

Harjoitustyö vaihe 1: Kehitysympäristön perustaminen

Kuvaan tässä harjoitustyön osan 1 raportissa, miten asensin kehitysympäristön ja testaan kehitysympäristöä ja tietokantaa. Minulla on jonkun verran ohjelmointikokemusta, mutta en ole kehittänyt web-sovelluksia aikaisemmin, joten päädyin suosiolla valitsemaan kurssilla suositellut teknologiat. En ole vielä valinnut lopullista harjoitustyöni aihetta, mutta minua kiinnostaa toteuttaa jonkunlainen koostepalvelu.

Ympäristö ja teknologiat

- Kehitysympäristö: Django 3.0.2
- Ohjelmointikieli: Python 3.8.1
- Editorina Visual Studio Code
- Versionhallinta: GitHub
- Tietokanta: SQLite
- Käyttöjärjestelmä: macOS Catalina 10.15.3

Visual Studio Coden asentaminen

Ennen kuin aloitin tekemään harjoitustyötä, asensin ensin Visual Studio Coden. Valitsin sen editoriksi muiden suosituksesta. Editorin asentaminen ei sujunut ongelmitta, sillä macOS Catalinalle on tyypillistä, että se ei pysty varmentamaan, onko tiedosto luotettavasta lähteestä. Järjestelmä ei suostunut avaamaan tiedostoa. Luin Visual Studio Coden dokumentaatiota ja siellä kerrottiin, että VS Codea ei ole todennettu, mutta se toimii silti hyvin macOS Catalinalla.

Dokumentaatiossa oli selkeät ohjeet, miten ongelman voi kiertää. *Järjestelmäasetukset* → *Suojaus ja yksityisyys* → *Yleiset* ja alhaalta avaa silti. Tämän jälkeen VS Code avautui normaalisti.

Python 3.8.1 asentaminen

Olin asentanut pythonin aikaisemmin koneelleni 2 vuotta sitten Johdatus ohjelmointiin -kurssia varten. Sen versio oli kuitenkin vanhentunut, joten päivitin pythonin uusimpaan versioon. Poistin vanhan version koneeltani. Olin asentanut aikaisemmin pythonin python.org-sivun kautta, joten asensin uusimman 3.8.1. version myös sitä kautta. Pythonin päivittäminen onnistuisi myös homebrew-paketinhallintaohjelmalla, mutta koska pythonia ei asennettu sitä kautta alun perin, päätin asentaa sen samalla tavalla kuin aikaisemmin.

Virtuaaliympäristön luominen

Noudatin virtuaaliympäristön luomisessa koodiklinikan ohjeita. Ne osoittautuivat tehokkaiksi eikä vastaani tullut suuria haasteita. Luennon ohjeiden mukaisesti asensin aluksi homebrew'n terminaalissa komennolla

```
/usr/bin/ruby -e "$(curl -fsSL  
https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

Tarkastin, oliko Pythonin asennus onnistunut komennolla

```
python3 -V
```

ja huomasin, että asennus oli onnistunut ja järjestelmä näyttää pythonin olevan versio 3.8.1. Seuraavaksi asensin virtuaaliympäristön komennolla

```
pip3 install virtualenv
```

Navigoin cd-komennolla kansioon, jonne virtuaaliympäristö luodaan. Loin virtuaaliympäristön.

```
virtualenv venv
```

Navigoin cd-komennolla venv-kansioon ja aktivoin virtuaaliympäristön komennolla

```
source bin/activate
```

ja seuraavaksi asensin Django komennolla.

```
pip3 install django
```

Kun Django oli asennettu, loin projektia varten oman kansion

```
django-admin startproject mysite
```

Kokeillaan, toimiiko asennus niin kuin pitää.

