Miskolci Egyetem

Gépészmérnöki és Informatikai Kar

Általános Informatikai Intézeti Tanszék

A képen szimbólum, embléma, Betűtípus, Védjegy látható

Automatikusan generált leírás

**Adminisztráló és oktató program**

**Szakdolgozat**

**Készítette:**

**Név:** Oláh Mózes

**Neptun-kód:** D9WPR7

**Szak:** Mérnökinformatikus BSc

Informatikai rendszermérnök szakirány

**Témavezető:** Smid László

**Tartalomjegyzék**

**Bevezetés**

A mai üzleti környezetben a hatékonyság és az automatizálás a siker elengedhetetlen összetevőjévé vált. Az adminisztratív feladatok, amelyek kötelezőek, rengeteg időt és erőforrást emésztenek fel, emiatt tökéletesek arra, hogy automatizáljuk őket. Az egyik ilyen feladat a környezetvédelmi, egészségügyi és biztonsági (EHS) képzés, amely számos szervezetnél kötelező. Az EHS képzéseket hagyományosan papíralapú módszerekkel, televízión vagy kivetítőn tartott előadásokkal, majd az oktatók által kézzel kijavított írásbeli tesztekkel végezték.

Szakmai gyakorlatom során azt a feladatot kaptam, hogy a jelenlegi képzési folyamatot korszerűbbé tegyem és egy alternatív módon valósítsam meg. Ennek eredményeként szakmai gyakorlatom témája egy olyan webalkalmazás elkészítése, amely teljes mértékben kiváltja a jelenlegi papíralapú elavult módszert. Az alkalmazás átalakítja a jelenlegi oktatási formát, sokkal egyszerűbbé, gyorsabbá és költséghatékonnyá téve a folyamatot.

Az ötlet abból a felismerésből született, hogy a hagyományos EHS képzési módszerek nemcsak idő- és költségigényesek, hanem nehézkesek és gyakran nem hatékonyak. Az automatizálás és digitalizáció előnyei inspirálták az elképzelést, hogy egy webalkalmazással helyettesítsük a régi módszereket, ezáltal javítva a képzés minőségét és hatékonyságát.

A projekt fő célja egy olyan webalkalmazás létrehozása, amely hatékonyabbá, gyorsabbá, pontosabbá és könnyebbé teszi az EHS képzést. A cél, hogy az alkalmazás felhasználóbarát és jól kialakított legyen, hogy a felhasználók könnyen és hatékonyan használhassák.

Az alkalmazás működéséhez számos követelményt kell figyelembe venni. Fontos, hogy a használt technológiák és programozási nyelvek korszerűek és széles körben támogatottak legyenek. Az adatstruktúra felépítése szintén kritikus szempont, mivel ez biztosítja az alkalmazás megbízhatóságát és hatékonyságát.

A szakdolgozatomban részletesen kitérek a webalkalmazás fejlesztési folyamatára, az alkalmazás működésére és az adatstruktúra felépítésére. Vizsgálni fogom a különböző technológiákat, programozási nyelveket és keretrendszereket, amelyek elengedhetetlenek ahhoz, hogy egy felhasználóbarát és jól kialakított alkalmazást készítsünk.

**Háttér információk**

**Digitális oktatás**

A digitális oktatás gyökerei az 1960-as évekig nyúlik vissza, amikor a számítógépek először kezdtek elterjedni az oktatási intézményekben. Az első számítógépes oktatási programok főként egyetemeken és kutatóintézetekben jelentek meg, ahol a technológia fejlődésével egyre több lehetőség nyílt a digitális tananyagok és oktatási eszközök kifejlesztésére.

Az 1980-as években a számítógépek megjelenése új lendületet adott a digitális oktatásnak. Az iskolák és a vállalatok kezdtek érdeklődni a számítógépes alapú oktatási programok iránt, amelyek lehetővé tették az interaktív tanulást és a tananyagok könnyebb hozzáférését. Az első e-learning rendszerek ebben az időszakban jelentek meg, és bár kezdetben korlátozott funkcionalitással rendelkeztek, megalapozták a későbbi fejlesztések alapjait.

A 1990-es évek végén és a 2000-es évek elején az internet elterjedése forradalmasította a digitális oktatást. Az online tanulási platformok és virtuális osztálytermek megjelenése lehetővé tette, hogy a diákok és munkavállalók a világ bármely pontjáról hozzáférhessenek a tananyagokhoz. Az első ilyen platformok, mint például a Blackboard és a Moodle, gyorsan népszerűvé váltak az oktatási intézmények és a vállalatok körében.

A 2010-es években az e-learning széles körben elterjedtté vált, köszönhetően a technológiai fejlődésnek. A mobil eszközök, például tabletek és okostelefonok megjelenése pedig lehetővé tette, hogy az emberek bárhol és bármikor tanulhassanak. Az olyan platformok, mint a Coursera, edX és Udemy, milliók számára tették hozzáférhetővé a világ vezető egyetemeinek és kutatóiknak a tananyagait.

Az utóbbi években, különösen a COVID-19 járvány idején, a digitális oktatás még nagyobb jelentőséget kapott. Az iskolák, egyetemek és vállalatok gyorsan átálltak az online oktatásra, hogy biztosítsák a folyamatos tanulást és képzést. A digitális oktatás ma már szerves része az oktatási rendszernek és a munkahelyi képzési programoknak, és várhatóan a jövőben is egyre fontosabb szerepet fog játszani.

**Népszerű e-learning felületek**

**Blackboard**

A Blackboard egy vezető e-learning platform, amelyet világszerte használnak egyetemek, főiskolák és vállalatok. Az oktatók számára lehetőséget biztosít tananyagok feltöltésére, online órák tartására és a tanulók teljesítményének nyomon követésére. Széles körű funkciókat biztosít, integrálható más rendszerekkel, és erős támogatást nyújt a tanárok és tanulók számára. Hátránya, hogy bonyolult a felhasználói felülete, magasak a költségei, és lassú a működése.

**Moodle**

A Moodle egy nyílt forráskódú tanulásmenedzsment rendszer, amelyet sok oktatási intézmény és vállalat használ. Lehetővé teszi az oktatók számára tananyagok létrehozását és megosztását, interaktív tesztek és fórumok használatát, valamint a tanulók haladásának követését. Ingyenes és nyílt forráskódú, valamint könnyen testre szabható. Azonban néha bonyolult lehet a beállítása és karbantartása, nem mindig felhasználóbarát.

**Coursera**

A Coursera egy online oktatási platform, amelyet vezető egyetemek és vállalatok támogatásával hoztak létre. Lehetővé teszi a felhasználók számára, hogy online kurzusokat végezzenek különféle témákban, beleértve a tudományt, technológiát, művészeteket. Magas minőségű kurzusokkal rendelkezik, különböző témákban. Viszont a költségek magasak lehetnek, és a platform kevésbé testre szabható.

**edX**

Az edX egy nonprofit online tanulási platform, amelyet a Harvard és a MIT indított el. Számos ingyenes és fizetős kurzust kínál különféle témákban, lehetőséget adva tanúsítványok és diplomák megszerzésére. Kiváló minőségű kurzusokkal rendelkezik, széles választékban. Hátránya, hogy bizonyos kurzusoknak magas a költsége és a felületen előfordulnak a technikai problémák.

