

Ibrahima Diallo
Kura Labs
Summer 2021
Instructors:
Daniel Adeyanju
Tyrone Sanderson

Deployment 4

I have used some open source resources in order to complete this assignment. The sources visited are linked below.

Setting up the environment in VS Code:

- I. Installation Flask in Python using visio code (Windows 10)
Flask requires Python 3

Setup virtual environment

1. Create an environment

```
> mkdir app_directory  
> cd app_directory  
> py -3 -m venv virtual
```

2. Activate the environment

```
> .\virtual\Scripts\activate
```

3. Install Flask

```
$ pip install Flask
```

- II. In the **app_directory** folder, create a folder **application_app** and a Python file where the code will be displayed.
- III. Create a static and templates folder in **application_app** folder
In the static folder, create a **CSS** file for the style if needed.
In the templates folder, create a HTML file

1 - Url Shortener using Flask

Goal for this deployment:

The web application displays a short version of a url. AWS Beanstalk and Jenkins are the tools we will use for the deployment.

Step 1: Set up the environment for the **url-shortner** web application.

Forked this repo (https://github.com/kura-labs-org/DEPLOY04_FLASK_APP) to have a copy of the **url-shortner** application.

All the source codes are in the repository [DEPLOY4_FLASK_APP](#).

Followed the steps provided from this pdf document link to set up the AWS Beanstalk and Jenkins.

https://github.com/kura-labs-org/DEPLOY4_FLASK_APP/blob/main/Deployment%204.pdf

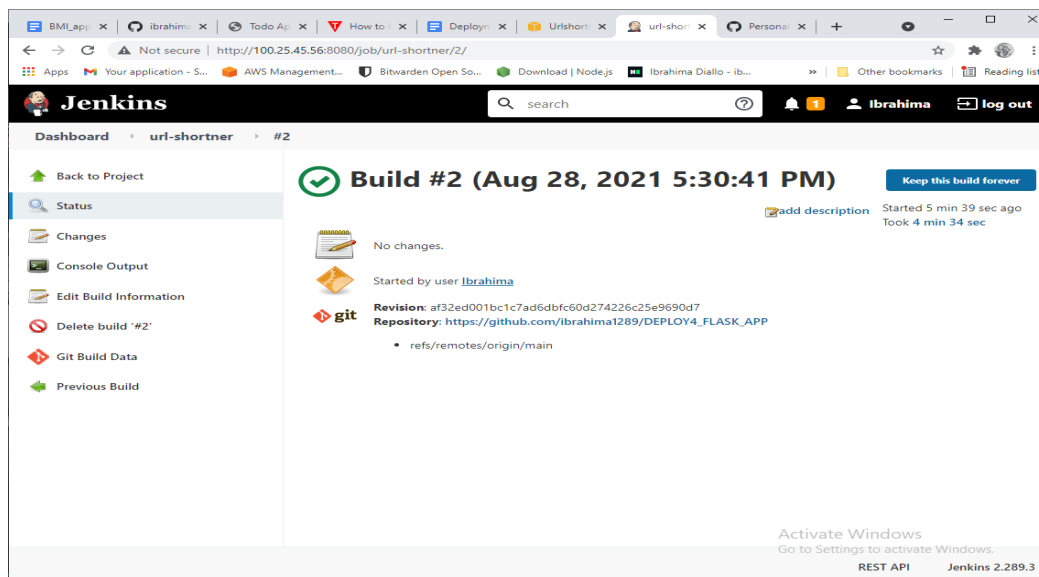
Step 2: Removed the README.md and Deployment4.pdf instructions from your forked repo before I start the build in Jenkins.

Created a `requirements.txt` file that is used for specifying what python packages are required to run the project.

Below is the command for the setup on VS code.

```
$ pip freeze > requirements.txt
```

Step 3: Take a screenshot of the **url-shortner** home page and add the screenshot to your screenshot file.

