**Beadandó Feladat Fejlesztési dokumentáció**

Tesztfeladat

Fekete Zoltán

1. A program célja, és feladata

A program célja a tesztfeladat követelményeinek való megfelelés, amelyek a következők voltak:

1. feladat

* Készíts egy futtatható PHP scriptet, ami a Port.hu oldalon elérhető tv műsor alapértelmezetten megjelenített első 5 csatornájának műsorát egy adatbázisba tölti.
* A következő adatokat rögzítsd
  + csatorna neve
  + műsor kezdete
  + műsor címe
  + rövid leírás
  + korhatár
* A scriptet lehessen paraméterezni, hogy melyik nap műsorát töltse le.

2. feladat

* Készíts egy webes felületet, ami megjeleníti az 1. feladat során elkészült script által adatbázisba töltött adatokat
* A felületen egyszerre 1 kiválasztott csatorna kiválasztott napi műsora jelenjen meg
* Legyen lehetőség másik csatornát és másik napot választani.
* A napok kiválasztásánál csak azok a napok legyenek elérhetők, amire vonatkozóan van adat az adatbázisban.
  1. Gépi és nyelvi igények:

A weblaphoz tartozó fájlok értelmezéséhez PHP 8, HTML5, CSS, Bootstrap és SQL ismeretek szükségeltetnek.

Teszt környezet gépleírása:

OS: Microsoft Windows 11 Pro

Processzor: i7-11800H @ 2.30GHz

Memória: 16GB RAM, 1 TB SSD

Használt böngészők: Google Chrome

PHP verzió: PHP 8.1.5

Szerver: Xamp – Apache server

Adatbázis: MySql

* 1. Telepítési leírás
* A fájlok kicsomagolása után a kicsomagolt mappát át kell helyezni a szerver elérési útvonalához (Xampp használatánál C:\xampp\htdocs)
* El kell indítani a szervert és az adatbázist
* Létre kell hozni az adatbázist (MySql-nél http://localhost/phpmyadmin/ felületen importálni kell a Port/Datas/create.sql fájlt, majd futtatni a létrehozáshoz)
* Az oldalt a <http://localhost/Port/> cím alatt érjük el

1. Mappa, fájlszerkezet, állományok bemutatása

A mappaszerkezet bemutatása során megkülönböztetünk többféle csoportot is. A mappák rendezettsége logikai struktúrát takar, ami segíti a fejlesztőt az átláthatóság végett. A feladat során az MVC (Modell–View-Controller) tervezési mintát használtuk, mely a mappák és állományok szerkezetét és kapcsolatait is meghatározta. Kapcsolati téren OOP jelenik meg.

* 1. index.php

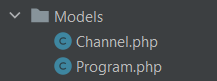
Az alkalmazás belépési pontja. A használt osztályok autoloadolása és az alkalmazáson belüli útválasztás megkezdése.

* 1. loader.php

A LoaderControllert itt példányosítjuk, valamint itt paraméterezzük fel a kívánt dátummal. Az if elágazás a program gyorsaságát szolgálja, ha betöltöttük a kívánt napot, akkor false megadásával kizárhatjuk a betöltést, ami gyorsítja az oldalt. Ha true-ként hagyjuk, akkor újra és újra ellenőrzi, hogy változott-e az alapértelmezett 5 csatorna vagy műsorváltozás történt-e.

* 1. Models

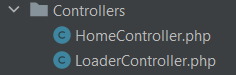
1. ábra



Adatok tárolására szolgáló osztályok, melyek a tárolni kívánt tartalomhoz adaptálódnak (1. ábra).

* Channel.php: az adatbázisból lekérdezett csatornák tárolásához használt osztály
* Program.php: az adatbázisból lekérdezett műsorok tárolásához használt osztály
  1. Controllers

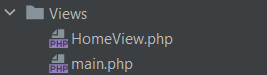
1. ábra



Eseményekre, felhasználói műveletekre reagál. A reakció lehet, az aktuális vezérlőhöz tartozó modell egy függvényhívása vagy adatbázisból lekérdezés és azon adatok átadása (2. ábra).

* LoaderController.php->loader(dátum): az alkalmazásban felhasználandó adatokat lekéri a https://port.hu/tvapi/ aloldalairól. Elsőnek lekérdezi az 5 alapértelmezett csatornát, majd feltölt az adatbázisba, de csak akkor, ha még nincs regisztrálva benne. Ezután csatornákként lekéri a kiválasztott napra letölthető adatokat majd rögzíti őket az adatbázisban, de csak akkor, ha még nincsenek rögzítve. A függvény paramétere a kívánt dátum, amit leszeretnénk kérdezni.
* HomeController.php: az oldal tartalmának elérésért és annak kezelésért felel. A konstruktorban hívódnak meg az adatbázis lekérések, majd lekérdező tulajdonságok segítségével adódnak át a nézetben. Az adatbázis lekérés megy elsőnek végre, lekéri az adatbázisban elérhető dátumokat és csatornákat, hogy ki lehessen őket választani a selectben. Ezután ellenőrzi, hogy a formtól kaptunk-e adatot, ha igen és az adatbázisban is megtalálható, akkor lekérdezzük hozzá a műsorokat, ha nem, akkor az adatbázisból lekérdezett dátum és csatorna első eleme alapján kérdezzük le a műsorokat.
  1. Views