**Udemy**

Az Udemy egy globális piac, ahol bárki létrehozhat és értékesíthet online kurzusokat. Sokféle kurzust kínál szinte minden témában, beleértve a szakmai és személyes fejlesztést. Nagy választékkal rendelkezik, megfizethető árakkal, gyakoriak a kedvezmények és élethosszig tartó hozzáférést biztosít a megvásárolt kurzusokhoz. Hátrányai, hogy változó minőségűek a kurzusok, ezáltal kevésbé elismert.

**Webalkalmazás szerepe**

**Cég életében**

A cég életében jelentős változást hoz a webalkalmazásom bevezetése, amelynek fő célja, hogy hatékonyabbá és megbízhatóbbá tegye a képzési folyamatokat. Korábban a vállalat egy saját fejlesztésű programot használt tabletekre, azonban annak instabil működése – gyakori lefagyások és váratlan kilépések – akadályozták a zökkenőmentes oktatást. Ennek következtében a cég visszatért a papír alapú oktatási formához és a TV-n keresztüli kivetítéshez, ami időigényes és kevésbé hatékony. A webalkalmazásom több szempontból is jelentős újítást hoz:

1. **Stabilitás és megbízhatóság:** Az új rendszer megbízhatóan működik, elkerülve a korábbi problémákat, így biztosítva a folyamatos és zökkenőmentes oktatást.
2. **Digitális innováció:** Az alkalmazás lehetővé teszi a papíralapú módszer teljes digitalizálását, ami nemcsak környezetbarát, hanem idő- és költséghatékony is.
3. **Könnyű hozzáférhetőség:** Az alkalmazásom a vállalati hálózaton belül bárhonnan elérhető, így a munkavállalók kényelmesen, saját ütemükben végezhetik el a képzéseket.
4. **Adatkezelés és nyomon követés:** Az adminisztrációs felület segítségével az EHS (Environment Health and Safety) oktatásért felelős csoport könnyedén kezelheti, nyomon követheti és elemezheti a képzések eredményeit és a munkavállalók fejlődését.

**Célcsoport**

A webalkalmazás fő célcsoportjai közé tartoznak:

1. **Új Munkavállalók:** Azok az újonnan érkező munkavállalók, akiknek kötelező képzéseken kell részt venniük. Az alkalmazás biztosítja számukra a szükséges oktatási anyagokat és teszteket, hogy gyorsan és hatékonyan elsajátíthassák a szükséges ismereteket.
2. **Cég dolgozói:** Azok a munkavállalók, akiknek lejárt a képzésük érvényességi ideje és újra kell végezniük a kötelező tanfolyamokat. Az alkalmazás lehetőséget biztosít számukra, hogy bármikor hozzáférjenek a frissítő képzésekhez.
3. **EHS Oktatási Csoport:** Az alkalmazás külön adminisztrációs felülete az EHS oktatásért felelős csoport számára készült, amely lehetővé teszi számukra a képzések kezelését, a felhasználók nyomon követését és az eredmények elemzését.

**Következtetés**

Az én alkalmazásom kifejezetten a cég számára készült, emiatt pontosan megfelel az adott igényeknek. Ellentétben a Blackboard, Moodle, Coursera, edX és Udemy általános megoldásaival, az én rendszerem célzottan egy specifikus képzésre van optimalizálva, ami nagyobb hatékonyságot biztosít. Mivel az oktató felületem egy bizonyos képzéshez készült, nincs szükség a nagyobb, komplexebb platformok megvásárlására, amelyek felesleges funkciókat tartalmazhatnak. Ezáltal jelentős költségmegtakarítás érhető el, mind a licencdíjak, mind a fenntartási költségek terén. A rendszerem egyszerű és könnyen használható felületet biztosít. Mivel az alkalmazás saját fejlesztés emiatt a technikai támogatása és karbantartása gyorsabb és rugalmasabb.

**Cég bemutatás**

Korábban már említettem, hogy a szakdolgozatom témája a szakmai gyakorlatom során alakult ki, amit a HELL ENERGY Magyarország Kft-nél töltöttem. A cég 2006-ban Magyarországon alakult meg. A gyors növekedéséről ismert, ennek köszönhetően Magyarországon és a világpiacon is jelentős szereplővé vált és közel 50 országba importál termékeket, amelyek közül például Romániában és Szlovákiában is vezető szerepet tölt be.

A HELL ENERGY termék portfóliója kellően változatos, egyedi ízeket tartalmaz, amely sok ember számára csábító. Zászlóshajó termékük a HELL Classic de ezen kívül elérhetőek cukormentes, koffeinmentes, vitaminnal dúsított és magasabb koffeintartalmú termékek is. Emellett rendelkezik kávés termékcsaláddal is, ami szintén nagyon népszerű.

A vállalat sikere a merész marketing stratégiájában rejlik és a magasabb szintű szponzorációkban. Ebbe tartozik például a Forma-1-es csapatokkal és a legjobb futball klubokkal kötött partnerségük, de akár nézhetjük a híres színészekkel kötött szerződéseket is. Ezek az erőfeszítések emelték meg a márka vonzerejét és ismertségét főként a fiatalabb fogyasztók körében.

A képen szöveg, embléma, Grafika, clipart látható

Automatikusan generált leírás

**Előkészületek**

A megfelelő webalkalmazás létrehozásához szükséges ismerni néhány előfeltételt, ami alapján létrehozható. Többek között, hogy kik fogják használni, milyen célra fogják használni, milyen szerepet fog betölteni és milyen környezetben fog működni. Ahogy már fentebb említettem, az alkalmazásom egy cég számára készült, azon belül is javarészt az újonnan érkező munkavállalók képzéséért lesz felelős. Mivel egy cégnél dolgoznak szellemi és fizikai munkát végző emberek emiatt eltérő lehet az informatikai képzettségük. Emiatt szükséges, hogy a program minden igényt kielégítsen és ne kelljen hozzá különösebb számítástechnikai tudás. Egyszerű, egyértelmű és egyirányú működést kell biztosítania. A felhasználását illetően a cégnél már erős infrastruktúra van kialakítva ezáltal a lokális hálózaton könnyűszerrel futtatható.

Megvalósítás

Egy webalkalmazás megvalósításához a mai digitalizált világban számtalan lehetőség áll rendelkezésre. Kezdve a szerver és a háttér megoldásoktól a kliens oldali látható megoldásokig. A backend fejlesztéséhez rendelkezésre állnak különböző programozási nyelvek, például a JavaScript, Ruby, PHP, C# és Java. Az adatbázisok szempontjából nézve elérhetőek relációs adatbázisok például a MySQL, PostrgeSQL vagy az SQLite amelyek SQL-t használnak adatbázis kezelésére, de léteznek NoSQL adatbázisok is mint a MongoDB, Firebase vagy a CouchDB amelyek a séma nélküli adattárolása szolgálnak. Különböző fejlesztői eszközök és felületek is rendelkezésre állnak, mint a verziókezelő rendszerek, konténerizációs felületek és a felhő szolgáltatások. A frontend fejlesztéséhez is különböző technológiák állnak rendelkezésre beleértve a HTML-t amely a weboldalak és webes alkalmazások készítésének a szabványos jelölő nyelve, a CSS amely a HTML-ben készített dokumentumok formázásáért felelős, a JavaScript amely a dinamikus tartalmak megjelenítéséért felel. Ezen alap típusokra épülnek a nagyobb keretrendszerek és könyvtárak, mint az Angular, Vue.js vagy a React amelyek mind JavaScript keretrendszerek.

**Fejlesztőeszközök**

Adatbázis

Az adatbázis strukturált információk vagy adatok szervezett gyűjteménye, amelyeket általában elektronikusan tárolnak egy számítógépes rendszerben. Lehetővé teszi az adatok tárolását, visszakeresését és kezelését.

