Table of Contents

[I. Introduction to Django 3](#_Toc120902608)

[II. Tools Setup 3](#_Toc120902609)

[1. Download django 3](#_Toc120902610)

[2. Create Django project and Django app 3](#_Toc120902611)

[3. Project Structure 3](#_Toc120902612)

[III. Code Development 5](#_Toc120902613)

[1. Database Setup and CRUD operations function definition 5](#_Toc120902614)

[2. Create URLs 6](#_Toc120902615)

[3. Create html templates 6](#_Toc120902616)

[4. Explanation of html format used to design UI 7](#_Toc120902617)

[5. Explanation of Django APIs used in development of CRUD operations 7](#_Toc120902618)

[IV. Execution and Testing 9](#_Toc120902619)

[1. Run server 9](#_Toc120902620)

[V. References: 12](#_Toc120902621)

1. Introduction to Django

In the mongoDB website(https://www.mongodb.com/compatibility/mongodb-and-django), it mentions django as the recommended web framework to build applications working with MongoDB databases. In this project we Django as the development web server together with pymongo for writing code to do CRUD operations on our database.

Django puts data in requests. We call it POST when a user submits a form with the attribute method=”post”

1. Tools Setup
2. Download django

**pip install pymongo**

**pip install dnspython**

1. Create Django project and Django app

django-admin startproject proj\_1

cd proj\_1

python manage.py startapp djangoapp

pip list

Make sure to have the same versions as the list below if not make sure to install with those specific versions.

***Verify Output from pip list:***

***pip 22.3.1***

***pymongo 4.3.3***

***dnspython 2.2.1***

***Django 4.1.3***

1. Project Structure

Text

Description automatically generated

Default files generated when creating a Django application are as follows:

settings.py - a normal Python module with module-level variables representing Django settings.

urls.py – define url paths and corresponding function to call

views.py – define views or so called

admin.py – not utilized (more likely used with djongo)

apps.py - not utilized (more likely used with djongo)

models.py - not utilized (more likely used with djongo)

tests.py – not utilized

1. Code Development
2. Adding our app for list of activated apps in the Django instance when it is run

Need to add the djangoapp in the installed apps so theat in the future the urls will be recognized.

We are using pymongo not djongo hence need to comment out the DATABASES declaration in the settings.

Text

Description automatically generated

1. Database Setup and CRUD operations function definition

Database setup is instead done on the views.py where we create the connection string, get the client from there and the instance of our database.

It is also in the views where we create the function definitions for each collection.

A screenshot of a computer

Description automatically generated with medium confidence

1. Create URLs

We assign different URLs accordingly to CRUD operation for each collection

Text

Description automatically generated

1. Create html templates

For our Gui, we create a new folder named templates to which we group the html files corresponding to the design of each page url

Text

Description automatically generated

1. Explanation of html format used to design UI
   1. <explain tags here>

<form method="POST" class="post-form" action="/customer/add">

we have an HTML form so that user can submit data and handle it in the view.

* 1. Improve Design through bootstrap

[Bootstrap · The most popular HTML, CSS, and JS library in the world. (getbootstrap.com)](https://getbootstrap.com/)

<!-- CSS only -->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-rbsA2VBKQhggwzxH7pPCaAqO46MgnOM80zW1RWuH61DGLwZJEdK2Kadq2F9CUG65" crossorigin="anonymous">

1. Explanation of Django APIs used in development of CRUD operations
   1. CREATE

In views we define the functions that will handle the data submitted from the POST in html.

Based on the corresponding collection html page from where POST came from we build the information and assign the value from the form

1. if request.method == "POST":
2. form = request.POST
3. print(form)
4. customer\_info = {
5. 'emailAddress' : form['emailAddress'],
6. 'firstName' : form['firstName'],
7. 'lastName' : form['lastName'],
8. 'billingAddress' :
9. {
10. 'bStreet' : form['bStreet'],
11. 'bUnit' : form['bUnit'],
12. 'bCity' : form['bCity'],
13. 'bProvince' : form['bProvince'],
14. 'bZip' : form['bZip'],
15. 'bMobileNumber' : form['bMobileNumber']
16. },
17. 'deliveryAddress' :
18. {
19. 'dStreet' : form['dStreet'],
20. 'dUnit' : form['dUnit'],
21. 'dCity' : form['dCity'],
22. 'dProvince' : form['dProvince'],
23. 'dZip' : form['dZip'],
24. 'dMobileNumber' : form['dMobileNumber']
25. },
26. 'paymentMethod' :
27. {
28. 'paymentType' : form['paymentType'],
29. 'nameOnCard' : form['nameOnCard'],
30. 'cardScheme' : form['cardScheme'],
31. 'cardLast4' : form['cardLast4'],
32. 'cardExpiryDate' : form['cardExpiryDate']
33. }
34. }

We then use .insert one to insert the new document onto the given collection.

        print(customer\_info)

        myCollection\_Customers.insert\_one(customer\_info)

After insertion, we want the user to go the the main page that shows the whole collection

        return redirect('/customer/show')

1. Execution and Testing
2. Run server

Python manage.py runserver

Text

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Table

Description automatically generated with medium confidence

Graphical user interface

Description automatically generated

1. References:

<https://www.youtube.com/watch?v=vNegKfkopoQ>

<https://docs.djangoproject.com/en/4.1/ref/models/options/#table-names>

<https://www.youtube.com/watch?v=FLZYLog369s>

<https://code.djangoproject.com/wiki/NoSqlSupport>

<https://www.youtube.com/watch?v=sjMlffIVGf8>

<https://www.youtube.com/watch?v=c45JJfgMyWo>

<https://www.youtube.com/watch?v=I17uA1sVQ2g>

<https://www.digitalocean.com/community/tutorials/how-to-connect-a-django-app-to-mongodb-with-pymongo>

<https://www.mongodb.com/compatibility/mongodb-and-django>

<https://www.mongodb.com/compatibility/mongodb-and-django>

https://www.javatpoint.com/django-crud-application