**A SubPart fejlesztői csapat projektfeladata:**

**A Trillance**

A Trillance-t egy olyan programnak képzeltük el, ami képes kielégíteni minden korosztályú, nemű, kultúrájú, és hangulatú ember igényét, hogy megtalálhassa a számára legmegfelelőbb zenét. Szerettünk volna egy diszkrét megoldást arra, hogy tudjunk a saját letöltött zenekönyvtárunkból, ingyenesen zenét streamelni az eszközeinkre, minimális helyigénnyel. Igyekszünk a szolgáltatásunkban nyújtott zene könyvtárat folyamatosan bővíteni, így nem maradhat le az előadók legfrissebb zenéiről. Szeretnénk, ha a zenét hallgató felhasználó személyre szabhatná a megjelenési felületet, valamint a hangszínt. Ezekhez szükséges lesz fejlesztenünk egy weboldalt, ami egy adatbázishoz van kötve. Szeretnénk, ha minden eszközzel kompatibilis lenne weboldalunk. A fejlesztéshez használt programok és programnyelvek részletesebben:

**Microsoft Visual Studio Code:** Ez az egyik leggyakrabban használt felhasználói környezet az általa támogatott széleskörű programnyelvek és operációs rendszereknek köszönhetően. A fejlesztők kedvükre bővíthetik a programot, hogy minél egyszerűbbé tehesse a munkájukat. Azért is szeretik, mert ezt bárki ingyen letöltheti, és használhatja. Tartalmazza az általunk használt verziókezelőt, a Git-et is. Ebben a programban írjuk meg a HTML, CSS, JavaScript, és React részét projektünknek.

**Microsoft Visual Studio:** A nevéből következtethetően ezt a terméket is a Microsoft fejlesztette, ez a fejlesztői környezet a **Code** változathoz képest inkább weblapokhoz képest asztali és mobil alkalmazások fejlesztésére alkalmas. Rendelkezik a .Net keretrendszer összes verziójával. Sok mindenben hasonlít a **Code** változathoz, de ennek részletesebb a hibakezelése, és ez a program sajnos csak a Windows operációs rendszerrel használható. Rendelkezik grafikus tervezőkkel is, ami hasznos, ha nem csak a kódot szeretnénk látni, szerkeszteni. Ebben írjuk projektünk ASP.NET Web API részét, és a Xamarin-t.

**XAMPP:** A XAMPP egy szoftvercsomag, amiben Apache helyi szervert futtatunk, és a MySQL adatbázis-kezelő programmal, a phpMyAdmin felületén keresztül készítjük el projektünk adatbázisát, annak tábláit, az adatszerkezetet, és itt is készíthetünk nézeteket lekérdezésekkel.

**Hypertext Markup Language(HTML):** A HTML egy jelölő nyelv, amit weblapfejlesztésre használunk, a világ összes weboldalához nélkülözhetetlen a HTML használata. Ezt a nyelvet már 1991. óta használják, azóta már rengeteg fejlődésen ment keresztül, a jelenlegi legújabb verzió a HTML 5. Célja, a weboldal alap szerkezetének kivitelezése, és tartalmának összesítése. A HTML-t szinte mindig összekapcsoljuk más nyelvekkel, például a Cascading Style Sheets(CSS)-el, vagy a JavaScript-el. Ezekkel "életre kelthetjük" weboldalunkat.

**Cascading Style Sheets(CSS):** A CSS egy stílusleíró nyelv, mely alkalmas a korábbiakban említett HTML weboldal megjelenésének elkülönítése nélkül. Ezzel sokkal egyszerűbben lehet az elemek stílusát változtatni, melyeket lehet elemtípus, osztály, és identitás alapján szűrni. A CSS-re hivatkoznunk kell pontos elérési útvonallal a HTML állományban, ha a CSS nem a HTML fájllal megegyező mappában van. Mostanság a Bootstrap helyettesítheti a CSS-t, mivel egyszerűbb díszes módosításokat gyorsan végrehajtani.

**JavaScript(JS):** A JavaScript egy programozási nyelv, melyet általában weboldal fejlesztéskor szoktunk használni. Sokan gondolják, azonban a JavaScript-nek nincs semmi kötődése a Java programozási nyelvhez. A JS segít megvalósítani a reszponzivitást, az interaktív funkciókat, mint például egy csúszka, vagy egy animáció. Lehetővé teszi azt is, hogy ne kelljen mindig új weboldalt betölteni a tartalom megváltozásakor, tehát egybe lehet ágyazni több weboldalt is. Az utóbbi három nyelv nélkül, vagy valamilyen alternatívájuk nélkül a weboldal nem számít elég fejlettnek, mert nem képes minden funkciót megvalósítani, amit egy manapság használt híresebb weboldal igen.

**React:** A React az egy JavaScript könyvtár, ami jelentősen megkönnyíti a weblapfejlesztést. Ez a Facebook által fejlesztett könyvtár magába foglalja az eddig említett nyelveket, így nincs szükségünk összekötni külön mind a három állományt. Ez a nyelv deklaratív, ami azt jelenti, hogy ellentétben az eddig felsorolt imperatív nyelvekkel, itt nem szükséges lépésről lépésre leírni a számítógép utasításait, itt elég azt leírni, hogy mi legyen a végeredmény. Természetesen mindezt a React szintaktikáján belül kell kivitelezni.

**MySQL:** Ez egy adatbázis-kezelő rendszer, ami relációs adatbázisok létrehozására, módosítására, és adatok lekérdezésére alkalmas. A MySQL szerverként használható, sok különböző felületen elérhető ez a szerver, de a leggyakrabban használt felület az a phpMyAdmin. Az adatbázissal rendelkező alkalmazásunkhoz nélkülözhetetlen egy adatbázis-kezelő rendszer használata, mivel segít az adatok közötti kapcsolatok felismerésében, a hozzáféréskezelésben, és a megfelelő adatok megtalálásában lekérdezésekkel. Az alkalmazásunk, és az adatbázis között az Objektum-relációs leképzés(ORM) biztosít egy köztes nyelvet, amit mindkét nyelv le tud fordítani. Mi a .NET Entity Framework-öt használjuk.

**ASP.NET Web API:** Ez egy keretrendszer, amely meg tud adatokat osztani a felhasználó böngészőjével. Az ASP.NET Web API képes a kliens által a webszerver felé küldött HTTP kéréseit kezelni és feldolgozni. Sokkal könnyebben meg tudjuk előzni vele a hacker támogatásokat, mint például az SQL injekció. Ez a Representational State Transfer(REST)-en alapul, JSON fájlt is tud kezelni. Az ASP.NET Web API működését a Swagger eszközkészlet segíti.

**Xamarin:** A Xamarin egy olyan keretrendszer, mely képes átalakítani számítógépre írt programokat kódbázis segítségével úgy, hogy az natív Android alkalmazássá váljon. A Xamarin használatával a szoftverfejlesztőknek nem kell annyi erőfeszítést belefektetni mindkét verzió frissítésébe, és karbantartásába.