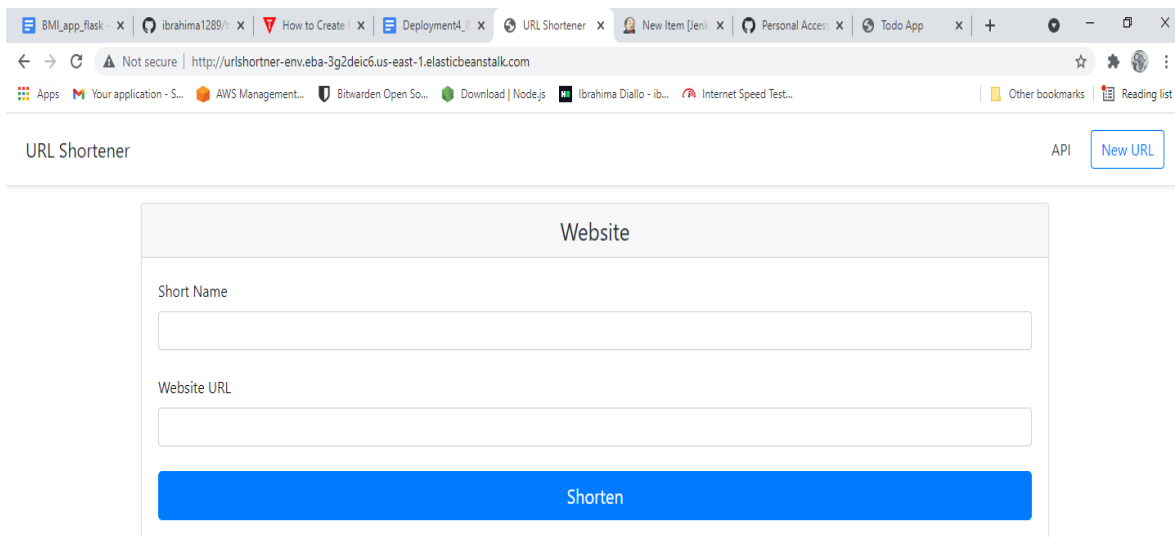


Troubleshooting: #1 build failed, had to change the Branch Specifier from “*/master*” to “*/main*” (Ricardo)



The screenshot shows the 'Branches to build' configuration section in Jenkins. At the top right is a button labeled 'Add repository'. Below it, the section title 'Branches to build' is followed by a help icon. A red 'X' icon and another help icon are visible. The 'Branch Specifier (blank for 'any')' field contains the text '*/main'. At the bottom right of this section is a button labeled 'Add Branch'.

The *url-shortner* home page



The screenshot shows the home page of a web application titled 'URL Shortener'. The browser's address bar shows the URL 'http://urlshortner-env.eba-3g2deic6.us-east-1.elasticbeanstalk.com'. The page has a header with the title 'URL Shortener' and an 'API' link next to a 'New URL' button. The main content area is a form titled 'Website' with two input fields: 'Short Name' and 'Website URL'. Below these fields is a large blue button labeled 'Shorten'.

Activate Windows
Go to Settings to activate Windows.

2 - Todo list using Flask

Goal for this deployment:

For this web application, a todo list is created, then items can be updated, and/or deleted from the list. We will use AWS Beanstalk and Jenkins to deploy the todo app.

Step 1: Upload the source codes in Github under this repository **todo-app-flask** created on Github :

<https://github.com/ibrahima1289/todo-app-flask>

Step 2: Followed the steps provided from this pdf document link to set up the AWS Beanstalk and Jenkins just as we did for the **url-shortener** deployment

https://github.com/kura-labs-org/DEPLOY4_FLASK_APP/blob/main/Deployment%204.pdf

Create a `requirements.txt` file using this command below.

```
$ pip freeze > requirements.txt
```

