Szoftverfejlesztés projekt

Projekt név: invPlayer

**Követelmény**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Modul | ID | Név | Verzió | Kifejtés |
| Regisztráció | A1 | Regisztrációs felület backendje | 0.1 | A regisztrációhoz szükséges megadni a felhasználó nevet, emailcímet, a jelszót és a megerősítő jelszót. |
| Regisztráció | A2 | Regisztrációs felület felhasználó név mező validációja | 0.2 | A felhasználó névben nem lehet megadni szóközöket és egyéb speciális karaktereket csak is az angol ABC betűiből valamit számokból állhat. Minimum 3 karakterből és maximum 15 karakterből állhat. |
| Regisztráció | A3 | Regisztrációs felület felhasználó jelszó mező validációja | 0.3 | A felhasználó jelszava állhat: speciális karakterekből, angol ABC betűiből és számokból. A jelszó hossza maximum 25 karakterből és legalább 4 karakterből állhat. |
| Regisztráció | A4 | Regisztrációs felület felhasználó e-mail címének a validációja | 0.4 | A felhasználó e-mail címe úgy kell kinézni, mint egy standard e-mail címnek, ennek ellenőrzése, hogy valid-e. |
| Regisztráció | A5 | Regisztrációs felület regisztráció gombja | 0.5 | A felhasználó amikor rákattint a regisztráció gombra akkor, ha minden regisztrációs feltételnek eleget tett, átdobja őt a bejelentkező oldalra. |
| Regisztráció | A6 | Regisztráció vissza igazoló e-mail | 0.6 | A felhasználó kap egy levelet a regisztrált email címére és a linkre kattintva igazolhatja az e-mail címét, ezáltal átirányítódik a bejelentkezési felületre. |
| Regisztráció | A7 | Regisztráció felület frontendje | 0.7 | Az oldal közepén jelenik meg a regisztrációs felület. Nagyban ki van írva az a szöveg, hogy regisztráció. Vannak bemeneti mezők, amik le lesznek kerekítve, továbbá felette olyan szöveg található, ami által tudjuk, hogy mit szükséges megadni az adott mezőben. A bejelentkezés egy lekerekített gomb. Van egy felirat a legalján, egy szöveg, amivel regisztráció helyett a bejelentkezés felületre mehetünk át. |
| Belépés | B1 | Belépő felület backendje | 0.8 | A belépéshez szükséges megadni a felhasználó nevet vagy az email címet és a jelszót. |
| Belépés | B2 | Belépési felület felhasználónév (vagy email cím) és jelszó validációja | 0.9 | Megnézzük, hogy a felhasználónév (vagy az email cím) benne van-e az adatbázisban.  Ha a megadott felhasználónév létezik, akkor megvizsgáljuk, hogy a hozzá megadott jelszó helyes-e. Viszont, ha a felhasználó név létezik az adatbázisban, de a hozzá megadott jelszó helytelen, akkor tudatjuk a felhasználót, hogy a megadott jelszó helytelen. |
| Belépés | B3 | Belépési felület belépés gombja | 1.0 | A felhasználónév és a jelszó validációja a belépés gombra rákattintva történik meg.  Ha jók a feltételek, akkor a felhasználót átirányítjuk az alkalmazás kezdőlapjára.  Ha rosszak a feltételek akkor tudatjuk a felhasználót.  || Felkínáljuk a felhasználónak az „Elfelejtettem a jelszóm” opciót, amivel kérhet az emailcímére egy új jelszót |
| Belépés | B4 | Belépési felület frontendje | 1.1 | Hasonló stílusokkal jelenik meg, mint a regisztrációs felület. Középen helyezkedik el a belépési felület, két bemeneti mező és egy gomb. A felső mező a felhasználó névnek vagy az email címnek, az alsó a jelszónak. Ezek a bemeneti mezők is ugyan olyan stílusúak, mint a regisztrációnál. A bemeneti mezők felett kiírva, hogy mit kérünk: „Felhasználó név vagy jelszó” és „Jelszó”. Majd a belépés gomb szintén olyan stílussal, mint a regisztráció gomb. |

**Ütemterv**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Funkció | Feladat | Prioritás | Becslés | Akt. becslés | Eltelt | Hátralévő |
| Regisztráció | Regisztrációs felület | 10 | 6 |  | 4 | 4 |
| Bejelentkezés | Bejelentkezés felület | 9 | 2 |  | 2 | 0 |
| Profil | Felhasználói profil | 7 | 15 |  | 0 | 15 |
| Fő oldal | Fő oldal | 4 | 20 |  | 0 | 20 |
| Funkcionális specifikáció | Maga a működő funkciók | 8 | 30 |  | 0 | 30 |
| Tesztelés | Alkalmazás tesztelése | 2 | 8 |  | 0 | 8 |
| Kinézet | Felhasználói felület javítása | 4 | 4 |  | 0 | 4 |
| Egyéb problémák | Egyéb problémák javítása | 1 | 2 |  | 0 | 2 |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

Fizikai környezet:

Az rendszer egy webalkalmazás formájában fog megvalósulni, ami kompatibilis lesz a mai napokban legelterjedtebb böngészőkkel: Mozilla Firefox, Google Chrome, és a további Chromium alapú böngészők.

