Requests response cycle

- Client: send HTTP request: URL, cookies to server
- Server: return HTTP response: HTML, Javascript

Model-View-Controller (M-V-C)

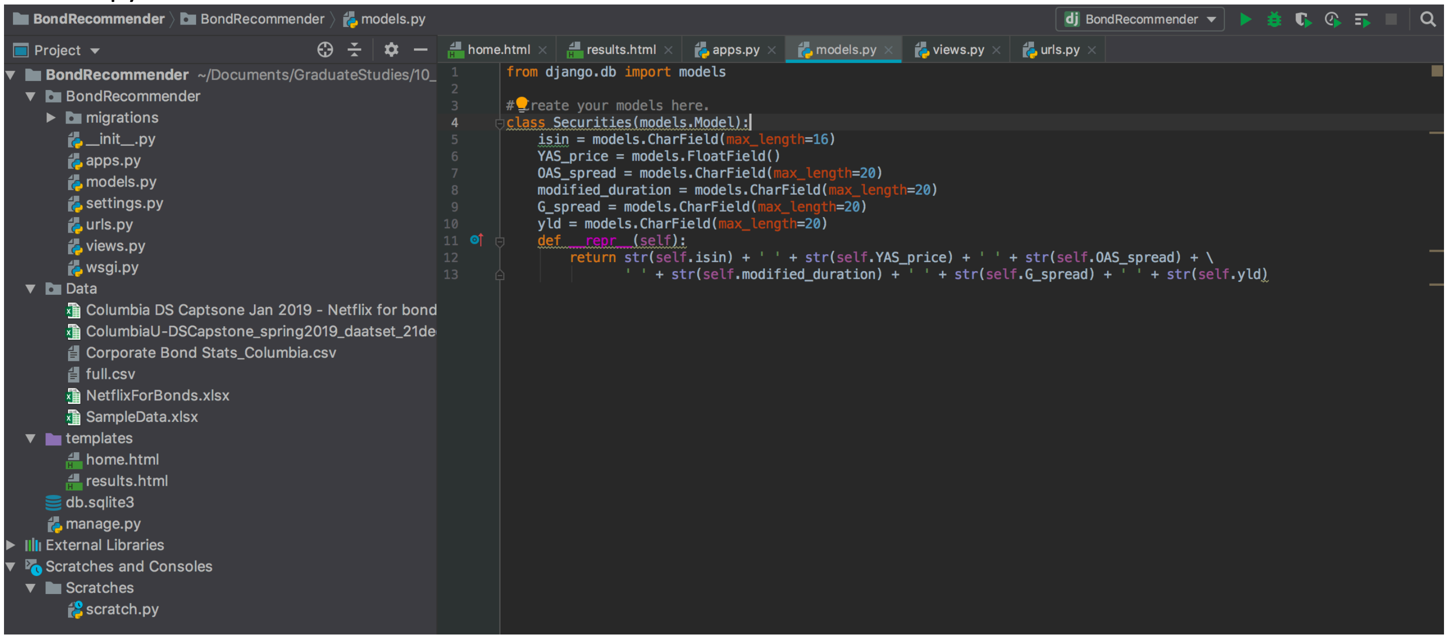
- Model: Data model. Handles communications with the databases
- View: User interface (or what the user sees)
- Controller: Application logic, or the code that handles the request- response cycle

Django

A full-service M-V-C web framework. Django is specifically written for Python

- Data Model (Model; models.py)
    - Represented by Python classes; Python function calls substitute for data queries
    - Possible to connect to many database backends (PostgreSQL, MySQl, Oracle, NoSQL, etc.)
- View (HTML Templates)
    - Maps URL between templates and the controller
    - Transfers data between templates and the controller
- Controller (views.py)
    - Implemented in Python
- Hosting of application
    - On local server vs on Cloud

Models.py – Model



```python
from django.db import models

#Create your models here.
class Securities(models.Model):
    isin = models.CharField(max_length=16)
    YAS_price = models.FloatField()
    OAS_spread = models.CharField(max_length=20)
    modified_duration = models.CharField(max_length=20)
    G_spread = models.CharField(max_length=20)
    yld = models.CharField(max_length=20)
    def __repr__(self):
        return str(self.isin) + ' ' + str(self.YAS_price) + ' ' + str(self.OAS_spread) + \
               ' ' + str(self.modified_duration) + ' ' + str(self.G_spread) + ' ' + str(self.yld)
```

