How to set up environment variables in GitPod

Instructions from Code Institute

Why are environment variables important?

- In your projects you are often dealing with sensitive information, such as SECRET_KEYs, or MONGDB_URIs that you do not want anyone else to have access to. If you send these values to github, they will be in plain text where anyone (or any bot) can pull them up and misuse them for example to hack your database.

Here is one way to set environment variables in gitpod:

1. Create a file named `env.py` in the root directory of your project. This is the file you will use to define your environment variables.

2. If you don't have one already, create a file named `.gitignore` in the root directory of your project.

3. Next we need to stop git from pushing this file to github, and so keep your environment variables secret. To do this, open your `.gitignore` file add the following text to it: `env.py`

4. At the top of your `env.py` file, you need to `import os` so that you can set the environment variables in the operating system. Once you have added the line "import os" underneath you can assign your environment variables using the following syntax: `os.environ["Variable Name Here"] = "Value of Variable Goes Here"` Example: `os.environ["SECRET_KEY"] = "ohsosecret"`

5. Then the following code imports this new `env.py` file where you need to use your environment variables. For example your `app.py` file for flask project or `settings.py` file for Django project. Add this under your other imports at the top of the file.
```
from os import path
if path.exists("env.py"):
  import env
```
The if statement here is so that the `env.py` file is only pulled when working on your code in your workspace, not when it is deployed on heroku. For deployment you can set your environment variables in the heroku dashboard in settings > config vars.

6. Now that your environment variables have been set in your `env.py` file, and the file has been imported into your project, you can use them as needed using the following syntax:
`SECRET_KEY = os.environ.get('SECRET_KEY')` Make sure you save all your files before testing if it works.