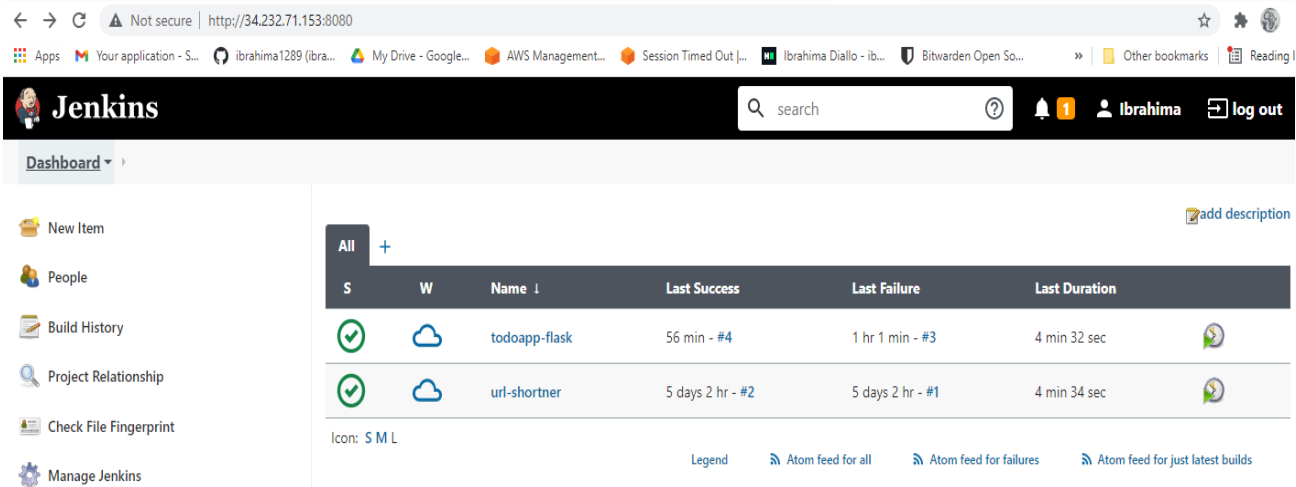
Step3: Setting up the database

In this project, I used SQLAlchemy which is *“the Python SQL toolkit and Object Relational Mapper that gives application developers the full power and flexibility of SQL.”*

The source about the documentation is in this link

<https://github.com/sqlalchemy/sqlalchemy> .

Step 4: Screenshot of the **todoapp-flask** on Jenkins.



The screenshot shows the Jenkins web interface. The top navigation bar includes the Jenkins logo, a search bar, and user information for 'ibrahima'. The left sidebar contains links to 'New Item', 'People', 'Build History', 'Project Relationship', 'Check File Fingerprint', and 'Manage Jenkins'. The main content area displays a table of build history for the 'All' job category. The table has columns for status (S), icon (W), name, last success, last failure, and last duration. Two builds are listed: 'todoapp-flask' and 'url-shortner', both with successful status and cloud icons. Below the table, there are links for 'Icon: S M L', 'Legend', and three Atom feed links.

S	W	Name	Last Success	Last Failure	Last Duration
✓	☁	todoapp-flask	56 min - #4	1 hr 1 min - #3	4 min 32 sec
✓	☁	url-shortner	5 days 2 hr - #2	5 days 2 hr - #1	4 min 34 sec

Troubleshooting:

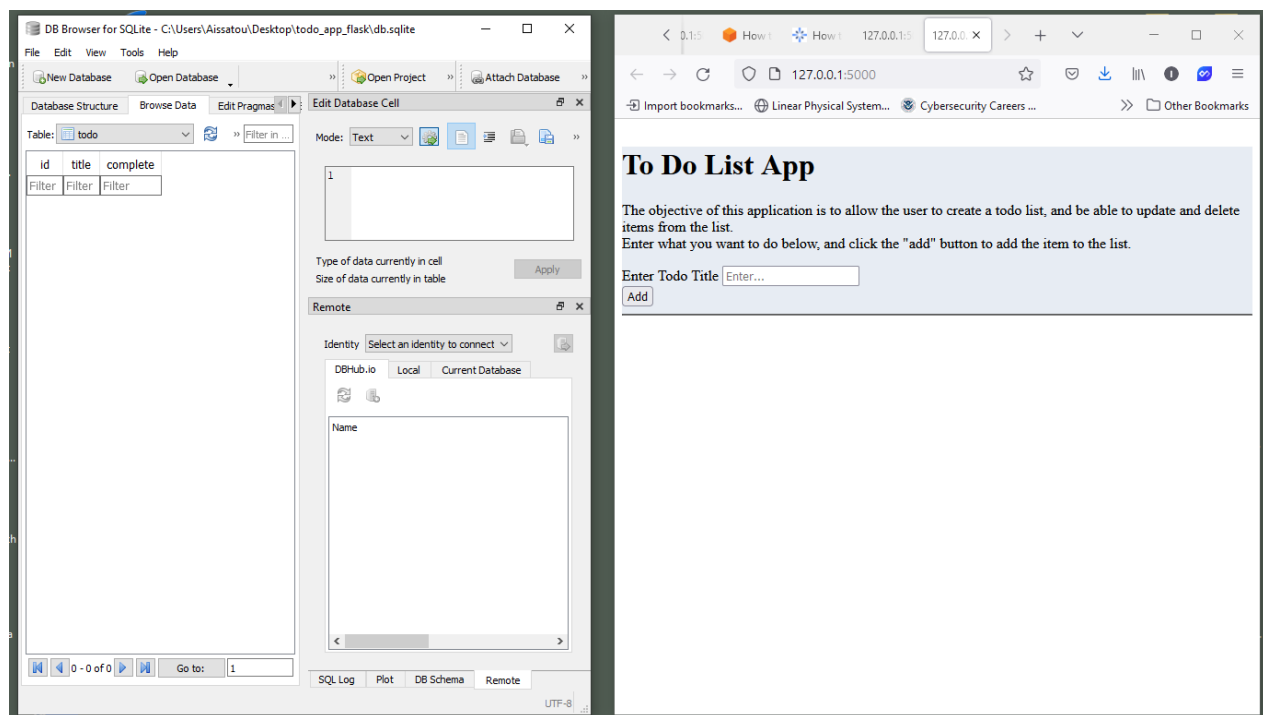
1. I ran into some errors when I was requesting a build in Jenkins. I found out that in the python file I was using the main clause which is not needed for the source file because python is not running directly to the local drive. (Sai)

```
if __name__ == "__main__":  
    db.create_all()  
    app.run(debug=True)
```