Az alkalmazás szerver oldali szükségleteit egy Apache 2 szerver fogja ellátni, a fejlesztés jelenlegi stádiumában Windows 10 operációs rendszeren, Xampp segítségével futtatva, a későbbiekben egy Linux operációs rendszert futtató szervergépen fog a webszerver üzemelni.

Az Apache szerver egy PHP 8.0.10 alapú backend szoftvert fog használni a bejövő kérések kezelésére, és megválaszolására. A backend szoftver kizárólag az alap PHP csomagot fogja használni, keretrendszer nélkül.

A rendszer perzisztenciai szükségleteinek kielégítését egy relációs SQL adatbázis fogja végrehajtani. Egy 10.4.21 verziójú MariaDB adatbázis amellyel a kapcsolatot a PHP alkalmazás a PHP Data Objects kiegészítővel fogja tartani. A fejlesztés jelenlegi stádiumában, ez is a fejlesztők gépén fut Xampp segítségével, a későbbiekben egy Linux alapú szervergépen fog futni.

Az alkalmazás felhasználói kliens oldalon működő, frontend része alap HTML 5 alapokon fog nyugodni.  
A megjelenésért saját CSS és Bootstrap 5 által szolgáltatott CSS fog felelni.  
Az oldalak funkcionalitását Javascript nyelven írt rendszer fogja adni, az általánost funkcionalitást a JQuery 3.6.1-es könyvtár segíti, a szerverrel való kommunikáció nagy részben AJAX kérésekkel fog megvalósulni. Ennek kezelésére az Axios könyvtár lesz használva.

A mindezek fejlesztéséhez használt eszköz a Visual Studio Code szoftver, illetve ennek a PHP, HTML, CSS, Javascript és JQuery kezelésére szolgáló bővítményei.

Adatbázis terv:

Users tábla:

ID: A felhasználó egyedi azonosítószáma, a tábla elsődleges kulcsa, integer, autómatikusan a legutóbb létrehozott user ID-jénél 1 el nagyobb szám az értéke.

Username: A felhasználó megjelenítendő neve, 100 karakter hosszúságú Varchar, Unique megszorítás van rajta, tehát minden username egyedi kell hogy legyen.

Email: A felhasználó email elérhetősége, 100 karakter hosszúságú Varchar, Unique megszorítás van rajta, tehát minden email egyedi kell hogy legyen.

Password: A felhasználó jelszava, hash-elve bcrypt algoritmussal, 255 karakter hosszúságú Varchar

Created\_at: A felhasználó létrehozásának időpontja, datetime típusú, alapértelemezett értéke a jelenlegi időpont.

Egyik oszlop értéke se lehet Null.

A tábla létrehozó kódja:

CREATE TABLE `users` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`username` varchar(100) NOT NULL,

`email` varchar(100) NOT NULL,

`password` varchar(255) NOT NULL,

`created\_at` datetime DEFAULT current\_timestamp(),

PRIMARY KEY (`id`),

UNIQUE KEY `username` (`username`),

UNIQUE KEY `email` (`email`)

) ENGINE=InnoDB AUTO\_INCREMENT=10 DEFAULT CHARSET=latin1

Architekturális terv:

A webalkalmazás a „3 rétegű alkalmazás” architektúrális tervezési minta alapján fog elkészülni, ezen három réteg a következő:

-Kliens réteg

-Alkalmazás szerver réteg

-Adat réteg

**A kliens réteggel** interaktál a felhasználó, ennek a feladata, hogy minél egyszerűbben és gyorsabban tudja megérteni az egyes oldalakon milyen információkat és hogy kell megadni az oldal használatához, illetve rendezi a szerverről visszakapott információkat egy könnyen emészthető, jól áttekinthető formátumba.

**Az alkalmazás szerver rétegben** található az üzleti logika, amely azért felel, hogy a kliens rétegről kapott információkat feldolgozza, és szükség esetén az adatbázisban eltárolja, illetve, hogy az adatbázisból kigyűjtse a szükséges adatokat a felhasználók igényeinek kielégítésére. Felelős továbbá a felhasználó adatainak biztonságáért, amit legfőképpen az oldal használóinak jogosultágai kezelésével tesz meg, biztosítja, hogy a személyes adatai a felhasználóknak nem kerülnek jogosulatlan kezekbe. Biztosítja ezt még jelszótitkosítással is, hogy a felhasználók fiókjai még esetleges adatszivárgás esetén se kerüljenek jogosulatlan irányítás alá. A rugalmasság érdekében az olyan konstansok mint az adatbázisszerver elérhetősége külön konfigurációs állományban kapnak helyet.

**Az adat rétegben** kap helyet az adatbázisszerver, ami az adatok perzisztenciájáért felel.

