

**Web application development**

**Student: Almabekuly Aibek**

**Date of submission: 10.11.2024**

**Git: https://github.com/Aibek2201/Web\_app\_dev\_MD.git**

Almaty, 2024

**Table of Contents**

1. **Executive Summary……………………………………………………….…3**
2. **Introduction .................................................................................................... 3**
3. **Code Snippets .............................................................................................….4**
4. **Results ..............................................................................................................7**
5. **Conclusion ...................................................................................................... 10**
6. **References ........................................................................................................10**
7. **Executive Summary**

This project focuses on developing a functional blog application using the Django web framework. The blog project serves as an exploration of core Django components Models, Views, and Templates which are fundamental for creating dynamic, database-driven web applications. This assignment requires implementing these elements, documenting the development process, and submitting a report and code repository.

1. **Introduction**

This project focuses on building a fully functional blog application using the Django web framework, a popular tool for creating robust, scalable web applications. Django is based on the Model-View-Template architectural pattern, which separates an application’s data, user interface, and business logic, making it an efficient and structured framework for web development. This project serves as an in-depth exercise in implementing core components of Django Models, Views, and Templates which form the backbone of any Django application. In modern web development, applications are expected to handle complex data relationships, provide an intuitive interface for users, and offer dynamic content that can be easily managed. With Django, developers can quickly set up these capabilities through reusable and maintainable code. This project was designed to explore these functionalities in a structured way, following a series of exercises to develop a simple but feature-rich blog application. The application allows users to view, categorize, and comment on blog posts, offering a practical example of Django’s power and flexibility.

The blog project not only highlights the importance of each component within Django’s MVT architecture but also demonstrates how they work together to create a dynamic, user-friendly web application. Through this assignment, we gain practical experience with essential Django features, reinforcing concepts like model relationships, view logic, form handling, and template rendering. This knowledge is invaluable for developing more complex and interactive applications in Django, providing a strong foundation for future projects.

1. **Code snippets**

To organize this project firstly I created the main base in Django using default commands *django-admin startproject*. And I have been base of my project. We can see the main apps of my project in figure 1.

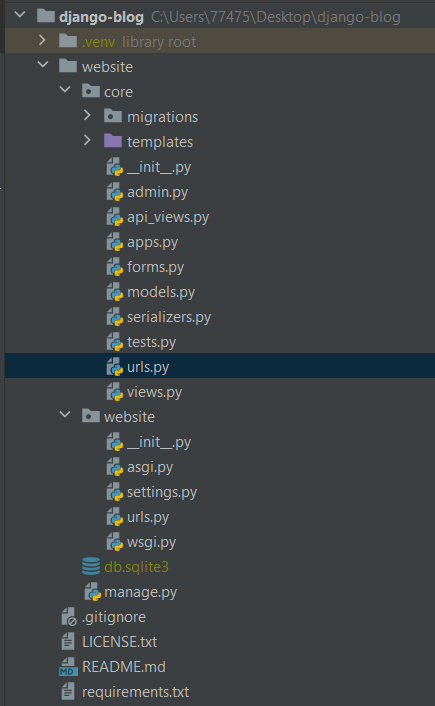


Figure 1. Architecture of blog project

After creating base for project, I started to arrange migration records to database, and created Models in Django inside of models.py. In the next figures we can see models which covered our project.

