**Beadandó Feladat Fejlesztési dokumentáció**

WEB Programozás II. GINFBAL-WEBPROG2-1

Lovas Bálint (RYGD2D)

Fekete Zoltán (3PCTLA)

2021/22/1 5. félév

Neumann János Egyetem GAMF Műszaki és Informatikai Kar

Mérnökinformatika - Levelező Bsc

Gyakorlat vezető: Dr. Subecz Zoltán

1. A program célja, és feladata

A program célja Webprogramozás-II. első beadandó feladatainak követelményeit megfeleljen. Témaválasztásunk Hatoslottóra esett, melynek hivatalos weboldala volt számunkra mintának követendő. A feladat megoldásához kapcsolódó igényeket kívánja kielégíteni, amiket pontokba szedve tárgyal ez a dokumentáció, hogy hasznos segédje legyen a fejlesztő vizsgálatának

* 1. Gépi és nyelvi igények:

A weblaphoz tartozó fájlok értelmezéséhez PHP, HTML5, Javascript, CSS, Bootstrap és SQL ismeretek szükségeltetnek. PHP-Strom használata javasolt és a futtatásához szükséges gépigény az irányadó a minimális és az ajánlott követelmények alapján.

[https://www.jetbrains.com/help/phpstorm/installation-guide.html#requirements](https://www.jetbrains.com/help/phpstorm/installation-guide.html%23requirements)

* 1. Teszt környezet gépleírása:

PC modellje: Aspire E5-573

OS: Microsoft Windows 10 Pro

Processzor: Intel(R) Core(TM)2 CPU 6420 @2.13GHz 2.13 GHz

Memoría: 8GB RAM

Használt böngészők: Brave, Edge

* 1. A weblap fejlesztéshez kapcsolódó elérhetőségek, hozzáférési adatok

URL[: http://noe.noealapitvany.nhely.hu/](http://noe.noealapitvany.nhely.hu/)

**Nethely**

account: [haraszti.gabor.zoltan@gmail.com](mailto:haraszti.gabor.zoltan@gmail.com)

password: 123Elendil123

**FTP**

account: csosco

password: Elendil123

**PHPmyadmin**

account: noe

password: Elendil123

GIT: <https://github.com/fekez/Noe.git>

* 1. Feladatok eloszlása

**Lovas Bálint**

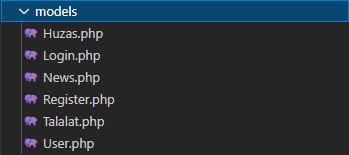
**Fekete Zoltán**

* „Kezdőoldal” tartalom
* „Előző sorsolások” év és hét alapú keresés
* „Hírek” oldal
* Képgaléria létrehozása
* Java és css állományok auto betöltése
* Ezen feladatok dokumentációja

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. Models

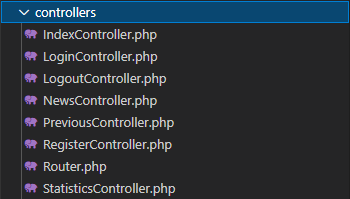


1. ábra

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

* News.php: Hírek/fórum oldal adatszerkezetéért felelős. News osztály a felhasználó nevét, azonosítóját, a bejegyzés dátumát és a bejegyzést képes tárolni. Tartozik hozzá egy konstruktor, mely, inicializálja az elemeket, ha minden paramétert megkap hozzá. Valamint, minden taghoz tartozik egy getter-setter tulajdonság.
  1. Controllers

1. ábra



Eseményekre, felhasználói műveletekre reagál. A reakció lehet, az aktuális vezérlőhöz tartozó modell egy függvényhívása vagy adat átadása a nézetnek.

* NewsController.php: Ennek az osztálynak 3 metódusa van. Az indexAction() függvény felelős a főbb adatokért: Átadja a sanitizeData() és validateData() függvényeknek a POST-olt kommentet, és azok feldolgozhatóvá teszik (szűrik a kapott adatokat, megelőzve a script injectálást vagy egyéb nem érvényes adatokat). Ellenőrzi, hogy kapott-e POST metóduson keresztül bejegyzeni való bejegyzést, valamit, hogy be van-e jelentkezve bármilyen felhasználó. Ha mindkettő igaz, és az adat is feldolgozható, akkor adatbázist hívjuk segítségül. Ellenőrizzük, hogy ez a felhasználó elküldte-e már ezt az üzenetet egyszer? Ha nem, akkor elkezdjük rögzíteni az adatbázisban a commentet az aktuális idővel és felhasználóval együtt. Ezután lekérjük a bejegyzéseket, és átadjuk a helyes nézetnek az adatokat.
  1. Views
  2. Data
  3. Styles
  4. Includes
  5. Scripts
  6. Soap