

OHSIHA 2020

Harjoitustyön loppuraportti



29. HUHTIKUUTA 2020

RIKU SUNDQVIST

SISÄLLYS

Vaihe 1: Kehitysympäristön perustaminen	2
Johdanto	2
Käyttämäni teknologia ja ympäristö.....	2
Pythonin ja Django:n asennus sekä kehitysympäristön perustaminen.....	2
Tietokannan käyttöönotto.....	5
Helppoudet & vaikeudet	6
Vaihe 2: Sisäänkirjautuminen, rekisteröityminen ja CRUD -toiminnot	8
Sisään- ja uloskirjautuminen	8
Rekisteröityminen	12
CRUD-toiminnot	14
Helppoudet ja vaikeudet	17
Vaihe 3: Datarajapinnan käyttö.....	18
Helppoudet ja vaikeudet	21
Lisäpiirteet: Palautus ja lisenssi Githubissa	22
Työn palautus Githubiin	22
Lisenssin lisääminen	23
Lähteet.....	24

VAIHE 1: KEHITYSYMPÄRISTÖN PERUSTAMINEN

JOHDANTO

En ole aiemmin paljoa koodaillut tai tuottanut ohjelmia. Pyrin luomaan avointa dataa rajapintojen avulla hyödyntävän koostepalvelun, koska tällaiseen tarkoitukseen löytyy paljon valmiita rajapintoja. Netistä myös löytynee Googlaamalla vinkkejä rajapintojen käyttöön Djangolla ja Visual Studiolla.

KÄYTTÄMÄNI TEKNOLOGIA JA YMPÄRISTÖ

Kehitysympäristö: Django

Ohjelmointikieli: Python 3.8.1

Editori: MS Visual Studio Code

Tietokanta: SQLite

Käyttöjärjestelmä: Windows 10

Valitsin Djangon, koska sitä suositeltiin aloittelijoille. Ohjelmointikielenä käytän Pythonia, koska olen sitä käyttänyt aiemmin toisella kurssilla ja sitä käytetään myös tämän kurssin koodiklinikoilla. Jälkimmäisestä syystä valitsin myös Visual Studion editoriksi. SQLite on oletuksena Djangossa tietokannaksi, enkä näe syytä vaihtaa sitä.

PYTHONIN JA DJANGON ASENNUS SEKÄ KEHITYSYMPÄRISTÖN PERUSTAMINEN

Aluksi latsin Pythonin koodiklinikan ohjeiden mukaisesti. Tämä onnistui, koska tietokoneeltani löytyi asennuksen jälkeen Pythonin uusin versio ja Powershell tunnisti kielen ja version kuvan 1 mukaisesti.

```
PS C:\WINDOWS\system32> python
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

1.1: PYTHONIN TUNNISTAMINEN POWERSHELLISSÄ

Seuraavaksi loin Powershellillä harjoitustyölle virtuaaliympäristön haluamaani tiedostosijaintiin. Aluksi sallittiin tietokoneelle scriptien suorittaminen antamalla kuvan 2 mukainen komento ja antamalla lupa ("Y") Powershellin varmistaessa olenko varma.

```
PS C:\WINDOWS\system32> Set-ExecutionPolicy RemoteSigned

Execution Policy Change
The execution policy helps protect you from scripts that you do not trust. Changing
the execution policy might expose you to the security risks described in the
about_Execution_Policies help topic at https://go.microsoft.com/fwlink/?LinkID=135170.
Do you want to change the execution policy?
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help
(default is "N"):Y
PS C:\WINDOWS\system32>
```

1.2: SCRIPTIEN SALLIMINEN

Pythonin toimivuus testattiin jo alussa, mutta testataan vielä pipin toimivuus *pip freeze* -komennolla.

```
PS C:\Users\Riku Sundqvist> python -V
Python 3.8.1
PS C:\Users\Riku Sundqvist> pip freeze
appdirs==1.4.3
distlib==0.3.0
filelock==3.0.12
six==1.14.0
virtualenv==20.0.1
PS C:\Users\Riku Sundqvist>
```

1.3: PIP TOIMII

Seuraavaksi asennettiin virtuaaliympäristö komennolla:

pip install virtual env

Seuraavaksi loin harkkatyötä varten kansion haluamaani tiedostosijaintiin. Valitsin koodiklinikan esimerkin mukaisesti kansiolle sijainnin C:\Users\Riku Sundqvist\documents. Powershellissä kansioden välillä navigoitiin komennoilla *cd [kansion nimi]* haluamaansa kansioon ja *cd ..* vei takaisin päin. Halutessaan saatavilla olevat kansiot ja tiedostot sai näkyviin komennolla *cd ls*, joka tulosti listan sen hetkisen sijainnin alavalikosta.

Tässä kohtaa merkittäviä vaikeuksia seurasi siitä, että nimelläni oleva kansio (Riku Sundqvist) käyttäjissä sisältää välilyönnin, jota ei käsitelty koodiklinikalla. Komento `cd Riku Sundqvist` tuotti siis virheviestin, koska *Riku* nimistä kansiota ei ole olemassa. Tätä pähkäilin hyvin kauan, kunnes Googlesta lopulta selvisi, että minun täytyy käyttää heittomerkkejä välilyönnin sisältävää kansiota käsitellessä. Oikea komento oli siis `cd "Riku Sundqvist"`. Seuraavaksi loin "OhsihaHarkka" -nimisen kansion komennolla `mkdir OhsihaHarkka`.

