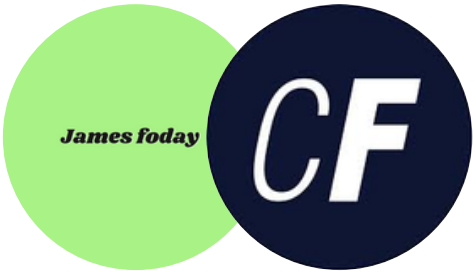
JAMES FODAY



REFLECTION QUESTION

Exercise 2.2 Reflection Questions and Answers

1. Suppose you're in an interview. The interviewer gives you their company's website as an example, asking you to convert the website and its different parts into Django terms. How would you proceed? For this question, you can think about your dream company and look at their website for reference.

Answer:
I would start by analyzing the company's website to identify its key components and functionalities. Then, I would break the website into distinct Django apps based on the features — for example, a user authentication app, a products app, a blog app, and a payments app. Each app would handle a specific domain of functionality, following Django's modular structure. The overall site would be the Django project containing these apps. This approach ensures code reusability and maintainability. Additionally, I would define models to represent the data, views to handle the business logic, and templates for rendering the user interface, all coordinated through URLs configured in the project's routing.

2. In your own words, describe the steps you would take to deploy a basic Django application locally on your system.

Answer:
To deploy a basic Django application locally, I would first ensure I have Python and a virtual environment set up. Next, I would create and activate a virtual environment to keep dependencies isolated. Then, I would install Django inside this environment. After that, I would use django-admin startproject to create a new Django project, navigate into the project folder, and run python manage.py migrate to apply initial database migrations. Then, I would start the development server using python manage.py runserver and access the application via the local URL provided. Finally, I would create a superuser with python manage.py createsuperuser to access the Django admin interface.

3. Do some research about the Django admin site and write down how you'd use it during your web application development.

Answer:
The Django admin site is an automatically generated, built-in interface for managing a Django application's data models. During development, I would use it to quickly add, edit, and delete database records without writing separate user interfaces. The admin site also supports user authentication and permission management, allowing different levels of access for users. It helps streamline development by providing an out-of-the-box CRUD interface, which accelerates testing and management of data. Moreover, the admin site can be customized to fit the specific needs of the application, making it a powerful tool for backend administration.