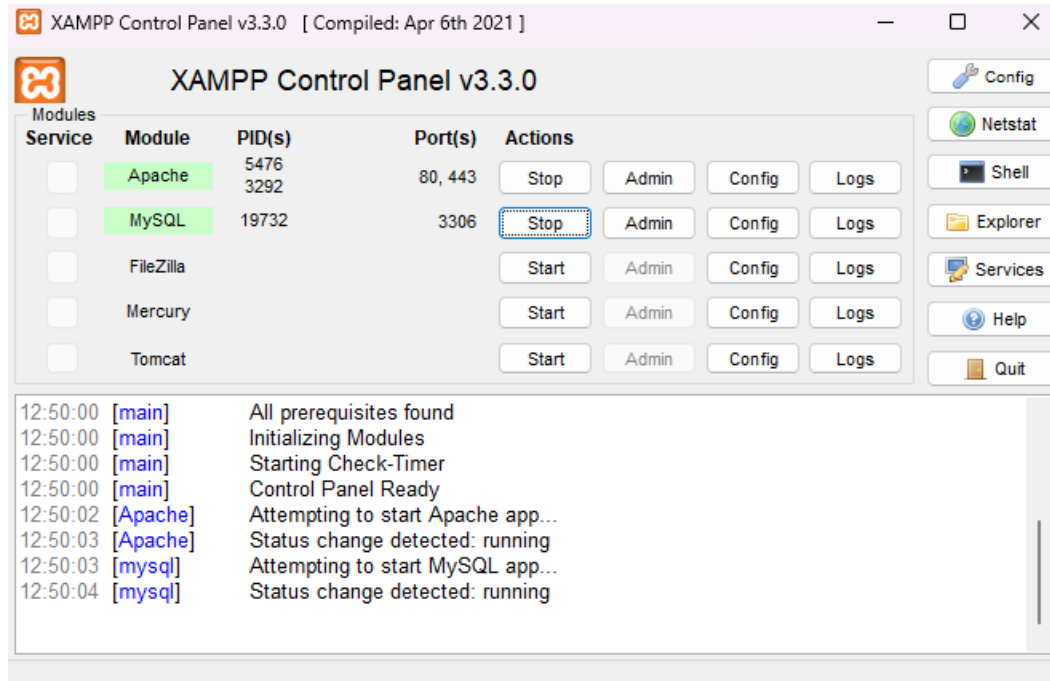


Backend környezet kialakítása

1. XAMPP helyi szerver letöltése, elindítása.



2. Frontend oldali helyi webszerveres futtatás beállítása.

Alapértelmezett elérési frontend út: **C:/xampp/htdocs** beállítása a konfigurációs menüpont alatt.

```
#  
#DocumentRoot "C:/xampp/htdocs"  
#<Directory "C:/xampp/htdocs">  
DocumentRoot "C:/xampp/htdocs"  
<Directory "C:/xampp/htdocs">
```

3. Idopontfoglalo mappa létrehozása az xampp\htdocs mappán belül.

C:\xampp\htdocs\idopontfoglalo\indexe

[szülőkönyvtár]

	Név	Méret	Módosítás dátuma
<input type="checkbox"/>	index.html	272 B	2025. 05. 30. 12:15:58
<input type="checkbox"/>	kapcsolat.php	393 B	2025. 05. 30. 14:28:11
<input type="checkbox"/>	teszt.php	144 B	2025. 05. 30. 14:31:19

4. Helyi XAMPP-s Frontend webszerver elindítása, tesztelése.

← → ↻ localhost/idopontfoglalo/ ★ 📁 📄 🌐 ⋮

Időpontfoglaló rendszer - XAMPP TESZT

5. Adatbázis létrehozása, tesztadatokkal való feltöltése.

A folyamat leírása megtalálható az **Adatbázis létrehozása.pdf** fájlban.

6. Kapcsolat létesítése PHP segítségével a Backend és az Adatbázis között.

```
<?php
// Beállítom az adatbázis szerver elérhetőségét és adatait.
$dbHost = 'localhost';
$dbName = 'idopontfoglalo';
$dbUser = 'root';
$dbPass = '';

try {
    // A PDO objektum létrehozása a MySQL adatbázishoz.
    $pdo = new PDO(
        "mysql:host=$dbHost;dbname=$dbName;charset=utf8", // DSN: szerver,
        $dbUser,
        $dbPass,
        [
            // Hibát íratok ki kivételként, ha bármi probléma adódik.
            PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
            // A lekérdezések eredményét asszociatív tömbként kérem.
            PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC
        ]
    );
} catch (PDOException $e) {
    // Ha nem sikerül a kapcsolódás, JSON formátumban küldöm vissza a
    hibaüzenetet.
    header('Content-Type: application/json'); // Beállítom a válasz
    tartalmát JSON-ra.
    http_response_code(500);
    echo json_encode([
        'error' => 'Adatbázis kapcsolódási hiba: ' . $e->getMessage()

    ]);
    exit;
}
```

7. Kapcsolat ellenőrzése, tesztelése.

```
<?php
// Betölti a kapcsolat.php fájlt, amely létrehozza az adatbázis-kapcsolatot.
require_once("kapcsolat.php");

// Beállítja, hogy a válasz JSON típusú legyen.
header('Content-Type: application/json');

// Egy JSON formátumú üzenetet küld vissza, amellyel jelzi, hogy a kapcsolat
sikeres.
echo json_encode(["status" => "sikeres kapcsolat"]);
?>
```

Kapcsolat sikeres megjelenítése

Olvashatóra formázás ☐

```
{"status": "sikeres kapcsolat"}
```