+36 1 783 2663
info@zerotohero.hu
www.zerotohero.hu
1095 Budapest, Haller Gardens,
Soroksári út 32-34. E ép.

Szia!

Ebben a dokumentumban a zeroToHero Programozás verseny második fordulójának a feladatsorát olvashatod.

Ne feledd, a feladatok megoldására egy hét áll a rendelkezésedre.
Beküldési határidő: 2020.08.02. 23:59

A feladatokat a következő lépések segítségével adhatod be:

Beadás módja

1. Töltsd le ezt a Google Docs dokumentumot a gépedre
2. Töltsd ki a Sárgán aláhúzott helyeket!
3. Miután kitöltötted a dokumentumot, mentsd le .pdf formátumban
4. Tedd elérhetővé az alkalmazásod bárki számára heroku segítségével. Segítséget a heroku ismertetőben kaphatsz hozzá!
5. Küldd el az így elkészült **feladatlapot** és a megoldott **zip fájlba csomagolt fájlok**at, mint csatolmányt emailben, a következő paraméterekkel:
    a. Tárgy: **[2.forduló] <Teljes neved>** (pl. “[2.forduló] Kovács András”)
    b. Cím: **programozas@pcf.hu**

6. Ellenőrizd, hogy tényleg megfelelően elküldted-e a levelet és csatoltad-e a megoldott feladatok forráskódját is :)

Sok sikert kívánunk a versenyhez!

Az első kitöltendő kérdésként kérlek add meg az neved és az email címed, melyről a megoldásokat küldöd:

Név:
Szalai Katalin

E-mail cím amiről a megoldást küldöd:
szkatalin@outlook.com

Üdvözöllek a zeroToHero programozás versenyének második fordulójában!

Add meg az általad választott nyelvet amivel a gyakorlati feladatokat fogod megoldani:

javascript

A beküldött projektnek fordíthatónak és futtathatónak kell lennie!

Technikai segítség

Python ismertető:

http://www.zerotohero.hu/python

Javascript ismertető:

http://www.zerotohero.hu/javascript

Java ismertető:

http://www.zerotohero.hu/java

A feladatmegoldáshoz szintén szükséges heroku ismertető:

http://www.zerotohero.hu/heroku

Előkészített keretrendszerek:
www.zerotohero.hu/keretrendszerek

Megoldásod heroku linkje:
https://zero-to-hero-1.herokuapp.com/

+36 1 783 2663
info@zerotohero.hu
www.zerotohero.hu
1095 Budapest, Haller Gardens,
Soroksári út 32-34. E ép.

Feladat 1 - Elméleti kérdések

Mik azok a rekurzív algoritmusok: (2 pont)

a rekurzió egy problémamegoldási módszer, ahol a megoldás az azonos probléma kisebb példányainak megoldásaitól függ

Mit nevezünk unit tesztnek?: (1 pont)

A unit-teszt, vagy más néven egységteszt, a metódusokat teszteli. Adott paraméterekre ismerjük a metódus visszatérési értékét (vagy mellékhatását).

Összesen: 3 pont

Feladat 2 - Alkalmazás készítése

Lépésszám meghatározás

Geek Elek egy panelház 3. Emeletén lakik,lépcsőzés közben elgondolkozott vajon hányféleképpen érhet fel a 3. Emeletre ha egyszerre maximum 1-et, 2-t, vagy 3-at léphet. A feladatod létrehozni egy Rest GET endpointot ami query paraméterben kap egy szám paramétert (a lelépendő lépcső számot) és egy szám lista paramétert (a lehetséges lépésközöke), az endpoint pedig visszaadja a hozzá tartozó lehetséges felmenetelek számát.
Azt feltehetjük, hogy 1 et mindig tud lépni Geek Elek (Tehát a szám listában az 1 biztosan szerepel)
A feladatot egy már előkészített keretrendszerben kell megoldanod. A query paraméterek neve amikkel dolgoznod kell, numberOfStair, stepSizeList. Nem használhatsz új függőségeket sem. A rest endpoint a “/number-of-steps/getNumberOfSteps” url en kell legyen.

Példa: Lépcső szám: 3, Lépésközök: 1, 2, 3
Lehetséges feljutás:
1 + 1 + 1
2 + 1
1 + 2

+36 1 783 2663
info@zerotohero.hu
www.zerotohero.hu
1095 Budapest, Haller Gardens,
Soroksári út 32-34. E ép.

