**Ötletek**

(saját ötletek lejjebb)

**ChatGPT**: Kérdésem volt, hogy „Milyen vizsgaremeket csináljunk Szoftverfejlesztő -és tesztelői szakképzésre?”

**1. Egyszerű ügyfélkapcsolati (CRM) rendszer**

* **Funkciók**: Ügyféladatok kezelése (hozzáadás, módosítás, törlés), keresés, ügyfélkapcsolatok követése, jegyzetek készítése az ügyfelekről.
* **Technológiák**: Webes vagy asztali alkalmazás, backend adatbázis (pl. MySQL, SQLite), frontend keretrendszer (pl. React, Angular).
* **Tesztelés**: Funkcionális tesztek, adatbázis tesztek, teljesítmény tesztelés.

**2. Feladatkezelő alkalmazás**

* **Funkciók**: Feladatok hozzáadása, szerkesztése, határidők kezelése, prioritások beállítása, feladatok állapotának követése.
* **Technológiák**: Webes vagy mobil alkalmazás, REST API, adatbázis (pl. PostgreSQL, MongoDB).
* **Tesztelés**: Unit tesztelés, API tesztelés, felhasználói felület tesztelése.

**3. E-commerce webáruház**

* **Funkciók**: Termékek böngészése, kosár kezelés, rendelés leadása, fizetési rendszer integráció, felhasználói fiók kezelés.
* **Technológiák**: Webes alkalmazás, különböző backend és frontend technológiák (pl. Django és Vue.js).
* **Tesztelés**: End-to-end tesztek, biztonsági tesztek, teljesítmény tesztek.

**4. Egyszerű közösségi média platform**

* **Funkciók**: Felhasználók regisztrációja, profilkezelés, posztolás, kommentelés, lájkolás.
* **Technológiák**: Full-stack fejlesztés, felhasználói hitelesítés, real-time funkciók WebSocket-ekkel.
* **Tesztelés**: Integrációs tesztelés, terhelési tesztek, biztonsági tesztek (pl. XSS támadások ellen).

**5. Automatizált tesztelő keretrendszer**

* **Funkciók**: Olyan eszköz vagy keretrendszer létrehozása, amely különböző alkalmazások automatizált tesztelését végzi el (pl. Selenium alapú webes tesztelés).
* **Technológiák**: Python vagy Java alapú fejlesztés, CI/CD integráció.
* **Tesztelés**: A keretrendszer saját tesztelése, regressziós tesztelés.

**6. Banki tranzakció szimulátor**

* **Funkciók**: Felhasználók regisztrációja, egyenleg nyomon követése, átutalások szimulálása, tranzakciós előzmények tárolása.
* **Technológiák**: Biztonságos hitelesítés, adatbázis kezelés, tranzakciók atomikus kezelése.
* **Tesztelés**: Adatkonzisztencia tesztek, biztonsági tesztek, stressz tesztek.

**7. Időjárás előrejelző alkalmazás**

* **Funkciók**: Aktuális időjárás és előrejelzések megjelenítése, térképes vizualizáció, API-k használata (pl. OpenWeatherMap).
* **Technológiák**: API integráció, frontend fejlesztés térképes megjelenítéssel.
* **Tesztelés**: API tesztelés, felhasználói felület tesztelése, mobil alkalmazás esetén eszközspecifikus tesztek.

**8. E-learning platform**

* **Funkciók**: Kurzusok létrehozása, felhasználók regisztrálása, videók és anyagok megosztása, tesztek készítése és kiértékelése.
* **Technológiák**: Webes alkalmazás, videó streaming, felhasználói jogosultságok kezelése.
* **Tesztelés**: Különböző tesztelési technikák, beleértve a biztonsági, funkcionális és teljesítmény teszteket is.

**9. Webalkalmazás egy kisvállalkozás számára:**

* **Leírás:** Készíts egy webalkalmazást egy kisvállalkozás (például étterem, szálláshely vagy könyvesbolt) számára. Az alkalmazás tartalmazhat rendelési rendszert, foglalási funkciót, termékkatalógust, felhasználói profilkezelést, stb.
* **Fejlesztési fókusz:** Backend (adatbázis-kezelés, API fejlesztés) és frontend (reszponzív design).
* **Tesztelési fókusz:** Funkcionális tesztelés, felhasználói tesztelés, automatizált tesztek (pl. Selenium).

