https://www.halvorsen.blog

Django Admin

Hans-Petter Halvorsen

Contents

Introduction

Hans-Petter Halvorsen

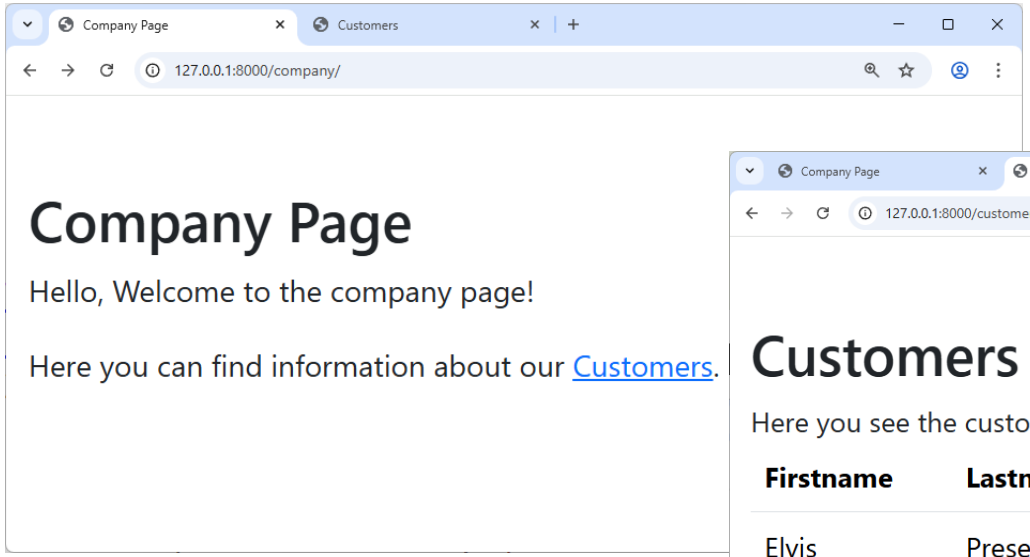
Table of Contents

Django

- Django is a Python web development framework.
- Django is a back-end/server-side web framework.
- Django has support for databases like SQLite, MySQL/MariaDB and PostgreSQL.
- Django includes an SQLite database, but for the others you need to install the database system you want to use from their respective websites and in addition install a Python package/driver for the specific database system.
- Django is free, open source and written in Python.
- Homepage: https://www.djangoproject.com

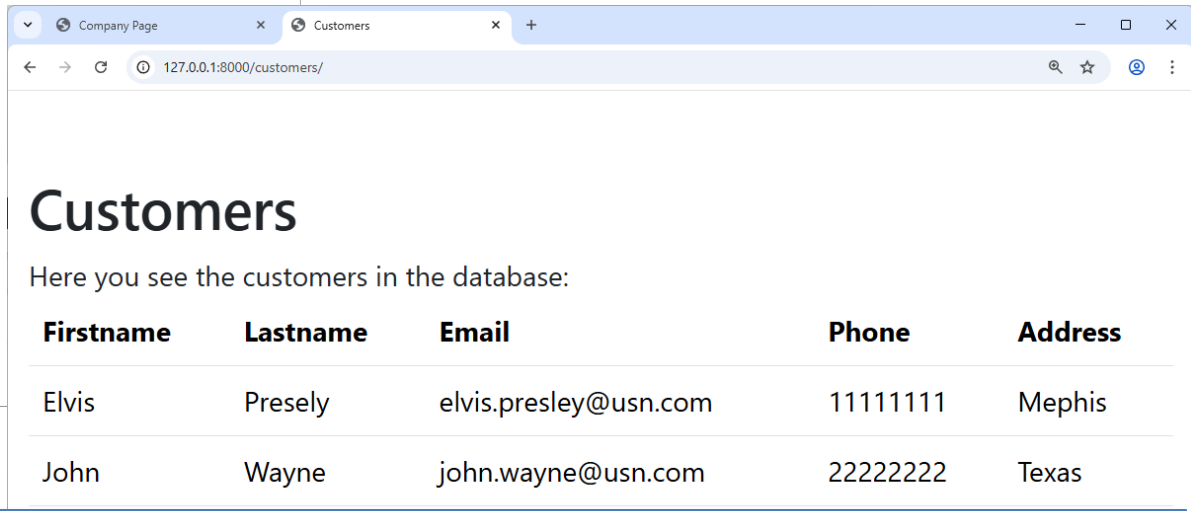
Django “Company” App

In the “Django Tutorial” we created the following Application with Django:



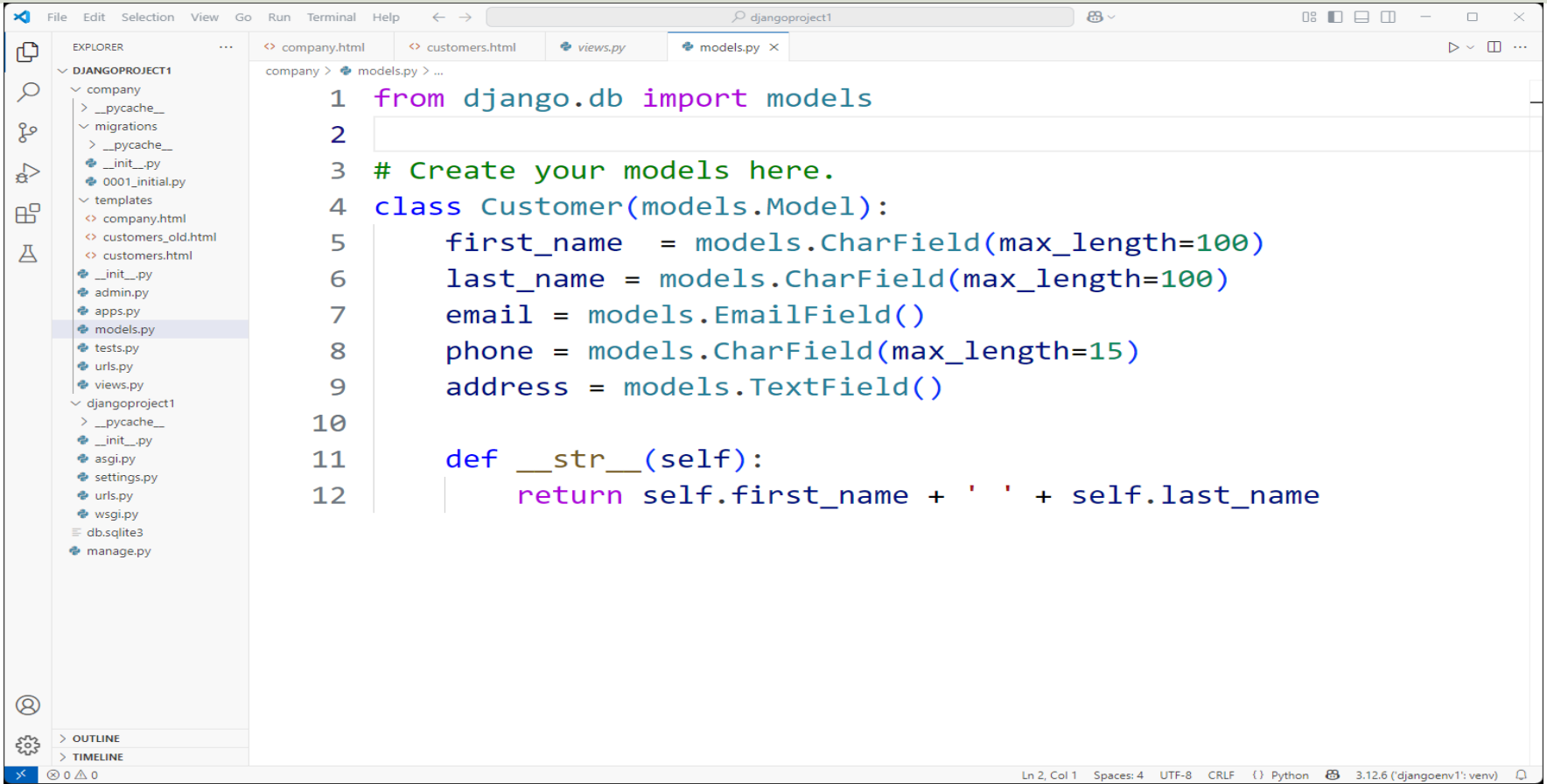
Main “Company” App

Subpage showing the “Customers” for the “Company” App:



We will use the “Django Admin“ interface to create CRUD functionality for the “Customer”, i.e., Create (insert), Read (select), Update and Delete data from the database.