```
python3 manage.py migrate
```

```
python3 manage.py runserver
```

Tämän jälkeen syötetään saatu `http://127.0.0.1:8000/` selaimen. Django oli asennettu onnistuneesti.

Tietokannan testaaminen ja testisovelluksen luominen

Olisin halunnut opetella käyttämään harjoitustyössä PostgreSQL:aa, koska koen sen opetteluun tärkeäksi tulevaisuutta varten. Kohtasin sen asentamisessa kuitenkin ongelmia, joten päädyin lopulta oletustietokanta SQLiteen.

Noudatin testisovelluksen luomisessa ja tietokannan testaamisessa Django:n dokumentaation ohjeita. Avasin mysite/settings.py VS codessa.

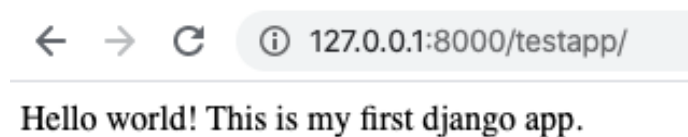
Tein testisovelluksen komennolla

```
python3 manage.py startapp testapp
```

Muokkasin tiedostoja views.py, urls.py (mysite) ja urls.py (testapp) Django:n dokumentaation mukaisesti. Testasin välissä, että toimiiko applikaatio.

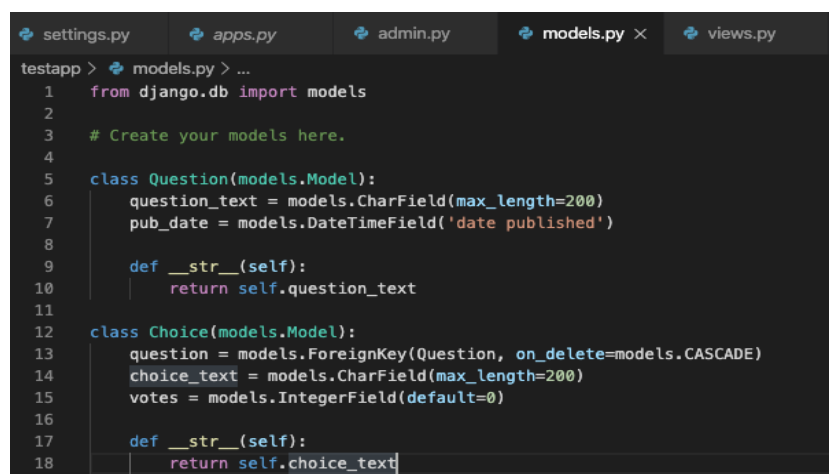
```
python3 manage.py runserver
```

Testasin ympäristön toimintaa selaimessa. Syötin saadun http://127.0.0.1:8000/testapp/ selaimeseen ja kuva 1 näyttää, että ympäristö toimii.



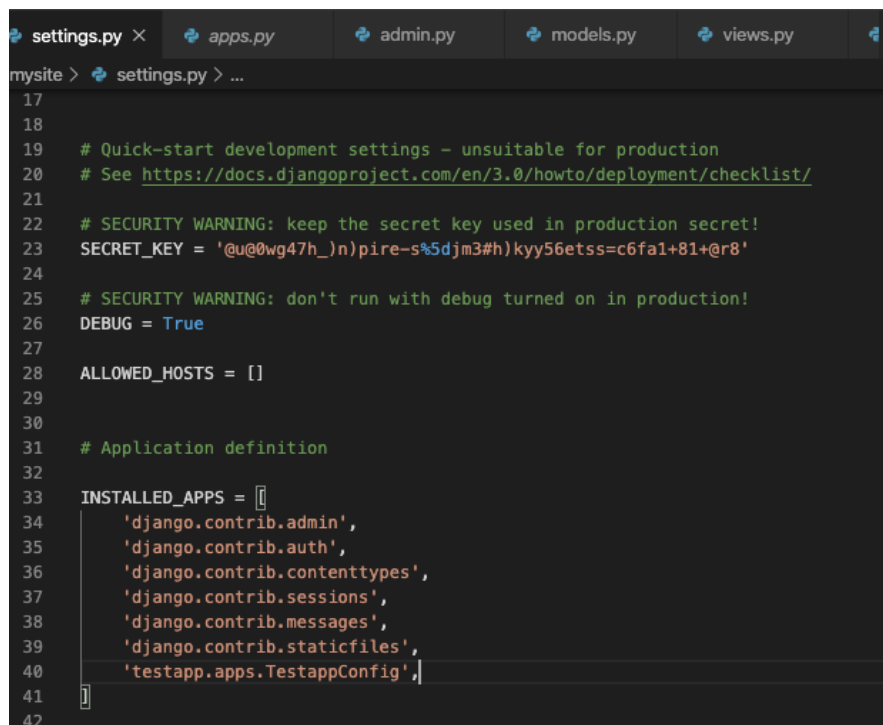
Kuva 1. Testappin testaus.

Muokkasin models.py-tiedostoa Django:n dokumentaation mukaisesti (kuva 2).



Kuva 2. Models-tiedoston muokkaukset.

Seuraavaksi tein tarvittavat muutokset setting.py tiedostoon (kuva 3).