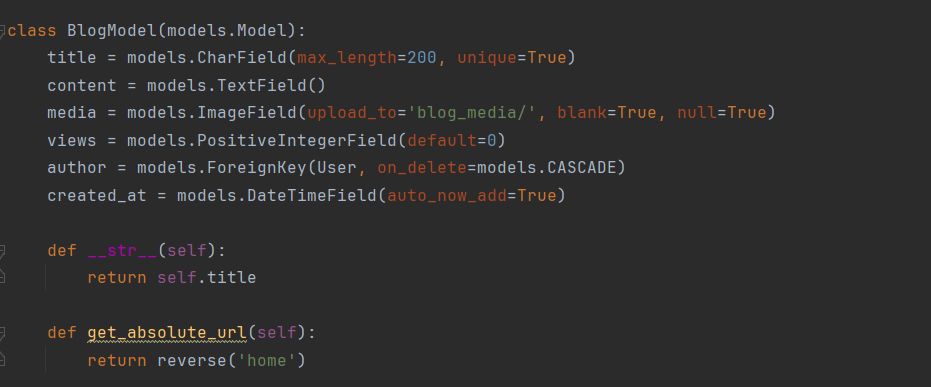


Figure 2. Model for blogs

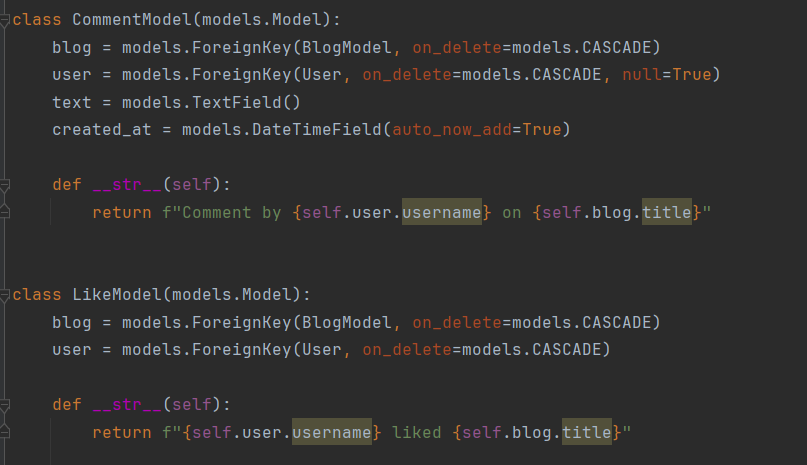


Figure 3. Model for comments and likes

After that I added views to arrange API connections between frontend and backend. This arranging we can see in the next figure.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Figure 4. Core urlpatterns

After arranging core urlpatterns, I created also one file for urlpatterns but in this case I put in inside of app. Here I wrote all of url addresses which related with this application.

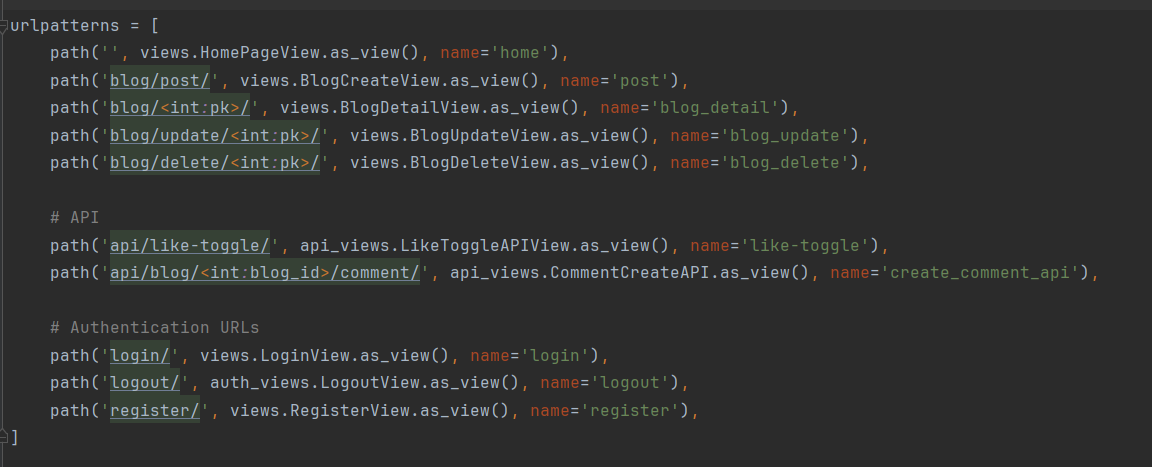


Figure 5. URL addresses related with application

I did a lot of works to arrange blog project, but still it’s not enough to see project in the browser, my next step was arrange views file to request data to frontend side. In this case I used Class based views file. What does it mean? Using this type of views I can put a lot of functions inside of class, and it can be so scalability. I will show several class based views which I used in my project.

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Figure 6. Class based views for login and register page