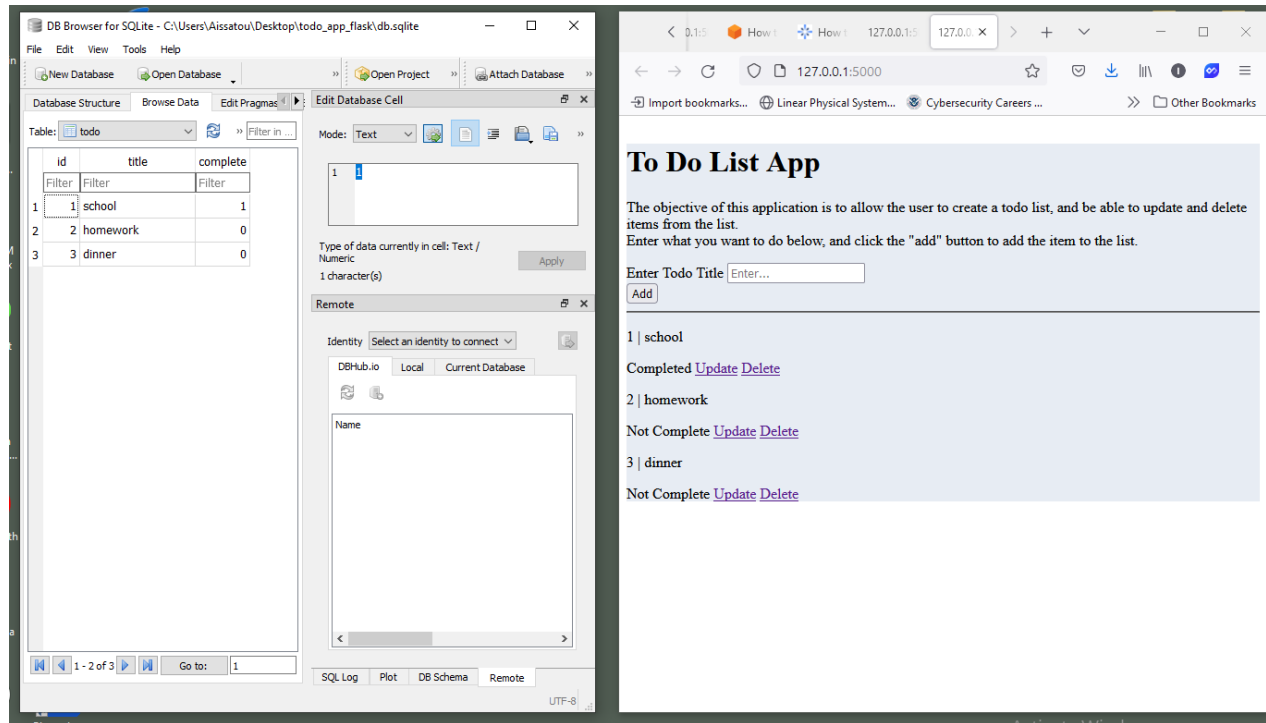
2. Also, I had to rename the python file **application.py** instead of **app.py** in order for it to be recognized.

The todo app web page is on the right, and on the left we have the database.

