Handout

project elkezdéséhez

Készítette: Nemes Tamás

KEZDÉS

```
python -m venv my_venv
```
Virtuális környezet létrehozása, hogy a projektünkket elszeparáltan dolgozhassunk

```
cd my_venv
Scripts\activate
```
Belépés a környezet mappájába, majd a környezet aktiválása (a sor elején zárójelben megjelenő my_venv jelzi a sikert)

```
pip install django
```
A Django keretrendszer legújabb stabil verziójának telepítése

```
django-admin startproject my_project
```
Projekt elkezdése (alap fájlok létrehozása)

```
rename my_project backend
cd backend
```
A külső mappa átnevezése backendre (opcionális, de így áttekinthetőbb lesz később)

```
python manage.py migrate
```
Adatbázis létrehozása (SQLite)

```
python manage.py createsuperuser
```
Admin jogosultságú felhasználó létrehozása

```
code .
```
Visual Studio Code elindítása az aktuális mappából

```
python manage.py runserver
```
Fejlesztői szerver elindítása

APP LÉTREHOZÁSA

```
python manage.py startapp my_app
```
Alkalmazás indítása (my_app mappa létrejön)

A my_project mappában hozzá KELL adni az INSTALLED_APPS listához a my_app-ot.

MODELL* LÉTREHOZÁSA

*Ez az ORM (Object Relational Mapping) eszköze, python class-ból generál adatbázist.

A my_app mappában a models.py fájlban hozhatunk létre modelleket.
A modellek a models.Model-től örökölnek, és az osztályváltozóikból generál a Django adatbázist. Az osztályváltozók szintén a models-ből jönnek. pl.:

```
class Dog(models.Model):
    name = models.CharField(max_length = 255)
    age = models.IntegerField()
```

MIGRATE

A létrehozott modell tükörképét az adatbázisban is el kell készítünk.

```
python manage.py makemigrations
```
Összehasonlítja az adatbázist és a modelljeinket, és logolja a kettő közti különbségeket.

```
python manage.py migrate
```
Szinkronba hozza a kettőt (módosítja az adatbázist a modellek alapján)

ADMIN

Ahhoz, hogy az admin felületen (http://*localhost:8000/admin/*) bejelentkezés után láthassuk a modelljeinket és módosíthassuk őket, “fel kell pakolnunk” őket az admin felületre.

Ehhez a my_app/admin.py fájlhoz kell hozzáírnunk a következőket:

```
from .models import Dog
admin.site.register(Dog)
```

PATH

A view létrehozása után módosítsuk az elfogadott URL-ek listáját.

```
from my_app import views

urlpatterns = [
... # eddigi kódot itt hagyjuk, a lista végére írunk
path(“pelda/”, views.hello),
]
```

(ezt a *my_project/urls.py* fájlba írjuk)

Ezután ha futtatjuk a szervert akkor a localhost:8000/pelda/ címen vissza kell kapnunk hogy hello.

VIEW

A Django-ban URL végződésekhez rendelünk függvényeket, amelyek az adott URL lekérésekor lefutnak. Ezeket a függvényeket view-oknak nevezzük, és minden esetben kapnak legalább egy paramétert (request) és visszatérnek valamivel (response).

A view-okat a views.py fájlba írjuk.

Példa view-ra:

```
def hello(request):
    return JsonResponse({“message”:”hello”})
```

(ezt a *my_app/views.py*-ba kell írni az előző elnevezések szerint)