DEPLOY A BERT QUESTION ANSWERING BOT ON DJANGO
On Windows
Based on the Coursera project
Starter steps:

path> mkdir wikibert && cd wikibert
path/wikibert> pipenv install django
path/wikibert> pipenv shell
path/wikibert> django-admin startproject bertbot .
(shell) path/wikibert>python manage.py runserver
<exit>

(shell) path/wikibert> python manage.py startapp main
(shell) path/wikibert> code .

bertbot/urls.py:

```
from django.contrib import admin
from django.urls import path
from django.urls import include

urlpatterns = [
    path('/', include('main.urls')),
    path('admin/', admin.site.urls),
]
```

(shell) path/wikibert> touch main/urls.py
main/urls.py

```
from django.urls import path
import main.views as views

urlpatterns = [
    # path('', views.index, name='index'),
    # path('history', views.history),
]
```

Register app:
bertbot/settings.py

```
INSTALLED_APPS = [
```

```
    'main.apps.MainConfig',...]
```

Question model: main/models.py

```
from django.db import models

# Create your models here.
class Question(models.Model):
    # five fields
    question = models.TextField(null=False)
    wiki_terms = models.CharField(max_length=200, null=False)
    wiki_text = models.TextField(null=False)
    answer = models.CharField(max_length=200)
    prediction_score = models.FloatField(default=0)
```

Migrate:
(shell) path/wikibert> python manage.py makemigrations main
(shell) path/wikibert> python manage.py sqlmigrate main 0001
(shell) path/wikibert> python manage.py migrate

Views:
main/views.py

```
from django.shortcuts import render
from django.conf import settings
from .models import Question
from .forms import QuestionForm
import wikipediaapi

def index(request):
    if request.method == 'POST':
        form = QuestionForm(request.POST)
        if form.is_valid():
            try:
                question = form.cleaned_data['question']
                wiki_terms = form.cleaned_data['wiki_terms']
                answer = ''
            except:
                answer = "There was an error!"
            return render(
                request, 'main/index.html',
```

```
                {
                    'form':form,
                    'answer': answer,
                    'score': prediction_score
                }
            )
        else:  # GET = fresh form, no data
            form = QuestionForm()  # new form
            return render(request, 'main/index.html', {'form':form})
```

(shell) path/wikibert> touch main/forms.py
main/forms.py

```
from django import forms
from .models import Question

class QuestionForm(forms.ModelForm):
    class Metha:
        model = Question
        fields = ['Question', 'wiki_terms']
```

Using wikipedia api:
(shell) path/wikibert> pipenv install wikipedia-api tensorflow pandas matplotlib numpy transformers

main/views.py

```
try:
                question = form.cleaned_data['question']
                wiki_terms = form.cleaned_data['wiki_terms']

                wiki = wikipediaapi.Wikipedia("en")
                wiki_text = wiki.page(wiki_terms).summary
                answer = ''
```

add this.

** test wikipedia api:
> pip install wikipedia-api
> import wikipediaapi
> wiki = wikipediaapi.Wikipedia("en")

> wiki.page("french revolution").summary

> pip install transformers
> from transformers import pipeline
> b = pipeline("question-answering")
> text = "Jon is 5 years old"
> q = "How old is Jon"
> b(question=q, context=text)

Put pipeline in settings.py for quick use
bertbot/settings.py

```
ALLOWED_HOSTS = []

from transformers import pipeline
BERT_PIPELINE = pipeline("question-answering")

# Application definition

INSTALLED_APPS = [
...
```

main/views.py

```
 # data from wikipedia
                wiki = wikipediaapi.Wikipedia("en")
                wiki_text = wiki.page(wiki_terms).summary

                # context is this data
                result = settings.BERT_PIPELINE(
                    question=question,
                    context=wiki_text
                )
                answer = ''
                prediction_score = result['prediction_score']

                q = Question()
                q.wiki_terms = wiki_terms
                q.wiki_text = wiki_text
                q.question = question
```

```
                q.answer = answer
                q.prediction_score = prediction_score
                q.save()  # saves into database
```

create templates:
(shell) path/wikibert>mkdir main/templates
(shell) path/wikibert>mkdir main/templates/main
(shell) path/wikibert>touch main/templates/main/index.html

main/templates/main/index.html

```
<html>
    <head>
        <title>Ask BERTBot</title>
    </head>
    <body>
        <form action="{% url 'index' %}" method="post">
            {% csrf_token %} {{form.as_p}}
            <input type="submit" value="Submit">
        </form>
        <p>The answer is: {{answer}}.</p>
        <p>The prediction score is: {{score}}.</p>
    </body>
</html>
```

get history:

main/views.py

```
def history(request):
    q = Question.objects.all()
    return render(request, 'main/history.html', {'questions':q})
```

(shell) path/wikibert>touch main/templates/main/history.html
main/templates/main/history.html

```
<html>
    <head>
        <title>History of question/answers</title>
    </head>
    <body>
```

```
        {% for q in questions %}
        <h1>{{q.question}}</h1>
        <p>Terms: {{q.wiki_terms}}</p>
        <p>Context: {{q.wiki_text}}</p>
        <p>Answer: {{q.answer}}</p>
        <p>Score: {{q.prediction_score}}</p>
        {% endfor %}
    </body>
</html>
```

bertbot/urls.py

```
urlpatterns = [
    path('', include('main.urls')),
    path('admin/', admin.site.urls),
]
```