MySQL

A MySQL egy nyílt forráskódú relációs adatbázis-kezelő rendszer (RDBMS), amelyet széles körben használnak webes alkalmazásokhoz és online adattároláshoz. A megbízhatóságáról, egyszerű használatáról és teljesítményéről ismert. Többféle platformot támogat, köztük Windows, Linux és macOS rendszereket. A webfejlesztők gyakran választják ezt az adatbázist. A MySQL robusztus funkciókészletéhez tartozik az indexelés az összetett lekérdezések, az és az ACID (Atomicity, Consistency, Isolation, Durability) megfelelés támogatása, amely biztosítja az adatok integritását és megbízhatóságát. A MySQL emellett nagy közösséggel rendelkezik és részletes dokumentációja van, így a kezdők és tapasztalt fejlesztők számára egyaránt jó választás.

PostgreSQL

A PostgreSQL egy fejlett, nyílt forráskódú RDBMS, amely a bővíthetőségre és az SQL-megfelelőségre helyezi a hangsúlyt. Támogatja az összetett lekérdezéseket, idegen kulcsokat, triggereket, nézeteket és tranzakciós integritást, így előnyös választás a robusztus és összetett adatkapcsolatokat igénylő alkalmazásokhoz. A PostgreSQL nagy adathalmazok kezelésére való képessége és a különböző adattípusok - köztük a JSON, az XML és az egyéni adattípusok - támogatása miatt sokféle alkalmazáshoz megfelelő, a webes szolgáltatásoktól a tudományos adatelemzésig. Gazdag funkciókészletéhez tartozik továbbá a teljes szöveges keresés, a földrajzi adatok támogatása a PostGIS segítségével, valamint a fejlett indexelési technikák, így kínálva hatékony eszközt a sokoldalú és skálázható adatbázis-megoldást kereső fejlesztők számára.

MongoDB

A MongoDB egy vezető NoSQL adatbázis, amely rugalmasságáról és skálázhatóságáról ismert. A hagyományos relációs adatbázis-kezelő rendszerekkel (RDBMS) ellentétben a MongoDB JSON-szerű dokumentumokban tárolja az adatokat, ami dinamikus és hierarchikus adatstruktúrákat tesz lehetővé. Ez különösen alkalmassá teszi olyan alkalmazásokhoz, amelyek nagy mennyiségű adatot, valós idejű elemzést kezelnek, és rugalmas séma kialakítást igényelnek. Kkiválóan kezeli a nagy mennyiségű strukturálatlan adatot, ezért népszerű választás a modern webes alkalmazásokhoz. A MongoDB emellett támogatja a horizontális skálázást a sharding révén, így hatékonyan kezelheti a nagyobb méretű, nagyobb forgalmú alkalmazásokat. Gazdag lekérdezési nyelve, robusztus aggregációs keretrendszere és a több dokumentumot tartalmazó ACID tranzakciók támogatása tovább növeli vonzerejét a változatos és összetett projekteken dolgozó fejlesztők számára.

Backend keretrendszer

A backend, más néven szerveroldal, a webes alkalmazás azon része, amely a logikát, az adatbázis-interakciókat, a hitelesítést, a szerver konfigurációját és más, a színfalak mögötti műveleteket kezeli. Felelős a frontendről (ügyféloldalról) érkező kérések feldolgozásáért, az üzleti logika végrehajtásáért és a megfelelő adatok vagy válaszok visszaküldéséért. A backend jellemzően egy szervert, egy adatbázist és egy alkalmazást foglal magában, és biztosítja, hogy minden megfelelően működjön együtt. Alapvető fontosságú az adattárolás, a biztonság és a teljesítményoptimalizálás kezelésében, így minden webes vagy mobilalkalmazás létfontosságú összetevője.

Spring Boot:

A Spring Boot egy hatékony, funkciókban gazdag keretrendszer Java-alapú alkalmazások építéséhez. A Spring Framework nagyobb ökoszisztémájának része, és a konfiguráció helyett a konvenciót biztosító megközelítéssel egyszerűsíti a fejlesztési folyamatot. A Spring Boot számos beépített eszközt és funkciót kínál, amelyek segítségével a fejlesztők önálló, gyártásra kész alkalmazásokat hozhatnak létre minimális konfigurációval.

Jellemzői:

1. Beágyazott szerverek: Olyan beágyazott szervereket tartalmaz, mint a Tomcat, a Jetty és az Undertow ezáltal az alkalmazások könnyedén üzembe helyezhetőek.
2. Automatikus konfiguráció: Automatikusan konfigurálja az alkalmazást a hozzáadott függőségek alapján, csökkentve a forráskódot és a beállítási időt.
3. Mikroszolgáltatások támogatása: Kitűnő támogatást nyújt a mikroszolgáltatások létrehozásához és telepítéséhez.
4. Robusztus ökoszisztéma: Tökéletesen integrálódik a Spring szélesebb ökoszisztémájába, hozzáférést biztosít a könyvtárak és eszközök sokaságához.

Laravel

A Laravel egy PHP keretrendszer, amelyet arra terveztek, hogy elegáns szintaxisával és fejlesztőbarát eszközrendszerével növelje a webfejlesztés hatékonyságát. Egyszerűsíti az olyan feladatokat, mint az útválasztás, a hitelesítés, a munkamenetek és a gyorsítótárazás, miközben tiszta, moduláris architektúrája révén elősegíti a karbantarthatóságot és a skálázhatóságot.

Jellemzői:

1. Eloquent ORM: Megkönnyíti az adatbázisokkal való interakciókat a modellek és kapcsolatok egyszerű PHP-szintaxisával.
2. Blade sablonmotor: Könnyű, nagy teljesítményű sablonozó motort kínál a tiszta, olvasható nézetekhez.
3. Artisan parancssori eszköz: A fejlesztési feladatokat egyszerűsíti az adatbázis-migrációra és az ütemezésre vonatkozó parancsokkal.
4. Átfogó ökoszisztéma: Tartalmazza a telepítést (Laravel Forge), a feladatütemezést (Horizon) és a teljes szöveges keresést (Scout) ezzel növelve a produktivitást.

Django

A Django egy magas szintű Python webes keretrendszer, amely gyors fejlesztésre és tiszta, pragmatikus tervezésre törekszik. A „batteries-included” filozófiát követi, ami azt jelenti, hogy számos beépített funkcióval és segédprogrammal rendelkezik, amelyek leegyszerűsítik a gyakori webfejlesztési teendőket. A Django az újra felhasználhatóságra, a kevesebb kódra és a „ne ismételd magad” (DRY) elvére helyezte a hangsúlyt, ezért népszerű választás a robusztus, skálázható webalkalmazásokat készítők számára.

Jellemzői:

1. ORM: A Django ORM lehetővé teszi a fejlesztőknek, hogy SQL helyett Python kóddal lépjenek kapcsolatba az adatbázissal, és így egyszerűsítsék az adatbázis-műveleteket.
2. Admin felület: Automatikusan generál egy admin felületet az alkalmazás adatainak kezeléséhez.
3. Biztonság: A beépített biztonsági funkciók védelmet nyújtanak a gyakori webalkalmazás-sebezhetőségek ellen, mint például az SQL befecskendezés.
4. Skálázható: Alkalmas nagy kiterjedésű projektekhez.

