Achievement 2 – Exercise 2.1

1. Write two to three sentences on why Django is so popular among web developers.

   Django has a large community that is constantly updating. It has a loosely coupled architecture so you can add components and hardware throughout development. It also follows the DRY (Don't repeat yourself) principle therefore developers can write reusable code that is easier to understand.

2. 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.

   Instagram - a large platform that shares photos and videos. Django helps with scaling the app because you can add custom tools and having a language that is known for its simplicity can help reduce code complexity.

   Spotify - is an audio streaming provider of music and podcasts. Django can help build the back-end and be used for data analysis.

   Pinterest - is also an image sharing app for inspiration. They can use Django to efficiently use the MVT architecture to display the GUI and add any tools later.

   National geographic - is a TV network, a publisher of magazines, and articles. They use Django CMS to help manage their articles and catalogue of magazines on their website.

   Mozilla - is a search engine built on Django. They use Django to help with load balancing and API consumption.

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, because with multiple users you have to take into consideration auth, DB storage, and multiple pages

- You need fast deployment and the ability to make changes as you proceed
  Yes, Django could get you going faster with everything included and could update the tools with having to change everything with your app.

- You need to build a very basic application, which doesn't require any database access or file operations.
  If you are building something simple, you don't want a bunch of things you won't use on your app. Plain python would be the best option.

- You want to build an application from scratch and want a lot of control over how it works.
  Stand-alone python would work better in this situation. If you were to use Django you might lose some of the controls and understanding of why something is made a certain way.

- You're about to start working on a big project and are afraid of getting stuck and needing additional support.
  Django has a large community to help when you get stuck.