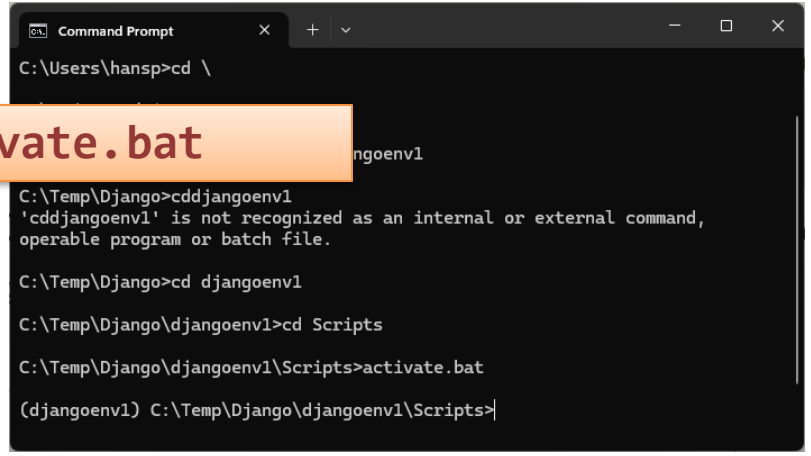
Django “Company” Project



Start Django “Company” Project

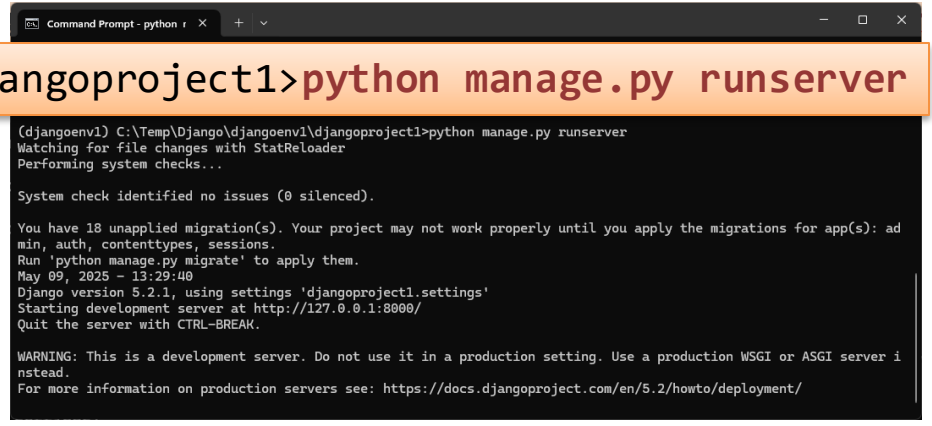
Activate the Virtual Python Environment

```
C:\Temp\Django\djangoenv1\Scripts>activate.bat
```



Run the Django Project:

```
(djangoenv1) C:\Temp\Django\djangoenv1\djangoproject1>python manage.py runserver
```



Open your Web Browser and enter the URL: **127.0.0.1:8000/company**

Django Tutorial and “Company App”

- Django Tutorial: https://www.youtube.com/playlist?list=PLdb-TcK6Aqj0enrpIQ0P4ISI8ofhk8Sza
- This Tutorial gives an overview of Django and create a “Company” Django App step by step.
- The ”Company” App is used as a starting point for this “Django Admin“ Tutorial.