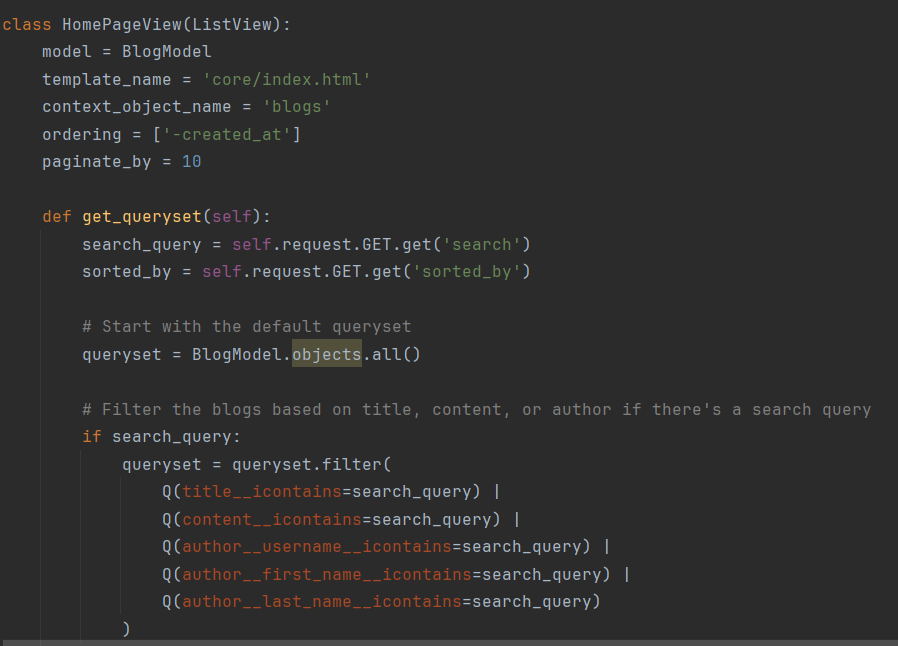


Figure 7. Class based Homepage view

Here we finished backend side of our project. Frontend side we can see in the result chapter.

1. **Results**

In this chapter we can see how works blog project, and how organized frontend side. Firstly I created templates file.

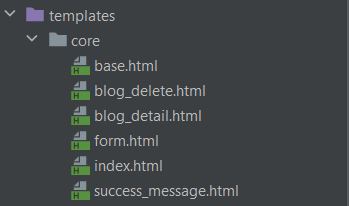


Figure 8. Architecture of template files

Here we can see base.html file. I used this file in this project like basement. Usign this file we won’t write same code in each file. After organizing base file, I created some files. For instance we can see code in the blog delete file which used base.html file.

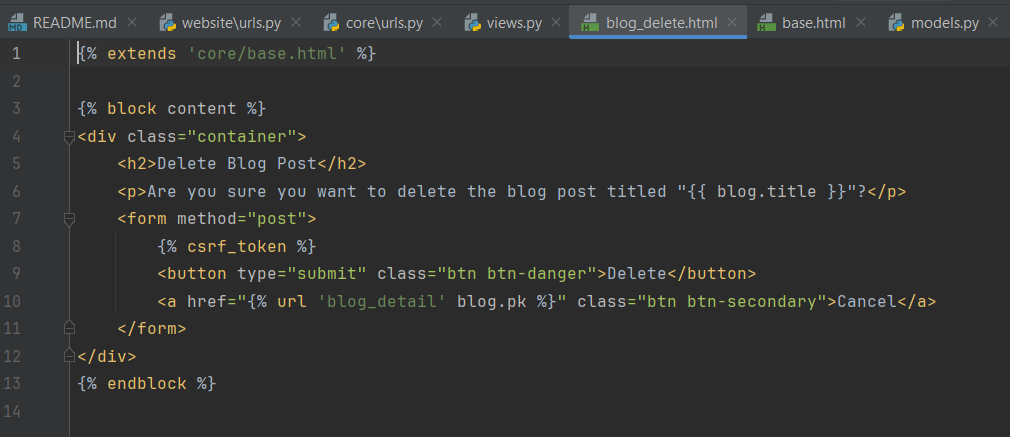
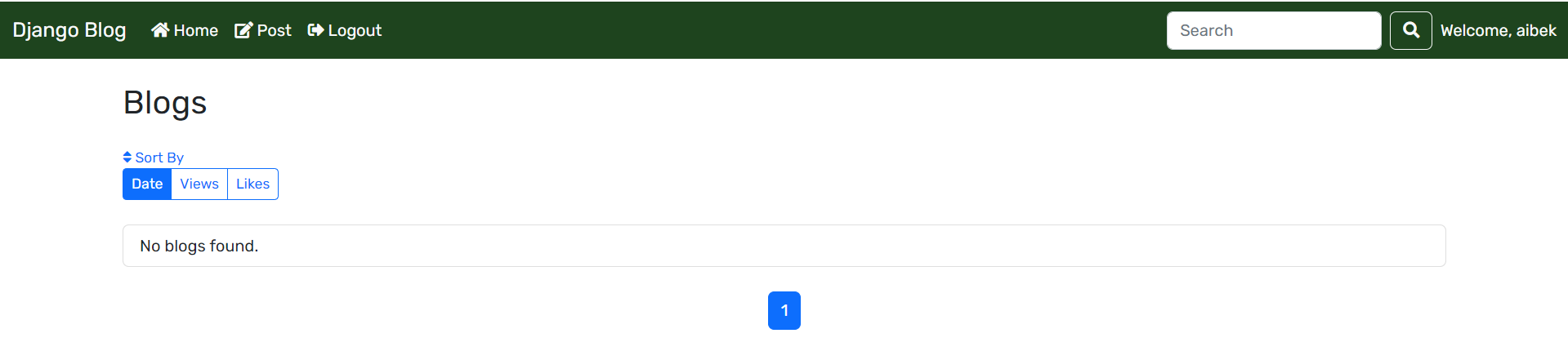


Figure 9. Blog delete file using base.html

Lets see how works this project in the browser.

