**QuickTick**

**Fejlesztői és felhasználói dokumentáció**

**Bevezető**

**Név eredete**

**A fejlesztés során használt programok**

* Visual Studio Code
* Google Chrome
* XAMPP
* GitHub
* Microsoft Word
* Microsoft PowerPoint

Visual Studio Code: A fejlesztéshez használt környezet.

Google Chorme: Weboldal megtekintéséhez használt szoftver.

XAMPP: Adatbázis illetve weboldal futtatása.

GitHub: A verzió követéshez használt szoftver.

Microsoft Word: A dokumentáció elkészítéséhez használt program.

Microsoft PowerPoint: A prezentáció, illetve bemutatáshoz használt program.

**A program célja**

A QuickTick egy feladatkezelő amely,

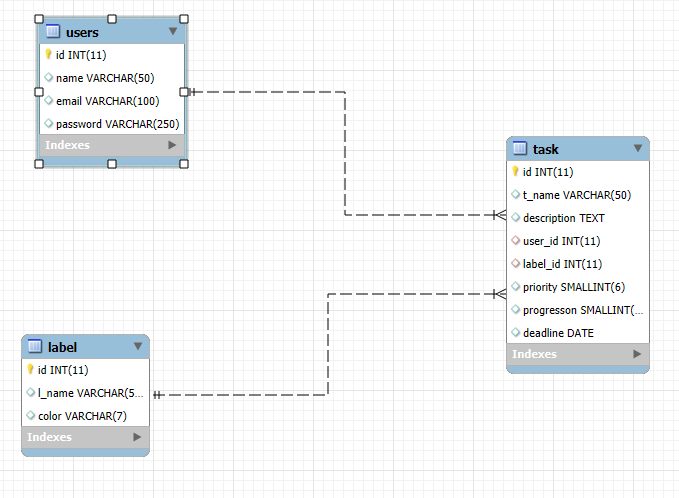
**Rendszerkövetelmény**

|  |  |  |
| --- | --- | --- |
| Rendszerkövetelmény | Minimum | Ajánlott |
| Processzor | Pentium(R) Dual-Core CPUE5700 @ 3.00GHz |  |
| Memória (RAM) | 4GB | 8GB |
| Merevlemez | HDD vagy SSD | HDD vagy SSD |
| Kijelző | 1080p (1920x1080) | 1080p (1920x1080) |
| Beviteli eszköz | Billentyűzet és egér | Billentyűzet és egér |
| Operációs rendszer | Windows 10 64bit | Windows 10 64bit |

**Adatmodell**

A megfelelő működéshez egy adatbázisra van szükség. A program egy MySQL adatbázissal rendelkezik, a XAMPP és a PHPMyAdmin használatával működik.

MySQL Workbench kép:



Az adatbázis 3 külön táblából áll:

1. users tábla: A felhasználó adatait tartalmazza
   * id: Elsődleges kulcsa a táblának, a felhasználó egyedi azonosítója.  
     Adattípus: INT (Auto Increment) automatikusan növekvő
   * name: A felhasználó nevét tartalmazza.  
     Adattípus: maximumim 50 karakterből álló szöveg VARCHAR (50)
   * email: A felhasználó e-mail címet tartalmazza.  
     Adattípus: maximum 100 karakterből álló szöveg VARCHAR (100)
   * password: A felhasználó jelszavát tartalmazza, amely belépéshez szükséges.  
     Adattípus: maximum 250 karakterből álló szöveg VARCHAR (250)
2. task tábla: A feladatok adatait tartalmazza
   * id:Eésődleges kulcsa a táblának, a feladat egyedi azonosítója.  
     Adattípus: INT (Auto Increment) automatikusan növekvő
   * t\_name: A feladat nevét tartalmazza.  
     Adattípus: maximumim 50 karakterből álló szöveg VARCHAR (50)
   * description: A feladat leírása. Adattípus: TEXT
   * user\_id: Idegen kulcsa a táblának, a felhasználó egyedi azonosítója. users tábla id kulcsára hivatkozik. Adattípus: INT (11)
   * label\_id: Idegen kulcsa a táblának, a felhasználó egyedi azonosítója. label tábla id kulcsára hivatkozik. Adattípus: INT (11)
   * priority: A feladat fontossága. Adattípus: SMALLINT (6)
   * progression: A feladat státusza, felvéve, elkészült, lejárt állapotot tartalmazza.  
     Adattípus: SMALLINT (6)
   * deadline: A feladat határideje. Adattípus: DATE
3. label tábla:
   * id: Eésődleges kulcsa a táblának, a címke egyedi azonosítója.  
     Adattípus: INT (Auto Increment) automatikusan növekvő
   * l\_name: A címke nevét tartalmazza.  
     Adattípus: maximum 50 karakterből álló szöveg VARCHAR (50)
   * color: A címke színét tartalmazza hexadecimális kódként  
     Adattípus: maximum 7 karakterből álló szöveg VARCHAR (7)

