben megőrizzük az adatbázis integritását és konzisztenciáját.

Backend

Alapelvek

A backend a rendszer olyan része, amely a felhasználói felülettől függetlenül működik. Ez azt jelenti, hogy nem kell minden egyes változtatást mind a frontend, mind a backend részen megváltoztatni, ami időt és erőforrásokat takarít meg. A kommunikációhoz használhatunk különböző módszereket, például az egyszerű HTTP-t vagy a biztonságosabb HTTPS-t. Az alkalmazást könnyen futtathatjuk Dockerrel vagy IIS szolgáltatás alatt, ami rugalmasságot biztosít a környezet kialakításában.

Adatkezelés

Az adatbázisban tárolt adatokhoz az Entity Framework-öt használjuk, ami lehetővé teszi az objektum-orientált adatkezelést. Fontos megjegyezni, hogy az Entity Framework nem engedi minden adatbázis-módosítást, különösen a kulcsok (elsődleges és idegen kulcsok) módosítását, amire figyelniük kell.

Kapcsolatok és API-k

A biztonságos kommunikáció érdekében a különböző platformok között CORS-t kell beállítanunk. Az adatok kezelését a CRUD műveletekkel végzünk (Create, Read, Update, Delete), amelyeket az API-k segítségével érünk el.

Adatátviteli Objektumok (DTO-k)

Az adatátviteli objektumok segítségével csak a szükséges adatokat küldjük el. Például, ha egy adott táblából nem minden információt szeretnénk elküldeni, használhatjuk a DTO-kat, ezáltal minimalizálva az adatok mennyiségét és növelve a kommunikáció hatékonyságát és biztonságát.

Felhasználói Azonosítás és Jogosultságok

Az autentikációt és az autorizációt a JWT (JSON Web Token) és BCrypt technológiákkal végzünk. A JWT token-alapú autentikációs megoldás, amely lehetővé teszi a felhasználók azonosítását és az információk biztonságos átvitelét.

A BCrypt jelszó-hashelési algoritmus biztosítja a jelszavak biztonságos tárolását és ellenőrzését, megvédve azokat a támadásoktól, például a brute force-tól és a rainbow támadásoktól. Ezáltal biztosítva, hogy csak azonosított felhasználók végezhesenek műveleteket, míg az adminisztrátoroknak különleges jogosultságok vannak a rendszerben.

Frontend

A weboldal fejlesztése a Visual Studio Code fejlesztői környezetben történik, mivel ez egy könnyen használható és sokoldalú kódszerkesztő, amely támogatja a modern webfejlesztési technológiákat. Bővítményei (extensions) rendkívül hasznosnak bizonyultak a munkafolyamat során. Csak az alább említett reacthoz 3 különböző, bővítményt használtunk. Az auto save funkció pedig rendkívül gyorsítja

Telepített Node.js Komponensek

A projekt működéséhez elengedhetetlenül szükséges Node.js komponensek telepítése a parancssorban történik. Ezek a komponensek biztosítják a fejlesztők számára a szükséges környezetet és eszközöket a frontend kód írásához és futtatásához.

Előre megírt Node komponensek: A projekt specifikus igényei alapján válogattuk össze azokat a Node.js csomagokat, amelyek szükségesek a projektben használt funkciók és követelmények megvalósításához.

Node verzió: A projekt szempontjából fontos, hogy a megfelelő Node.js verziót használjuk, hogy biztosítsuk a kompatibilitást és a stabilitást. Az akkor még legfrissebb verziót installáltuk, amely a v20.10.0-as kiadás.

Bootstrap

A Bootstrap keretrendszert választottuk a weboldal megjelenítésének és a reszponzív design kialakításának céljából. A Bootstrap egy jól ismert és elismert CSS keretrendszer, amely lehetővé teszi a gyors és egyszerű reszponzív weboldalak készítését.

Megjelenítés: A Bootstrap segítségével könnyen létrehozhatunk vonzó és esztétikus felhasználói felületeket, anélkül, hogy túl sok időt kellene fordítanunk a CSS stílusok írására.