Backend dependency

JPA

A JPA (Java Persistence API) egy specifikáció a relációs adatok kezelésére Java alkalmazásokban. Absztrakciót biztosít a különböző adatok felett, ami lehetővé teszi, hogy bonyolult SQL-lekérdezések írása nélkül működjön együtt az adatbázis-objektumokkal.

Jellemzői:

1. Entitáskezelés: A JPA lehetővé teszi, hogy entitásokat definiáljunk, amelyek könnyű, tartós domain objektumok és egy adatbázis táblához vannak leképezve.
2. Annotációk: Az olyan megjegyzésekkel, mint az @Entity, @Table, @Id és @Column, Java osztályokat képezhetünk le adatbázis táblákhoz és oszlopokhoz.
3. Entitáskezelő: Ez a JPA központi interfésze, amely a perzisztencia-környezettel való interakcióra szolgál. Metódusokat biztosít a CRUD műveletekhez és a lekérdezések végrehajtásához.
4. JPQL (Java Persistence Query Language): Az SQL-hez hasonló lekérdezési nyelv, de nem közvetlenül az adatbázis tábláin, hanem az entitásobjektumokon dolgozik.
5. Tranzakciók kezelése: A JPA támogatja a deklaratív tranzakciókezelést az adatok integritásának és konzisztenciájának biztosítása érdekében.

Hibernate

A Hibernate egy népszerű nyílt forráskódú objektum-relációs leképező (ORM) keretrendszer, amely a JPA specifikációját valósítja meg. Leegyszerűsíti a Java-alkalmazások és a relációs adatbázisok közötti adatkezelést.

Jellemzői:

1. ORM (Object Relational Mapping): A Hibernate automatizálja a Java osztályok és az adatbázis táblái közötti leképezést, csökkentve ezzel a forráskódok szükségességét.
2. Caching: A Hibernate támogatja mind az első szintű (munkamenet), mind a másodszintű (globális) gyorsítótárazást az adatbázishoz való hozzáférés optimalizálása érdekében.
3. API-kritériumok: A JPQL alternatívája a típusbiztos lekérdezések létrehozásához programozási feltételek felhasználásával.

Thymeleaf

A Thymeleaf egy modern szerveroldali Java sablonmotor webes és önálló környezetekhez. HTML, XML, JavaScript, CSS és egyszerű szöveg feldolgozására tervezték.

Jellemzői:

1. Természetes sablonok: A Thymeleaf sablonok statikus prototípusként közvetlenül megnyithatók a böngészőkben, így a fejlesztési és tervezési folyamat gördülékenyebbé teszi.
2. Kifejező nyelv: A JSP-ben használt EL-hez hasonló, hatékony kifejezési nyelvet biztosít a sablonokon belüli adat- és logikai manipulációhoz.
3. Integráció a Springgel: Ez a Spring keretrendszerrel való zökkenőmentes integráció miatt népszerű választás a Spring MVC alkalmazások építéséhez.

JavaScript könyvtárak

A JavaScript könyvtárak olyan előre elkészített kódgyűjtemények, amelyek megkönnyítik a webfejlesztők munkáját bizonyos funkciók és feladatok megvalósításában.

jQuery

jQuery egy gyors és kis méretű JavaScript könyvtár, amely leegyszerűsíti a HTML dokumentumok kezelését, az eseménykezelést, az animációkat és az Ajax műveleteket. Használatával kevesebb kóddal lehet elérni összetett funkciókat, valamint biztosítja a böngészők közötti kompatibilitást, ami megkönnyíti a fejlesztést.

DataTables

DataTables egy jQuery plug-in, amely jelentősen kibővíti a HTML táblázatok funkcionalitását. Automatikusan hozzáad olyan funkciókat, mint az oldalszámozás, az oszlopok rendezése, valamint a keresés és szűrés. Ezek a funkciók segítenek kezelhetővé és átláthatóvá tenni a nagy adathalmazokat, és jelentősen javítják a felhasználói élményt.

Chart.js

Chart.js egy egyszerű, mégis rugalmas JavaScript könyvtár, amely különféle típusú grafikonokat képes megjeleníteni, beleértve a vonal-, oszlop-, kör-, radar- és buborékgrafikonokat. Az HTML5 canvas elemet használja, és könnyen testreszabható, így lehetővé teszi az interaktív és reszponzív adatelemzési vizualizációk létrehozását.

Frontend keretrendszer

A frontend keretrendszer olyan szoftvereszköz, amelyet a webfejlesztők használnak a weboldalak és webalkalmazások felhasználói felületének fejlesztéséhez. Ezek a keretrendszerek előre definiált kódokat, könyvtárakat és komponenseket tartalmaznak, amelyek megkönnyítik és felgyorsítják a fejlesztési folyamatot.

Bootstrap

Bootstrap egy népszerű, nyílt forráskódú frontend keretrendszer, amelyet a Twitter fejlesztői hoztak létre. Célja, hogy egyszerűsítse és felgyorsítsa a reszponzív és mobilbarát weboldalak és webalkalmazások fejlesztését.

Jellemzői:

1. JavaScript pluginok: Beépített JavaScript pluginokat biztosít (jQuery alapúak), amelyek interaktív funkciókat kínálnak, mint például modálok és eszköztippek.
2. Testreszabhatóság: Könnyen testreszabható a Sass változók és mixinek segítségével, lehetővé téve, hogy a projekt igényeinek megfelelően módosítsuk a megjelenést.
3. Előre definiált komponensek: Számos előre definiált HTML és CSS elemet tartalmaz, mint például gombok, űrlapok és navigációs menük.
4. Reszponzív rácsrendszer: A Bootstrap egy rugalmas, 12 oszlopos rácsrendszert kínál, amely lehetővé teszi a könnyű elrendezés kialakítását különböző képernyő méretekhez.

Bulma

Bulma egy modern CSS keretrendszer, amely a Flexbox rendszert használja a reszponzív dizájnhoz.

Jellemzői:

1. Flexbox alapú: Teljes mértékben a Flexbox layout rendszert használja, ami megkönnyíti a rugalmas és adaptív elrendezések kialakítását.
2. Egyszerű használat: Könnyen érthető és alkalmazható szintaxisú, ami gyorsítja a fejlesztést.
3. Moduláris felépítés: Minden komponens külön CSS fájlban található, így csak a szükséges részeket kell betölteni, ami csökkenti a weboldalak méretét.
4. Előre elkészített elemek: Előre definiált stílusok és komponensekkel rendelkezik, mint például gombok, űrlapok és menük.

Fejlesztő eszközök

Spring Tool Suite 4 (STS4)

A Spring Tool Suite 4 (STS4) egy integrált fejlesztőkörnyezet, amely kifejezetten a Spring keretrendszerben történő fejlesztéshez készült. Az Eclipse alapjaira épül és a Pivotal fejlesztette ki. STS4 kiterjedt támogatást nyújt a Spring keretrendszer különböző moduljaihoz, mint például Spring Boot, Spring MVC, Spring Data és Spring Cloud. Egyedülálló funkciója a Spring Boot Dashboard, amely megkönnyíti a Spring Boot alkalmazások kezelését és futtatását. Az IntelliSense alapú kódkiegészítése és intelligens javaslatai különösen hasznosak a Spring specifikus kódokhoz. Az IDE keresztplatformos, így Windows, macOS és Linux rendszereken egyaránt futtatható.