```
settings.py × apps.py admin.py models.py views.py
mysite > settings.py > ...
17
18
19 # Quick-start development settings - unsuitable for production
20 # See https://docs.djangoproject.com/en/3.0/howto/deployment/checklist/
21
22 # SECURITY WARNING: keep the secret key used in production secret!
23 SECRET_KEY = '@u@0wg47h_)n)pire-s%5djm3#h)kyy56etss=c6fa1+81+@r8'
24
25 # SECURITY WARNING: don't run with debug turned on in production!
26 DEBUG = True
27
28 ALLOWED_HOSTS = []
29
30
31 # Application definition
32
33 INSTALLED_APPS = [
34     'django.contrib.admin',
35     'django.contrib.auth',
36     'django.contrib.contenttypes',
37     'django.contrib.sessions',
38     'django.contrib.messages',
39     'django.contrib.staticfiles',
40     'testapp.apps.TestappConfig',
41 ]
42
```

Kuva 3. Settings-tiedoston muokkaukset.

Tämän jälkeen otin kaikki muutokset käyttöön komennolla.

```
python3 manage.py makemigrations testapp
```

Seuraavaksi testasin tietokantaa terminaalissa (kuva 4).

```
[(venv) ~anutamminen@Anu-MacBook-Air mysite % python3 manage.py shell
Python 3.8.1 (v3.8.1:1b293b6006, Dec 18 2019, 14:08:53)
[Clang 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
(InteractiveConsole)
[>>> from testapp.models import Choice, Question
[>>> from django.utils import timezone
[>>> Question.objects.all()
<QuerySet []>
[>>> q = Question(question_text="Miten menee?", pub_date=timezone.now())
[>>> q.save()
[>>> q.question_text
'Miten menee?'
[>>> q.id
1
[>>> q.pub_date
datetime.datetime(2020, 2, 16, 10, 59, 13, 745605, tzinfo=<UTC>)
[>>> q.question_text = "Hyvin!"
[>>> q.save()
[>>> Question.objects.all()
<QuerySet [<Question: Hyvin!>]>
[>>> q.id
1
[>>> ]
```

Kuva 4. Tietokannan testaus API:a hyödyntäen.

Testasin aikaisemmin ympäristöä myös selaimessa, joten näiden testausten perusteella kaikki toimii niin kuin pitääkin.

Helpot/vaikeat asiat

Helppoa

- Django:n asentaminen oli helppoa koodiklinikan ohjeilla ja Django:n dokumentaation seuraaminen auttoi paljon.
- Minulla on ennestään kokemusta komentorivitulkkin/terminaalin käytöstä, joten sen käyttö oli helppoa.

Haastavaa

- En meinannut saada VS codea toimimaan aluksi raportissa kuvatun macOS Catalina -ongelman takia. Ongelmaa ei ollut varsinaisesti vaikea ratkaista, mutta siihen meni paljon aikaa.
- Olin alun perin suunnitellut käyttäväni tietokantana PostgreSQL:aa, mutta kohtasin lukuisia haasteita asennuksen aikana, mikä oli turhauttavaa, ja päädyin lopulta SQLiteen.
- Yritin tehdä testisovelluksen aluksi koodiklinikan ohjeiden mukaan, mikä ei onnistunut ja tähän meni paljon aikaa. Django:n dokumentaatiota oli kuitenkin helppo seurata ja sain testattua tietokantaa ja sovellusta onnistuneesti.
- Näin jälleenkäin harjoitustyön dokumentointi olisi ollut miellyttävämpää, jos olisin tajunnut ottaa kuvia heti alussa, kun asensin Django:n ja loin virtuaaliympäristön.

Hyviä lähteitä

Django:n dokumentaatio (testisovellus ja tietokannan testaaminen)

<https://docs.djangoproject.com/en/3.0/intro/tutorial01/>

<https://docs.djangoproject.com/en/3.0/intro/tutorial02/>

Koodiklinikan opetusvideo Django:n asentamiseen

<https://echo360.org.uk/lesson/209132bb-7ea0-4bc8-8280-130623fee143/classroom#sortDirection=desc>