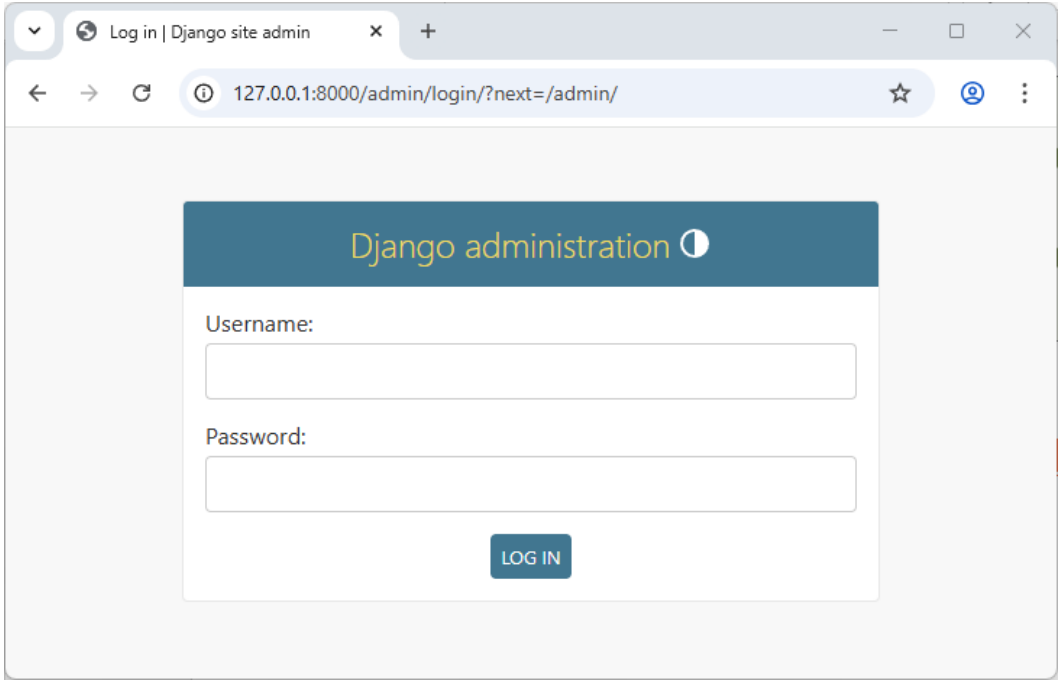
Django Admin

Hans-Petter Halvorsen

Table of Contents

Django Admin

http://127.0.0.1:8000/admin



“Django Admin“ is included with Django.

Before we can start using “Django Admin”, we need to create a User.

Django Admin

- “Django Admin“ is included with Django.
- Django provides a ready-to-use user interface for administrative activities. Django automatically generates admin UI based on your Django Models.
- “Django Admin” offers a CRUD user interface for all your Models.
- CRUD is short for Create, Read, Update and Delete.
- The “Django Admin“ interface is intended for “superusers” and not for ordinary users of your application.
- To use it you need to create a User.

Django Admin - Create User

Type the following to create a new User:

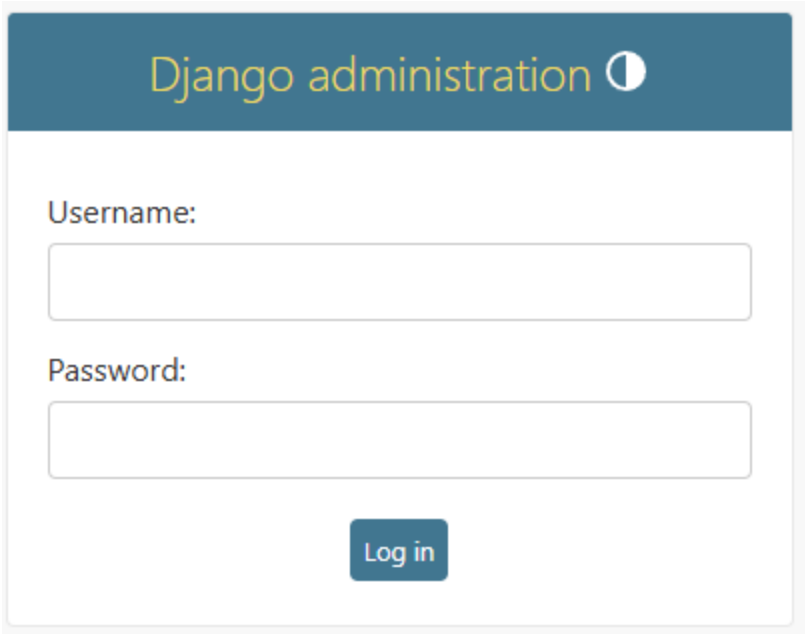
```
...>python manage.py createsuperuser
```

Theb you are asked to create a Username and a Password.

