Szerveroldal

A szerveroldal feladata a kliensek felől érkező kérések kiszolgálása és az adatok központi tárolása, kezelése. Az alkalmazás webes és mobil verzióját is ugyanaz a rendszer szolgálja ki. Ez a rendszer áll egy webszerverből és egy adatbázis szerverből.

Webszerver

A kliens oldalról érkező HTTP kéréseket egy Apache webszerver szolgálja ki, amelyre PHP értelmező is van telepítve. Ezzel a szerver képes szerveroldali PHP kódok futtatására. Hálózati operációs rendszerként az Ubuntu 18.04 szerverekre szánt verziója szolgál, amelyre még különböző kiegészítő szolgáltatások lettek telepítve, mint pl. SSH a távoli eléréshez, SFTP a távoli biztonságos fájlátvitelhez. Ezen kívül a szerveren a hozzáférés szabályozás, jogosultági rendszer, valamint tűzfal nehezíti meg az esetleges támadó dolgát. A kliensek és a szerver közötti biztonságos kommunikációért a HTTPS protokoll felel, amely egy hitelesített titkosított csatornán keresztül továbbítja az adatokat, megakadályozva az adatokhoz való illetéktelen hozzáférést. A projekt során egy ingyenes webtárhely szolgáltatás is lett alkalmazva, amely megkötésekkel ugyan, de biztosítja azokat a szolgáltatásokat, amelyek a rendszer működéséhez szükségesek, gyakorlatilag 0-24 órás rendelkezésre állással, szemben a saját szerverrel.

Adatbázis

Az adatbázisok kezelése a MYSQL relációs adatbáziskezelő-rendszer segítségével történik. Az adatok egy milliomos nevű adatbázisban vannak, amelyben a következő táblák találhatóak:

|  |  |  |
| --- | --- | --- |
| users | | |
| id | int | Felhasználó azonosítója |
| username | varchar(150) | Felhasználónév |
| pwhash | varchar(150) | A jelszó lenyomata |
| email | varchar(150) | Email cím |

|  |  |  |
| --- | --- | --- |
| question | | |
| id | int | Kérdés azonosítója |
| question | varchar(150) | Kérdés szövege |
| right0 | varchar(150) | Helyes válasz |
| wrong1 | varchar(150) | Helytelen válaszok |
| wrong2 | varchar(150) |
| wrong3 | varchar(150) |
| level | tinyint | Kérdés szintje |

|  |  |  |
| --- | --- | --- |
| score | | |
| id | int | Azonosító |
| userid | int | Felhasználó azonosító |
| money | int | Legmagasabb megnyert összeg |
| score | int | Pontszám |

PHP

sqlcredits.php

Ebben a fájlban találhatók meg azok az adatok, amelyek az adatbázishoz történő kapcsolódáshoz szükségesek, pl.: adatbázis címe, felhasználónév, jelszó, adatbázis neve. Célszerű ezeket az adatokat külön tárolni, és minden használatnál erre a fájlra hivatkozni, mivel így egy esetleges módosításnál elég, csak ebben az állományban módosítani, valamint biztonsági szempontból is előnyösebb, mivel ezt a fájl kívülre tehetjük azon a könyvtáron, amely az internet felől elérhető.

signup.php

Ez a fájl felel a felhasználók regisztrációjáért. A kliensen lévő regisztrációs űrlapból ide érkeznek be az adatok egy HTTP POST kéréssel. Ebből a POST kérésből a kód kinyeri a paramétereket (felhasználónév, jelszó, email), majd egy trim függvény segítségével levágja a végéről az esetleges whitespace karaktereket (pl szóköz, tabulátor). Az SQL injection nevű támadási módszer ellen véd a real\_escape\_string() függvény, amely megakadályozza, hogy a SQL kódot injektáljanak a formon keresztül. Ezután megvizsgálja, hogy minden paraméter kitöltésre került-e, majd ellenőrzi, hogy van-e már regisztrált felhasználó ezzel a névvel és/vagy emaillel. Amennyiben nincs, úgy az adott felhasználó regisztrálásra kerül az adatbázisba és bejelentkezhet. Biztonsági okokból a jelszó nem kerül tárolásra az adatbázisban, hanem csak annak a lenyomata. Így még az sem tudhatja meg a jelszót, aki hozzáfér az adatbázishoz.

login.php

A bejelentkezésnél a felhasználónevet és jelszót a regisztrációnál használt módszerhez hasonlóan kapja meg és ellenőrzi. A megadott jelszóból generál egy lenyomatot, amelyet összehasonlít az adatbázisban tárolt lenyomattal. A bejelentkezés akkor sikeres, ha az megadott felhasználónév létezik, és a hozzá tartozó jelszó lenyomata egyezik a megadott jelszó lenyomatával.

getQuestions.php

A játék megkezdésekor a kliens küld egy kérést, amellyel a játék kérdéseit kérdezi le. Paraméterként egy számot kap, hogy hány kérdést adjon vissza. Ezután az adatbázisból szintenként kérdezi le a kérdéseket véletlenszerűen, amelyet JSON formátumban küld vissza a kliensnek.

getScores.php

A TOP10 játékost adja vissza a kliens számára.

highscore.php

Minden játék végén a kliens elküldi az elért nyereményt, és a kód összehasonlítja az adott felhasználó legmagasabb nyereményét az éppen elért nyereménnyel. Ha utóbbi magasabb, akkor az új nyeremény íródik az adatbázisba.