**Task-1:**

**POST gressql reflection:**

**1) What is the need for Add-ons in Heroku?**

**Ans)** Heroku add-ons are components that support our application, such as data storage, monitoring, analytics, data processing, and more. These are fully maintained for you by either a third-party provider or by Heroku.

**2) What exactly happenes when you click on provision while configuring the Postgres addon?**

**Ans)** When we click on provision while configuring the postgres add-on, as part of the provision process it will allocate a database url for our application.

**3) What is use of Adminer? How does it work?**

**Ans)** Adminer is a database management tool that can be accessed via your web browser. It’s a singular file written in PHP which can be deployed wherever you need it just by moving the file to wherever you need it to be.

* First we need to install Adminer PHP file.
* Upload PHP file to our root web directory.
* We can either zip and upload our SQL file to directory, where we’ve uploaded adminer.php.
* Open web browser and access Adminer PHP file.
* Log into Adminer with your credentials. You’ll see a window similar to phpMyAdmin home window
* Here we can do all the necessary actions like dump, import, and create tables, run SQL commands etc.

**TASK-2:**

**Python and Flask – Reflection:**

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

**Ans)** First install python 3.6 in your system, then add python 3.6 path in system variables. Open command prompt and run command python3 -V . It will switch to python 3.6. If you want to switch back to python 2.7 then in command prompt just type python2 -V, it will automatically switches to python 2.7 version.

**2) What is the role of pip and how does it work?**

**Ans)** pip is a de facto standard package-management system used to install and manage software packages written in Python. Most distribution of python come with pip preinstalled. One major advantage of pip is the ease of its command-line interface. To install a package we just need to type pip install package-name in command-line.

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

**Ans)** The requirements.txt file is used for specifying what python packages are required to run the project you are looking at. pip has a feature to manage full lists of packages and corresponding version numbers, possible through a “requirements.txt” file. This permits the efficient recreation of an entire group of packages in a separate environment or virtual environment. This can be achieved by the following command

pip install –r requirements.txt

**4) Which packages are installed and why are they required?**

**Ans)**

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

**Ans)** By default flask\_env variable is set to production mode. Setting flask\_env variable to development will enable debug mode. The debug mode is useful when we make a change in our application if debug mode is activated flask will automatically refresh the server and reflects the changes in he webpage. We can even control the debug mode separately by using FLASK\_DEBUG flag. If it is set to True then the debug mode will be activated.

**6) What happens when the Flask run command is issued in the terminal?**

**Ans)** When we issue Flask run command in the terminal the flask server will start and it will generate an ip address. If we copy and paste the ip address generated by flask in the web browser our application will be rendered and output will be displayed in the web browser.

**7) On which port is Flask running and can it be changed?**

**Ans)** By default flask will run on port 5001. We can change the port of the flask by flask run –port=port-number

**8) How is Flask different from the tiny web server?**

**Ans)** Tiny webserver is a basic http server. This server can accept multiple requests at once. It supports only GET request and it also does not support any server side processing. Where as Flask is a microframework which supports both GET and POST requests as well as it can do server side processing.

**TASK-3:**

**Goodreads API-Reflection:**

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

**Ans)** REST API,

**2) Is there a limit on the use of the web API? What are the limits?**

**Ans)** Yes there are limits on the use of web API.

In the API console, there is a similar quota referred to as **Requests per 100 seconds per user**. By default, it is set to 100 requests per 100 seconds per user and can be adjusted to a maximum value of 1,000. But the number of requests to the API is restricted to a maximum of 10 requests per second per user.