**Az elkészítéshez használt számítógépek specifikációja**

Az iskolában lévő két külön számítógép, illetve egy személyes laptop amely előre összeállított konfigurációval rendelkezik (Asus TUF Gaming A15, Modellje: FA506II) amely a vizsgán a projekt bemutatásához is szükséges, ezen kívül két személyes asztali számítógép egyedi konfigurációval, volt használva a fejlesztéshez.

Első iskolai számítógép specifikációja:

|  |  |
| --- | --- |
| Processzor | Pentium(R) Dual-Core CPUE5700 @ 3.00GHz |
| Memória (RAM) | 8GB |
| Merevlemez | HDD |
| Kijelző | 1080p (1920x1080) |
| Beviteli eszköz | Billentyűzet és egér |
| Operációs rendszer | Windows 10 64bit |

Második iskolai számítógép specifikációja:

|  |  |
| --- | --- |
| Processzor | Intel(R) Core(TM) i5-14500 |
| Memória (RAM) | 32GB |
| Merevlemez | SSD |
| Kijelző | 1080p (1920x1080) |
| Beviteli eszköz | Billentyűzet és egér |
| Operációs rendszer | Windows 10 64bit |

Laptop specifikációja:

|  |  |
| --- | --- |
| Processzor | AMD Ryzen™ 5 4600H |
| Videókártya | NVIDIA GeForce GTX 1650 TI |
| Memória (RAM) | 8GB |
| Merevlemez | SSD |
| Kijelző | 1080p (1920x1080) |
| Beviteli eszköz | Billentyűzet és egér |
| Operációs rendszer | Windows 11 64bit |

Első otthoni számítógép specifikációja:

|  |  |
| --- | --- |
| Processzor | AMD Ryzen™ 7 7800X3D |
| Videókártya | NVIDIA GeForce RTX 2060 |
| Memória (RAM) | 32GB |
| Merevlemez | SSD |
| Első kijelző | 24’ 1080p (1920x1080) |
| Második kijelző | 27’ 1440p (2560x1440) |
| Beviteli eszköz | Billentyűzet és egér |
| Operációs rendszer | Windows 11 64bit |

Második otthoni számítógép specifikációja:

|  |  |
| --- | --- |
| Processzor | Intel(R) Core(TM) i7-8700 |
| Videókártya | NVIDIA GeForce RTX 3060 |
| Memória (RAM) | 32GB |
| Merevlemez | SSD |
| Első kijelző | 24’ 1080p (1920x1080) |
| Második kijelző | 49’ 1440p(5120x1440) |
| Beviteli eszköz | Billentyűzet és egér |
| Operációs rendszer | Windows 10 64bit |

Program főbb részleteinek áttekintése

A következő részben a program által használt általunk írt API-ok bemutatása következik röviden milyen bemenő paramétereket várnak milyen lehetséges hibaüzenetek jöhetnek vissza illetve ha minden rendben milyen adatok érkeznek vissza

# **register.php**

1. Az Api ellenőrzi a metódust illetve átveszi a bemenő paramétereket

2. Ellenőrzi jött e üres adat ha igen hibát dob

3. titkosítja a jelszót

4. Ellenőrzi létezik e már a email cím

5. beszúrja az új felhasználót

|  |  |  |  |
| --- | --- | --- | --- |
| Metódus | URL | Body | Válasz |
| POST | ./Assets/php/register.php | JSON | JSON |