**10. Mobilalkalmazás fitnesz nyomon követésére:**

* **Leírás:** Egy mobilalkalmazás, amely lehetővé teszi a felhasználók számára, hogy nyomon kövessék edzéseiket, étrendjüket, és fejlődésüket. Az alkalmazás tartalmazhat edzésterveket, értesítéseket, célkitűzéseket és statisztikákat.
* **Fejlesztési fókusz:** Mobil app fejlesztés (iOS/Android), adatbázis-kezelés.
* **Tesztelési fókusz:** Unit tesztek, integrációs tesztek, felhasználói élmény tesztelése.

**Saját:** én valamilyen foglalós rendszerre gondoltam, amibe bele lehetne integrálni fizetési opciót

* **Maszek orvosi szakrendelés**
  + Milyen fajta? -> ortopédus, kardiológus, stb.
  + Melyik orvoshoz?
  + Orvos bemutatkozása -> felsorolás, hogy az adott orvos mivel foglalkozik
  + Szabad időpontok megjelenítése
  + Kezelés hossza
  + Árak
  + fizetési opcióval (foglalás után megkérdezi, hogy online vagy személyesen szeretnénk fizetni)
  + Visszaigazoló e-mail a foglalásról
  + Előtte való nap vagy aznap x órával előtte figyelmeztető SMS/e-mail
  + Ha x időn belül nem mondja le, akkor is ki kell fizetni, ha nem megy el
  + Rendelési idő változás esetén vagy ha nincs a doki, új időpont egyeztetés (felhívnak)

**(GPT) ↓**

**Funkciók:**

1. **Felhasználói regisztráció és bejelentkezés**
2. **Orvosok és szakrendelések keresése**
3. **Időpontfoglalás kiválasztása és visszaigazolása**
4. **Online fizetési lehetőség (konzultációs díj)**
5. **Emlékeztetők küldése a pácienseknek**
6. **Admin felület az orvosoknak az időpontok kezelésére**
7. **Adatok biztonságos tárolása és kezelése**

**Technológiák:**

* **Backend**: Flask (Python), Express.js (Node.js) vagy Django (Python)
* **Frontend**: React vagy Angular
* **Adatbázis**: PostgreSQL vagy MongoDB
* **Fizetési integráció**: Stripe vagy PayPal
* **E-mail szolgáltatás**: SendGrid vagy SMTP alapú rendszer

**Fejlesztési lépések:**

1. **Adatbázis tervezése**: Táblák létrehozása a felhasználók, orvosok, időpontok és tranzakciók számára.
2. **Backend fejlesztés**: API végpontok készítése a regisztrációhoz, bejelentkezéshez, időpontfoglaláshoz és fizetésekhez.
3. **Frontend fejlesztés**: Felhasználói felület kialakítása a foglaláshoz, kereséshez és fizetéshez.
4. **Fizetési integráció**: Stripe vagy PayPal integrálása a fizetési folyamatokhoz.
5. **Tesztelés**: Egységtesztek, integrációs tesztek és végpontok tesztelése.
6. **Biztonsági intézkedések**: Adatvédelem és felhasználói adatok titkosítása.

**VAGY**

**Szállásfoglalási rendszer**

**Funkciók:**

1. **Felhasználói regisztráció és bejelentkezés**
2. **Szálláshelyek keresése és listázása**
3. **Foglalási időpontok kiválasztása**
4. **Online fizetési lehetőség (előleg vagy teljes összeg)**
5. **Visszaigazoló e-mailek küldése**
6. **Admin felület a szállásadóknak a foglalások kezelésére**
7. **Vélemények és értékelések a szálláshelyekről**

**Technológiák:**

* **Backend**: Laravel (PHP), Node.js (Express.js) vagy Spring Boot (Java)
* **Frontend**: Vue.js vagy React
* **Adatbázis**: MySQL vagy MongoDB
* **Fizetési integráció**: PayPal vagy Stripe
* **E-mail szolgáltatás**: Mailgun vagy SMTP alapú rendszer

**Fejlesztési lépések:**

1. **Adatbázis tervezése**: Táblák létrehozása a felhasználók, szálláshelyek, foglalások és tranzakciók számára.
2. **Backend fejlesztés**: API végpontok készítése a regisztrációhoz, bejelentkezéshez, foglaláshoz és fizetésekhez.
3. **Frontend fejlesztés**: Felhasználói felület kialakítása a kereséshez, foglaláshoz és fizetéshez.
4. **Fizetési integráció**: PayPal vagy Stripe integrálása a fizetési folyamatokhoz.
5. **Tesztelés**: Egységtesztek, integrációs tesztek és végpontok tesztelése.
6. **Biztonsági intézkedések**: Adatvédelem és felhasználói adatok titkosítása.