Visual Studio Code (VS Code)

A Visual Studio Code egy könnyű és erőteljes, nyílt forráskódú kódszerkesztő, amelyet a Microsoft fejlesztett ki. A VS Code egyik kiemelkedő jellemzője a kiterjedt bővítménytámogatása, amely lehetővé teszi, hogy szinte bármilyen programozási nyelvet és fejlesztési feladatot támogasson a Visual Studio Code Marketplace-en belül elérhető kiegészítők segítségével. A beépített funkciói intelligens kódkiegészítést és szintaxiskiemelést nyújt, míg a beépített terminál közvetlen hozzáférést biztosít a parancssorhoz az editorból. Emellett a beépített Git integráció lehetővé teszi a kód változásainak nyomon követését és megkönnyíti a csapatmunkát.

IntelliJ IDEA

Az IntelliJ IDEA egy erőteljes és integrált fejlesztőkörnyezet, amelyet a JetBrains fejlesztett és elsősorban Java fejlesztésre optimalizált. Az IntelliJ IDEA intelligens kódelemzési és refaktoring eszközökkel rendelkezik, amelyek segítenek a kód minőségének javításában és a fejlesztési hatékonyság növelésében. Ez az IDE is keresztplatformos, így Windows, macOS és Linux operációs rendszereken is elérhető. Számos programozási nyelvet támogat a Java mellett, mint például Kotlin, Groovy és Scala. Beépített eszközei között megtalálhatóak a hibakereső, verzió kezelő, adatbázis kezelő eszközök és building eszközök, mint például a Maven és Gradle.

Verziókezelő eszközök

Azure DevOps

Azure DevOps egy Microsoft által fejlesztett felhőalapú szolgáltatáscsomag, amely a teljes szoftverfejlesztési életciklust támogatja. Az Azure DevOps különböző eszközöket kínál, mint például az Azure Repos, amely Git és TFVC alapú verziókezelést biztosít. Szorosan integrálódik más fejlesztői eszközökkel és szolgáltatásokkal, biztosítva a rugalmas és hatékony szoftverfejlesztést.

GitHub

A GitHub egy kiemelkedő webes felület, amelyet verziókezelésre és közös szoftverfejlesztésre terveztek, és ami a Git-et használja alaprendszerként. A 2008-ban alapított és 2018-ban a Microsoft által felvásárolt GitHub olyan alapvető funkciókat biztosít, mint a tárolók, a pull-kérelmek és hiba keresés, amelyek egyszerűsítik a projektmenedzsmentet és javítják a csapatmunkát. A fejlesztők egy egységes felületen belül hosztolhatják és vizsgálhatják felül a kódot, kezelhetik a projekteket, és hozzájárulhatnak a nyílt forráskódú kezdeményezésekhez.

Subversion (SVN)

A Subversion egy központosított verziókezelő rendszer, amelyet az Apache Software Foundation fejlesztett ki. Lehetővé teszi, hogy nyomon kövessük a fájlok és könyvtárak történetét, valamint könnyen kezeljük a projekt különböző verzióit egy központi adattárban. A rendszer támogatja a változások egyesítését (merge) és az ágazások kezelését (branching), így a csapatok hatékonyan dolgozhatnak együtt nagy projekteken is. Az erős jogosultságkezelése biztosítja, hogy csak az illetékes személyek férjenek hozzá a kritikus fájlokhoz. Az SVN különösen hasznos nagy csapatok és szervezetek számára, ahol a központosított verziókezelés és az erős kontroll fontos szempontok.

Összegzés

Ebben a fejezetben ismertettem néhány fejlesztői eszközt, amelyek szükségesek ahhoz, hogy megfelelő alkalmazást tudjunk készíteni. A fentebb bemutatott technológiák és eszközök között szerepelnek az általam választott megoldások is.

Az adatbázis kiválasztásánál azt vettem legfőképpen figyelembe, hogy könnyen kezelhető legyen. Erre tökéletes megoldást nyújtott a MySQL és a MySQL Workbench mivel a kettő kombinációja ideális környezetet biztosít az adatbázisok struktúrájának vizuális megjelenítésére. Másik okom az volt, hogy ingyenesen elérhető legyen, amely ennek a kitételnek is eleget tesz hiszen nyílt forráskódú.

A backend kiválasztásánál fontos szerepet játszott, hogy az egyetemi éveim alatt Java-t tanultam. Ezáltal egy olyan integrált megoldásokkal rendelkező keretrendszert kerestem, amely nagyban megkönnyíti a fejlesztés folyamatát. Erre tökéletes volt a Spring Boot kiválasztása, hiszen minimális konfigurációval és a beépített konvencióival lehet egyszerűen alkalmazásokat létrehozni.

A fejlesztői környezet megválasztásakor szerettem volna már egy ismerős környezetet használni. A tanulmányaim során megismertem az Eclipse-t, emiatt esett a választásom a Spring Tool Suite 4-re. Az STS4 az Eclipse alapjaira épül és kifejezetten a Spring keretrendszerhez készült.

A frontend fejlesztése során szerettem volna egy olyan keretrendszert alkalmazni, amellyel rendkívül egyszerűen és gyorsan tudok jól kinéző oldalakat készíteni. Emiatt esett a választásom a Bootstrap-ra és az egyszerűen integrálható JavaScript könyvtárakra, mint például a jQuery. Könnyű a kezelésük, sok dokumentáció áll rendelkezésre és könnyen integrálhatóak a Spring Boot-al.

Verziókezelő rendszer tekintetében az egyik legnépszerűbb opciót választottam a GitHub-ot. Rendkívül egyszerű a használata és a hosszas fejlesztés folyamatában biztos pontot nyújt a régebbi verziók visszaállítására.