```
PS C:\> cd Users
PS C:\Users> cd "Riku Sundqvist"
PS C:\Users\Riku Sundqvist> cd documents
PS C:\Users\Riku Sundqvist\documents> ls

Directory: C:\Users\Riku Sundqvist\documents

Mode                LastWriteTime         Length Name
----                -
d-----         10.2.2020    20.15              OhsihaHarkka
```

1.4: HARKKATYÖKANSIONI SIJAINTI JA SINNE NAVIGOINTI. ERITYISHUOMIO HEITTOMERKKIEN KÄYTÖLLE VÄLILYÖNNIN SISÄLTÄVÄÄ KANSION NIMEÄ KÄSITELTÄESSÄ.

Viimeinen komento virtuaaliympäristön luomiseksi oli `virtualenv env`, jossa `env` oli ympäristölle antamani nimi. Virtuaaliympäristö käynnistettiin komennolla

`.\Scripts\activate`

Onnistuneen käyttöönoton merkiksi Powershellin komentoriviin ilmestyi `\env` -pääte.

```
PS C:\Users\Riku Sundqvist\documents\OhsihaHarkka> cd env
PS C:\Users\Riku Sundqvist\documents\OhsihaHarkka\env> .\Scripts\activate
(env) PS C:\Users\Riku Sundqvist\documents\OhsihaHarkka\env>
```

1.5: VIRTUAALIYMPÄRISTÖN KÄYNNISTYS

Seuraavaksi asennettiin Django komennolla `pip install django`. Lisäksi Visual Studion käyttäjänä asensin PyLintin, joka on Python -koodauksessa apuväline virheiden tarkastamiseen. Asennus tapahtui komennolla `install pylint`.

Seuraavaksi loin Harkkatyö -nimisen Djangoprojetkin komennolla `django-admin startproject Harkkatyö`. Ennen tätä menin kansiossa yhden askeleen taaksepäin komennolla `cd ..`.

Seuraavaksi asensin Python lisäosan Visual Studioon. Tämän jälkeen avasin projektin Visual Studioissa komennolla

`code`. (Huom! välilyönti ennen pistettä)

Ennen testausta valittiin vielä oikea tulkki:

ctrl+shift+p -> Python select interpreter

Testatakseni ympäristön toimivuuden navigoin takaisin projektiini ja ajoin Powershellissä komennon *py manage.py runserver*. Seuraavaksi syötettiin saatu URI (korostettuna kuvassa 6) selaimeen ja saatiin kuvassa 7 näkyvä ilmoitus ympäristön virheettömästä toiminnasta.

```
(env) PS C:\Users\Riku Sundqvist\documents\OhsihaHarkka\env> cd ..
(env) PS C:\Users\Riku Sundqvist\documents\OhsihaHarkka> cd Harkkatyö
(env) PS C:\Users\Riku Sundqvist\documents\OhsihaHarkka\Harkkatyö> py manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

You have 17 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin,
auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
February 20, 2020 - 20:54:11
Django version 3.0.3, using settings 'Harkkatyö.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

1.6: KOMENNOT YMPÄRISTÖN TESTAAMISEKSI



The install worked successfully! Congratulations!

You are seeing this page because `DEBUG=True` is in your settings file and you have not configured any URLs.

1.7: SELAIMEEN SYÖTETYLÄ URILLA NÄKYVÄ VIESTI TOIMIVASTA VIRTUAALIYMPÄRISTÖSTÄ

Seuraavaksi luotiin django applikaatio komennolla *py manage.py startapp applikaatio* ja tehtiin koodiklinikan mukaiset muutokset tarvittaviin kansioihin.

TIETOKANNAN KÄYTTÖÖNOTTO

Otettiin SQLite oletustietokannaksi käyttöön komennolla *python manage.py migrate* kuvan 8 mukaisesti.

```
(env) PS C:\Users\Riku Sundqvist\documents\ohsihaharkka\harkkatyö> python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  No migrations to apply.
```

1.8: ASETETAAN OLETUSTIETOKANNAKSI SQLITE

Seuraavaksi tehtiin tarvittavat muutokset applikaation *models.py* tiedostoon, joka käsittelee tietokantoihin liittyvät seikat:

```
applikaatio > models.py > ...
1  from django.db import models
2
3  class questions(models.Model):
4      question_text = models.CharField(max_length=200)
5      pub_date = models.DateTimeField('date published')
6
7  class Choice(models.Model):
8      question = models.ForeignKey(questions, on_delete=models.CASCADE)
9      choice_text = models.CharField(max_length=200)
10     votes = models.IntegerField(default=0)
11
```

1.9: TARVITTAVAT MUUTOKSET MODELS.PY TIEDOSTOON

Muutokset otettiin käyttöön Powershellillä komennolla *py manage.py makemigrations applikaatio*. Viimeisenä vielä testattiin toimivuus komennolla *python manage.py shell*. Ainoa virhe oli kuvan 1.10 mukainen `NameError`. Tämä johtui siitä, että luokan nimi oli `questions` eikä `question`.

```
NameError: name 'Question' is not defined
```

KUVA 1.10: VIRHE TIETOKANTAA TESTATESSA

HELPPOUDET & VAIKEUDET

Helppoa:

- Ohjeistus koodiklinikalla selkeää, mikä helpotti mukana pysymistä ympäristössä luodessa
- Powershellin komennot jäivät nopeasti mieleen
- Visual Studion käyttö helppoa selitteineen ja automaattisine täyttöehdotuksineen komentoja kirjoittaessa.