*In order to set up database, we need following two steps;

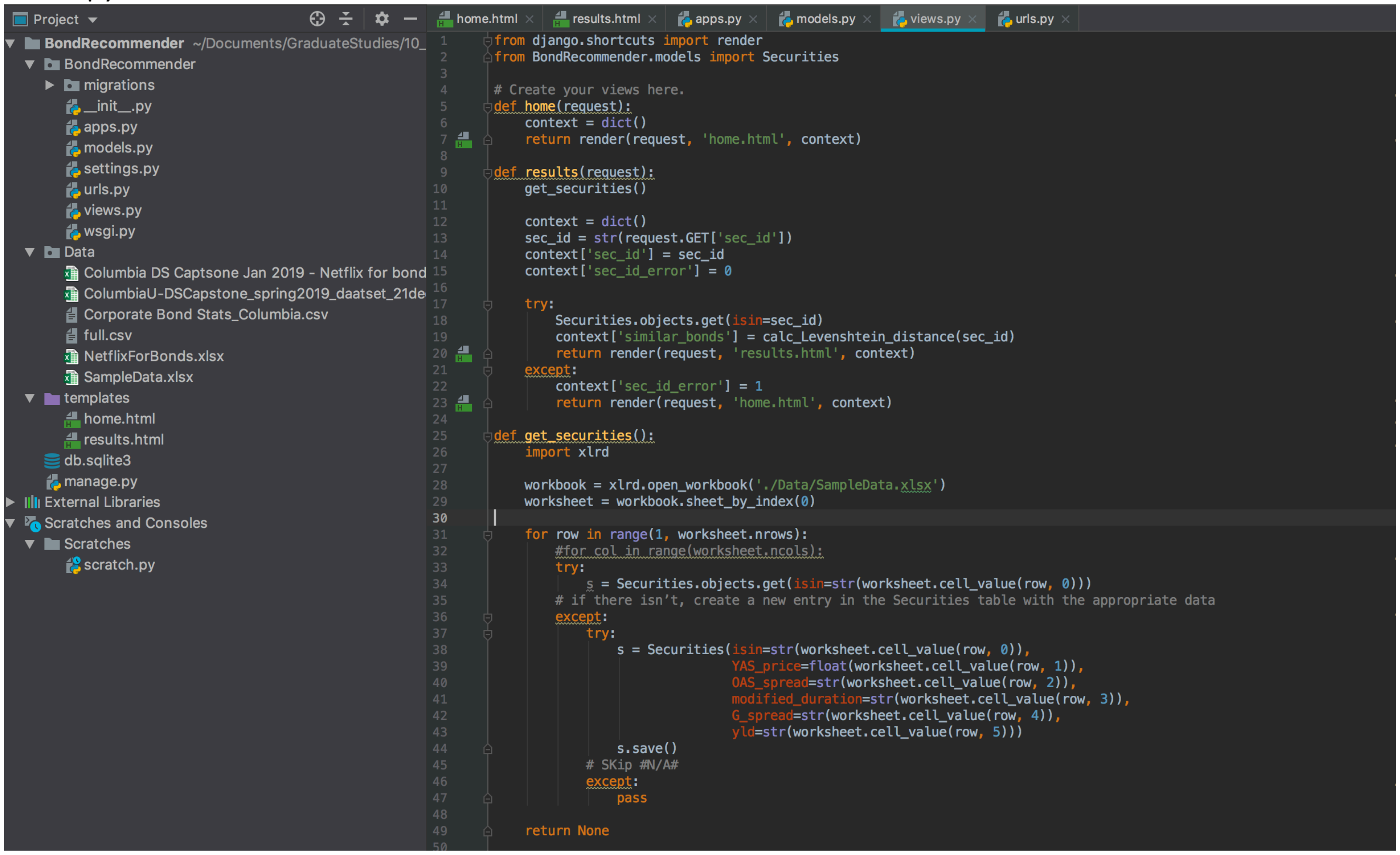
```
python manage.py makemigrations
```

Creates a record of changes that need to be made to the database. (the actual database stays unchanged.)

```
python manage.py migrate
```

