1) Látogatási napló fájlba (append mód)

Cél: fájlba írás hozzáfűzéssel, soronkénti beolvasás, napló megjelenítése.

Feladat:

Készítsen egy PHP oldalt, amely minden megnyitáskor naplózza a látogatást egy szövegfájlba.

1. Az oldal minden betöltésekor írjon egy új sort a data/visits.log fájl végére.
2. A sor tartalmazza:
    - aktuális dátum és idő (YYYY-MM-DD HH:MM:SS),
    - látogató IP-címe (ha elérhető),
    - böngésző azonosító (User-Agent, ha elérhető).
3. A fájlírás **hozzáfűzéssel** történjen (ne írja felül a korábbi adatokat).
4. Az oldalon jelenítse meg:
    - az éppen most hozzáadott sort,
    - az utolsó 15 bejegyzést (a legfrissebb legyen legfelül).
5. Hozzon létre egy „Napló törlése” funkciót, amely törli a fájlt, majd visszairányít az oldalra.

Extra feladat:
Használjon fájlzárolást az írás során, hogy párhuzamos kérések esetén se sérüljön a napló.

---

2) Vendégkönyv szövegfájlban

Cél: adatok mentése és visszaolvasása egyszerű szövegfájlból.

Feladat:

1. Készítsen űrlapot két mezővel:
    - Név (kötelező),
    - Üzenet (kötelező, maximum 300 karakter).
2. Beküldés után:
    - Ellenőrizze a mezőket szerveroldalon.
    - Hibás mező esetén jelenítsen meg mezőnkénti hibaüzenetet.
3. Siker esetén mentse az adatokat a data/guestbook.txt fájlba úgy, hogy egy sor egy bejegyzést jelentsen.

4. A sor formátuma legyen: időbélyeg | név | üzenet.
5. Az oldalon jelenítse meg az utolsó 20 bejegyzést táblázatban.
6. Legyen lehetőség az összes bejegyzés törlésére.

Extra feladat:
Gondoskodjon arról, hogy a felhasználó ne tudja tönkretenni a fájl szerkezetét (pl. sortörés vagy elválasztó karakter beszúrásával).

---

3) Regisztráció mentése JSON fájlba

Cél: JSON fájl használata strukturált adat tárolására.

Feladat:

1. Hozzon létre egy regisztrációs űrlapot:
   - Felhasználónév (minimum 3 karakter),
   - E-mail (érvényes formátum).
2. Beküldéskor:
   - Töltse be a data/users.json fájl tartalmát (ha létezik).
   - Ellenőrizze, hogy az e-mail cím még nem szerepel a fájlban.
3. Siker esetén:
   - Hozzon létre új rekordot (id, username, email, created_at).
   - Mentse vissza az összes rekordot JSON formátumban.
4. Jelenítse meg az összes regisztrált felhasználót táblázatban.
5. Legyen „JSON törlése” funkció.

Extra feladat:
A JSON fájlt olvasható (szépen formázott) formában mentse.

---

4) Todo lista JSON-ben (állapotváltással)

Cél: fájl alapú „mini adatbázis” CRUD műveletekkel.

Feladat:

1. Készítsen oldalt új feladat rögzítésére (max. 120 karakter).
2. A feladat mentése JSON fájlba történjen (data/todos.json).
3. Minden feladat tartalmazza:

- egyedi azonosító,
- szöveg,
- állapot (kész / nem kész),
- létrehozási idő.

4. A listában minden sor mellett legyen:
   - „Állapot váltása” gomb,
   - „Törlés” gomb.
5. Legyen „Összes törlése” funkció.

Extra feladat:
Minden POST művelet után alkalmazzon PRG mintát.

---

5) Jegyek tárolása CSV fájlban

Cél: CSV fájl írása és beolvasása.

Feladat:

1. Készítsen űrlapot:
   - Név,
   - Neptun-kód (6 karakter),
   - Jegy (1-5).
2. Beküldéskor:
   - Ellenőrizze az adatokat.
   - Írjon egy új sort a data/grades.csv fájl végére.
3. Olvassa vissza a CSV fájlt, és jelenítse meg az adatokat táblázatban.
4. Számolja ki és jelenítse meg az átlagot.
5. Legyen törlési lehetőség.

Extra feladat:
Alkalmazzon fájlzárolást írás közben.

---

6) Logfájl elemzése (napi összesítés)

Cél: fájl beolvasása és adatok csoportosítása.

Feladat:

1. Olvassa be a data/access.log fájl sorait.
2. A sorok elején szereplő dátum alapján számolja ki, hogy egy adott napon hány esemény történt.
3. Jelenítse meg táblázatban:
    - Dátum,
    - Események száma.
4. Legyen „Új sor hozzáadása” funkció.
5. Legyen „Log törlése” funkció.

Extra feladat:
Rendezze a táblázatot dátum szerint növekvő sorrendbe.

---

7) Képfeltöltés és fájltárolás

Cél: fájlfeltöltés kezelése szerveroldalon.

Feladat:

1. Készítsen fájlfeltöltő űrlapot (multipart/form-data).
2. Csak képfájlokat engedélyezzen (JPG/PNG/GIF).
3. Legyen maximális méretkorlát (pl. 2MB).
4. Mentse a fájlt uploads/ mappába.
5. Ne az eredeti fájlnevet használja.
6. Jelenítse meg az eddig feltöltött képeket táblázatban előnézettel.
7. Legyen törlési lehetőség.

Extra feladat:
Védje a törlési funkciót egyszerű névellenőrzéssel (ne lehessen mappán kívüli fájlt törölni).

---

8) Konfiguráció mentése JSON fájlban

Cél: konfigurációs adatok tárolása és szerkesztése.

Feladat:

1. Kezelje a következő beállításokat:
    - site_title (szöveg),
    - items_per_page (1-100),
    - maintenance (igen/nem).
2. Ha nincs config fájl, hozza létre alapértékekkel.
3. A form mezői legyenek előtöltve.
4. Mentés után írja vissza a JSON fájlba.
5. Jelenítse meg az aktuális beállításokat táblázatban.

---

9) Egyszerű cache rendszer fájlban

Cél: számítás eredményének mentése és újrafelhasználása.

Feladat:

1. Kérjen be egy N értéket (100-50000).
2. Számolja ki, hány prímszám van 2 és N között.
3. Mentse el az eredményt egy JSON fájlba:
    - N értéke,
    - számítás időpontja,
    - eredmény.
4. Ha ugyanarra az N-re érkezik kérés 30 másodpercen belül, ne számoljon újra.
5. Jelenítse meg, hogy az eredmény cache-ből vagy újraszámolással jött.
6. Legyen cache törlési lehetőség.

---

10) Mini „adatbázis” JSON-ben (keresés + rendezés)

Cél: összetett fájl alapú adattárolás.

Feladat:

1. Tároljon könyv adatokat JSON fájlban:

   - id,
   - cím,
   - szerző,
   - év.
2. Legyen lehetőség új könyv hozzáadására.
3. Listázza a könyveket táblázatban.
4. Legyen törlési lehetőség.
5. Készítsen GET alapú keresést (címben vagy szerzőben).
6. Készítsen GET alapú rendezést:
   - cím A→Z / Z→A,
   - év növekvő / csökkenő.
7. Legyen teljes fájl törlés.