Vaikeaa:

- Pitää olla huolellinen: jokainen kirjoitusvirhe on kriittinen lopputuloksen kannalta Powershelliä ja Visual Studioa käyttäessä.
- Pitää tarkastaa, että tekee oikeat komennot oikeassa kansiossa.
- Jos putoaa kärryiltä koodiklinikkaa seurattessa, joutuu kelaamaan paljon taakse, että pääsee taas kiinni

VAIHE 2: SISÄÄNKIRJAUTUMINEN, REKISTERÖITYMINEN JA CRUD -TOIMINNOT

SISÄÄN- JA ULOSKIRJAUTUMINEN

Käytän Django'n omaa authentication -sovellusta, jonka käyttöönottoon löytyi learndjango.com -sivulta opas rekisteröitymisen ja sisäänkirjautumisen luomiseksi. Setting.py tiedostosta löytyvä Authentication -sovellus otettiin käyttöön lisäämällä urls.py tiedostoon rivit *from django.urls import path, include* ja *path('accounts/', include('django.contrib.auth.urls'))*, :

```
16 from django.contrib import admin
17 from django.urls import path, include
18
19 urlpatterns = [
20     path('admin/', admin.site.urls),
21     path('', include('applikaatio.urls')),
22     path('', include("django.contrib.auth.urls")),
23 ]
```

2.1: URLS.PY

Seuraavaksi loin oman html tiedoston sisäänkirjautumista varten. Loin tutoriaalin mukaisesti Registration -kansion Template -tiedostojen alle ja sinne "login.html" -tiedoston. Tiedostoon tein html-komennoilla otsikon ja napin kirjautumiselle seuraavilla komennoilla:

```
templates > registration > <> login.html > form
1 <h2>Login</h2>
2 <form method="post">
3     {% csrf_token %}
4     {{ form.as_p }}
5     <button type="submit">Login</button>
6 </form>
```

2.2: LOGIN.HTML

Muutoksia tarvitsi tehdä myös settings.py tiedostoon. Aluksi lisäsin *os.path.join(BASE_DIR, 'templates')* -tekstin DIRS -asetuksiin, jotta Django huomioi projektissa Templates -kansion:

```

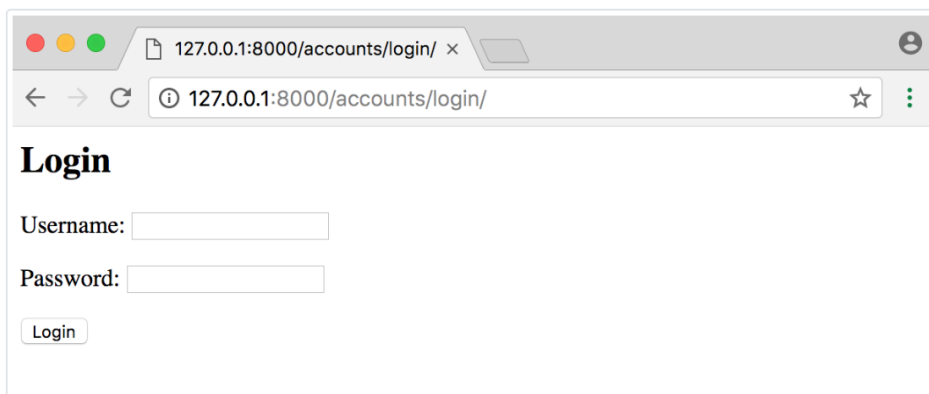
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR, 'templates')],
        'APP_DIRS': True,
        'OPTIONS': {

```

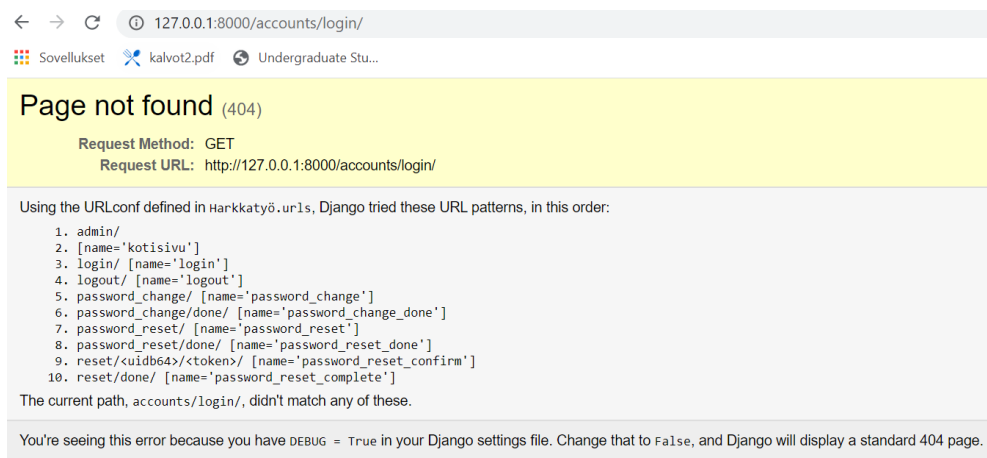
2.3: SETTINGS.PY

Lisäksi tiedoston loppuun lisättiin vielä komento `LOGIN_REDIRECT_URL = '/'`, jotta sisäänkirjautuminen ohjaa käyttäjän aloitussivulle.