**Kéréstörzs (Request Body) JSON:**

{

username:”Ex Pista”

email:”[ex@gmail.com](mailto:ex@gmail.com)”

password:”123”

}

**Lehetséges Hibaüzenetek:**

{

//Üres mező esetén

'error' => 'Minden mező kitöltése szükséges']);

//Már használatban lévő email esetén

'error' => 'Email már használatban van'

}

**Sikeres Regisztráció esetén**

{

'message' => 'Felhasználó sikeresen regisztrált'

}

# **login.php**

1. Az Api ellenőrzi a metódust illetve átveszi a változókat
2. Ellenőrzi jött e üres adat ha igen hibát dob
3. Ellenőrzi az e-mailt és a jelszót
4. Beállítja a Session változókat

|  |  |  |  |
| --- | --- | --- | --- |
| Metódus | URL | Body | Válasz |
| POST | ./Assets/php/login.php | JSON | JSON |

**Kéréstörzs (Request Body) JSON:**

{

email:”[ex@gmail.com](mailto:ex@gmail.com)”

password:”123”

}

**Lehetséges Hibaüzenetek:**

{

//Üres mező esetén

'success' => false, 'message' => 'Email és jelszó szükséges'

//Nem helyes felhasználónév vagy jelszó

'success' => false, 'message' => 'Helytelen email cím vagy jelszó'

}

**Sikeres Regisztráció esetén**

{

'success' => true,

        'message' => 'Login successful',

        'logged\_in' => true,

        'user\_id' => $\_SESSION['user\_id'],

        'email' => $\_SESSION['email']

}

# **task.php**

1.Az Api ellenörzi be van e lépve felhasználó és átveszi a dátumot illetve a címke id-át egy $\_GET en keresztül

2. Ellenőrzi a dátumot ha nincs hibát dob

3. Futtatja az alap lekérdezést illetve ha a címke nem nulla hozzáfűzi azt a paramétert is illetve sorba rendezi fontosság alapján

4. Ha érkezik vissza feladat azt visszaküldi de ha nem akkor külön üzenetet küld vissza

|  |  |  |  |
| --- | --- | --- | --- |
| Metódus | URL | Body | Válasz |
| GET | ./Assets/php/task.php | Üres | JSON |

**GET paraméterek:**

{

date:2025-04-16

label\_id:1 //Nem minden esetben kell használni

}

**Lehetséges Hibaüzenetek:**

{

//Nincs bejelentkezve felhasználó

'success' => false, 'message' => 'felhasználó azonosító szükséges'

//Nincs dátum

'success' => false, 'message' => 'Dátum szükséges'

//Sikertelen lekérdezés

'success' => false, 'message' => 'Query failed: ' . $e->getMessage()

}

**Sikeres lekérdezés esetén**

{

// nincs az adott napon feladata a felhasználónak

'success' => true, 'tasks' => [], 'message' => 'Nincs feladatod erre a napra'

//Van feladat

'success' => true, 'tasks' => $tasks

}

# **add\_task.php**

1. Az Api ellenőrzi be van e lépve felhasználó és átveszi az adatokat

2. Ellenőrzi a feladat neve illetve a dátum meg van e

3. Ellenőrzi a CheckForExistingTask functional létezik e már ilyen című feladat aznap

|  |  |  |  |
| --- | --- | --- | --- |
| Metódus | URL | Body | Válasz |
| POST | ./Assets/php/add\_task.php | JSON | JSON |

**Kéréstörzs (Request Body) JSON:**

{

t\_name:”Teszt”

description:”Teszt leírása”

label\_id: 1 // nem minden esetben szükséges

priority:1

progresson:0

deadline:2025-04-16

}

**Lehetséges Hibaüzenetek:**

{

//Nincs bejelentkezve felhasználó

'success' => false, 'message' => 'felhasználó azonosító szükséges'

