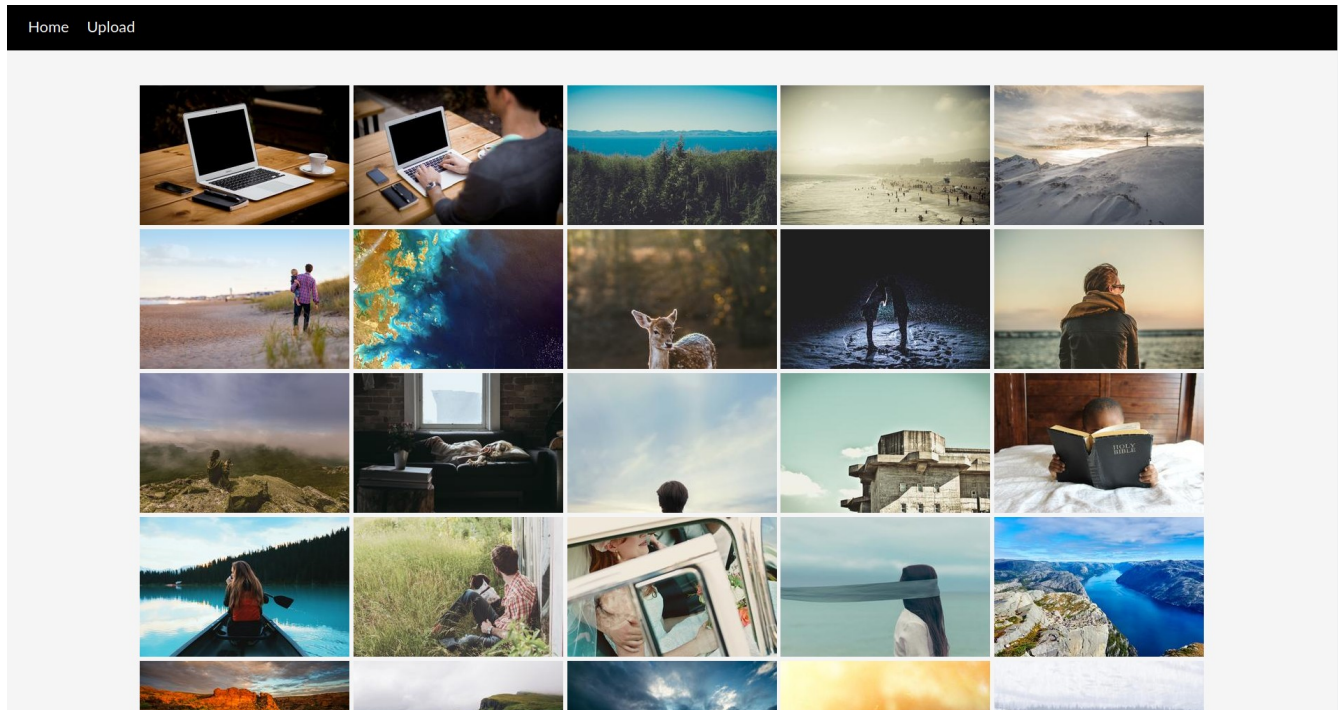


Képek

Valósítsa meg az alábbi veboldalt!



Home

Upload

Add file

Thumbnail

Author

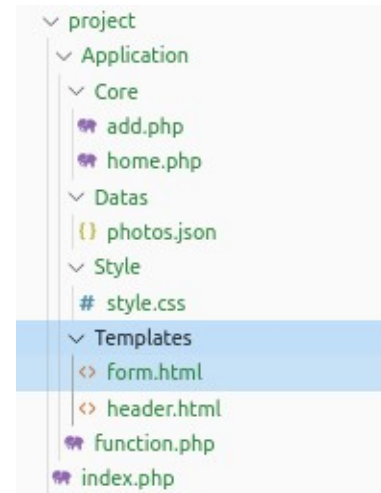
Küldés

A weboldal mappaszerkezete:

Az index.php a gyökérkönyvtárban, az alkalmazás az Application mappában lakik. A Core mappában található található a műveletekért felelős home.php és az add.php. A Datas mappa az adatokat tároló .json fájlt, a Style mappa a stíluslapot tartalmazza. A Template mappába kerülnek a html fájlok(add.html, header.html)

A mellékelt könyvtárban a struktúra már készen található.

A Style mappában megtalálható formázások tetszés szerint alakíthatóak, módosíthatóak. **Az oldal használhatóan legyen megformázva!**



1. Az alkalmazás két oldalból áll. Az index.php egy statikus front kontrollerrel kezeli a \$_GET-el, az url querystring-jéből kapott változókat. Ezek lehetnek home, vagy pl. add, de tetszőleges az elnevezés.
2. Linkelje be a functions és az add.php fájlokat az index.php tetjén és a header.html állományt a body törzsébe. A bodyban egy switch-el csak a kimeneteket generáló állományok legyenek belinkelve, attól függően, hogy a page változó milyen értéket ad vissza.
3. Az index.php-ban csak az oldalváltáshoz szükséges változó és a .json fájl elérési útja legyen definiálva, más művelet itt ne történjen!
4. A header rész két linket tartalmaz, ezek legyenek Home és Upload, melyek az oldal irányításáért felelősek. A szükséges \$_GET értéket, mely lehet pl. page, ezek a linkek adják át az url-hez illesztve → [index.php?home=add](#).
5. Készítsen a functions.php állományban két függvényt, melyek a fájl olvasását illetve írását végzik a szükséges json konvertálással! Használhat magas szintű író/olvasó beépített függvényeket. A beolvasó **függvény PHP tömbben adja vissza a tartalmat, akkor is ha a fájlban nincs adat!**
6. A képek kilistázását az index.php HTML törzsében a home.php végzi, miután beolvastatta a functions.php fájlban található beolvasó függvénnyel a .json állományát.
7. Miután az upload linkre kattintunk, a kapott \$_GET változó értéke alapján az index.php az add.php fájlt hívja meg, melyben ellenőrizze le, hogy a form **submit** változója elküldésre került-e. A form POST metódussal küldje el az adatokat, vagyis azok ne jelenjenek meg az url-ben biztonsági szempontból!

8. Ha a submit változó létezik, az add.php beolvassa a .json fájl tartalmát, hozzáadja egy újabb kételemű tömbként(asszociatív, vagyis név => érték) a kapott értékeket és visszatölti azt a fájlba!

Megj.: Ez kizárólag a fájl felülírása lehet, mert a .json fájl egyetlen objektumot tartalmaz, vagyis össze kell fésülni a már benne lévő objektummal.

9. Miután a fájlbaírás sikeresen lezajlott(ellenőrzés), `header('Location: index.php');` függvénnyel visszairányítjuk az oldalt az index.php-ra, mely már a POST érték hiányában újra kilistázza a fájl tartalmát!

Beadás előtt tesztelje le, hogy a feltöltött kép megjelenik-e a főoldalon.

Megj.. Célszerű a tesztkép url-t más arányértékekkel lementeni, az ellenőrizhetőség kedvéért.

A projektet töltse fel egy saját felhasználóval létrehozott git repository-ba, a további munkák