Seuraavaksi Powershellissä käynnistettiin sovellus komennolla `python manage.py runserver`, jonka jälkeen kopiottiin saatu osoite selaimeen ja lisättiin osoitteen perään `/accounts/login/`. Tässä vaiheessa minulla ilmeni ongelma, koska sivun kuului näyttää tältä:



Ja minulla se näytti tältä:



Ensimmäinen käyttäjä palveluun tehtäisiin Powershellissä komennolla `python manage.py createsuperuser`. Seuraavaksi syötettiin tilin tiedot. Turvallisuussyistä, salasana ei näy edes peitettynä:

```
(env) PS C:\Users\Riku Sundqvist\documents\ohsihaharkka\harkkayö> python manage.py createsuperuser
Username (leave blank to use 'rikusundqvist'):
Email address: riku.sundqvist@tuni.fi
Password:
Password (again):
Superuser created successfully.
```

1.4: ENSIMMÄISEN KÄYTTÄJÄN LUOMINEN

Tässä vaiheessa tutoriaalia kerrottiin, että aiempi ongelmani sisäänkirjautumissivulla johtui siitä, ettei kotisivuni ollut vielä valmis. Onneksi seuraavanlaisin ohjein sen tekeminen onnistui. Loin uudet html tiedostot *base.html* ja *home.html* . Niihin kirjoitin seuraavat koodinpätkät:

```
templates > <> base.html > ...
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="utf-8">
5      <title>{% block title %}Django Auth Tutorial{% endblock %}</title>
6  </head>
7  <body>
8      <main>
9          {% block content %}
10             {% endblock %}
11      </main>
12  </body>
13  </html>
```

2.5: BASE.HTML

```
templates > <> home.html > ...
1  {% extends 'base.html' %}
2
3  {% block title %}Home{% endblock %}
4
5  {% block content %}
6      {% if user.is_authenticated %}
7          Hi {{ user.username }}!
8      {% else %}
9          <p>You are not logged in</p>
10         <a href="{% url 'login' %}">login</a>
11     {% endif %}
12 {% endblock %}
```

2.5: HOME.HTML

Lisäksi muutin *login.html* tiedostoa lisäämällä sen alkuun seuraavat komennot:

```

templates > registration > <> login.html > ...
1  {% extends 'base.html' %}
2
3  {% block title %}Login{% endblock %}
4
5  {% block content %}
6  <h2>Login</h2>
7  <form method="post">
8      {% csrf_token %}
9      {{ form.as_p }}
10     <button type="submit">Login</button>
11 </form>
12 {% endblock %}

```

2.6: LOGIN.HTML

Jotta kotisivu saadaan näkyviin, lisätään *urls.py* tiedostoon komennot *from django.views.generic.base import TemplateView* ja *path("", TemplateView.as_view(template_name='home.html'), name='home'), :*

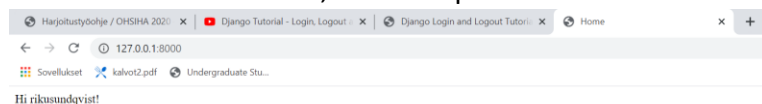
```

16 from django.contrib import admin
17 from django.urls import path, include
18 from django.views.generic.base import TemplateView
19
20 urlpatterns = [
21     path('admin/', admin.site.urls),
22     path('', include('applikaatio.urls')),
23     path('', include("django.contrib.auth.urls")),
24     path('', TemplateView.as_view(template_name='home.html'), name='home'),
25 ]

```

2.7: KOTISIVUNÄKYMÄN TOTEUTUS (URLS.PY)

Nyt pitäisi olla valmista kirjautumisen osalta. Ajoin Powershellissä komennon *python manage.py runserver* ja kopioin saadun URL:n. Se vei minut sivulle, joka ilmaisi, etten ollut kirjautunut sisään sekä ohjasi kirjautumissivulle. Syötin luomani tilin tunnuksen ja salasanan ja pääsin seuraavanlaiselle sivulle, niin kuin pitikin



2.8: ETUSIVU KIRJAUTUMISEN JÄLKEEN

Seuraavaksi tein uloskirjautumistoiminnon. Aluksi lisäsin "Logout" vaihtoehdon kotinäkymään muokkaamalla *home.html* tiedostoon seuraavan rivin if -ehdon loppuun: `<p>logout</p>`

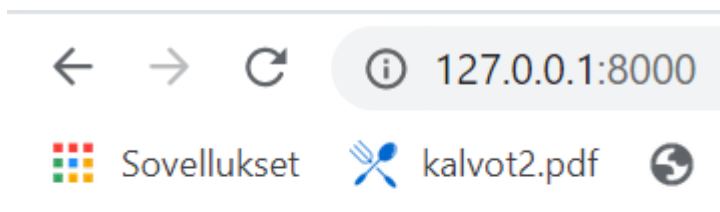
Myös *setting.py* tiedoston loppuun lisäsin `LOGOUT_REDIRECT_URL = '/'`

Muokkasin tiedostosta vielä sisään- ja uloskirjautumista koskevat komennot järkevämpään muotoon

```
123 LOGIN_REDIRECT_URL = 'home'
124 LOGOUT_REDIRECT_URL = 'home'
```

2.9: SISÄÄN- JA ULOSKIRJAUTUMISEN OHJAUS ETUSIVULLE

Nyt myös uloskirjautuminen oli valmis:



2.10: LINKKI ULOSKIRJAUTUMISELLE

REKISTERÖITYMINEN

Yritin luoda aluksi Powershellillä rekisteröitymistä varten uuden appin *accounts* , mutta se ei onnistunut ainakaan komennolla

```
py manage.py startapp accounts
```

Powershell antoi virhekoodin "ModuleNotFoundError: No module named 'accounts'"

Ilmeisesti virhe oli komennon etuliitteessä. Sen tulisi viitata työhöni, mutta en silti löytänyt oikeaa. Alla olevassa kuvassa kokeilujen jälkeä:

```
(env) PS C:\Users\Riku Sundqvist\documents\ohsihaharkka\harkkatyö> (applikaatio) python manage.py startapp accounts
At line:1 char:15
+ (applikaatio) python manage.py startapp accounts
+ ~~~~~
Unexpected token 'python' in expression or statement.
+ CategoryInfo          : ParserError: (:) [], ParentContainsErrorRecordException
+ FullyQualifiedErrorId : UnexpectedToken