Then run the server:

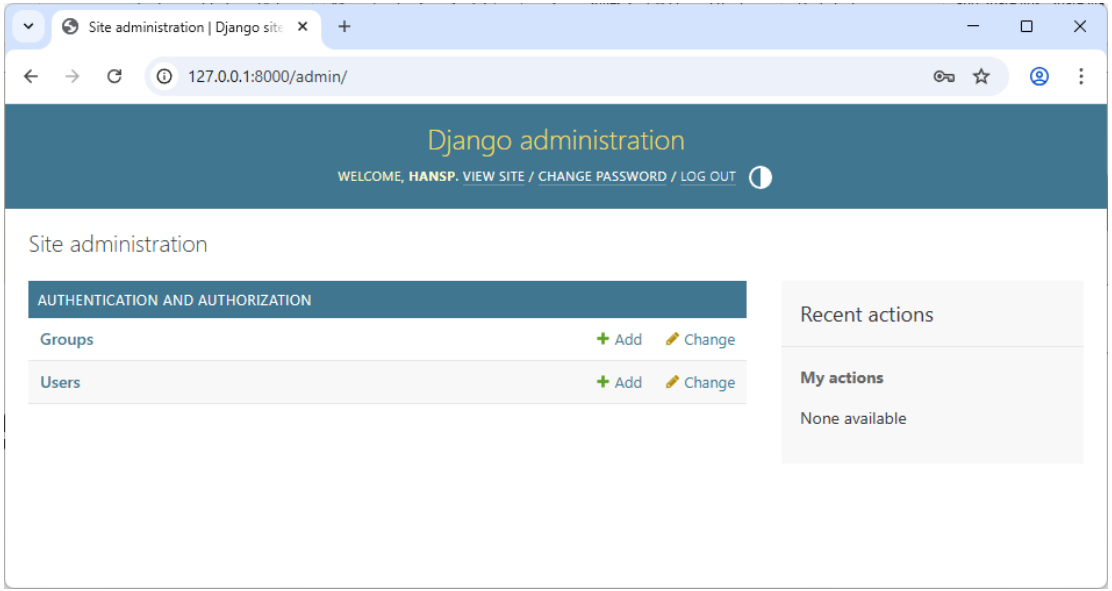
```
...>python manage.py runserver
```

Goto http://127.0.0.1:8000/admin and enter your Username and Password:



Django Admin

The following will appear:



Here you can Add, Change and Delete (CRUD functionality) Groups and Users.

Existing **Customer** Model

models.py

company > models.py > ...

```python
from django.db import models

# Create your models here.
class Customer(models.Model):
    first_name  = models.CharField(max_length=100)
    last_name = models.CharField(max_length=100)
    email = models.EmailField()
    phone = models.CharField(max_length=15)
    address = models.TextField()

    def __str__(self):
        return self.first_name + ' ' + self.last_name
```

By default, all Models created in the Django project will be created as tables in the default SQLite database (“db.sqlite3”) that is installed with Django.

Add “Customer” Model to Admin

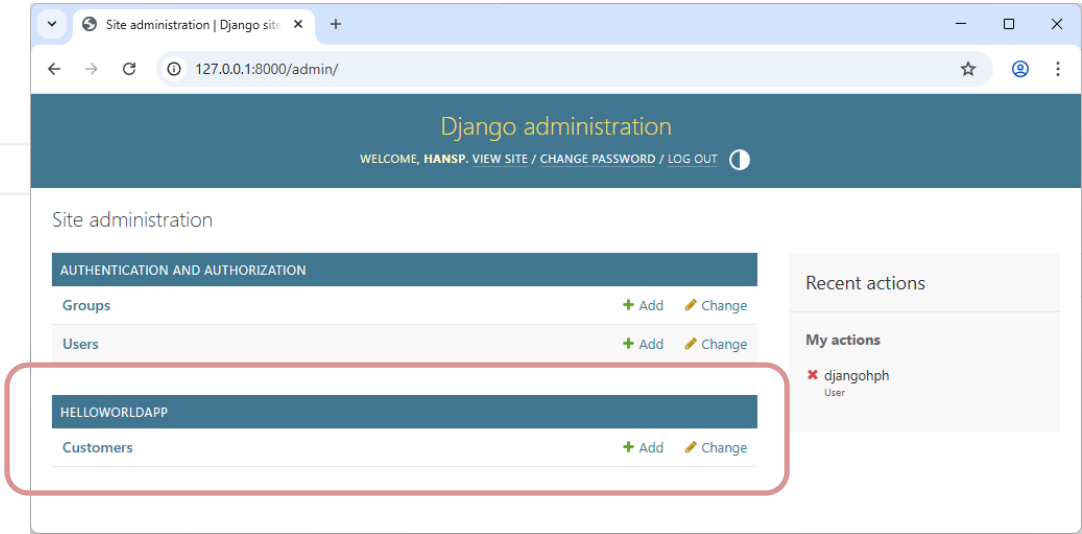
Django Admin offers a CRUD user interface for all your Models.
Let's add CRUD functionality for our Customer Model.

