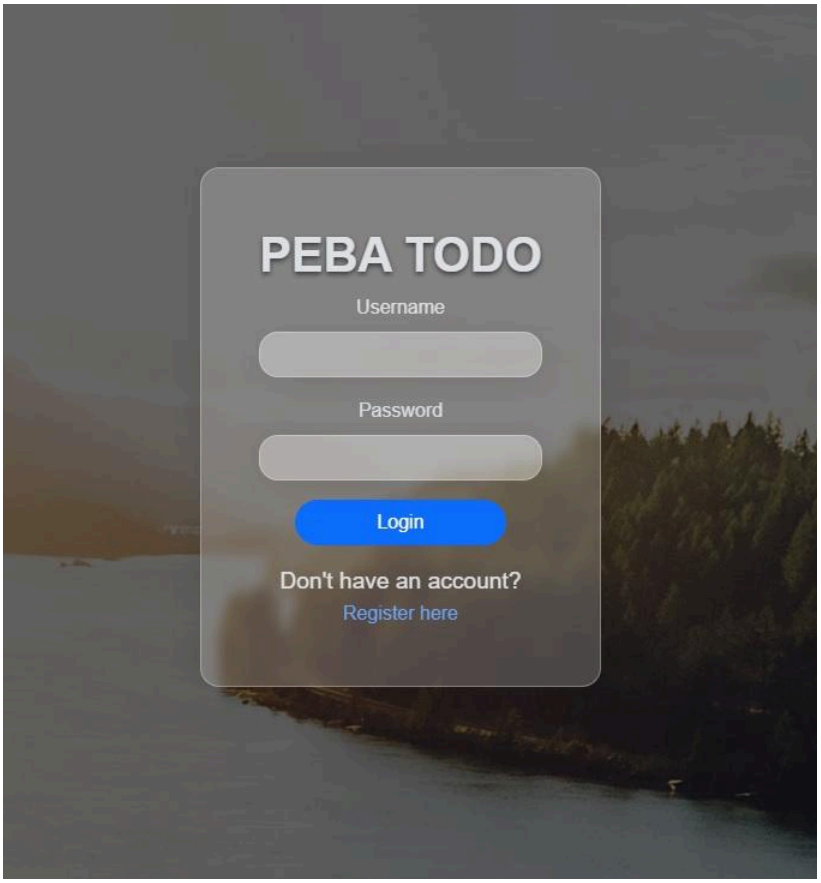
PEBA TODO
Dokumentáció



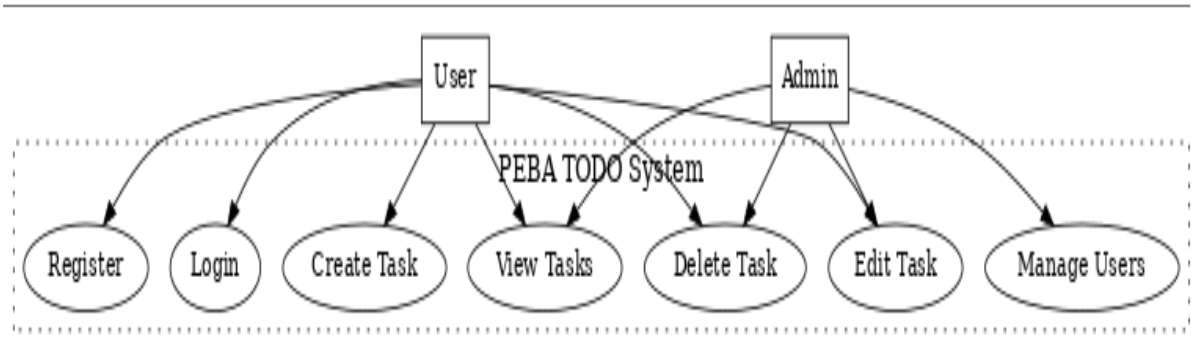
Tartalomjegyzék

1. Bevezetés

Projekt neve: PEBA TODO
Cél: A PEBA TODO alkalmazás célja, hogy segítsen a felhasználóknak nyomon követni és kezelni mindennapi feladataikat, elkerülve az elfelejtett teendők problémáját. A projekt bemutatja, hogy milyen képességekkel rendelkezel, és rávilágít azokra a megoldásokra, amelyek képesek kezelni a felhasználók gyakori problémáit.