3

Összesen 4 féleképpen tud eljutni Geek Elek a 3. lépcsőfokra.

Összesen: 38 pont

Feladat 3 - Alkalmazás készítése

CRUD folytatása

Ebben a feladatban az előző héten elkészített CRUD feladatot fogod folytatni, azt kiegészítve kis logikával, input validációval és pár unit teszttel. Abban az esetben, ha előző héten nem sikerült ezt a feladatot befejezned mellékelünk egy megoldást aminek segítségével kezdheted egyből az eheti feladatokat, bár jobban örülnénk ha a sajátodat folytatnád :).

A mellékelt megoldást ezen a linken találod: www.zerotohero.hu/keretrendszerek

Extra logika:

- Employee létrehozásnál mostantól szakács csak sütőt üzemeltethessen, pénztáros meg csak pénztárgépet.
- Egészítsd ki az employee sémát egy fizetés oszloppal (salary, int, not null)
- A napi minimálbér 300 kopejka, ennél kevesebbért tilos munkaerőt alkalmazni, ezért ennél kevesebb fizetésért nem lehet a rendszerbe alkalmazottat felvenni.
- A manager fizetése nem lehet alacsonyabb mint az azonos helyen dolgozó bármelyik alkalmazotté.
- A viszályok elkerülése végett nem előnyös túl nagy eltérést engedni a megegyező pozíciót ellátó emberek fizetése között, új szakács illetve pénztáros felvételekor ellenőrizni kell hogy a fizetése nem térhet ki jobban az eddigi ugyanazon poszton dolgozó emberek átlagfizetésénél 20%-al.

Input validációs feladatok:

- Megvizsgálni, hogy a művelethez szükséges összes mező érkezik-e a kérésben, megfelelő adatokkal.
- Abban az esetben ha érkezik egy szöveg típusú mező a requestben az ne legyen üres vagy csak láthatatlan karakterek.
- Speciális validációk:
    - Helyszín neve maximum 2 szóból álljon
    - Helyszín címe irányítószámmal kezdődjön (4 szám)
    - Dolgozók neve kettő vagy három szóból álljon

+36 1 783 2663
info@zerotohero.hu
www.zerotohero.hu
1095 Budapest, Haller Gardens,
Soroksári út 32-34. E ép.

Unit tesztek:

- A unit tesztekkel azt vizsgáld, hogy az employee létrehozásnál a feltételek megfelelően működnek-e. Az adatbázis hívásokat a megfelelő mock funkcionalitással egyszerűen felülírhatod.
    - Írj egy tesztet a felszerelés/alkalmazottak vizsgálatára 1 helyen. (több felszerelés mint alkalmazott / ugyanannyi felszerelés mint alkalmazott)
    - Írj tesztet arra az esetre ha nem ugyanazon helyen lévő felszereléshez akarnak embert hozzárendelni.
    - Írj tesztet ami vizsgálja új emberek felvételénél a fizetésüket a minimálbérhez képest.
    - Írj tesztet új manager felvételére ami megnézi hogy többet keres-e az ugyanazon helyen dolgozó többi alkalmazottnál.
    - Írj tesztet arra az esetre amikor az alkalmazottak átlagfizetésétől több mint 20% eltérést mutató fizetéssel rendelkező embert vesznek fel.

Websocket:

- Készíts egy endpointot, melynek meghívásával elindul egy rendelés folyamat. Ennek menete, hogy request-ben érkezik egy rendelés (string). Ezt a pincér átadja a szakácsnak, aki kis idő múlva válaszol hogy elkészült a rendelés, ezután a pincér válaszol a feladónak hogy átvehető a rendelés.
- Ehhez a folyamathoz nem kell módosítani az adatbázis sémán, két websocket közötti kommunikációt kell megvalósítani.
- Ehhez a feladathoz tetszőleges websocket könyvtárat használhatsz. (pythonosoknak, vegyétek fel a lib pontos nevét a requirements.txt-be, az alapján gyűjti be a heroku remote-on a csomagokat)

Összesen: 60 pont

Feladat 1:	
Feladat 2:	
Feladat 3:	
Feladat 4:	
Összesen:	