1. ábra

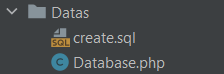


A felhasználónak jeleníti meg a Controllerek által átadott adatokat, valamint egyéb elemeket. Felelős, a felhasználóval való kapcsolattartásért. Html tartalom ténylegesen itt jelenik meg. (3. ábra)

* main.php: a html tartalom kerete vagy váza, ide tudjuk betölteni a nézetek html tartalmait.
* HomeView.php: a főoldalunk kinézete és adatok megjelenítését, hibaüzenetek megjelenítését szolgálja.
  1. Data

Az adatbázis létrehozásához és kezdeti adatfeltöltésre szolgáló állományok találhatóak itt. (4. ábra)

1. ábra



* create.sql: az adatbázist, táblákat és megszorításokat létrehozó SQL utasításokat tartalmazza.
* Database.php: az adatbázis kapcsolatok regisztrálása, osztályba zárása, hívásának egyszerűsítése

1. **Tesztelési terv, és eredményei**  
   A tesztelési terv tartalmazza:

Adatbázis hibás elérése

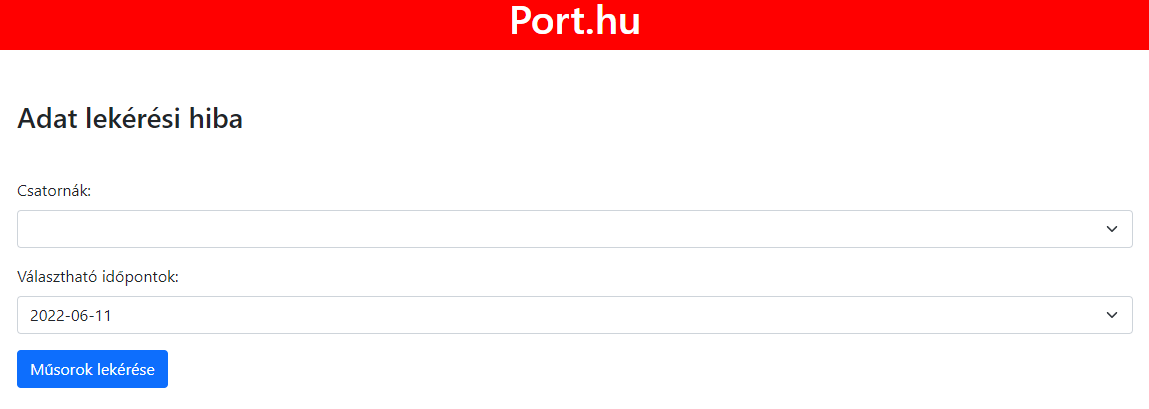
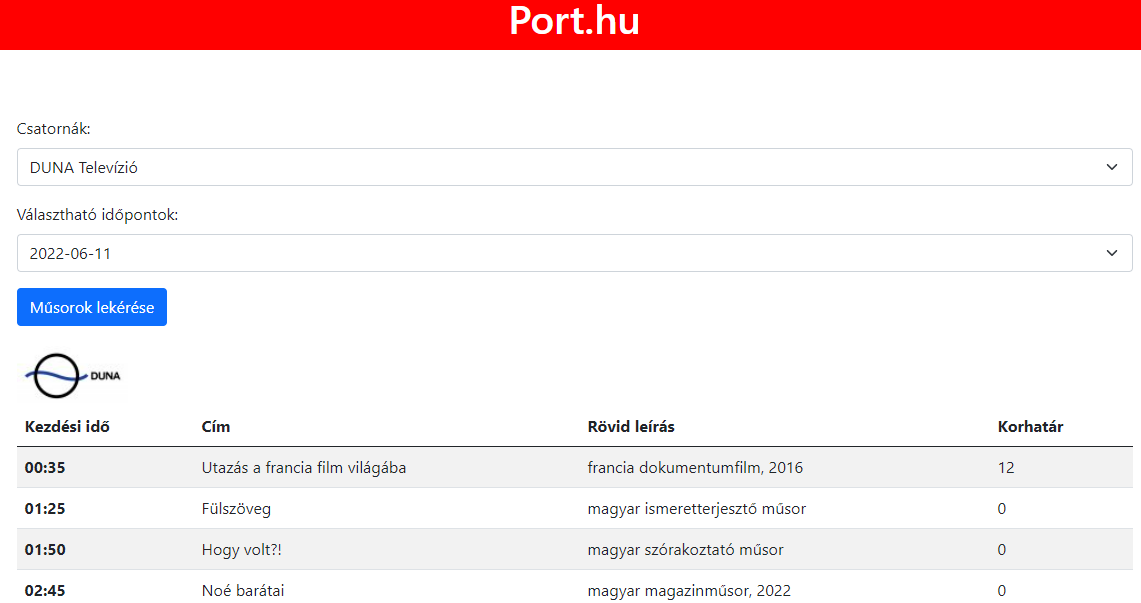
Adatok nélküli kinézet

Betöltés ki- és bekapcsolása

Rossz adatok elérésénél kapott reakció (GET átírása az URL-ban)

Napok bővítése

Csatornák, napok választása

(1)(2)(3)(4)(5)

1. **Program továbbfejlesztési lehetőségei**

A program potenciális továbbfejlesztése során lehetne az oldalakat stílusosabbá tenni, valamint képeket feltölteni, útvonalválasztást deklarálni.