

Python programozás – 2. Zh (Flask)

Feladat: egy egyszerű Moodle rendszer prototípusának fejlesztése, amely lehetővé teszi a hallgatók kurzusokra való jelentkezését. A fejlesztés során a következő feladatokat kell megvalósítani:

1) Kurzusok listázása:

- a) A megjelenő weboldalon a „Courses” menüre kattintva táblázatos formában kell megjelennie a kurzusoknak a következő képen látható formában. A „/courses” hívást kezelő függvényt nem kell módosítani, csak a *courses.html*-be kell a megfelelő feldolgozást megírni Jinja2 template használatával. (4 pont)

Moodle Courses: [Home Subscription](#)

Code	Course	Students
VEMISAB254ZF	Python programozas	John Robert Mary Kevin
VEMISAB146AP	Programozas alapjai	William Thomas
VEMISAB156GF	Programozas I.	Bob

2) Felíratkozás űrlap létrehozása:

- a) Hozzunk létre egy *SubscriptionForm* osztály a *"forms/subscriptionForm.py"* fájlban, amely a következő mezőket kezeli:

- **coursecode** : string, kitöltése kötelező
 - **neptun**: string , kitöltése legyen kötelező, a megadott szöveg hossza pontosan 6 karakter
 - **name**: string, kitöltése közelező
 - **iamhuman**: boolean, kitöltése kötelező
- (4 pont)

Moodle Courses: [Home Courses](#)

Subscription

Course Code

Neptun

Name

☐ I'm human!

Subscription

- b) A `"templates/subscription.html"`-en belül valósítsuk meg a form megjelenítéséhez szükséges Jinja2 templatot. (2 pont)
- c) A template jelenítse meg a hibaüzeneteket! (2 pont)
- d) Egészítsük ki a `„/subscription”` kérést fogadó függvényt úgy, hogy képes legyen megjeleníteni a `SubscriptionForm`-ot (2 pont)
- e) Egészítsük ki a `„/subscription”` kérést fogadó függvényt úgy, hogy kezelje a form által elküldött adatokat, amelyeket validál.
- Ha az adatok nem megfelelőek, vagy nincs a megadott kódnak megfelelő kurzus, akkor jelenítse meg újra a formot.
 - Ha az adatok megfelelőek, akkor a hallgatót adja hozzá a megadott kurzushoz (`test_courses` lista), majd irányítson át az index oldalra a következő flash üzenettel: `„You have successfully subscribed!”` (4 pont)

3) Adatbázis létrehozása a kurzusok és hallgatók tárolására:

- a) A `„models.py”` fájlban készítsünk két modell osztályt:
- Course osztály a következő adattagokkal: (4 pont)
 - i. id: integer, primary key
 - ii. name: string, unique
 - iii. code: string, unique
 - iv. students: hivatkozás a kurzushoz tartozó hallgatókra (Student osztály)
 - Order osztály a következő adattagokkal: (4 pont)
 - i. id: integer, primary key
 - ii. name: string
 - iii. neptun: string, unique
 - iv. course_id: integer, idegen kulcs (course.id)
 - v. course: hivatkozás a hallgatóhoz tartozó kurzusra (Course osztály)
- b) Az adatbázis létrehozásához hozzunk létre migrációs scripteket, amelyhez a következő parancsokat kell futtatni (projekten jobb-klikk -> Open Command Prompt Here...)
- `set FLASK_APP=run.py`
 - `flask db init`
 - `flask db migrate -m „_init”`
 - `flask db upgrade`

Eredményként létrejön a WebApp mappába a `mydb.db`, eredmény ellenőrizhető a SQLiteStudio segítségével (1 pont)

- c) A Course táblát töltsük fel úgy, hogy futtassuk a projektben lévő dbmanger.py scriptet. (set as startup file) (1 pont)
- d) Módosítsuk az 1) feladatban megvalósított „/courses” végpontot úgy, hogy az adatbázisból kérje le a termékeket (2 pont)
- e) Módosítsuk a 2) feladatban megvalósított „/subscription” végpontot úgy, hogy adatbázisban ellenőrizze, hogy a formban átadott kurzus kód megfelelő-e. Ha nincs a kódnak megfelelő kurzus az adatbázisba, akkor irányítson át a „/courses” oldalra a következő felíráttal: “{} not found! Check the list!”, ahol a paraméter a hibásan megadott kurzuskód legyen. (4 pont)
- f) Megfelelő adatok esetén mentjük a hallgatót az adatbázisba a kódnak megfelelő kurzushoz. Az eredmény ellenőrizzük a /courses oldalon. (2 pont)

4) API fejlesztés

- a) észítsünk egy “/students” API végpontot, amely segítségével lekérhetjük az egy kurzushoz tartozó hallgatók listáját. A végpont path változóként kapja meg kurzus kódját (a megvalósítás során használható a test_courses lista is. Az eredmény konvertálja JSON formátumba és adja vissza a kliens számára (2 pont)
- b) Módosítsuk a /courses oldal úgy, hogy a táblázathoz adjunk még egy oszlopot, amely olyan linkeket tartalmaz, amelyre rákattintva az API hívódik meg a táblázat sorának megfelelő kurzus azonosítójával. (2 pont)

Moodle Courses: [Home](#) [Subscription](#)

code	Course	Members	Link
VEMISAB254ZF	Python programozas	John	List students
		Robert	
		Mary	
		John	
VEMISAB146AP	Programozas alapjai	Kevin	List students
		William	
		Thomas	
VEMISAB156GF	Programozas I	Bob	List students
		Günter	

```

1 [
2   {
3     "name": "John",
4     "neptun": "JJJJJJ"
5   },
6   {
7     "name": "Robert",
8     "neptun": "RRRRRR"
9   },
10  {
11    "name": "Mary",
12    "neptun": "MMMMMM"
13  }
14 ]

```