You need to register the Customer Model in the file “**admin.py**”:

admin.py

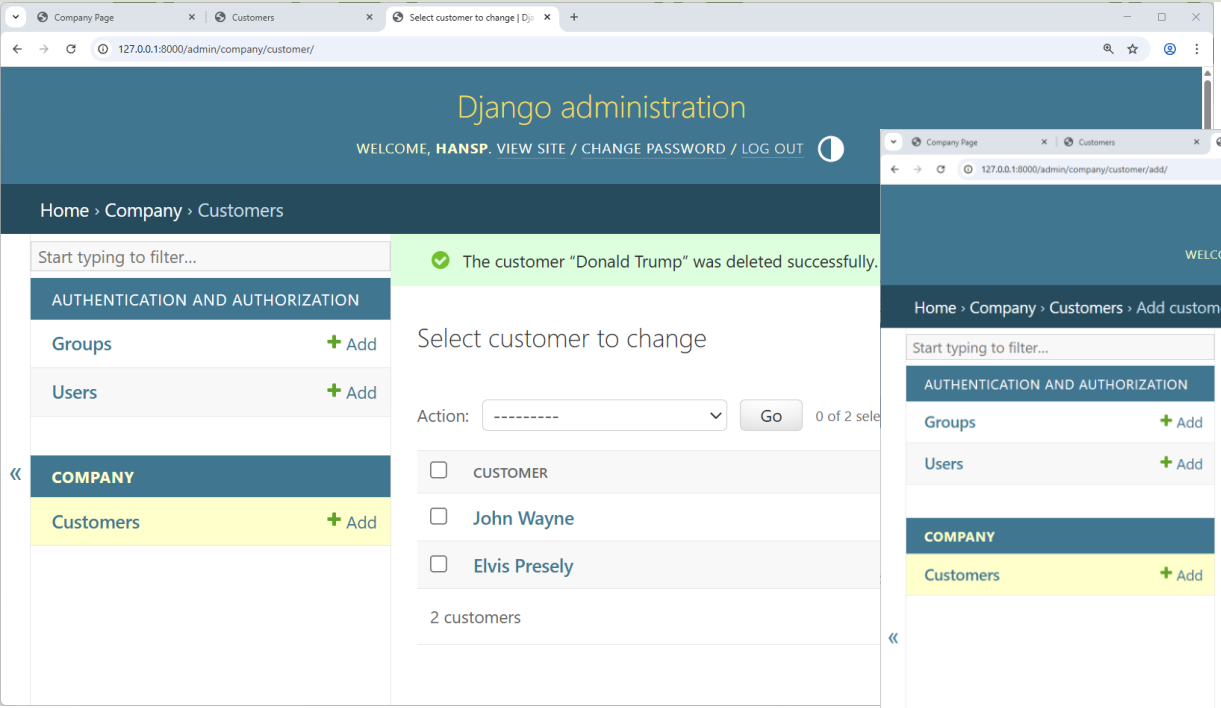
helloworldapp > admin.py

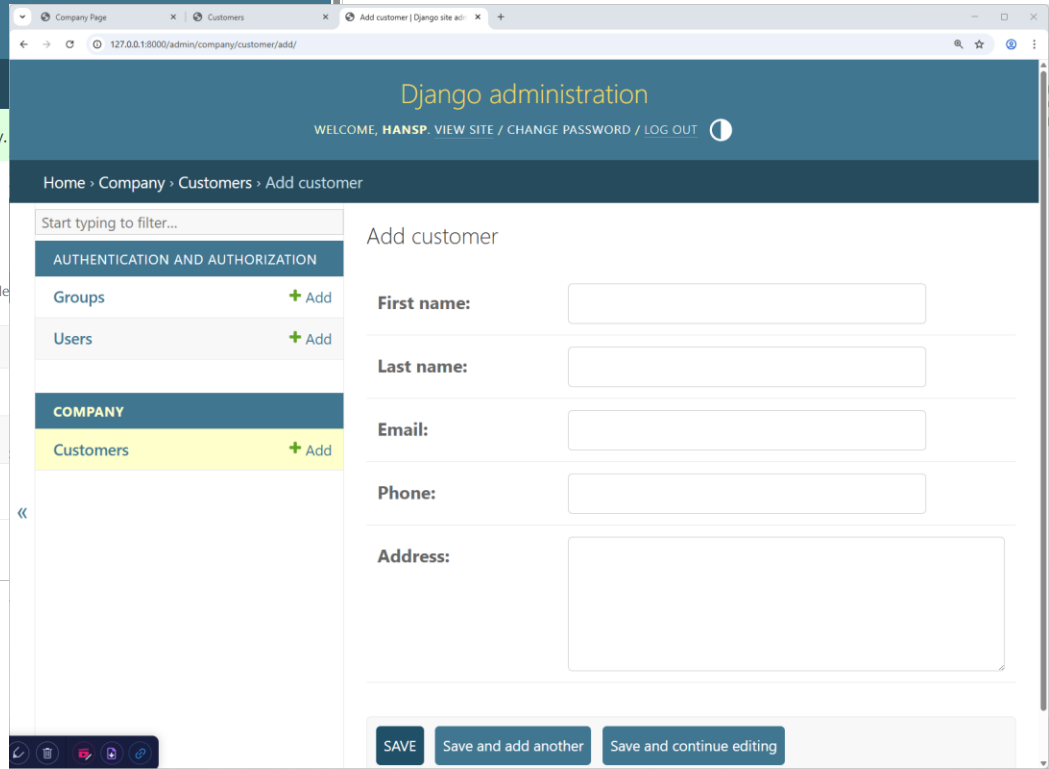
```
1  from django.contrib import admin
2  from .models import Customer
3
