Exercise 2.1:

1. Why Django is so popular among web developers

Django is loved because it offers a “batteries-included” experience—authentication, routing, ORM, and admin panel all built in—making development much faster and more efficient. It also excels at handling large-scale traffic and offers strong security features out of the box.

2. Five large companies using Django

- Instagram – A social media platform for sharing photos and stories. They use Django to scale and manage billions of user interactions.

- Spotify – A global music-streaming service offering playlists and podcasts. They use Django for backend services including user authentication and playlist management.

- Mozilla – The organization behind Firefox, using Django for various web properties and API services thanks to its scalability and security.

- Pinterest – A visual discovery platform with millions of pins and boards. They chose Django for its flexibility and ability to handle massive data loads.

- Bitbucket – A code hosting and Git repository service by Atlassian. Django powers the backend infrastructure for collaborative code workflow

3. For each of the following scenarios, explain if you would use Django (and why or why not):

- **You need to develop a web application with multiple users.**
  Yes, Django is a great choice because it includes built-in user authentication, permissions, and session management, making it easy to handle multiple users securely and efficiently.
- **You need fast deployment and the ability to make changes as you proceed.**
  Yes, Django’s framework allows for rapid development. Its built-in tools and clear structure make it easy to update and modify features quickly without breaking the entire application.
- **You need to build a very basic application, which doesn’t require any database access or file operations.**
  No, Django might be overkill for something simple. A lightweight framework like Flask would be better for small projects that don’t need a database or complex structure.

- **You want to build an application from scratch and want a lot of control over how it works.**
  No, Django's structure can feel restrictive. If you want more freedom and customization, Flask or FastAPI would give you more flexibility to design everything from the ground up.
- **You're about to start working on a big project and are afraid of getting stuck and needing additional support.**
  Yes, Django is perfect for large projects. It has a huge community, extensive documentation, and many ready-made solutions that can help you avoid getting stuck and save development time.