//Nincs dátum vagy feladat név

'success' => false, 'message' => 'Feladat név és dátum szükséges'

//Már vettünk fel ilyen néven feladatot arra a napra

'success' => false, 'message' => 'Ilyen nevű feladat már létezik ezen a napon'

//Sikertelen lekérdezés

'success' => false, 'message' => 'Query failed: ' . $e->getMessage()

}

**Sikeres beszúrás esetén**

{

'success' => true, 'message' => 'Feladat sikeresen hozzáadva '

}

# **logout.php**

1.Ellenőrzi a kérést típust

2.Nullázza a Sessiont

|  |  |  |  |
| --- | --- | --- | --- |
| Metódus | URL | Body | Válasz |
| POST | ./Assets/php/logout.php | Üres | JSON |

**Lehetséges Hibaüzenetek:**

{

// Nem megfelelő metódus

'error' => 'Method not allowed']

}

**Sikeres kijelentkezés esetén**

{

''success' => true

}

# **labels.php**

1. Lekérdezi az összes címkét

|  |  |  |  |
| --- | --- | --- | --- |
| Metódus | URL | Body | Válasz |
| GET | ./Assets/php/labels.php | Üres | JSON |

**Lehetséges Hibaüzenetek:**

{

// Sikertelen lekérdezés

''success' => false, 'message' => 'Query failed: ' . $e->getMessage()]

}

**Sikeres Lekérdezés esetén**

{

'success' => true, 'labels' => $labels

}

**profile\_details.php**

1. Ellenőrzi a metódust illetve be van e jelentkezve a felhasználó
2. Végrehajtja a lekérdezést

|  |  |  |  |
| --- | --- | --- | --- |
| Metódus | URL | Body | Válasz |
| GET | ./Assets/php/profile\_details.php | Üres | JSON |

**Lehetséges Hibaüzenetek:**

{

//Nincs bejelentkezve felhasználó

'success' => false, 'message' => 'felhasználó azonosító szükséges'

// Sikertelen lekérdezés

''success' => false, 'message' => 'Query failed: ' . $e->getMessage()

}

**Sikeres Lekérdezés esetén**

{

//Megvan a felhasználó

'success' => true, 'user' => $user

//Nincs ilyen felhasználó

'success' => false, 'message' => 'User not found'

}

# **get\_task.php**

1. Átveszi  a feladat id-át Get paraméterből
2. ellenőrzi hogy sikeresen át lett e véve

|  |  |  |  |
| --- | --- | --- | --- |
| Metódus | URL | Body | Válasz |
| GET | ./Assets/php/get\_task.php | Üres | JSON |

**GET paraméterek:**

{

task\_id:1

}

**Lehetséges Hibaüzenetek:**

{

//Nincs feladat azonosító

'success' => false, 'message' => 'Feladat azonosító szükséges'

// Sikertelen lekérdezés

''success' => false, 'message' => 'Query failed: ' . $e->getMessage()

}

**Sikeres lekérdezés esetén**

{

//megvan a feladat

'success' => true, 'task' => $task

//Nincs meg a feladat

'success' => false, 'message' => 'Task not found'

}

# **delete\_task.php**

1.Ellenőrzi be van e jelentkezve a felhasználó és bekéri a feladat id-át

2. Ellenőrzi megvan e a feladat id-ját

|  |  |  |  |
| --- | --- | --- | --- |
| Metódus | URL | Body | Válasz |
| GET | ./Assets/php/delete\_task.php | Üres | JSON |

**GET paraméterek:**

{

task\_id:1

}

**Lehetséges Hibaüzenetek:**

{

// Nincs bejelentkezve a felhasználó

'success' => false, 'message' => 'Feladat azonosító szükséges'

//Nincs feladat azonosító

'success' => false, 'message' => 'Feladat azonosító szükséges'

// Sikertelen lekérdezés

''success' => false, 'message' => 'Query failed: ' . $e->getMessage()

}

Sikeres lekérdezés esetén

{

//Sikeres törlés

'success' => true, 'message' => 'Feladat sikeresen törölve'