4  # Register your models here.
5  admin.site.register(Customer)
```



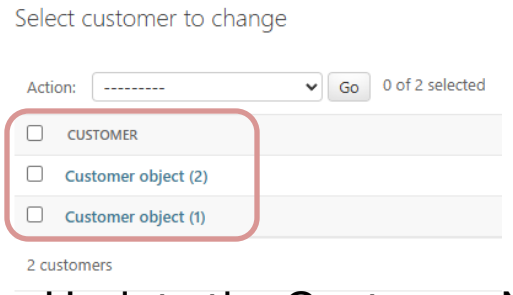
Then go back to
http://127.0.0.1:8000/admin

Customer CRUD Interface





Change Display Name

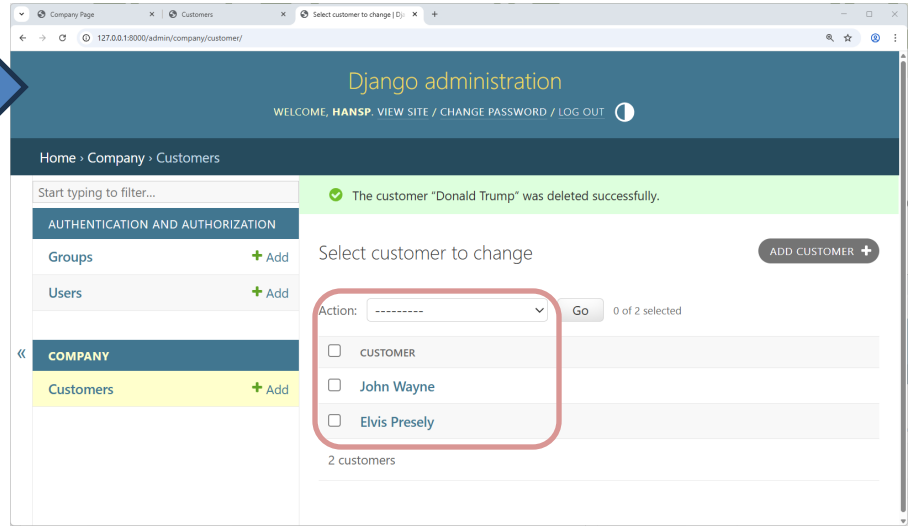


Update the Customer Model in models.py:

models.py

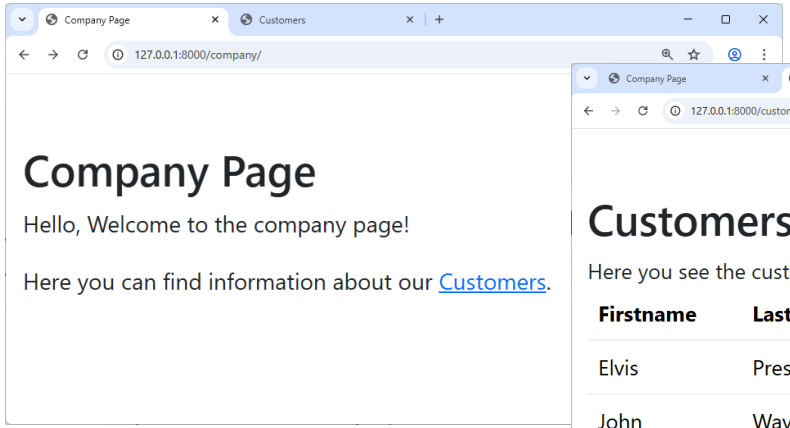
helloworldapp > models.py > ...

```python
from django.db import models

class Customer(models.Model):
    firstname = models.CharField(max_length=100)
    lastname = models.CharField(max_length=100)
    address = models.CharField(max_length=100)
    city = models.CharField(max_length=100)
    email = models.EmailField(max_length=100)
    phone = models.CharField(max_length=20)

    def __str__(self):
        return f'{self.firstname} {self.lastname}'
```

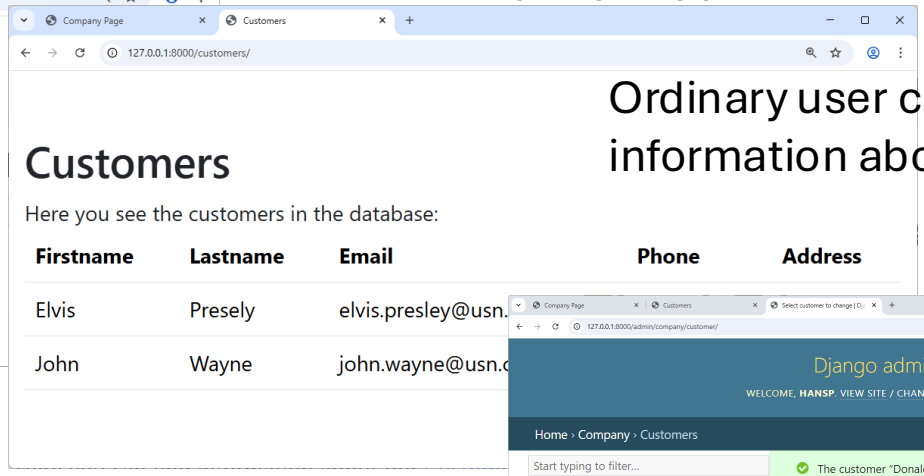


Final Django “Company” App

Main “Company” App

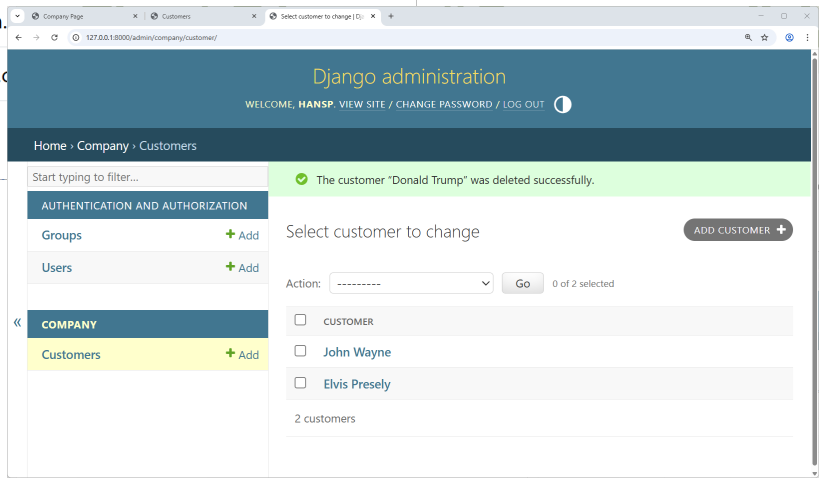


Subpage showing the “Customers” for the “Company” App:



Ordinary user can only see information about the customers

Admin interface for creating, changing and deleting Customers (this part is only available for superusers, and administrators and you need to login):



Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

E-mail: hans.p.halvorsen@usn.no

Web: https://www.halvorsen.blog