Figure 10. Main page of project

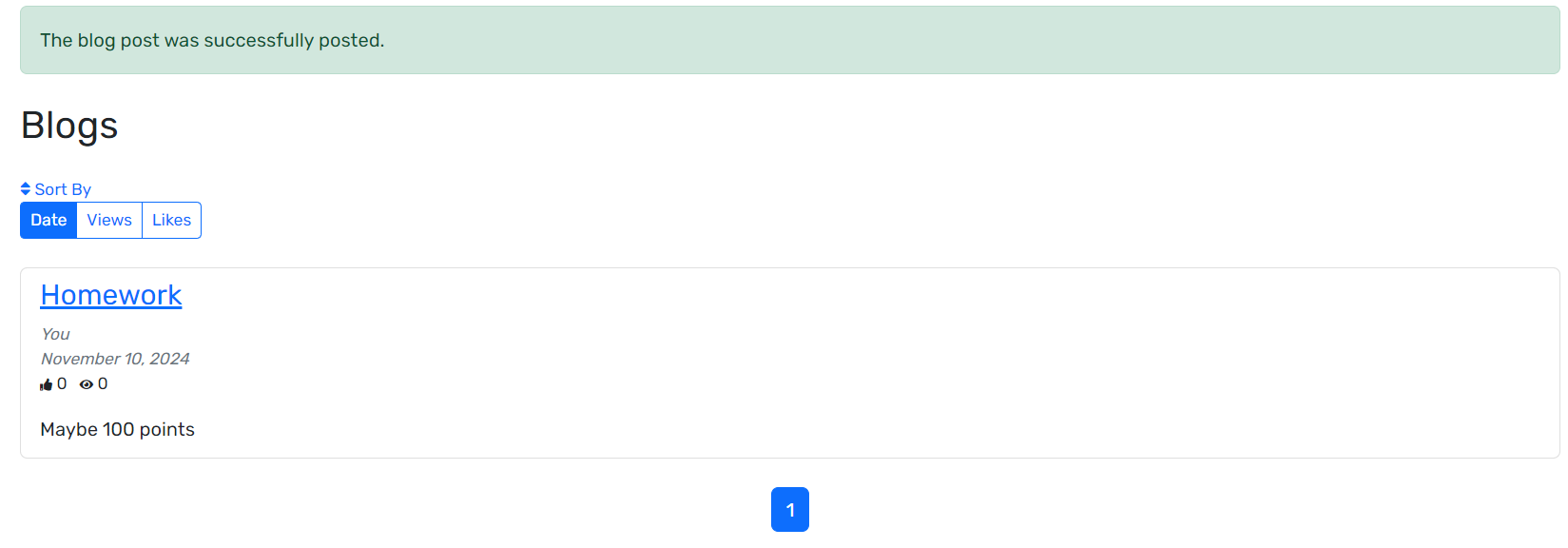


Figure 11. Page after creating blog with dialog window

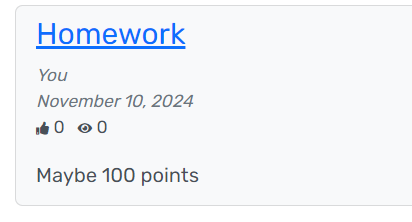


Figure 12. Content details with likes and views

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание

Figure 13. Blog records with sorting

1. **Conclusion**

This Django blog project has been a comprehensive exploration of the fundamental components involved in developing a modern web application. By implementing models, views, and templates, we have gained hands-on experience with Django’s Model-View-Template architecture, which is essential for creating structured, maintainable, and efficient web applications.

Overall this project has reinforced the significance of the MVT components in Django. By working with models, views, and templates, we have acquired a deep understanding of how these elements work together to produce a complete, functional web application. This foundational knowledge is essential for building more advanced applications in the future, equipping us with the skills to design and implement sophisticated, scalable solutions with Django.

1. **References**
2. Docker Documentation: https://docs.docker.com
3. Django Documentation: <https://docs.djangoproject.com>
4. Official Python Docker Image: https://hub.docker.com/\_/python