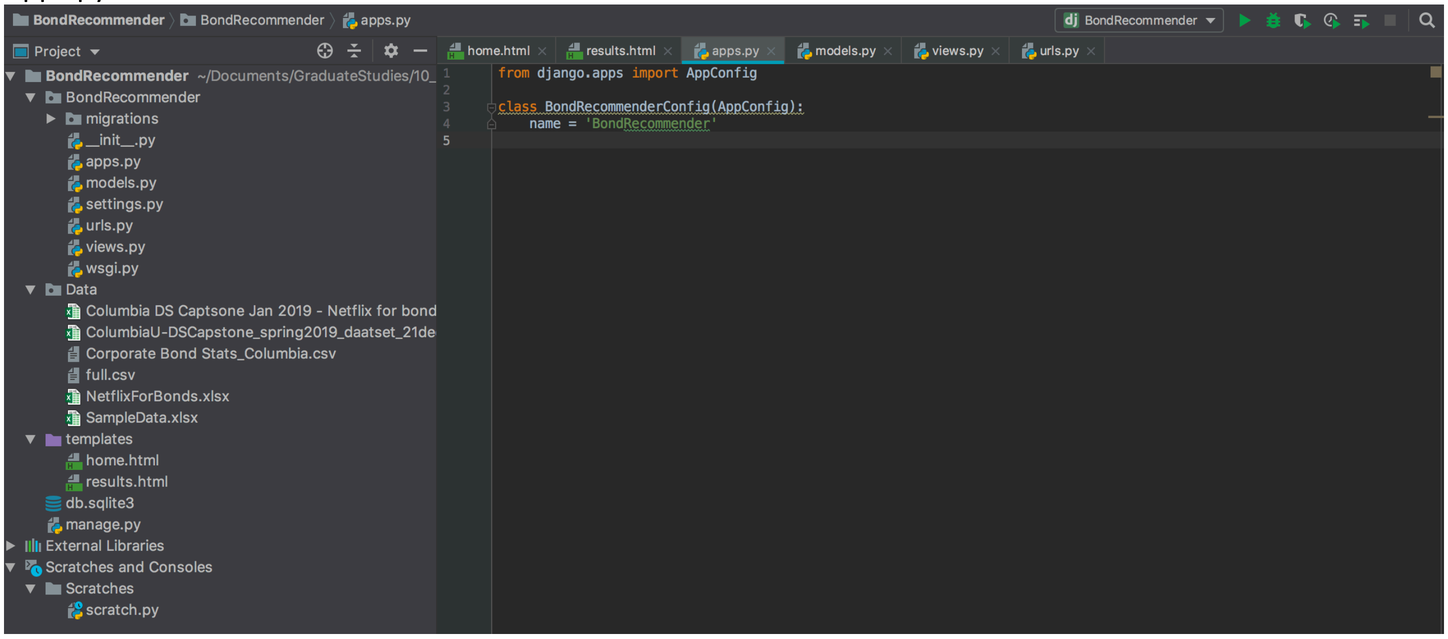
Applies any unapplied migrations

Views.py - Controller



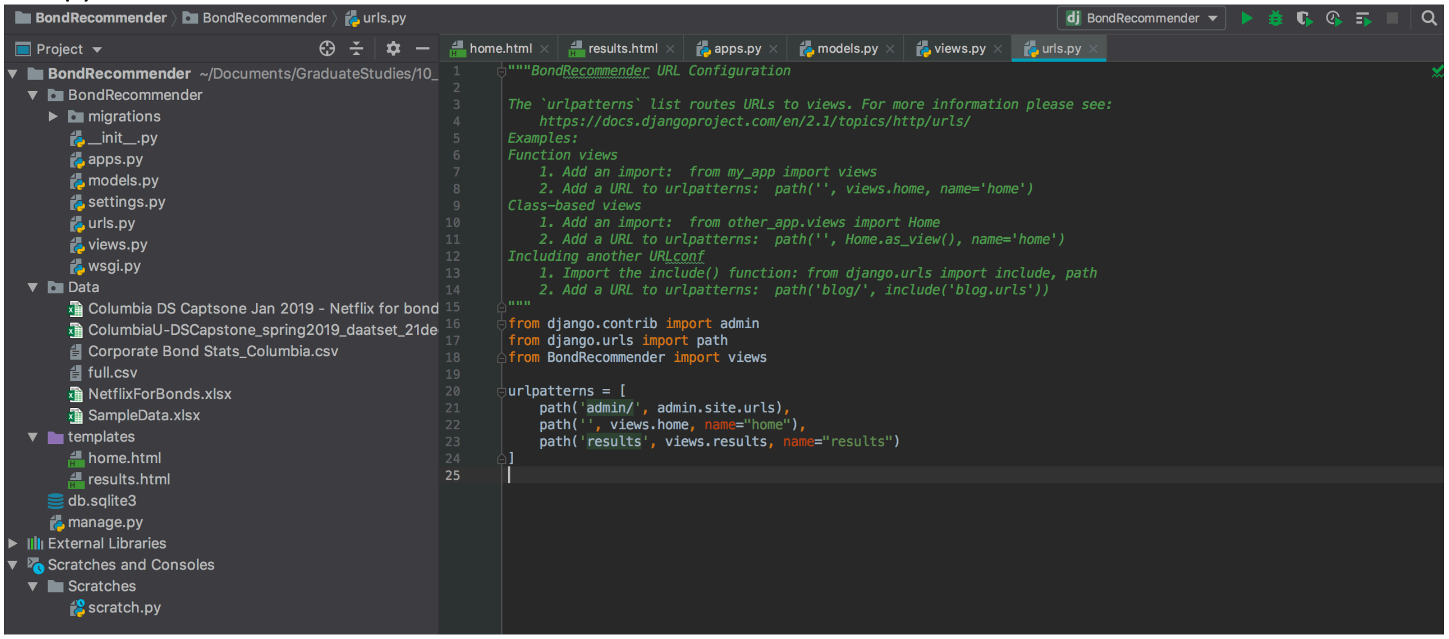
```python
from django.shortcuts import render
from BondRecommender.models import Securities

# Create your views here.
def home(request):
    context = dict()
    return render(request, 'home.html', context)

def results(request):
    get_securities()

    context = dict()
    sec_id = str(request.GET['sec_id'])
    context['sec_id'] = sec_id
    context['sec_id_error'] = 0

    try:
        Securities.objects.get(isin=sec_id)
        context['similar_bonds'] = calc_Levenshtein_distance(sec_id)
        return render(request, 'results.html', context)
    except:
        context['sec_id_error'] = 1
        return render(request, 'home.html', context)

def get_securities():
    import xlrd

    workbook = xlrd.open_workbook('./Data/SampleData.xlsx')
    worksheet = workbook.sheet_by_index(0)

    for row in range(1, worksheet.nrows):
        #for col in range(worksheet.ncols):
        try:
            s = Securities.objects.get(isin=str(worksheet.cell_value(row, 0)))
        # if there isn’t, create a new entry in the Securities table with the appropriate data
        except:
            try:
                s = Securities(isin=str(worksheet.cell_value(row, 0)),
                               YAS_price=float(worksheet.cell_value(row, 1)),
                               OAS_spread=str(worksheet.cell_value(row, 2)),
                               modified_duration=str(worksheet.cell_value(row, 3)),
                               G_spread=str(worksheet.cell_value(row, 4)),
                               yld=str(worksheet.cell_value(row, 5)))

                s.save()
            # SKip #N/A#
            except:
                pass

    return None
```

Apps.py



```python
from django.apps import AppConfig

class BondRecommenderConfig(AppConfig):
    name = 'BondRecommender'
```

