

## 1) Alap validáció mezőnként + értékek visszatöltése

**Cél:** mezőnkénti hibaüzenetek, hibák gyűjtése, beküldött értékek megtartása.

**Feladat:**

1. Készítsen űrlapot a következő mezőkkel: **név, e-mail, életkor**.
2. Beküldés után szerveroldalon ellenőrizze:
  - o a név nem lehet üres,
  - o az e-mail nem lehet üres és legyen érvényes formátumú,
  - o az életkor nem lehet üres, csak egész szám lehet, és ésszerű tartományban legyen (pl. 0-120).
3. A hibákat **mezőnként** jelenítse meg közvetlenül a mező alatt.
4. Hibás beküldés esetén az űrlap mezői **maradjanak kitöltve** a korábban megadott értékekkel.
5. Sikeres beküldéskor jelenítsen meg egy összefoglalót a kapott adatokról.

**Extra feladat:** legyen egy általános hibaüzenet is („Kérem, javítsa a hibás mezőket.”).

---

## 2) Regisztráció validációval + sikeres mentés sessionbe (demó)

**Cél:** összetettebb validáció, „Redirect-After-POST” minta, session adattárolás.

**Feladat:**

1. Készítsen regisztrációs űrlapot mezőkkel: **felhasználónév, e-mail, jelszó, jelszó ismét**.
2. Ellenőrizze szerveroldalon:
  - o felhasználónév: kötelező, minimum 3 karakter,
  - o e-mail: kötelező, formailag érvényes,
  - o jelszó: kötelező, minimum 8 karakter, és legyen benne legalább 1 kisbetű, 1 nagybetű, 1 szám,
  - o a két jelszó egyezzen.
3. Hiba esetén jelenítsen meg mezőnkénti hibaüzeneteket és töltse vissza a felhasználónév/e-mail mezőket.
4. Sikeres beküldés esetén tárolja az adatokat **sessionben** (oktatási demóként), majd:

- végezzen átirányítást ugyanarra az oldalra (hogy frissítésre ne ismétlődjön a POST),
  - és jelenítsen meg egy „sikeres mentés” üzenetet.
5. Az oldalon külön blokkban jelenítse meg az **utolsó rögzített regisztráció** adatait a sessionből.

**Extra feladat:** jelezze a felületen, hogy valós rendszerben ez adatbázisba kerülne.

---

### 3) Radio / checkbox / select mezők validálása

**Cél:** különböző űrlapelemek kezelése, megengedett értékek ellenőrzése.

**Feladat:**

1. Készítsen űrlapot:
  - teljes név (szöveg),
  - képzési szint (radio: pl. BSc/MSc),
  - érdeklődési témák (checkbox: legalább 4 opción, pl. PHP/HTML/CSS/JS),
  - időzóna (select: legalább 3 opción).
2. Beküldés után ellenőrizze:
  - a teljes név nem lehet üres,
  - a radio értéke csak a megengedett opciónok egyike lehet,
  - legalább 1 checkbox legyen kijelölve,
  - a select értéke csak a megadott listából jöhet.
3. Hibás mezőknél mezőnként írjon hibaüzenetet.
4. Siker esetén jelenítsen meg összefoglalót, ahol a választások szépen listázva láthatók.

**Extra feladat:** biztosítsa, hogy a visszatöltés korrekt legyen (a korábbi jelölések megmaradjanak).

---

### 4) Többlépcsős űrlap (step 1-2) session tárolással

**Cél:** több oldal/lépés állapotának kezelése sessionnel.

**Feladat:**

1. Készítsen 2 lépéses űrlapot:

- **1. lépés:** teljes név, Neptun-kód,
  - **2. lépés:** város, irányítószám.
2. Első lépés beküldése után:
- validálja a mezőket (Neptun: 6 karakter, betű/szám),
  - mentse a valid adatokat sessionbe,
  - majd léptessen a 2. lépéstre.
3. Második lépésnél:
- validálja a várost (nem üres),
  - validálja az irányítószámot (4 számjegy, ésszerű tartomány),
  - mentse sessionbe,
  - és menjen az összegzés oldalra.
4. Az összegzésben jelenítse meg táblázatban a 4 adatot.
5. Legyen „Újrakezdés” gomb/link, ami törli az adott feladathoz tartozó session adatokat.

