

# TODO Lista

Készíts egy (többrétegű) alkalmazást. A kódot töltsd fel egy általunk is elérhető git repository-ba, amelynek linkjét küld el nekünk. (Egy-egy kész feladatrésszel után commitolj, ne csak a végeredményt töltsd fel.)

A kódolás során tartsd be az objektumorientált tervezés és programozás szabályait. Olyan kódot szeretnénk kérni, ami megítélésed szerint jól reprezentálja a fejlesztői tudásodat. Válassz olyan architektúrát, ami illik a feladathoz, ha úgy gondolod, használj nyugodtan 3rd party library-eket.

## Technológia

Alap esetben az általad választott, legjobban ismert technológiákat alkalmazd, ami jól reprezentálja a tudásodat, valamint a fejlesztési ismereteidet mind adatbázis, mind backend oldalon. Amennyiben az email kíséző szövegében szerepel konkrét technológia megjelölés, úgy azt tekintsd mérvadónak.

## Feladat leírás

Az alkalmazás tudjon egy relációs adatbázisból dolgozva usereket és hozzájuk tartozó todo-kat kezelni (Create, Read, Update, Delete) REST API-n keresztül.

A feladat megoldásának csak opcionális része a UI készítése, de az API tervezését a frontend igények alapján kell elvégezni. A backend főleg tervezett felületen lehessen kilistázni a usereket és egy userre kattintva a hozzá tartozó todo-kat. A userhez lehessen feltölteni profilképet. A UI-hoz további vonalvezető, amit a backendnek támogatnia kell: Az első képernyő egy user listából áll, ahol lehet felvenni/módosítani/törölni usereket. Egy userre rákattintva modal-ban jelenjenek meg a hozzá tartozó todo-k és itt is lehessen felvenni/módosítani/törölni todo-kat.

Az adatszerkezet megtervezése és a REST API elkészítése a feladat központi része.

## Opcionális extra feladatok

- Regisztráció/Login: user/password alapú bejelentkezés
- Jogosultság kezelés: admin szerepkörrel mindent láthasson a user, a szerepkör nélkül pedig csak a saját elemeit
- Email küldés: pl. egy todo határidejének közeledtéről
- Logolás: minden TODO elemen végzett műveletről készüljön log bejegyzés egy általad választott logolási megoldással (ki, mikor, melyik element, mit változtatott)
- Integrációs tesztek: tesztelési célt szolgáló alkalmazás felállítása a build részeként, API tesztek
- Unit tesztek: az alkalmazás üzleti logikáját reprezentáló valamely komplexebb osztályhoz
- UI készítés

Legyél benne biztos:

- Az alkalmazás lefut és buildelhető
- Minden követelménynek megfelel
- A kód szervezett, jól érthető
- Az alkalmazáshoz készíts egy rövid leírást, hogyan tudjuk lebuildelni, futtatni
- Készíts sql dump-ot az adatbázisról és tölts fel a git repository-ba

*Ha kérdésed van olyannal kapcsolatban, ami nincs specifikálva, nyugodtan adhatsz rá saját kútfőből megoldást, hogy ne kelljen válaszra várnod.*