Usman Zia #24087

24087@student.dorset-college.ie

https://github.com/flast8021/CA1Project24087

Geoff Wright

CA1 Report

This application is created to perform CRUD functions. this is a application to be used by libraries or book stores in order to store book names and availability, Which is done using web site and the application itself would be able to handle all the four functions including create read update and delete functionality of my SQL database.

Procedure:

In order to start with this project first of all I used following commands to start the project

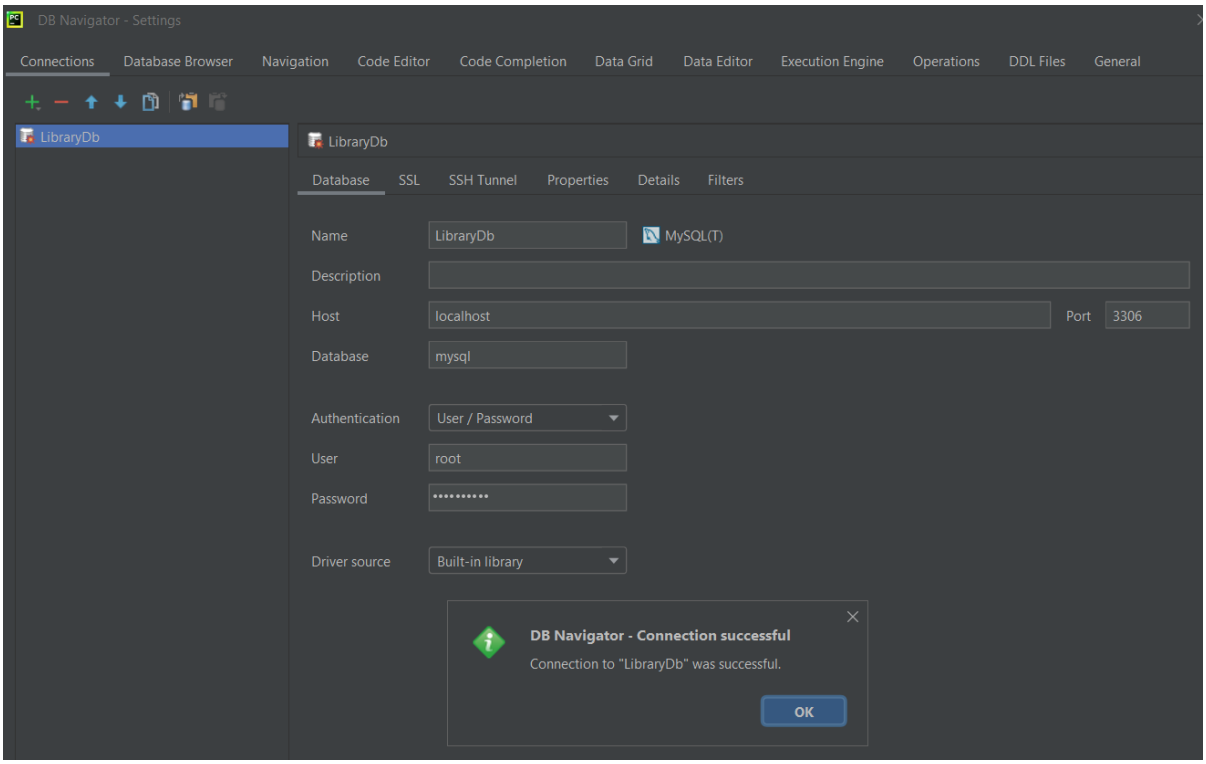
Cd desktop

Mkdir CA1Project

Django-admin startproject library

Py manage.py startapp booksData

Then after this I added path to python interpreter, In order for this project to work then I created a database connection to store the books data in our application, database configuration was done



Database configuration settings in settings.py file of our project are as follows

```
# Configuring personal database
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'libraryDb',
        'USER':'root',
        'PASSWORD':'Usman.8021',
        'HOST':'localhost',
    }
}
```

After this I created a model and updated models.py file of our app and the configurations can be seen below

Usman Zia #24087

24087@student.dorset-college.ie

https://github.com/flast8021/CA1Project24087

Geoff Wright

CA1 Report

```python
from django.db import models

# Creating models here to add fields and variable for our online library bookData application.

class Library(models.Model):
    # variable, variable type and thier length etc.
    bookId = models.CharField(max_length=20)
    bookName = models.CharField(max_length=100)
    bookAvailability = models.BooleanField()
    #to make our DB table (library)
    class Meta:
        db_table = "library"
```

Then the next thing I did was to write a view in view.py file of our application and the configuration can be seen in the project, as it's a very long file so instead of adding multiple screenshot I have referenced them GitHub link so the file can be accessed easily.[1]

After I was done with the views I decided to go and give the routing to pages in the urls.py file of the project, url configuration can be seen in screenshot below

```python
from django.contrib import admin
from django.urls import path
from booksData import views

#urls to access our online application and to perform CURD operations
urlpatterns = [
    path('admin/', admin.site.urls),
    path('book', views.book),
    path('show',views.show),
    path('edit/<int:id>', views.edit),
    path('update/<int:id>', views.update),
    path('delete/<int:id>', views.destroy),
]
```

After configuring URLs file the next step I considered was to organise the templates so I created three HTML files

1. Index.html
2. show.htlml

[1] https://github.com/flast8021/CA1Project24087/blob/master/booksData/views.py

3. Edit.html

The files mentioned above was created in a new directory added in the project folder named as templates which carry these three files so that user can interact with the application an application can be responsive I also created another and directory called as static which has style.css file but for some reasons it didn't worked and due to lack to time it couldn't be sorted.

moving on to the next step that I took in the process was to migrate which created multiple tables in the database including session authentication etc then I also used make migrations command both of these commands are as follow

Py manage.py migrate

Py manage.py makemigrations

after this I was able to have a working online application and was able to connect to correct database.

The URL accessible links are as follows the first link shows us the number of books we have, their unique Id's and if it's available or not, It can be updated using the third link which will also be functional when we are in the /show url and we choose added option so it will take us to link 3 with the **/edit/** book ID at the end

Url link:

127.0.0.1:8000/show

127.0.0.1:8000/book

127.0.0.1:8000/edit/(book-Id)

To sum up super user is also created with credentials as follows

Username: admin

Password: admin

Just above are credentials for super user to make changes and work true as an admin and to conclude I was able to build a proper working online application which is being hosted in the local PC and works as expected and required.[i]

*** End of Project***

[i] https://docs.djangoproject.com/en/4.1/intro/tutorial03/
https://stackoverflow.com/questions/42888366/error-in-loading-static-files-in-django-project