1. Create a text document where you can write your answers to the following questions. You'll want to do some research online to help you develop your answers.
2. Write two to three sentences on why Django is so popular among web developers.

It offers a comprehensive and scalable framework that simplifies the development of complex, data-driven web applications.

Its built-in features, like the ORM, authentication, and admin interface, allow developers to focus on building features rather than reinventing the wheel.

Django's emphasis on security, scalability, and rapid development makes it a go-to choice for both small projects and large, enterprise-level applications.

3. After some research, list five large companies that use Django. Specify what the company's product or service is and what they use Django for.

1. Instagram:
   Product/Service: Social media platform for photo and video sharing.
   Use of Django: Instagram uses Django for its backend, handling millions of active users and enabling rapid development of new features.

2. Spotify:
   Product/Service: Music streaming service.
   Use of Django: Spotify uses Django for its web applications and APIs, supporting features like playlist management and user interactions.

3. Pinterest:
   Product/Service: Visual discovery and social media platform.
   Use of Django: Pinterest uses Django for its backend to manage content and handle user interactions efficiently.

4. Mozilla:
   Product/Service: Software company known for the Firefox web browser.
   Use of Django: Mozilla uses Django for various internal tools and systems, as well as parts of its website and developer platform.

5. Dropbox:
   Product/Service: Cloud storage and file-sharing service.
   Use of Django: Dropbox leverages Django for handling user interfaces, file handling, and portions of its web-based services.

4. 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.
   - You need fast deployment and the ability to make changes as you proceed.
   - You need to build a very basic application, which doesn’t require any database access or file operations.
   - You want to build an application from scratch and want a lot of control over how it works.
   - You’re about to start working on a big project and are afraid of getting stuck and needing additional support.

You need to develop a web application with multiple users.

- **Use Django:** Yes, because Django provides built-in support for 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.

- **Use Django:** Yes, because Django’s "batteries-included" approach with built-in features, combined with its support for rapid development, allows for quick deployment and iterative changes.

You need to build a very basic application, which doesn’t require any database access or file operations.

- **Not use Django:** No, because Django might be overkill for a basic application. A simpler framework or even plain Python with a lightweight framework like Flask might be more appropriate for such minimal requirements.

You want to build an application from scratch and want a lot of control over how it works.

- **Not use Django:** No, because Django's structure and conventions might limit the level of control you have. If you need granular control, a microframework like Flask or using vanilla Python would offer more flexibility.

You're about to start working on a big project and are afraid of getting stuck and needing additional support.

- **Use Django:** Yes, because Django has a large, active community, extensive documentation, and numerous third-party packages, which means you're more likely to find support and solutions if you encounter issues.