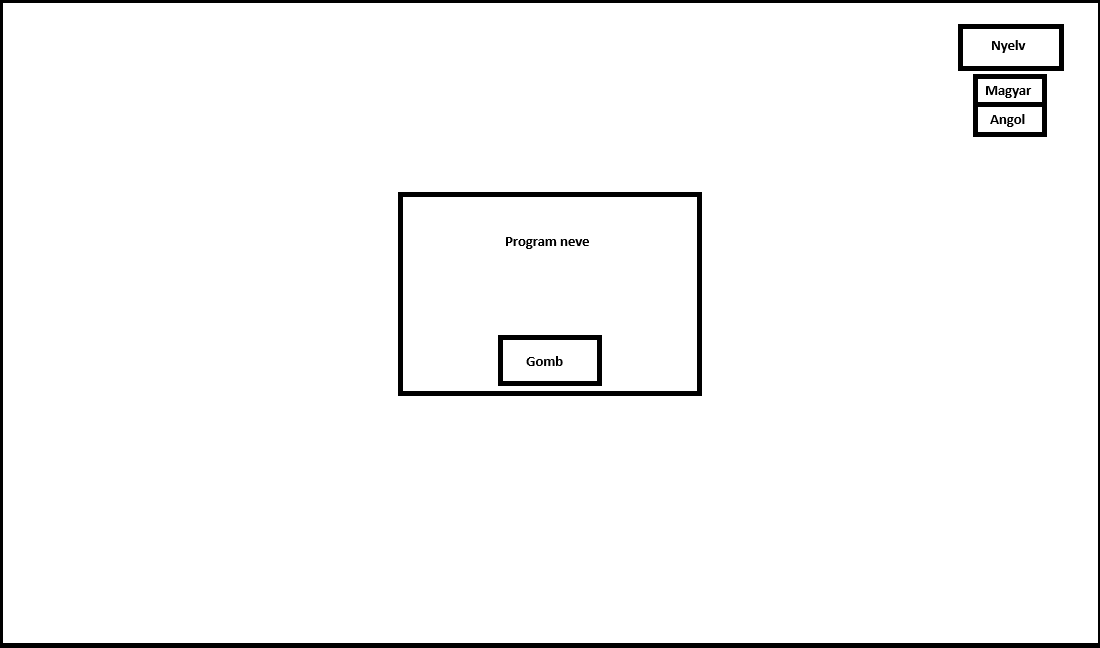
Programspecifikáció

Képernyőtervek

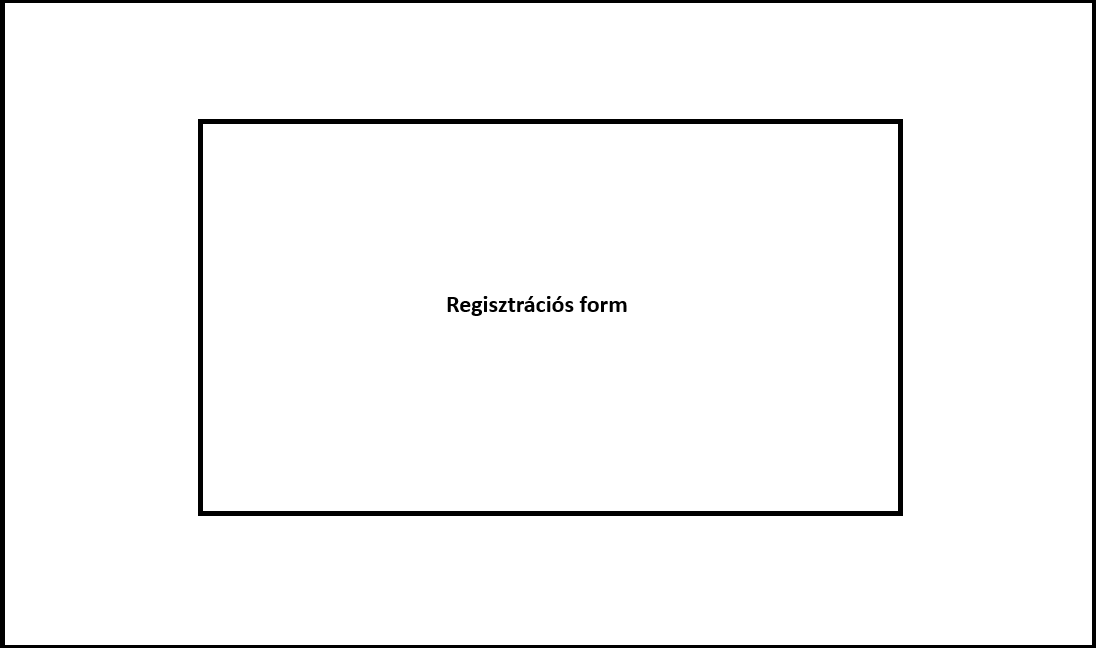
A program két felülettel rendelkezik. Az egyik felülete az átlag felhasználók számára elérhető oktatási felület, ahol az oktatásokat lehet elvégezni. A másik felület az admin felület, ahol az oktatásért felelős csoport ellenőrizni tudja az oktatások eredményét. A következőkben a program kinézetének a tervét fogom bemutatni.

Oktatási felület

A következő ábrákon az oktatási felület különböző oldalainak a tervét fogom bemutatni.



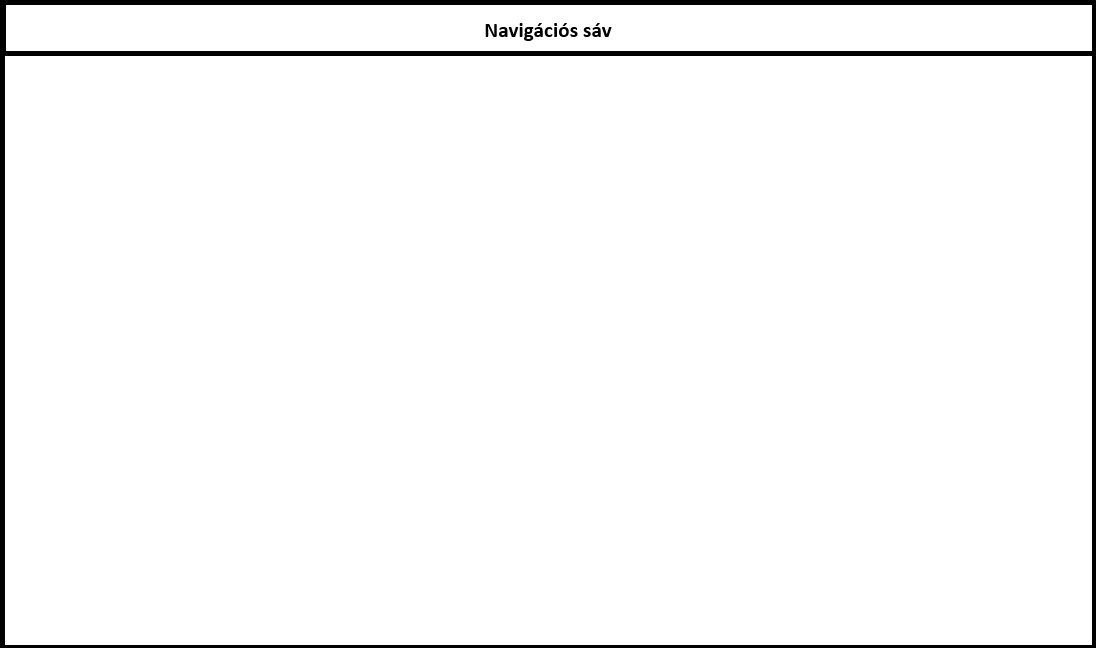
A fentebb látható ábrán látható, hogy az oktató felület belépési pontja, milyen elemekkel fog rendelkezni.



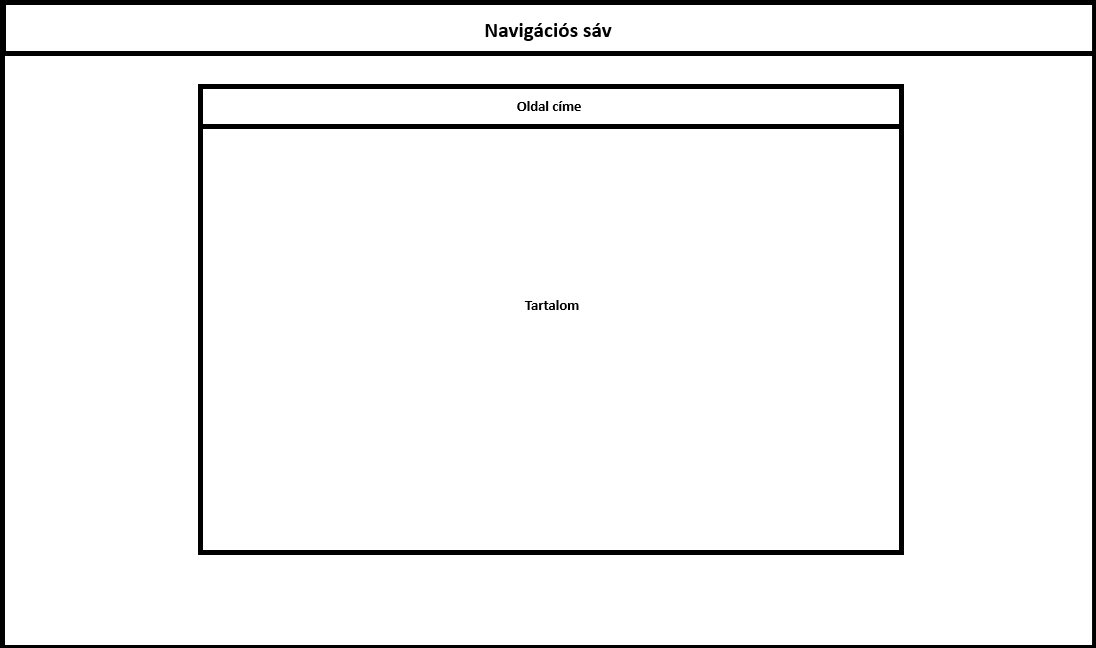
A x. ábrán látható, hogy az oktatási felület második és a többi pontja csak egy darab mezőt fog tartalmazni, amely a program során különféle adatokkal lesz feltöltve és különböző méretű lesz majd, aa megjelenített adatmennyiségtől függően.

