**Task 1 Postgresql - Reflection**

1. **What is the need for Add Ons in Heroku?**

Ans: Heroku add-os support data storage, monitoring, analytics and data processing. These are maintained by 3rd party. Add ons offer different plans with different features and capabilities.

1. **What exactly happens when you click on provision while configuring the Postgres addon?**

Ans: Provision while configuring the postgres addon facilitates us to find whether the app has Heroku postgres provisioned. Hence, we provision to hobby-dev plan and then the database will be connected.

1. **What is the use of Adminer? How does it work?**

Ans: Adminers are the web-based frontend to the Heroku space postgres. It helps in managing the contents of the database.

**Task 2 Python and Flask - Reflection**

1. **How do I manage to use python 3.6 if I already have python 2.7?**

Ans: By design, python installs to a directory with the version number embedded. So we can have multiple versions of python on the same system without conflicts. Or we can upgrade the lower version to a higher version.

1. **What is the role of pip and how does it work?**

Ans: Pip is a package management system used to install and manage software packages. Pip has different uses like installing a package, removing and upgrading a package, searching and checking the status of a package.

1. **What is the role of requirements.txt and how does it work with pip?**

Ans: requirements.txt is a text document that contains different python libraries or packages to run python code. We link them with pip after installations and set the environment accordingly.

1. **Which packages are installed and why are they required?**

Ans: Packages like flask, Flask-Session, psycopg2-binary and SQLAlchemy were installed for flask and database interaction using SQL. These wrap around the python database and facilitates interaction between python modules and databases.

1. **Which environment variables set for Flask to work? What is the purpose of each variable?**

Ans: The FLASK\_APP sets the path of the current working directory, a python file and an optional name instance. Here we are setting the environment flask path with the application.py file.

1. **What happens when the Flask run command is issued on the terminal?**

Ans: The flask executable is a simple command line runner for flask apps. This sets the name to main and when run is called, it will run the development server.

1. **On which port is Flask running and can it be changed?**

Ans: When request comes to the server on port 80 or port 443, a web server handles the connection of the request. Yes, we can change the port using export FLASK\_RUN\_PORT = new\_port\_number

1. **How is Flask different from the tiny web server?**

Ans: Flask app doesn’t actually run as a tiny web server would. It wouldn’t wait for requests and reacting to them. It can be seen as a function which is called by the application server.

**Task 3 Goodreads API - Reflection**

1. **What are the various categories of web APIs available on good reads?**

Ans: There are different APIs available regarding authorization, events search, follow unfollow a user, finding a group, getting reviews etc.

1. **Is there a limit on the use of the web API? What are the limits?**

Ans: General quota limits are 50,000 requests per project per day, which can be increased according to your opted plan.