**BACKEND PROGRAMOZÁS**

**TANFOLYAM feladat**

1. Hozzon létre backend szerver projektet az Ön által választott programnyelven, illetve fejlesztési környezetben! A projektmappát 2025\_03\_04 néven nevezze el!
2. Hozzon létre a lokális SQL szerveren **tanfolyam** néven adatbázist! Az adatbázis alapértelmezett rendezési sorrendje a magyar szabályok szerinti legyen! Állítsa be az UTF-8 kódolást alapértelmezettnek az adatbázis létrehozásánál! Az SQL mondatot másolja a *lekerdezesek.sql* állományba (2. feladat:)

Az *tanfolyam.sql* állomány tartalmazza a táblákat létrehozó, és az adatokat a táblába beszúró SQL parancsokat! Futtassa a parancsfájlt az *tanfolyam* adatbázisban!

***Adatbázis szerkezete:***

1. tanulok **(Tanulók tábla)**

Tárolja a tanulók alapadatait:

* + id (INT) – Egyedi azonosító
  + nev (VARCHAR) – Tanuló neve
  + telefonszam (VARCHAR) – Tanuló telefonszáma
  + szuletesido (DATE) – Tanuló születési dátuma
  + email (VARCHAR) – Tanuló e-mail címe

1. ertekelesek **(Értékelések tábla)**

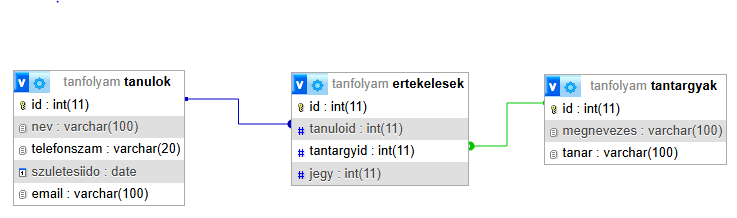
Összekapcsolja a tanulókat a tantárgyakkal és a kapott jegyekkel:

* + id (INT) – Egyedi azonosító
  + tanuloid (INT) – Kapcsolat a tanulók táblával
  + tantargyid (INT) – Kapcsolat a tantárgyak táblával
  + jegy (INT) – Tanuló által kapott jegy

1. tantargyak **(Tantárgyak tábla)**

Tárolja a tantárgyakat és az oktatókat:

* + id (INT) – Egyedi azonosító
  + megnevezes (VARCHAR) – Tantárgy neve
  + tanar (VARCHAR) – Tantárgyat oktató tanár neve



1. Hozzon létre egy **PHP alapú API-t**, amely ***JSON formátumban visszaadja az összes tanulói értékelést*** az adatbázisból. Az **API** működése:
2. Csatlakozik az adatbázishoz
3. Az API lekérdezi *az összes értékelést tanulónként és tantárgyanként* az adatbázisból.
4. **JSON** formátumban visszaadja az adatokat

|  |  |  |  |
| --- | --- | --- | --- |
| Metódus | URL | Body | Válasz |
| **GET** | **/api/tanulok.php** | **üres** | **JSON** |

Válasz az adott mintának legyen megfelelő:

[

{

"nev": "Kovács Péter",

"tantargy": "Matematika",

"erdemjegy": "3",

},

…

]

1. Készítsen egy **PHP API végpontot**, amely egy adott ***tantárgy neve* alapján** visszaadja a *tantárgyi átlagot*. Az **API**:
2. Kliens küldi a kérést a "tantargy" paraméterrel.
3. Az **API** ellenőrzi, hogy van-e ilyen tantárgy a tantargyak táblában, ha nem, akkor hibaüzenetet küld.
4. Az SQL lekérdezés:

* Összekapcsolja az ertekelesek és tantargyak táblákat.
* Kiszámítja a tantárgyi átlagot.

1. Az API visszaadja a JSON választ, amely tartalmazza a *tantárgy nevét és az átlagot*, vagy hibaüzenetet küld az alábbi mintának megfelelően.

|  |  |  |  |
| --- | --- | --- | --- |
| Metódus | URL | Body | Válasz |
| **POST** | **/api/tanulok.php** | **JSON** | **JSON** |

**Kéréstörzs (Request Body) (JSON):**

{

"tantargy": "Matematika"

}

**Válasz (Response JSON)**

***Sikeres lekérdezés esetén (példa):***

[

{

"tantargy\_neve": "Metematika",

"atlag": 3.5

}

]

***Hiba esetén:***

{

"error": "A megadott tantárgy nem található az adatbázisban."

}

1. Készítsen egy PHP **API** végpontot, amely *új értékelést vesz fel az adatbázisba*. Az **API** működése:

* **A kliens elküldi az *új értékelés adatait*** *JSON* formátumban.
* **Az API ellenőrzi, hogy *minden kötelező mező meg van-e adva (isset)*.**
* **Ha egy mező hiányzik**, hibaüzenetet küld vissza JSON-ben.
* Ellenőrzi, hogy a *tantárgy neve és a tanuló azonosítója* szerepel-e a táblákban (tantargyak, tanulok), ha nem, akkor hibaüzenetet küld
* **Ha az ellenőrzések nem találtak hibát,** akkor beszúrja az új értékelést az adatbázisba.
* **JSON válaszban visszaadja a sikeres beszúrás eredményét vagy hibaüzenetet.**

|  |  |  |  |
| --- | --- | --- | --- |
| Metódus | URL | Body | Válasz |
| **POST** | **/api/ujertekeles.php** | **JSON** | **JSON** |

**Kéréstörzs (Request Body) (JSON):**

{

"tantargy": "Matematika",

"tanuloId": "1",

"jegy": "2"

}

**Válasz (Response JSON)**

***Sikeres lekérdezés esetén:***

{

"message": "Sikeresen hozzáadtuk az értékelést az adatbázishoz.",

"tanulo": {

"nev": "Kovács Elek",

"email": "kovacs.elek@example.com",

"tantargy": "Matematika",

"jegy": "2"

}

}

***Hiba esetén:***

{

"error": "A megadott tantárgy nem létezik."

}

{

"error": "A megadott tanuló nem létezik."

}

{

"error": "Hiányzó kulcs (tantargy, tanuloId, jegy)."

}

{

"error": "Nem sikerült felvenni az értékelést az adatbázisba!"

}