

Szerveroldali fejlesztés féléves feladat

A féléves feladat témáját és funkcióit nektek kell megtervezni! Az elvárás az alábbi táblázat szerint minél több elem felhasználása!

A feladatot githubon kell fejleszteni és meg kell hívni rá az **oenikprog** usert! A repository elnevezése: **ABC123_SOF_2022231**

(ABC123 egy példa neptunkód, SOF a szerveroldali fejlesztés rövidítése, 2022231 a jelenlegi szemeszterkód). Több főss munká esetén a csapattagok listáját (név és neptunkód) a repository gyökerében egy szöveges fájlban kell elhelyezni! Ekkor a csapatvezető neptunkódja szerepeljen a repó nevében!

Az alkalmazást MVC minta segítségével kell elkészíteni, a feladat teljesítéséhez szükséges modulok: 2-6 és 9-10. Az API alapú fejlesztés (8) és Autorizáció API környezetben (9) modulok a „Kliensoldali fejlesztés” tantárgy szerveroldali ismereteit tartalmazzák. Majd ezen a ráépülő tárgyon kell API végpontot és kliensalkalmazást fejleszteni!

Kötelező elemek, melyek nélkül a feladat nem fogadható el (összesen 50 pont)

Az alkalmazás...	Pont
az MVC tervezési minta segítségével készült, legalább 2 Controllert és összesen legalább 10 Actiont tartalmaz	5
legalább SQL adatbázisra vagy valamilyen NOSQL adatbázisra épül	5
legalább lokálisan tárolt felhasználó és szerepkörkezelést tartalmaz	5
adatbázisában seedelt adatok találhatóak	5
adatbázisában (az identity táblákon felül) legalább 2 táblát tartalmaz	5
publikusan elérhető egy domain címen keresztül bárki számára (nem szükséges saját domain, azure-os is megfelelő)	5
tartalmaz a template-ben biztosított felül valamilyen alapvető formázásokat CSS-el (saját vagy library)	5
validációkat tartalmaz mind kliens – mind szerveroldalon	5
a controller rétegtől elválasztott logic réteget tartalmaz (de ez egybeolvadhat a repository réteggel is)	5
githubon készült el	5

Szabadon választott elemek, amelyekkel pluszpontok gyűjthetőek (50 pont maximum / fő)

Az alkalmazás...	Pont
rendelkezik API végponttal is	3
rendelkezik API kliensalkalmazással is, amelyen valamilyen rész-adatok megjelennek (pl. aggregált adat, statisztika)	5
az API kliensalkalmazás signalR segítségével folyamatosan frissíti a felületét	4

tartalmaz kép kezelést (feltöltés/listázás)	3
használ valamilyen SQL-től eltérő adatbázist (is) pl. cassandra, mongodb, blob storage, stb.	5
használ email küldést a regisztrációk megerősítésére és/vagy bizonyos események teljesülésekor a felhasználók értesítésére	4
időzített feladatot is tartalmaz Azure Functions vagy valamilyen alternatív megoldással (pl. crontab). pl: akciók beállítása pontban éjfélkor	5
használ social login providert is a belépéshez	5
profilkép kezelést tartalmaz, social login provider esetén letölti onnan a képet	5
használ valamilyen külső CSS library-t (pl. bootstrap) és az oldal felépítése során legalább 3 különböző komponenst vett onnan igénybe	4
a logic rétegben használ valamilyen tanult tervezési mintát 3p/pattern	3
valamelyik view-ban használ önállóan írt JS kódot (pl. egy űrlapelemhez autocomplete szerveroldalról letöltött adatokkal)	5
ModelBinderbe szervezi a modellek legyártását a form adatokból	3
reszponzívan jelenik meg telefonról nézve	5
logic rétegében használ valamilyen haladó algoritmust (pl. gráf, backtrack, optimalizáció, mohó algoritmus)	6
valamilyen hosszan futó feladatot úgy old meg, hogy adatbázisban tárolja az input adatokat és egy worker (pl .virtuális gép, azure functions) valamikor kiszámolja az eredményt és visszaírja az adatbázisba	8
használ valamilyen CI/CD megoldást. Pl. githubon masterre mergelve elindul egy folyamat (pl. github actions), amely az éles szerveren cseréli a buildet (lásd: Azure App Service)	5
elérhető docker hubon image formájában is	2
virtuális gépen/felhőben és azon belül docker engine-en van futtatva	5
használ valamilyen terheléelosztást/replica setet (pl. kubernetes)	6

Plusz feladatnak saját ötleteket is fogadok óra idejében. Illetve a pluszpont táblázat irányadó, kiemelkedő megoldások még több pontot érhetnek (pl. CSS frameworköt használ valaki legalább 3 komponenssel → 4 pont jár, de ha még ráadásul iszonyat szép is az oldal, akkor ez felmehet több pluszra is).

A féléves feladat összefügghet futó szakdolgozattal/projektmunkával/TDK munkával is. Tehát egyáltalán nem elvárás egy izolált projektet készíteni a tárgyra (sőt, kifejezetten hülyeség).

A féléves feladat csoportmunkában (max. 3 fő) is megoldható. Ekkor viszont egy leírást kell készíteni githubra, hogy melyik pluszt ki fejlesztette és az a személy kapja meg rá a pontot.

Csapatmunka esetén a kommunikációs „veszteségek”, meetingek, közös tervezések kompenzálására minden csapattag **+20** pontot kap.

Osztályzás:

0-50p: elégtelen

50-62p: elégséges

63-73p: közepes

74-85p: jó

86p felett: jeles

A féléves feladatnak határidejei nincsenek, a vizsgaidőszakban kell bemutatni a vizsgaalkalmon. Csapat esetén kérem, hogy azonos vizsgát vegyetek fel!

Témaötletek – ha nem jönne az ihlet

Tantárgyi követelmény szerkesztő app, amelyen keresztül a tanárok az O365 címükkel belépve képesek új tantárgyi követelményt szerkeszteni (leírás, ütemezés, stb.) teljesen újonnan vagy egy előző féléves követelmény deep copyjából indulva.
OE fórum – topicok az egyes tárgyaknak, feltehető kérdések, kiosztható pontok
Projektmenedzser app – csapat szervezés, kiosztható taskok, valamilyen ütemezési probléma megoldása
Szenzoros statisztikai app, valamilyen szenzor/fake szenzor/adatforrás adatokat küld az api végpontnak folyamatosan, mvc felületen statisztikák és grafikonok rajzolása
Gáz fogyasztás tervező: előző évek naptári alapú napi hőmérséklet gyűjteményéből kiindulva egy fűtési-plan tervezése. Melyik mennyit fűtsünk, hogy elérjük év végén a lakossági átlagfogyasztást (kellene valamilyen hővesztési együttható).