(env) PS C:\Users\Riku Sundqvist\documents\ohsihaharkka\harkkatyö> Harkkatyö python manage.py startapp accounts
Harkkatyö : The term 'Harkkatyö' is not recognized as the name of a cmdlet, function, script file, or operable program.
Check the spelling of the name, or if a path was included, verify that the path is correct and try again.
At line:1 char:1
+ Harkkatyö python manage.py startapp accounts
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (Harkkatyö:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

(env) PS C:\Users\Riku Sundqvist\documents\ohsihaharkka\harkkatyö> applikaatio python manage.py startapp accounts
applikaatio : The term 'applikaatio' is not recognized as the name of a cmdlet, function, script file, or operable program.
Check the spelling of the name, or if a path was included, verify that the path is correct and try again.
At line:1 char:1
+ applikaatio python manage.py startapp accounts
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (applikaatio:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

(env) PS C:\Users\Riku Sundqvist\documents\ohsihaharkka\harkkatyö> (Harkkatyö) python manage.py startapp accounts
At line:1 char:13
+ (Harkkatyö) python manage.py startapp accounts
+ ~~~~~
Unexpected token 'python' in expression or statement.
+ CategoryInfo          : ParserError: (:) [], ParentContainsErrorRecordException
+ FullyQualifiedErrorId : UnexpectedToken

(env) PS C:\Users\Riku Sundqvist\documents\ohsihaharkka\harkkatyö> Ohsihaharkka python manage.py startapp accounts
Ohsihaharkka : The term 'Ohsihaharkka' is not recognized as the name of a cmdlet, function, script file, or operable program.
Check the spelling of the name, or if a path was included, verify that the path is correct and try again.
At line:1 char:1
+ Ohsihaharkka python manage.py startapp accounts
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (Ohsihaharkka:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException
```

2.11: VIRHEKOODEJA YRITTÄESSÄNI LUODA "ACCOUNTS" SOVELLUSTA

Ongelma selvisi, kun käynnistin Powershellin uudestaan ja poistin Visual Studiosta "accounts" sovelluksen. Nyt loin uudestaan Powershellissä "accounts" sovelluksen ja lisäsin sen vasta tämän jälkeen Installed Apps luetteloon.

Seuraavaksi loin *views.py* tiedostoon signup -luokan:

```
views.py > ...
1 from django.shortcuts import render
2 from django.contrib.auth.forms import UserCreationForm
3 from django.urls import reverse_lazy
4 from django.views import generic
5
6
7 def HomePageView(request):
8     return render(request, 'home.html', {})
9
10 class SignUp(generic.CreateView):
11     form_class = UserCreationForm
12     success_url = reverse_lazy('login')
13     template_name = 'signup.html'
```

2.12: SIGNUP

Seuraavaksi lisäsin rekisteröitymisen *urls.py* tiedostoon komennolla *path('signup/', views.SignUp.as_view(), name='signup')*, (kuvassa rivillä 26)

```

21 urlpatterns = [
22     path('admin/', admin.site.urls),
23     path('', include('applikaatio.urls')),
24     path('', include("django.contrib.auth.urls")),
25     path('', TemplateView.as_view(template_name='home.html'), name='home'),
26     path('signup/', views.SignUp.as_view(), name='signup'),
27 ]

```

2.13: REKISTERÖITYMISEN LINKITYS

Ja alkuun *from . import views* :

```

16 from django.contrib import admin
17 from django.urls import path, include
18 from django.views.generic.base import TemplateView
19 from applikaatio import views

```

Luodaan vielä *signup.html* tiedosto, joka täytetään seuraavilla komennoilla:

```

templates > <> signup.html > ...
1  {% extends 'base.html' %}
2
3  {% block title %}Sign Up{% endblock %}
4
5  {% block content %}
6      <h2>Sign up</h2>
7      <form method="post">
8          {% csrf_token %}
9          {{ form.as_p }}
10         <button type="submit">Sign up</button>
11     </form>
12 {% endblock %}

```

2.14: REKISTERÖITYMISEN HTML TEMPLATE

CRUD-TOIMINNOT

Aluksi myöskään CRUD-toimintoja varten en pystynyt luomaan uutta sovellusta, koska edelleen sama virheviesti oli voimassa:

ModuleNotFoundError: No module named 'accounts'

Tämä korjautui samalla, kuin edellisenkin vaiheen ongelma. Loin Powershellissä sovelluksen "books" ja lisäsin sen projektikansion sovelluslistaan. Loin sinne html -templatet "book_confirm_delete", "book_detail", "book_confirm" ja "book_list".

models.py tiedostoon loin luokan "Book":

```

models.py > ...
1  from django.db import models
2  from django.urls import reverse
3
4  class Book(models.Model):
5      name = models.CharField(max_length=200)
6      pages = models.IntegerField()
7
8      def __str__(self):
9          return self.name
10
11     def get_absolute_url(self):
12         return reverse('book_edit', kwargs={'pk': self.pk})

```

2.15: LUOKKA BOOK

views.py tiedostoon loin seuraavan kuvan mukaiset luokat:

```

views.py > ...
1  from django.http import HttpResponse
2  from django.views.generic import ListView, DetailView
3  from django.views.generic.edit import CreateView, UpdateView, DeleteView
4  from django.urls import reverse_lazy
5
6  from books.models import Book
7
8  class BookList(ListView):
9      model = Book
10
11     class BookView(DetailView):
12         model = Book
13
14     class BookCreate(CreateView):
15         model = Book
16         fields = ['name', 'pages']
17         success_url = reverse_lazy('book_list')
18
19     class BookUpdate(UpdateView):
20         model = Book
21         fields = ['name', 'pages']
22         success_url = reverse_lazy('book_list')
23
24     class BookDelete(DeleteView):
25         model = Book
26         success_url = reverse_lazy('book_list')