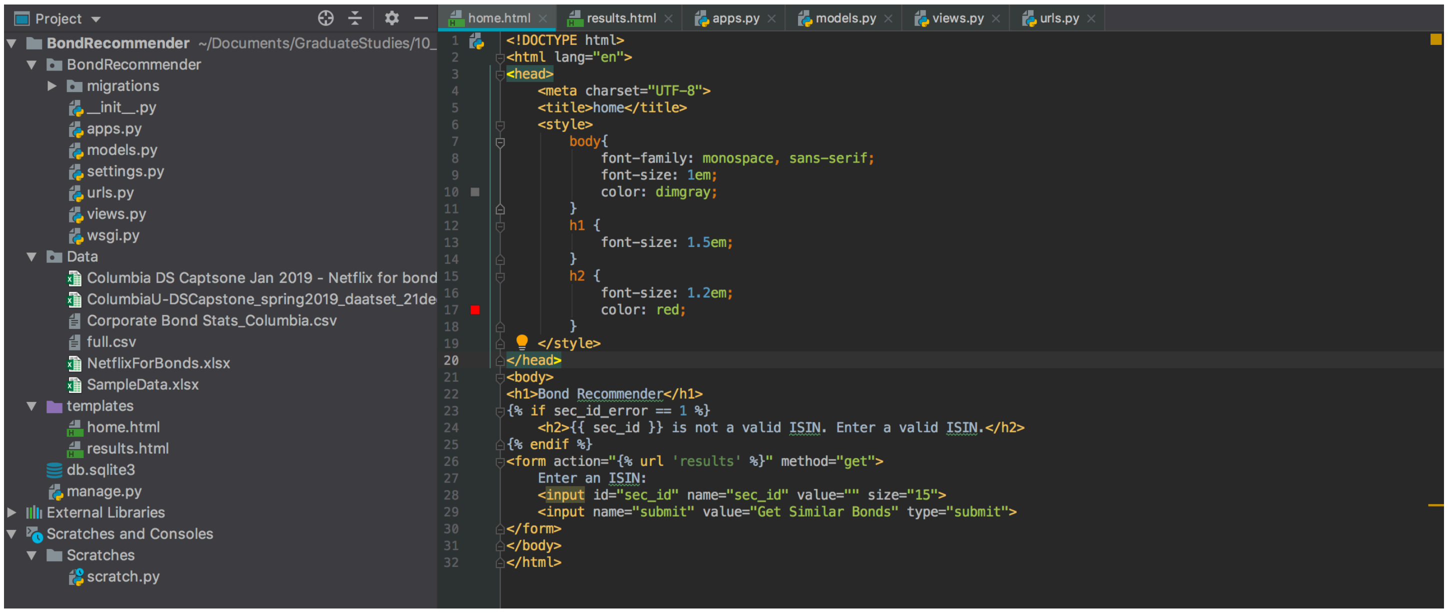
Urls.py



```python
"""BondRecommender URL Configuration

The `urlpatterns` list routes URLs to views. For more information please see:
    https://docs.djangoproject.com/en/2.1/topics/http/urls/
Examples:
Function views
    1. Add an import:  from my_app import views
    2. Add a URL to urlpatterns:  path('', views.home, name='home')
Class-based views
    1. Add an import:  from other_app.views import Home
    2. Add a URL to urlpatterns:  path('', Home.as_view(), name='home')
Including another URLconf
    1. Import the include() function: from django.urls import include, path
    2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
"""
from django.contrib import admin
from django.urls import path
from BondRecommender import views

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', views.home, name="home"),
    path('results', views.results, name="results")
]

