Django Take-Home Coding Assignment

Overview

In this assignment, you will build a simple Django project that models products, categories, and tags. You'll set up the appropriate relationships between these models, populate the database with sample data using the Django admin interface, and create a basic HTML page that allows users to search and filter products based on description, category, and tags. The main goal is to demonstrate your ability to work with Django models, querysets, and views to build functional features.

Objectives

- Create Django models with correct relationships.
- Populate the database using Django's admin interface.
- Implement search and filter functionality.
- Build an HTML page for user interaction.
- Demonstrate proficiency in writing Django queries.

Requirements

Data Population

- Use the Django admin interface to populate the database with sample data.
- Create at least **5 categories**, **10 tags**, and **20 products**.

Search and Filter Functionality

- Create a simple HTML page that allows users to:
    - **Search** products by description.
    - **Filter** products by category.
    - **Filter** products by tags.
- Users should be able to combine search and filter options.

Front-End

- You may use Django templates or any front-end framework of your choice.
- The design and styling are **not** important.
- The focus is on functionality and query implementation.

Deliverables

- A complete Django project with all source code.
- A `README.md` file that includes:
  - Instructions on how to set up and run the project.
  - Any assumptions or additional notes.
- If using a front-end framework, include build instructions.

Submission Guidelines

- Upload your project to a Git repository (e.g., GitHub, GitLab).
- Ensure all files are committed and pushed to the repository.
- Provide the repository link for review.

Evaluation Criteria

- **Functionality**: Correct implementation of models, relationships, and querying.
- **Code Quality**: Cleanliness, readability, and organization of code.
- **Documentation**: Clarity of setup instructions and comments within the code.
- **Query Proficiency**: Effective use of Django querysets to achieve required functionality.
- **Completion**: Fulfillment of all specified requirements.

AI Policy

Through this assignment, we aim to evaluate your non-AI-assisted technical skills. While AI tools (such as ChatGPT, Copilot, or similar) are **permitted** for assistance, they should not replace your own problem-solving and coding abilities. If you use any AI tools, the following conditions must be met:

1. **Understanding and Ownership**: You must **fully understand** any AI-generated code that you include in your submission. You will be asked questions about your implementation, and inability to explain your code will result in disqualification.
2. **Enhancement and Originality**: AI-generated code must be **modified and expanded** upon. Submitting raw AI-generated code without significant personal contributions will not be accepted and will result in immediate disqualification.
3. **Attribution**: If you use AI to generate any part of the code, clearly document how and where it was used in your `README.md` file. You can also provide attribution as in-line comments or docstrings.

By submitting this assignment, you acknowledge that you understand and adhere to this AI policy.

Note: Styling and visual design are not the focus of this assignment. The primary goal is to assess your ability to work with Django models, querysets, and views to build functional features.

If you have any questions or need clarification on any aspect of this assignment, please feel free to reach out.