//Nem sikerül a törlés

'success' => false, 'message' =>'Feladatot nem sikerült törölni

}

# **change\_progress.php**

1.Ellenőrzi a metódust és átveszi a két bemenő adatot

2. Ha ’POST’ a kiválasztott feladat előrehaladását 1 (Kész)-re teszi

3. Ha ’GET’ a aznap előtt lejárt feladatok előrehaladását 2 re teszi (Nincs kész) illetve a 30 napnál régebbi feladatokat törli

|  |  |  |  |
| --- | --- | --- | --- |
| Metódus | URL | Body | Válasz |
| POST/GET | ./Assets/php/change\_progress.php | JSON | JSON |

**Kéréstörzs (Request Body) Json:**

{

task\_id:1

}

**Lehetséges Hibaüzenetek:**

{

'success' => false, 'message' => 'Nem megfelelő feladat azonosito',"task\_id"=>$task\_id

// Sikertelen lekérdezés

''success' => false, 'message' => 'Query failed: ' . $e->getMessage()

}

Sikeres lekérdezés esetén

{

//sikeres előrehaladás frissités

'success' => true, 'message' => 'Fealdat sikeresen frissitve'

}

**GET paraméterek:**

{

//Üres

}

**Lehetséges Hibaüzenetek:**

{

//Nem támogattot metódus

'success' => false, 'message' => 'Methódus nem megengedett'

// Sikertelen lekérdezés

''success' => false, 'message' => 'Query failed: ' . $e->getMessage()

}

Sikeres lekérdezés esetén

{

'success' => true, 'message' => 'Feladatok sikeresen frissitve régi feladatok törölve'

}

# **all\_task.php**

1. Ellenőrzi be van e jelentkezve a felhasználó

2. Átveszi a bejövő adatokat

3. 2 lehetőség van vagy a jövőbeli feladatokat kérjük le vagy a múlt bélieket ezt az type bejövő paraméter dönti el

|  |  |  |  |
| --- | --- | --- | --- |
| Metódus | URL | Body | Válasz |
| GET | ./Assets/php/all\_tasks.php | JSON | JSON |

**Get paraméterek**

{

//alapesetben future

type: ’past’

}

**Lehetséges Hibaüzenetek:**

{

'success' => false, 'message' => 'Felhasználó nincs bejelentkezve'

'success' => false, 'message' => 'Nem megfelelő típus'

// Sikertelen lekérdezés

''success' => false, 'message' => 'Query failed: ' . $e->getMessage()