```

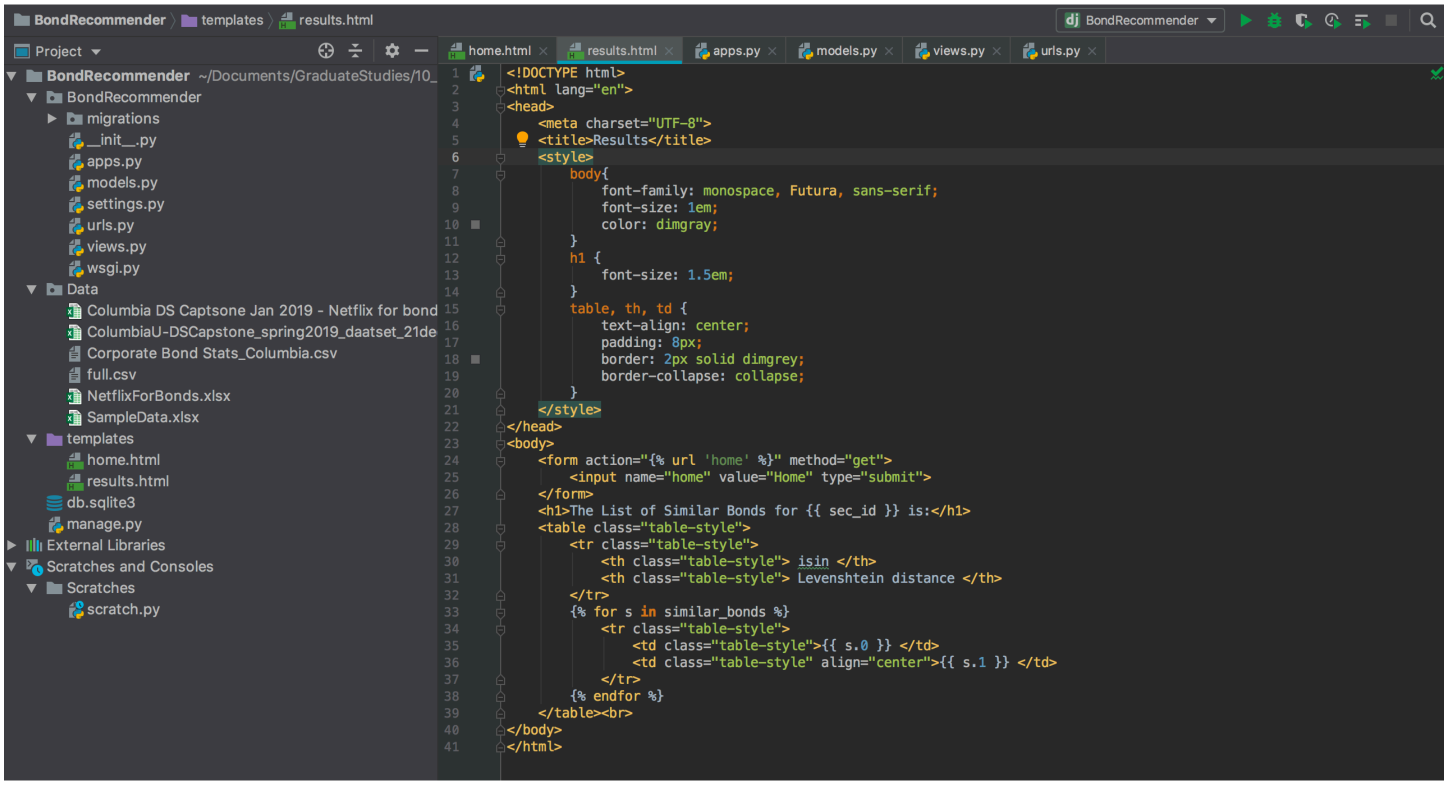
Templates (View)

Home.html



```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>home</title>
    <style>
        body{
            font-family: monospace, sans-serif;
            font-size: 1em;
            color: dimgray;
        }
        h1 {
            font-size: 1.5em;
        }
        h2 {
            font-size: 1.2em;
            color: red;
        }
    </style>
</head>
<body>
<h1>Bond Recommender</h1>
{% if sec_id_error == 1 %}
    <h2>{{ sec_id }} is not a valid ISIN. Enter a valid ISIN.</h2>
{% endif %}
<form action="{% url 'results' %}" method="get">
    Enter an ISIN:
    <input id="sec_id" name="sec_id" value="" size="15">
    <input name="submit" value="Get Similar Bonds" type="submit">
</form>
</body>
</html>
```

Results.html



```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Results</title>
    <style>
        body{
            font-family: monospace, Futura, sans-serif;
            font-size: 1em;
            color: dimgray;
        }
        h1 {
            font-size: 1.5em;
        }
        table, th, td {
            text-align: center;
            padding: 8px;
            border: 2px solid dimgrey;
            border-collapse: collapse;
        }
    </style>
</head>
<body>
    <form action="{% url 'home' %}" method="get">
        <input name="home" value="Home" type="submit">
    </form>
    <h1>The List of Similar Bonds for {{ sec_id }} is:</h1>
    <table class="table-style">
        <tr class="table-style">
            <th class="table-style"> isin </th>
            <th class="table-style"> Levenshtein distance </th>
        </tr>
        {% for s in similar_bonds %}
            <tr class="table-style">
                <td class="table-style">{{ s.0 }} </td>
                <td class="table-style" align="center">{{ s.1 }} </td>
            </tr>
        {% endfor %}
    </table><br>
</body>
</html>
```

Reference:
Course material from Python Web Application Programming
Django documentation: https://docs.djangoproject.com/en/2.1/