Products Backend Framework

Introduction

Backend development is typically the first step after the planning phase of the creation of a new application. The frontend application is dependent on the backend server's data and functionality. This is an exercise of building a backend Web API prior to the start of the frontend of the application. We will be testing and interacting with this backend Web API using Postman to ensure everything is working as expected. This will also help us visualize what the structure of the data will be that the frontend will be consuming.

Learning Objectives

Understand how to build a Web API for a full-stack application with database integration.

Task

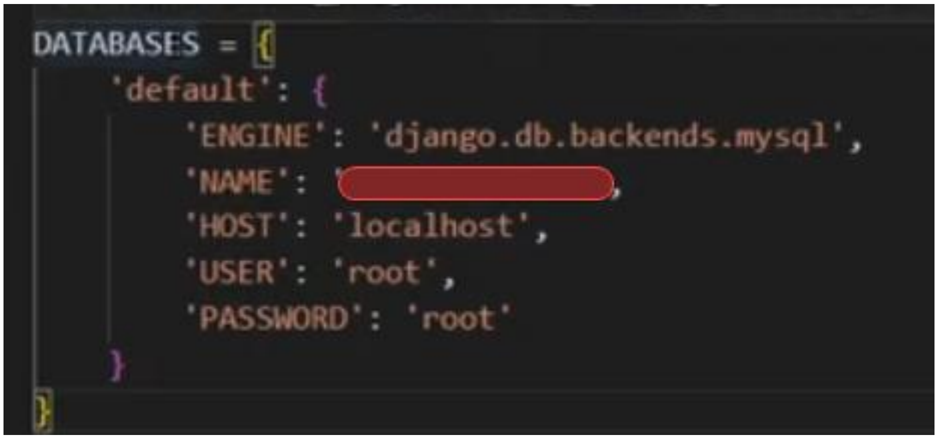
Create a Django REST API connected to a MySQL database allowing a frontend application to POST comments and replies to the MySQL database relative to the video the user is currently watching.

Setup Steps

1. Make a GitHub Repository (** with Python gitignore **)
2. Clone down repository to local computer
3. Open folder in VS code and create/activate a local venv
    a. "pipenv install"
    b. "pipenv shell"
4. Select the correct Python interpreter for the project
5. Install all necessary packages
    a. "pipenv install django"
    b. "pipenv install djangorestframework"
    c. **WINDOWS**- "pipenv install mysqlclient"
    d. **MAC** - "pipenv install mysql-connector-python==8.0.26"
6. Create an initial Django project
    a. "django-admin startproject products_project ."

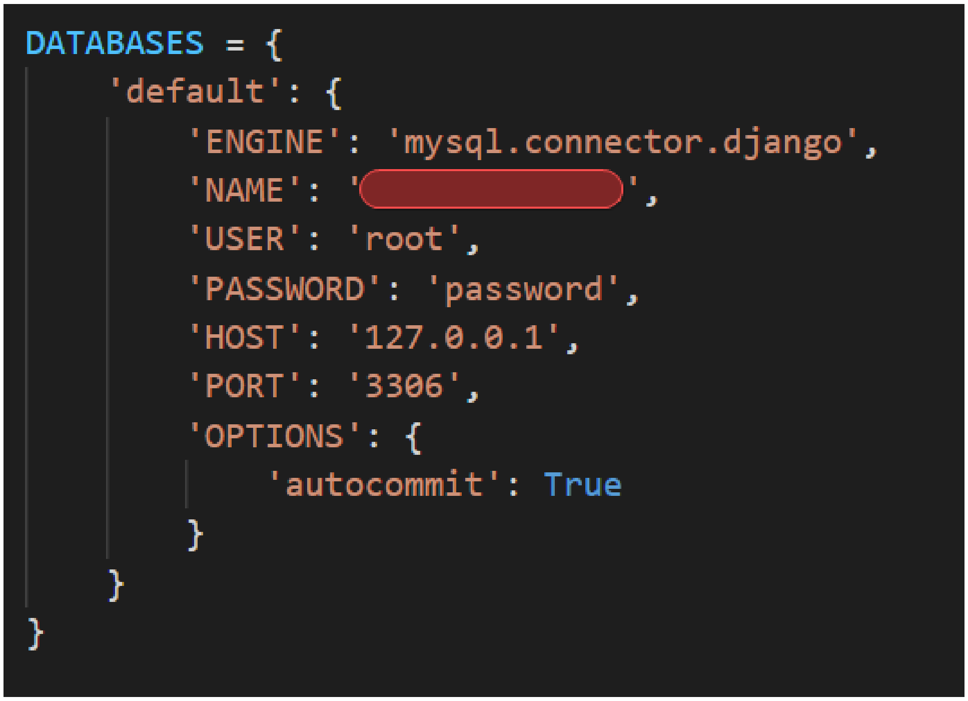
7. Create a `local_settings.py` file and import it into your settings.py file to prevent your settings from being pushed to the public repository.
8. Cut & Paste DATABASES and SECRET_KEY from settings.py to local_settings.py. Change DATABASES to reflect the appropriate database NAME, ENGINE,USERNAME,PASSWORD, etc.

Windows:



```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': '[redacted],
        'HOST': 'localhost',
        'USER': 'root',
        'PASSWORD': 'root'
    }
}
```

Mac:



```
DATABASES = {
    'default': {
        'ENGINE': 'mysql.connector.django',
        'NAME': '[redacted]',
        'USER': 'root',
        'PASSWORD': 'password',
        'HOST': '127.0.0.1',
        'PORT': '3306',
        'OPTIONS': {
            'autocommit': True
        }
    }
}
```

9. Push project to GitHub repo.
10. Create database in MySQL Workbench
11. Execute migrations commands
    a. “python manage.py migrate”
12. Create a new app for the products
    a. “python manage.py startapp products”

13. Follow user stories and continually refer back to video tutorial series while completing and testing your view functions!

End Result

The result of your Products API backend will be the execution of requests made in Postman.

You must test your Products API by executing each request you create in Postman.