```

2.16: VIEWS.PY

Lisäsin Book -luokan vielä projektin models.py tiedostoon:

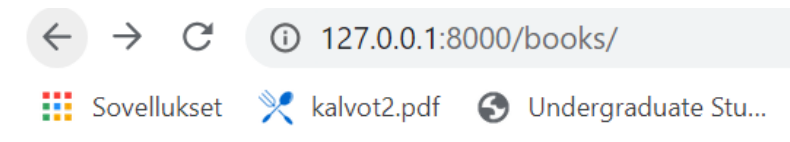
```

13 class Book(models.Model):
14     name = models.CharField(max_length=200)
15     pages = models.IntegerField()
16
17     def __str__(self):
18         return self.name
19
20     def get_absolute_url(self):
21         return reverse('book_edit', kwargs={'pk': self.pk})

```

2.17: BOOK LUOKKA PROJEKTIN MODELS.PY TIEDOSTOSSA

Testattaessa sovelluksen toimivuutta (ajamalla ensiksi *py manage.py runserver*) pääsee kirjalistaan, jossa näkee lisätyt kirjat ja pystyy tarkastelemaan ja editoimaan sen ominaisuuksia:



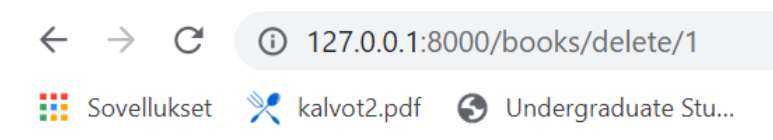
Books

Name	Pages	View	Edit	Delete
Satukirja	666	view	edit	delete

[New](#)

2.18: BOOKS ETUSIVU

Jos kirjan haluaa poistaa ("delete") kirjastosta, sovellus kysyy vielä, onko käyttäjä varma toiveestaan:

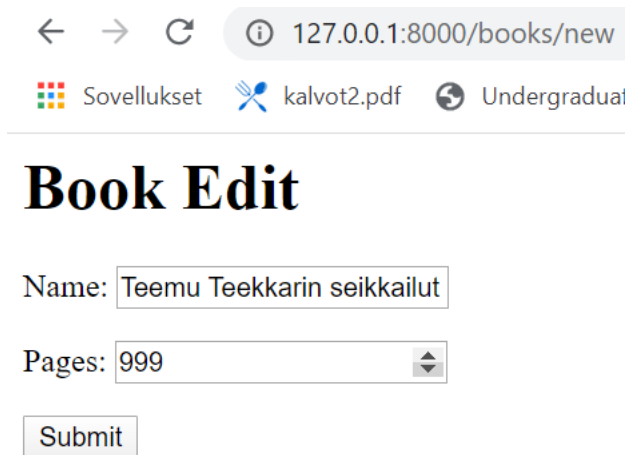


Book Delete

Are you sure you want to delete "Satukirja" ?

2.19: KIRJAN POISTAMINEN

Painamalla kirjavalikossa "New" pääsee käyttäjä lisäämään listaan kirjan:



← → ↻ ⓘ 127.0.0.1:8000/books/new

Sovellukset kalvot2.pdf Undergraduat

Book Edit

Name:

Pages:

2.20: UUDEN KIRJAN LISÄYS

HELPPOUDET JA VAIKEUDET

Helppoa:

MS Visual Studiassa tiedostojen lisäys ja muokkaus, koska siihen oli hyvin helpot step-by-step ohjeet.

Vaikeuksia:

Powershellissä täytyy kiinnittää huomiota siihen, että tekee komennon oikeassa kansiossa, ja että tarviiko virtaaliympäristö olla aktivoituna.

Ylitsepääsemätön vaikeus oli rekisteröitymisosion alussa iskenyt "ModuleNotFoundError: No module named 'accounts'", josta en päässyt eroon edes uudelleen aloittamalla. Sama virhe esti myöhemmin myös CRUD-toimintojen luomisen loppuun viemisen.

VAIHE 3: DATARAJAPINNAN KÄYTTÖ

Tässä vaiheessa harjoitustyötä lisäsin avoimen datan rajapinnan. En oikein keksinyt, mitä käyttäisin, joten päätin aluksi hyödyntää Spotifyn avointa dataa, koska olen musiikista hyvin kiinnostunut. Ajan puutteen vuoksi en kerennyt tähän kuitenkaan perehtyä, joten käytin Tampereen kaupungin dataa sen kirjastoista, koska se oli hyvin yksinkertainen, suppea ja helppo käyttää. Sovellukseni näyttää Tampereen kirjastojen nimet, osoitteet, postinumeron, puhelinnumeron ja linkin verkkosivulle. Lisäksi sivulla on linkki etusivulle palaamista varten.

Aluksi loin Powershellissä uuden applikaation *Api* tutulla komennolla *py manage.py startapp Api*. Tämän jälkeen lisäsin applikaation projektin pääkansion *setting.py* tiedoston applikaatiolistaan. Varsinaisen *Api* -sovelluksen muokkauksen aloitin sen *views.py* kansiota lisäämällä sinne ominaisuudet datan keräykseen avoimen datan sivustolta.

```
views.py > home
1  from django.shortcuts import render
2  import requests
3  import json
4
5  def home(request):
6      url = 'https://data.tampere.fi/data/api/action/datastore_search?resource_id=0747926f-1593-423f-b351-5fa2ae477eb48'
7
8      response = requests.get(url)
9      fielddata = response.json()
10     fielddata = fielddata['result']
11     fielddata = fielddata['records']
12     data = []
13
14     for field in fielddata:
15         data.append(field)
16     return render(request, 'api.html', {'fielddata' : data})
```

3.1: DATAN KERÄYS DATALÄHTEESTÄ

datan keruu *views.py* tiedostossa. "url" jatkuu vielä pitkälle kuvan rajojen ulkopuolelle, se on copy-paste datarajapinnan osoitteesta.