Responsive: A Bootstrap használatára ösztönző tulajdonság, hogy a Bootstrap alapú weboldalak reszponzív jellege garantálja, hogy azok megfelelően jelenjenek meg és működjenek különböző eszközökön és képernyőméreteken, így javítva a felhasználói élményt, valamint használata a munkafolyamatot is gyorsítja.

React

A React keretrendszert választottuk a weboldal frontendjének fejlesztéséhez, mivel ez egy hatékony és korszerű eszköz a dinamikus felhasználói felületek készítéséhez és kezeléséhez.

Projekt Készítése: A React segítségével könnyen létrehozhatunk egy új projektet a `create-react-app` paranccsal, amely előre beállítja a projekt struktúráját és inicializálja a szükséges fájlokat.

Build, Start, Fejlesztés

A React alkalmazás fejlesztése során a szokásos műveleteket végezzük, beleértve a projekt buildelését, a helyi fejlesztői szerver indítását és a változtatások figyelését és újrafordítását.

Előnyök és Hátrányok: A React keretrendszernek számos előnye van, mint például a könnyű tanulhatóság és az újrafelhasználható komponensek, ugyanakkor néhány hátrány is lehet, például a komplexitás és a kezdeti beállítások bonyolultsága.

Telepített Komponensek

A projekt során telepítettünk különböző kiegészítő modulokat, például az Axios-t, amely segítségével könnyen kommunikálhatunk a szerverrel aszinkron HTTP kérésekkel. Különböző megjelenítéshez szükséges komponenseket is töltöttünk le, pl. a `react-bootstrap`, `Autosuggest`, `mdc-react-ui-kit`, stb.

Nagyon fontos volt még a react-router-dom amely lehetővé teszi dinamikus útvonalak megvalósítását egy webalkalmazásban. Lehetővé teszi az oldalak megjelenítését és a felhasználók navigálását közöttük. A React Router Domot arra használják, hogy egyoldalas alkalmazásokat hozzanak létre, azaz olyan alkalmazásokat, amelyeknek sok oldala vagy komponense van, de az oldal soha nem frissül, hanem a tartalom dinamikusan lekérdezésre kerül az URL alapján.

Tesztelés: A projekt fejlesztése során rendszeres teszteléseket végeztünk, beleértve a komponensek és funkciók, integrációs tesztjeit és felhasználói felület tesztjeit is.

WPF

Az alkalmazás célja lehetővé tenni az események és felhasználók kezelését, valamint biztosítani egy biztonságos bejelentkezési mechanizmust az adminisztrátori jogosultságokhoz.

Funkciók

Bejelentkezés

A felhasználók bejelentkezhetnek az alkalmazásba a felhasználónevével és jelszavával.

A bejelentkezési adatokat ellenőrzi az alkalmazás az adatbázisban tárolt adatok alapján.

Sikeres bejelentkezés esetén az adminisztrátori felületre irányítja a felhasználót.

Adminisztrátori Felület (AdminWindow)

Az adminisztrátori felületen az adminisztrátor további tevékenységeket végezhet el.

Lehetőség van az események és felhasználók kezelésére.

Eseménykezelés (EventWindow)

Az eseményeket lehet megtekinteni és törölni az EventWindow ablakon.

Az események betöltődnek az adatbázisból és megjelennek egy ComboBox-ban.

A felhasználók az egyes eseményeket kiválaszthatják, majd törölhetik őket az alkalmazásból.

Felhasználókezelés (UserWindow)

A felhasználókat lehet megtekinteni és törölni a UserWindow ablakon.

Lehetőség van egy felhasználót adminisztrátorrá tenni.

Adatbázis és Adatkezelés

Az alkalmazás MySQL adatbázist használ az adatok tárolására.

A kapcsolódáshoz és az adatlekérdezésekhez szükséges SQL műveleteket a megfelelő C# osztályok végzik.

A felhasználói jelszavakat biztonságosan tárolják a BCrypt hashelés segítségével.

Hibakezelés és Biztonság

Az alkalmazás különböző hibákat kezel, beleértve az adatbázis kapcsolat hibáit és a felhasználói input validálását.

A felhasználói jelszavakat biztonságosan tárolják hashelt formában, hogy megakadályozzák a jelszavak illetéktelen hozzáférését.

Továbbifejlesztési lehetőségek

Az alkalmazás további funkciókkal bővíthető, például részletes esemény- és felhasználói adatlapokkal.