**Extra feladat:** ha a 2. lépést úgy próbálja elérni, hogy az 1. lépés nincs kitöltve, kezelje hibával.

---

## 5) Flash üzenetek + Redirect-After-POST (PRG)

**Cél:** egyszeri (következő betöltésig élő) üzenetek sessionben.

**Feladat:**

1. Készítsen egy oldalt, ahol a felhasználó egy rövid üzenetet (note) írhat be és elmenthet.
2. Beküldéskor:
  - ha üres az üzenet, tároljon egy „hiba” flash üzenetet,
  - ha nem üres, mentse el az üzenetet sessionben egy listába, és tároljon „siker” flash üzenetet.
3. A beküldés után mindig irányítson vissza ugyanarra az oldalra (PRG minta).
4. Az oldalon jelenítse meg:
  - a flash üzenetet (siker/hiba),
  - a mentett üzenetek listáját táblázatban.

5. Legyen „lista törlése” funkció (session lista törlése).

**Extra feladat:** a lista elemeinél jelenjen meg mentési időpont is.

---

## 6) Egyszerű login/logout + védett oldal

**Cél:** bejelentkezési állapot sessionben, belépéskor biztonságos session-kezelés.

**Feladat:**

1. Készítsen bejelentkezési űrlapot felhasználónév + jelszó mezőkkel.
2. Legyen legalább 2 demó felhasználó (pl. admin, user) előre definiálva.
3. Beküldés után ellenőrizze:
  - o a felhasználó létezik-e,
  - o a jelszó helyes-e.
4. Siker esetén:
  - o tárolja a bejelentkezett felhasználó adatait a sessionben,
  - o irányítson a „védett” oldalra,
  - o jelenítsen meg üdvözlést és belépési időt.
5. Ha nincs bejelentkezés, a védett oldal legyen tiltott és kérje bejelentkezésre.
6. Legyen kijelentkezés funkció, ami törli a sessiont.

**Extra feladat:** belépéskor használjon session-azonosító frissítést ( jó gyakorlat).

---

## 7) Szerepkör alapú hozzáférés (admin/user)

**Cél:** jogosultság-kezelés session alapján.

**Feladat:**

1. A login rendszerben tároljon szerepkört (pl. admin vagy user).
2. Készítsen két védett oldalt:
  - o dashboard: minden bejelentkezett user láthatja,
  - o admin oldal: csak admin láthatja.
3. Ha a user megpróbál admin oldalt nyitni:
  - o kapjon „Hozzáférés megtagadva” üzenetet,
  - o és legyen link vissza a dashboardra.

4. A dashboardon jelenítse meg a felhasználónevét, szerepkörét és belépési időt.

**Extra feladat:** állítsan be megfelelő HTTP státuszkódot tiltás esetén (403).

---

## 8) Látogatásszámláló + utolsó látogatás időpontja (session)

**Cél:** sessionben egyszerű perzisztens számlálók, időpontok kezelése.

**Feladat:**

1. minden oldalbetöltéskor növeljen egy sessionben tárolt számlálót (hányadik megnyitás).
  2. Tárolja sessionben az utolsó látogatás időpontját.
  3. Jelenítse meg:
    - o aktuális megnyitások száma,
    - o aktuális idő,
    - o előző látogatás időpontja (ha volt).
  4. Legyen „nullázás” funkció, ami törli a számlálót és a last-visit adatot.
- 

## 9) Session alapú kosár (add / remove / update)

**Cél:** összetett session adatstruktúra kezelése, több művelet.

**Feladat:**

1. Készítsen egy fix katalógust legalább 4 termékkel (név + ár).
2. Jelenítse meg a katalógust táblázatban, minden termékhez legyen „Kosárba” gomb.
3. A kosár sessionben tárolódjon (termék azonosító → mennyiség).
4. A kosár résznél jelenítse meg:
  - o termék neve, egységár,
  - o mennyiség (módosítható),
  - o részösszeg,
  - o soronként törlés,
  - o végösszeg.
5. Legyen „mennyiségek frissítése” és „kosár ürítése” funkció.

6. Műveletek után használjon PRG mintát (átirányítást), és jelenítsen meg sikeres üzenetet.
- 

## 10) CSRF védelem demo: token sessionben + ellenőrzés

**Cél:** alap webbiztonsági szemlélet, session tokenek.

**Feladat:**

1. Készítsen egy űrlapot (pl. üzenetküldés) mezőkkel:
  - tárgy,
  - üzenet (minimum 10 karakter).
2. Az űrlapban szerepeljen egy rejtett mező, amely egy sessionben tárolt „CSRF token” értékét tartalmazza.
3. Beküldéskor ellenőrizze:
  - a token létezik és egyezik a sessionben tárolttal,
  - ha nem egyezik, utasítsa el a kérést és jelezze hibával.
4. Sikeres beküldéskor mentse az üzenetet session listába és jelenítse meg táblázatban a mentett üzeneteket.
5. Jó gyakorlatként sikeres mentés után generáljon új tokent.