Admin felület

Az következőkben az admin felület elrendezési tervét fogom bemutatni.



A fentebbi ábrán látható, hogy az admin felületen lesz egy navigációs bár ami felül helyezkedik el. A felületen belül mindenhol elérhető lesz és innen el lehet navigálni különböző oldalakra.



Ezen az ábrán látható, hogy hogyan fog elhelyezkedni egy oldalon a tartalom. Mindig lesz egy főcím és alatta fognak elhelyezkedni a különféle információk.

Adatbázis struktúra

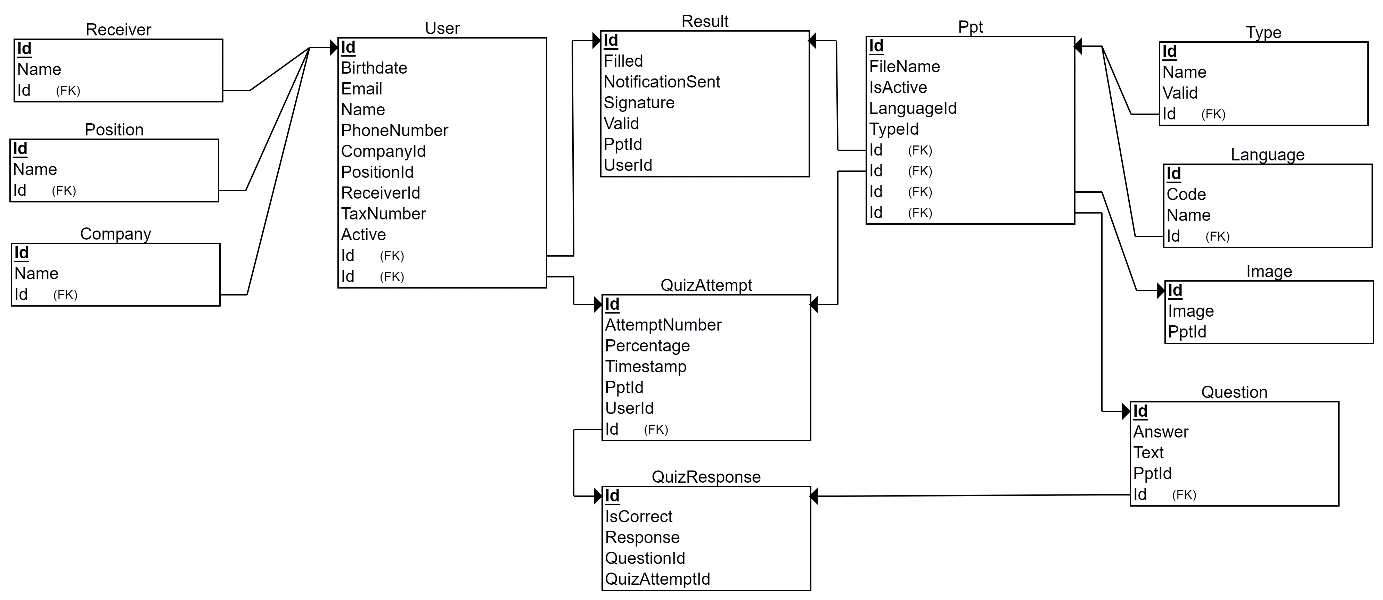
ER-modell.

A képen diagram, rajz, vázlat, minta látható

Automatikusan generált leírás

Relációs adatmodell.

A következő ábrán látható az az átkonvertált ER modell.



Position

A Position táblában tárolom az elérhető pozíciókat. Amikor egy felhasználó megadja az adatait a pozíciók közül kell választani egyet. A Position tábla rendelkezik Id-val és a pozíció nevével.

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

Company

A Company táblában tárolom azokat a cégeket, amelyek közül lehet választani. Rendelkezik egy Id-val és a cég nevével.

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

Receiver

A Receiver táblában tárolom a fogadó fél nevét és Id-ját. A fogadó fél egy személy, aki felelős az új kollégáért a szervezetben.

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

User

A User táblában tárolom a felhasználók személyes adatait.

A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírás

Language

A Language táblában találhatóak a webalkalmazás számára elérhető különböző nyelvek jelölői. Ezen mezők meghatározzák, hogy milyen nyelvű egy oktatás és milyen nyelven jelenik meg a felhasználói felület.

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

Type

A Type táblában van meghatározva, hogy egy oktatás milyen típusú és hány hónapig érvényes

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

Image

Az Image táblában találhatóak az adott oktatáshoz tartozó képek.

A képen szöveg, képernyőkép, Betűtípus, tervezés látható

Automatikusan generált leírás

Ppt

A Ppt táblában található az oktatás neve és megmutatja, hogy az adott oktatás aktív-e.

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

Question

A Question táblában találhatóak az eldöntendő kérdések és a hozzájuk tartozó helyes válaszok.

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

QuizAttempt

A QuizAttempt tábla jelzi egy adott felhasználóhoz és oktatáshoz tartozó próbálkozásokat. Jellemzője, hogy van benne egy próbálkozás szám, egy százalék érték, amely megmutatja, hogy milyen eredménnyel zárult a próbálkozás, valamint egy időbélyeg, ami a próbálkozás idejét adja meg, hogy mikor kezdte el a kvízt.

A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírás

QuizResponse

A QuizResponse tábla felelős egy próbálkozásnak a válaszainak a tárolásában. A tábla rendelkezik két jelölővel, hogy helyes volt – e a válasz és hogy mit válaszolt a felhasználó.