Az UI/UX javítása érdekében újabb dizájn elemek bevezetése lehetséges.

A kód optimalizálása és struktúrájának további tisztázása a karbantarthatóság és kiterjeszthetőség érdekében.

Megjegyzések

Az alkalmazás jelenlegi verziója a legfontosabb alapfunkciókat valósítja meg.

Az alkalmazás főbb komponensei (MainWindow, AdminWindow, EventWindow, UserWindow) külön osztályokba vannak szervezve a jobb karbantarthatóság és olvashatóság érdekében.

5. Webalkalmazás bemutatása és használata

1. Főoldal

- a. Home komponens**
- b. Navbar**
- c. Sidebars Menük**

2. Profil menü komponensek

3. Esemény komponensek

6. Kommunikációs eszközök

A csapat hatékony működéséhez elengedhetetlenek voltak az olyan kommunikációs eszközök, amelyek lehetővé teszik a gyors és strukturált információ-, és fájláramlást, verziókezelést, valamint a közös munkát és annak vázolását. Az alábbiakban öt ilyen eszközt használtunk fel projektmunkánk során.

Trello

A Trello egy digitális kanban tábla, amely lehetővé teszi a feladatok és projekt részleteinek vizuális szervezését és nyomon követését. A felhasználók egyszerűen hozzáadhatnak kártyákat különböző listákhoz, és mozgathatják azokat az előrehaladásuknak megfelelően. Használata átláthatóvá teszi a feladatok elosztását és követését, valamint elősegíti a csapatmunkát és a hatékony projektmenedzsmentet.

Github

A Github egy online verziókezelő platform, amely lehetővé teszi a fejlesztők számára a projektforrások verzióinak nyomon követését és kezelését. A Git alapú rendszer lehetővé teszi a kódbázisok közös munkáját, a változtatások kezelését és azok visszavonását. A pull requestek és ágak segítségével a fejlesztők könnyen együtt dolgozhatnak, és az alkalmazás fejlesztése strukturált és nyomon követhető marad.

Messenger

Az üzenetküldő alkalmazások, mint például a Messenger, azonnali kommunikációt biztosítanak a csapattagok között. Ez lehetővé teszi az egyszerű kérdések gyors megválaszolását, az ötletek megbeszélését és azonnali visszajelzést a projektekkel kapcsolatban. Azonnali és informális kommunikációs csatornaként szolgálhat, ami felgyorsíthatja a döntéshozatalt és a problémamegoldást.

Discord

A Discord egy olyan kommunikációs platform, amely lehetővé teszi a csapatok számára a szöveges, hangos és video alapú kommunikációt. Különböző szerverek létrehozásával és csatornák létrehozásával a csapatok könnyen szervezhetik és kategorizálhatják a kommunikációt a projektjeikhez kapcsolódóan. A Discord rugalmas platformként szolgál a csapatoknak, ahol egyszerre lehetőség van informális beszélgetésekre és hivatalos megbeszélésekre is.

Google Drive

A Google Drive egy felhőalapú dokumentumtároló és megosztó platform, amely lehetővé teszi a felhasználók számára a dokumentumok, prezentációk, táblázatok és egyéb fájlok tárolását és közös munkáját. A csapatok könnyen megoszthatják és egyidejűleg szerkeszthetik a fájlokat, valamint könnyen hozzáférhetnek hozzájuk bármilyen eszközről. A Google Drive hatékony eszköz a dokumentumok megosztásához és együttműködéshez, amely elősegíti a csapatok hatékonyabb munkáját és a projektmenedzsmentet.

Ezek az eszközök számos előnnyel járnak a csapatok számára, segítve őket a hatékonyabb kommunikációban, együttműködésben és projektmenedzsmentben. A megfelelő eszközök kiválasztása és alkalmazása kulcsfontosságú a sikeres csapatmunka és projektvezetés szempontjából.

7. Összefoglalás

Célkitűzések és teljesítések:

Összességében a projekt a célja megvalósult, legyen eseménykezelő, oldal ahol felhasználó találhat magának programot, ahol közösségekhez csatlakozhat, ahol

9.Melléklet

10. Forráslista