Seuraavaksi muokkasin sovelluksen urls.py näyttämään dataa:

```
urls.py > ...  
1  from django.urls import path  
2  from . import views  
3  
4  urlpatterns = [  
5      path('', views.home, name='home'),  
6  ]  
7
```

3.2: SOVELLUKSEN URLS.PY TIEDOSTO

Oli myös tärkeää muistaa lisätä pääprojektikansioon polku uuteen sovellukseen:

```
urls.py  
1  from django.urls import path  
2  from . import views  
3  from django.urls import include  
4  
5  urlpatterns = [  
6      path('', views.HomePageView, name='kotisivu'),  
7      path('books/', include('books.urls')),  
8      path('api/', include('api.urls')),  
9  ]
```

3.3: VARSINAISEN PROJEKTIN URLS.PY TIEDOSTOON LISÄSIN POLUN UUTEEN "API" SOVELLUKSEEN

Jotta visuaalisesti sovellus olisi toimiva, tein Api sovellukseen html tiedoston datan näkymää varten:

```

urls.py  views.py  models.py  tests.py  api.html  apps.py
Templates > api.html > html > body > div.container > a
1  <!doctype html>
2  <html lang="en">
3      <head>
4          <meta charset="utf-8">
5          <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
6          <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css" integrity=
7
8          <title>Libraries in Tampere</title>
9      </head>
10     <body>
11         <div class="container">
12             <div class="row justify-content-center">
13                 <div class="col-8">
14                     <hr class="mt-0 mb-4">
15                     {% block content %}
16                     <table class="table table-striped"><tr><th>Nimi</th><th>Osoite</th><th>Postinumero</th><th>Puheli
17                         {% for field in fielddata %}
18                         <tr>
19                             <td>{{ field.NIMI }}</td>
20                             <td>{{ field.OSOITE }}</td>
21                             <td>{{ field.POSTINUMERO }}</td>
22                             <td>{{ field.PUHELIN }}</td>
23                             <td>{{ field.URL }}</td>
24                         </tr>
25                         {% endfor %}
26                     </table>
27                     {% endblock %}
28                 </div>
29             </div>
30             <a href="{% url 'home' %}">Etusivu</a>

```

3.4: HTML TIEDOSTON ALKU

```

Templates > api.html > html > body > div.container > div.row.justify-content-center > div.col-8
1
2
3
4
5
6  .css" integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdKnLPMO" crossorigin="anonymous">
7
8
9
10
11
12
13
14
15
16  <th>Puhelinnumero</th><th>Verkkosivu</th></th></tr>
17

```

3.5: HTML TIEDOSTO JATKUU OIKEALLE NÄIN

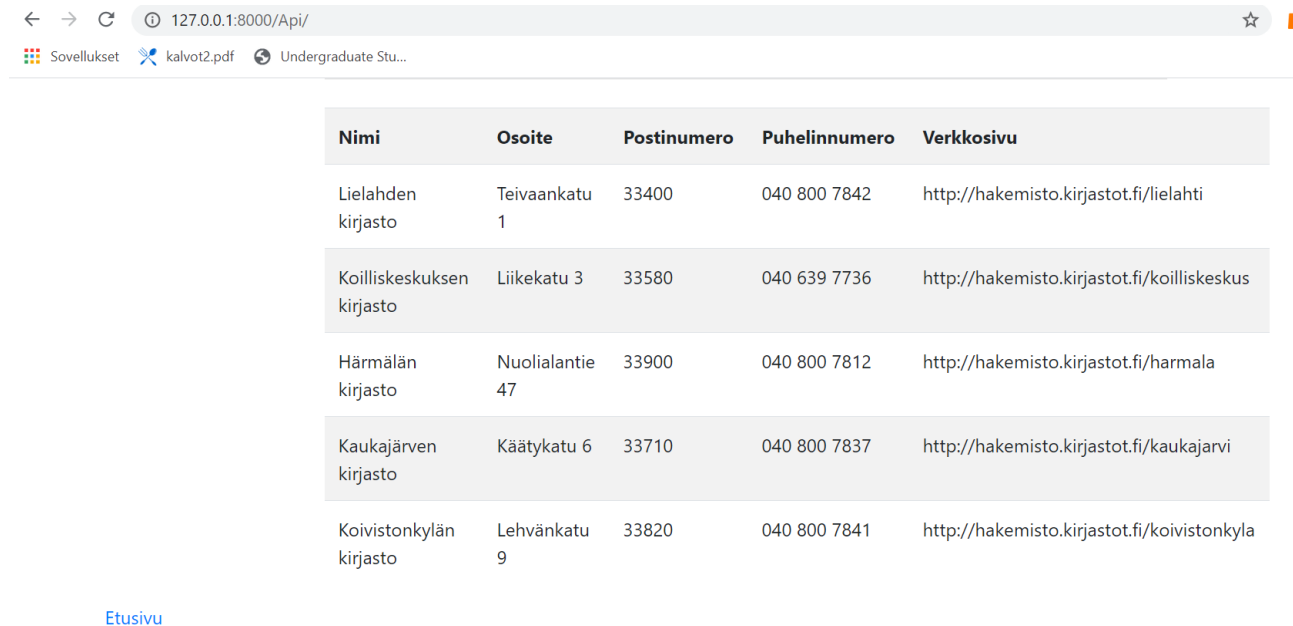
```

31  </div>
32  </body>
33  </html>

```

3.6: VIIMEISET KOLME RIVIÄ, JOTKA EIVÄT MAHTUNEET ENSIMMÄISEEN KUVAAAN

Nyt koodi oli valmis. Seuraavaksi käynnistin serverin tuttuun tapaan `py manage.py runserver` komennolla ja pääsin tarkastelemaan lopputulosta:



Nimi	Osoite	Postinumero	Puhelinnumero	Verkkosivu
Lielahden kirjasto	Teivaankatu 1	33400	040 800 7842	http://hakemisto.kirjastot.fi/lielahti
Koilliskeskuksen kirjasto	Liikekatu 3	33580	040 639 7736	http://hakemisto.kirjastot.fi/koilliskeskus
Härmälän kirjasto	Nuolialantie 47	33900	040 800 7812	http://hakemisto.kirjastot.fi/harmala
Kaukajärven kirjasto	Käätykatu 6	33710	040 800 7837	http://hakemisto.kirjastot.fi/kaukajarvi
Koivistonkylän kirjasto	Lehvänkatu 9	33820	040 800 7841	http://hakemisto.kirjastot.fi/koivistonkyla

[Etusivu](#)

3.7: API -SOVELLUS SELAIMESSA

HELPPOUDET JA VAIKEUDET

Helppoa:

- Sivuston ulkoasun muokkaus html templateissa
- Esitettävien tietojen rajausta ja järjestely

Vaikeaa:

- Internetistä löytyvien ohjeiden soveltaminen omaan työhön
- Mielikuvituksen käyttäminen datan aihetta keksiessä

LISÄPISTEET: PALAUTUS JA LISENSSI GITHUBISSA

Koska en harjoitustyön neljättä vaihetta osannut kovasta yrittämisestä huolimatta, päätin pelastavani osan pisteistä lisäpistetehtävällä. Palautin siis työni myös Githubiin ja lisäsin siihen lisenssin.

TYÖN PALAUTUS GITHUBIIN

Aloitin palautuksen lataamalla tietokoneelleni Git Bashin. Tässä pystyin lisäämään kätevästi koko harjoitustyökansioni Github -tililleni muutamalla komennolla.

Aluksi navigoin harjoitustyökansiooni samalla tavalla, kuin Powershellissä:

```
cd documents
cd ohsihaharkka
cd harkkatyö
```

Seuraavaksi aloitin palautuksen kirjautumalla githubtililleni. Tämän tein komennoilla

```
git config --global user.email riku.sundqvist@hotmail.fi
git config --global user.name RikuSun
```

Aloitin "palautus" -nimisen palautuksen komennolla

```
git commit -m "palautus"
```

Lopuksi lisäsin tiedostot komennolla

```
git remote add origin https://github.com/RikuSun/OhsihaOIKEA.git
```

, jossa linkki on Githubin osoite palautukselle

Lopullisesti tiedostot latautuivat komennolla

```
git push origin master
```

, minkä jälkeen pyydettiin Github käyttäjätunnukseni ja salasananani. Nämä syötettyäni harjoitustyö oli Githubissani.

RikuSun / OhsihaOIKEA

Unwatch 1 Unstar 1 Fork 0

Code Issues 0 Pull requests 0 Actions Projects 0 Wiki Security 0 Insights Settings

No description, website, or topics provided. [Edit](#)

Manage topics

1 commit 1 branch 0 packages 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

RikuSun palautus		Latest commit 6292fb8 26 minutes ago
Api	palautus	26 minutes ago
Harkkатыö	palautus	26 minutes ago
accounts	palautus	26 minutes ago
applikaatio	palautus	26 minutes ago
books	palautus	26 minutes ago
Harjoitustyö, vaihe 1.pdf	palautus	26 minutes ago
Harkkатыön raportti.docx	palautus	26 minutes ago
db.sqlite3	palautus	26 minutes ago
index.html	palautus	26 minutes ago
manage.py	palautus	26 minutes ago

Linkki repoon: <https://github.com/RikuSun/OhsihaOIKEA>

LISENSSIN LISÄÄMINEN

Lisäsin vielä työhöni MIT-lisenssin, koska se oli koettu perushyväksi aiemmissa töissä, enkä nähnyt oman työni kohdalla syytä epäillä tätä.

Lisenssin lisäsin repossa seuraavalla tavalla:

Create new file -> nimeksi LICENCE.txt -> Want to use a template? Choose a licence

Vetovalikosta valitsin MIT lisenssin. Lopuksi painoin "Commit New File" ja lisenssi ilmestyi repooni.

Api	palautus	38 minutes ago
Harkkатыö	palautus	38 minutes ago
accounts	palautus	38 minutes ago
applikaatio	palautus	38 minutes ago
books	palautus	38 minutes ago
Harjoitustyö, vaihe 1.pdf	palautus	38 minutes ago
Harkkатыön raportti.docx	palautus	38 minutes ago
LICENCE	LICENCE	4 minutes ago
db.sqlite3	palautus	38 minutes ago
index.html	palautus	38 minutes ago
manage.py	palautus	38 minutes ago

LÄHTEET

Sisään- ja uloskirjautuminen: <https://learndjango.com/tutorials/django-login-and-logout-tutorial>

Rekisteröityminen: <https://learndjango.com/tutorials/django-signup-tutorial>

CRUD: <https://rayed.com/posts/2018/05/django-crud-create-retrieve-update-delete/>

Avoimen datan rajapinnan käyttöönotto:

<https://simpleisbetterthancomplex.com/tutorial/2018/02/03/how-to-use-restful-apis-with-django.html>

Datalähde: <https://data.tampere.fi/data/fi/dataset/tampereen-kirjastot>

Kurssin luento- ja koodiklinikatallenteet