}

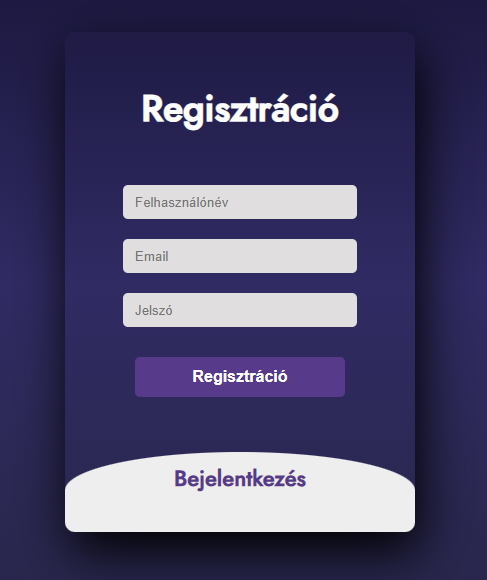
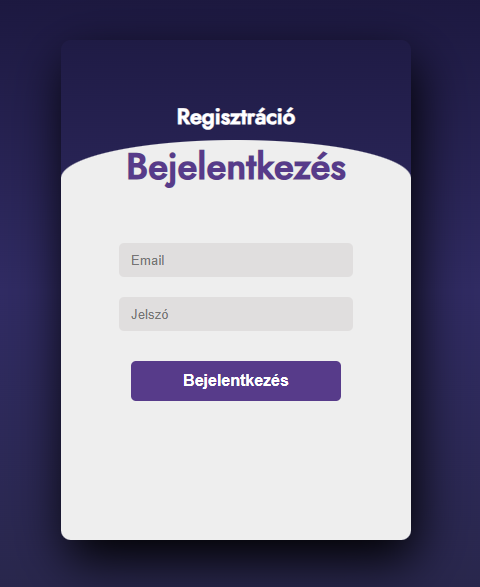
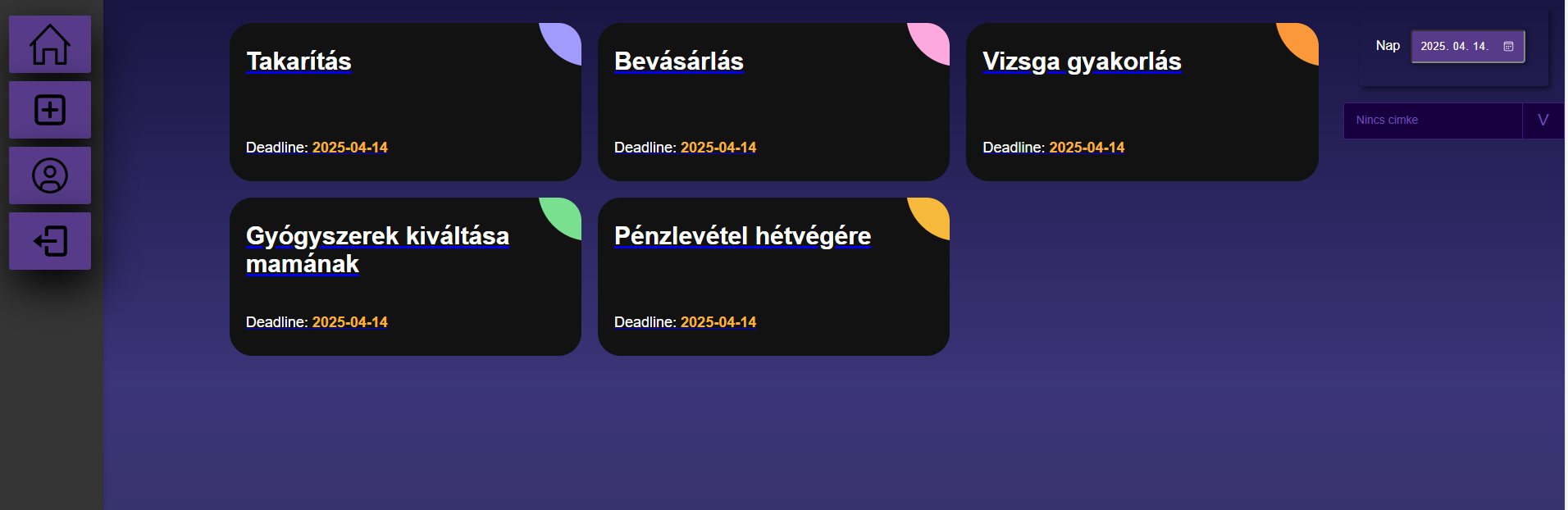
Sikeres lekérdezés esetén

{

'success' => true, 'események' => $events

}

**A program főbb részeinek áttekintése:**

1. **Regisztráció**A regisztrációkor a felhasználónak meg kell adnia ez felhasználónevet, egy emailt, és egy jelszót. Az email és a jelszó a bejelentkezéshez lesz szükséges a későbbiekben ****
2. **Bejelentkezés**Abejelentkezéshez meg kell adnia egy már regisztrált email-t és az ahhoz tartozó jelszót. A bejelentkezés után a felhasználó láthatja a saját oldalát, ahol a saját feladatait tekintheti meg. ****
3. **Főoldal**Ha egy új felhasználó lép be, vagy egy olyan nap van kiválasztva akkor a következő kinézet fogadja**(KEPET IDE BE KELL RAKNI)**:   
     
   Ha az adott napon van feladata a felhasználónak akkor pedig az alábbi kinézettel találkozik: 
4. **Új feladat hozzáadása**
5. **Feladatt állapotának szerkesztése**
6. **Kijelentkezés**