A képen szöveg, Betűtípus, képernyőkép látható

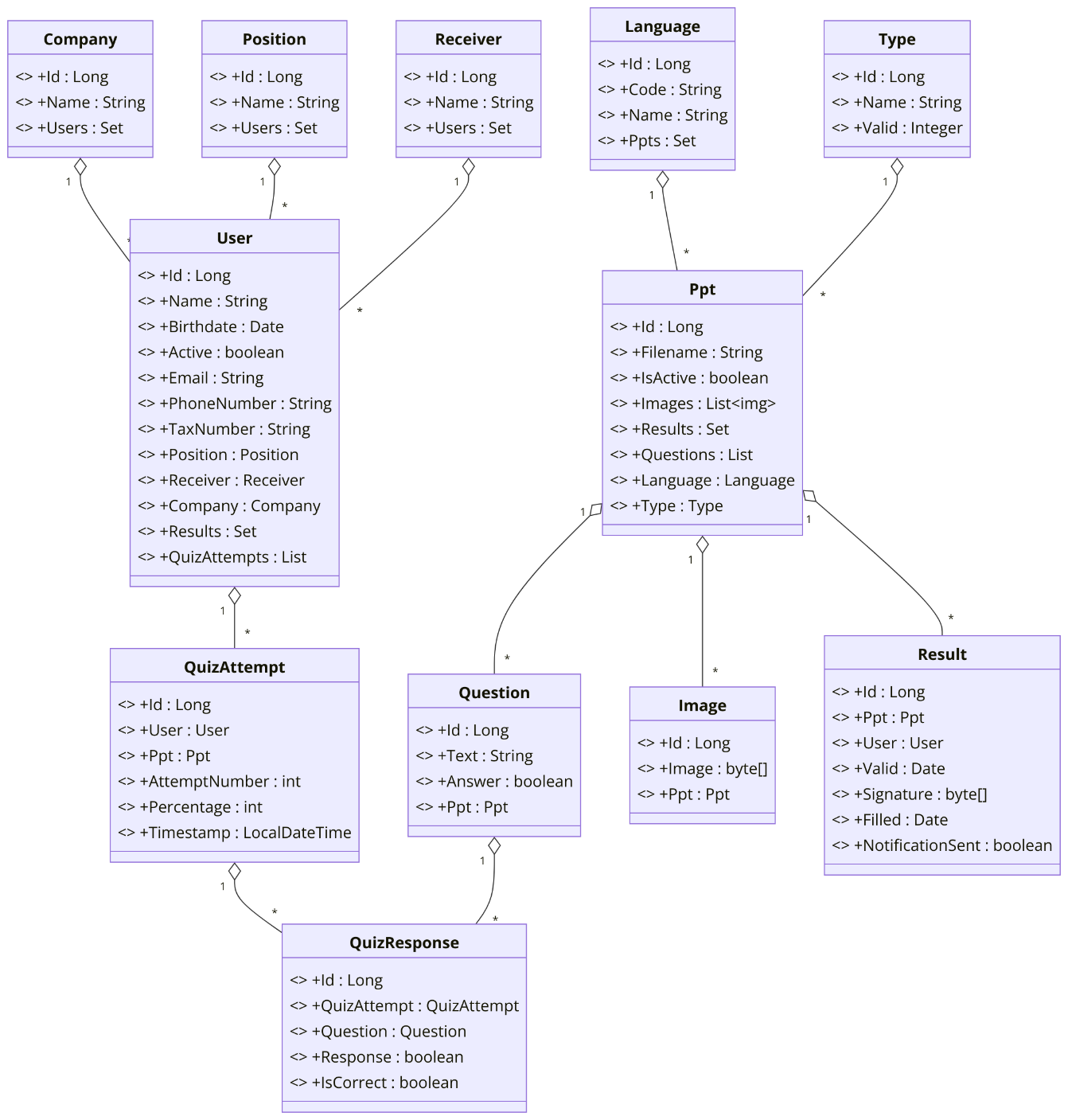
Automatikusan generált leírás

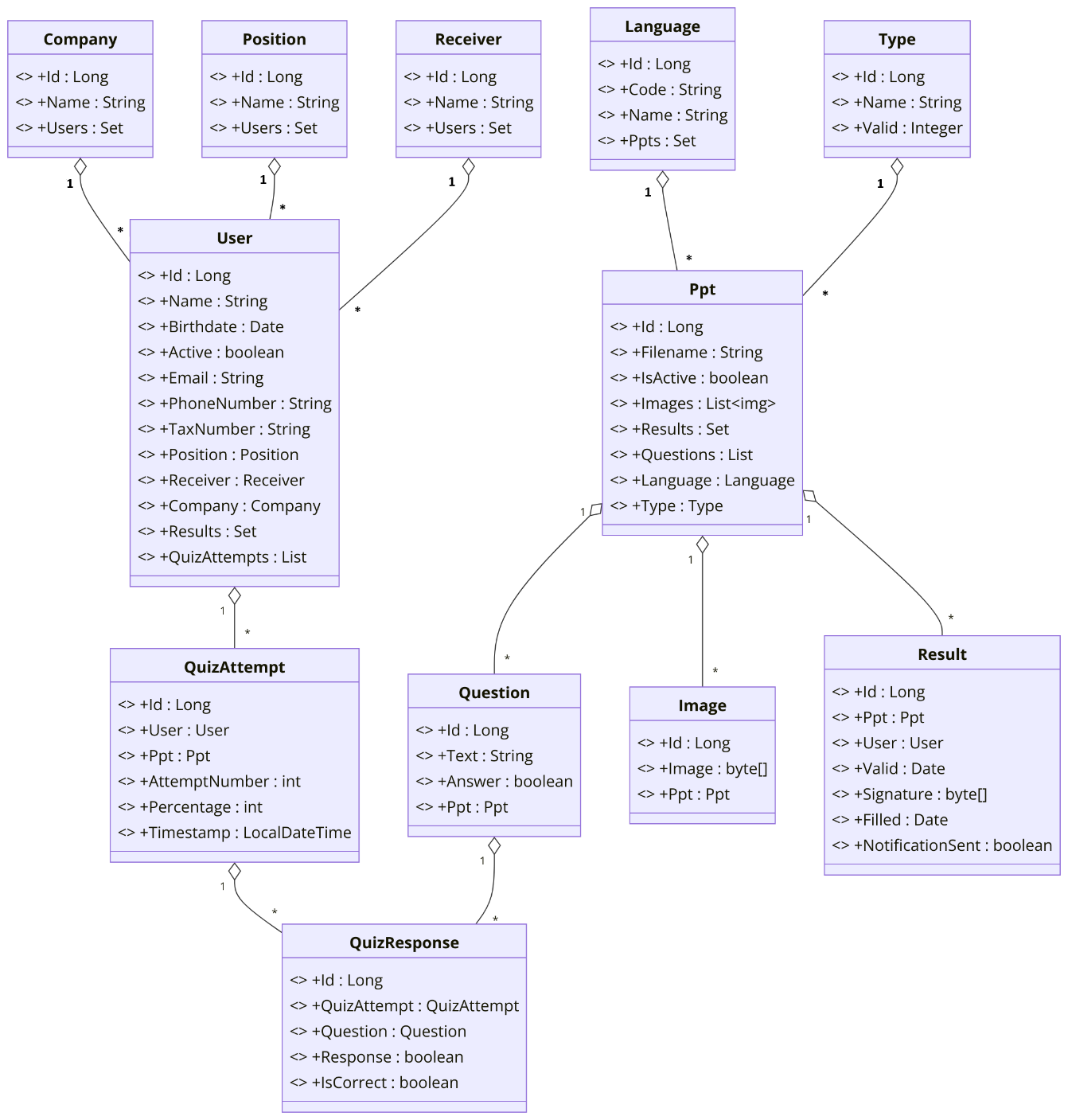
Result

A Result táblában tároljuk el az egy felhasználóhoz tartozó legfrissebb sikeres oktatásokat. Egy eredmény rendelkezik azzal, hogy melyik oktatásra vonatkozik az eredmény, mikor töltötte ki a felhasználó, melyik dátumig érvényes és egy aláírással ami hitelesíti. Emellett található benne egy értesítés jelző, amely azt jelzi, hogy az oktatásért felelős csoport kapott-e már értesítőt arról, hogy az oktatás 30 napon belül érvényességét veszíti.

A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírás

Entity

írok az osztálydiagramról,

szükséges alapfeltételek:

mik a rendszerkövetelmények: Db storage, STS4 helyigény memória processzor.

terv ismertetése:

felépítés, validálás