Fő funkciók:

- Felhasználókezelés: Regisztráció, bejelentkezés, szerepkörök (admin/user).
- Feladatkezelés: Feladatok létrehozása, módosítása, törlése.
- Adminisztrációs panel: Felhasználók és feladatok teljes körû kezelése.



2. Telepítési útmutató

2.1. Követelmények

- **PHP verzió:** 7.4 vagy újabb.
- **Adatbázis:** MySQL.
- **Webszerver:** Apache vagy más kompatibilis szerver.

2.2. Telepítési lépések

1. Klónozd a projektet a gépedre:
   gh repo clone balintpethe/vizsgaremek
2. Állítsd be az adatbázist a projekt számára:
   - Hozz létre egy MySQL adatbázist (név: peba_todo).
   - Importáld az adatbázis inicializáló szkriptet.
3. Konfiguráld a db.php fájlt az adatbázis kapcsolathoz:

```
$host = 'localhost';
$dbname = 'todo_app';
$username = 'root';
$password = '';
```

4. Indítsd el a fejlesztõi szervert
5. A szoftver belépési pontja:
   http://localhost/vizsgafeladat/public/login

---

3. Mappastruktúra

A PEBA TODO projekt mappastruktúrája az MVC (Model-View-Controller) architektúrát követi.

Fő mappák:

- controllers/
  - TaskController.php: A feladatokhoz kapcsolódó mûveletek kezelése.
  - UserController.php: A felhasználókhoz kapcsolódó mûveletek kezelése.
- models/
  - TaskModel.php: Az adatbázis mûveletek feladatokhoz.
  - UserModel.php: Az adatbázis mûveletek felhasználókhoz.
- public/
  - assets/: Statikus fájlok (CSS, JS, képek).
  - .htaccess: URL átirányítások és biztonsági szabályok.
  - index.php: Az alkalmazás belépési pontja.
- views/
  - admin.php: Adminisztrációs felület.
  - login.php: Bejelentkezési felület.
  - register.php: Regisztrációs felület.
  - tasks.php: Felhasználói feladatok megjelenítése.

---

4. Adatbázis Struktúra

Az alkalmazás két fő táblát használ: users és tasks.

users tábla

Oszlop neve	Típus	Leírás
id	INT (PK)	Egyedi azonosító.
username	VARCHAR(50)	A felhasználó neve.
password	VARCHAR(255)	A felhasználó jelszava (hash-elt).
role	ENUM(user, admin)	Felhasználói szerepkör.
created_at	DATE	A felhasználó létrehozásának dátuma.

tasks tábla

Oszlop neve	Típus	Leírás
id	INT (PK)	Egyedi azonosító.
user_id	INT (FK)	A feladat létrehozójának azonosítója.
title	VARCHAR(255)	A feladat címe.
created_at	DATE	A feladat létrehozásának dátuma.

Kapcsolat: Egy felhasználóhoz (users) több feladat (tasks) is tartozhat (1:N).

5. Routing

Az alkalmazás dinamikus routingot használ a Router osztály segítségével.

Főútvonalak

HTTP Metódus	Útvonal	Funkció	Nézet / Vezérlő
GET	/login	Bejelentkezési oldal	../public/login
GET	/register	Regisztrációs oldal	../public/register
GET	/admin	Admin panel megjelenítése	../public/admin
POST	/index.php	Felhasználói mûveletek	UserController és TaskController
GET	/tasks	Feladatok listázása	../public/tasks

---

6. Tesztelés

Tesztelési forgatókönyvek

1. **Regisztráció**
    - Hibás adat bevitele (pl. több mint 50 karakteres felhasználónév).
    - Helyes adat bevitele, sikeres regisztráció.
2. **Bejelentkezés**
    - Hibás felhasználónév/jelszó.
    - Helyes adatokkal történő belépés.
3. **Feladatok kezelése**
    - Feladat létrehozása, módosítása, törlése.
4. **Admin funkciók**
    - Felhasználó módosítása és törlése.

---

7. Jövőbeli fejlesztési lehetőségek

- Logolási rendszer implementálása.
- Userek képesek legyenek profiljukat módosítani.
- Avatarok létrehozása.
- Taskok megosztása több userrel.
- Email értesítések támogatása.
- További UI/UX fejlesztések.
- Taskok fejlesztése, lejárati dátum, hosszas leírás.
- A rendszer okos otthonba való integrálása.