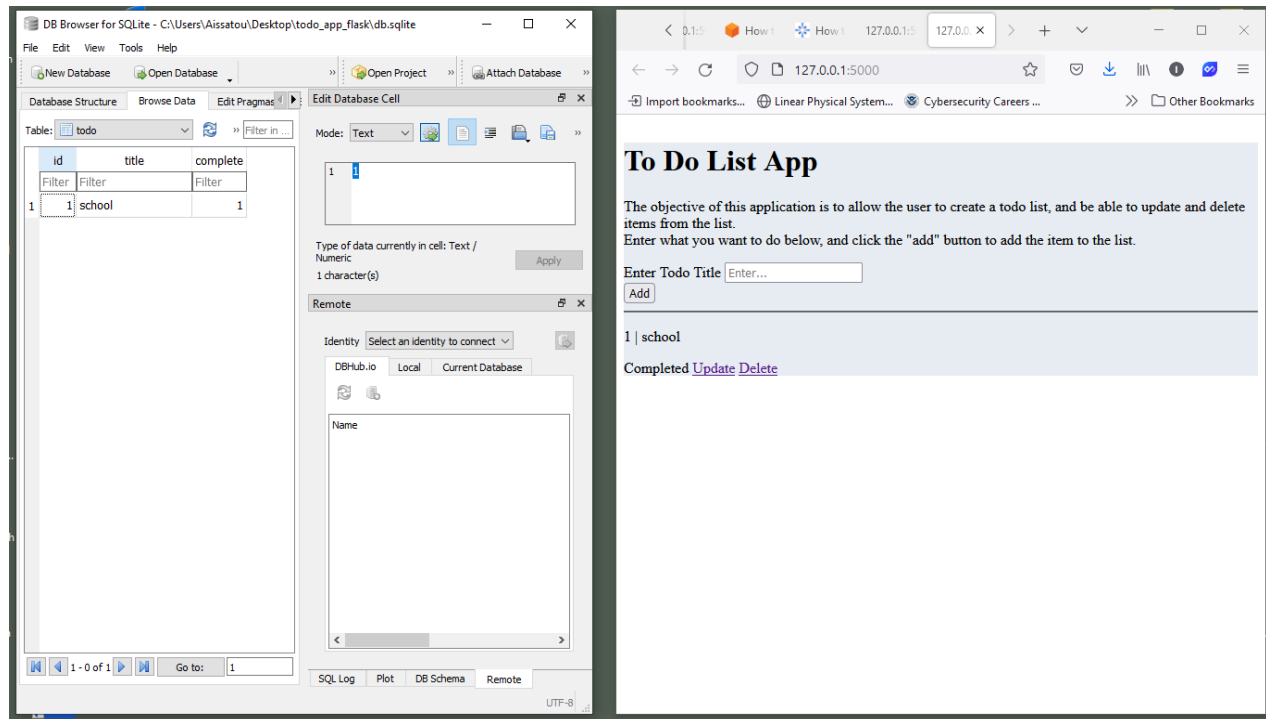
Here is the initial state:



Here, we added some items to the list, and we can see that the database is being updated as well.



And after we deleted some items from the list, they are also being deleted from the database.



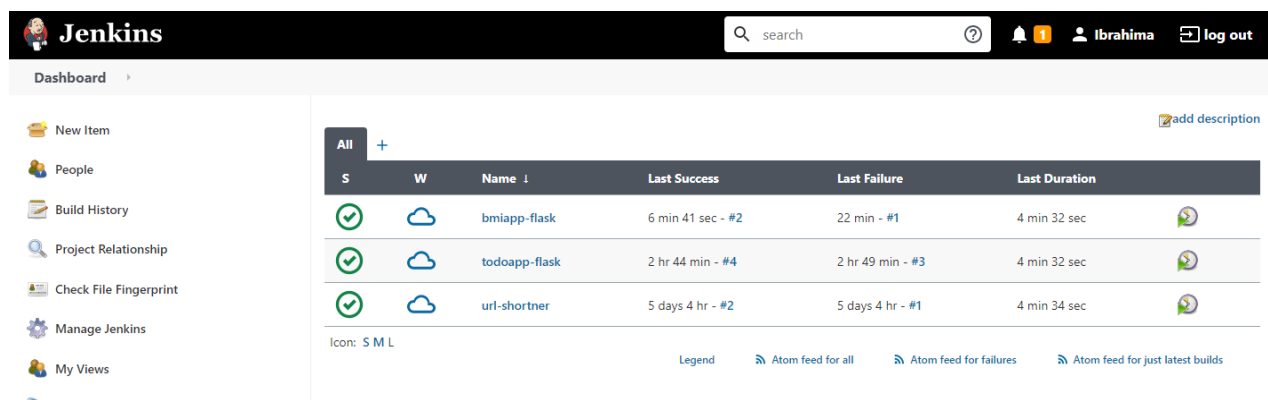
3 - Costum app: Calculator

Goal for this deployment:

1. The BMI (Body Mass Index) calculator gives the user some advice whether or not to lose/gain weight to stay healthy.
2. Simple calculator
3. Units converter (not finished yet)

Setups: We will repeat the same steps as the previous two deployments except for the database setup.

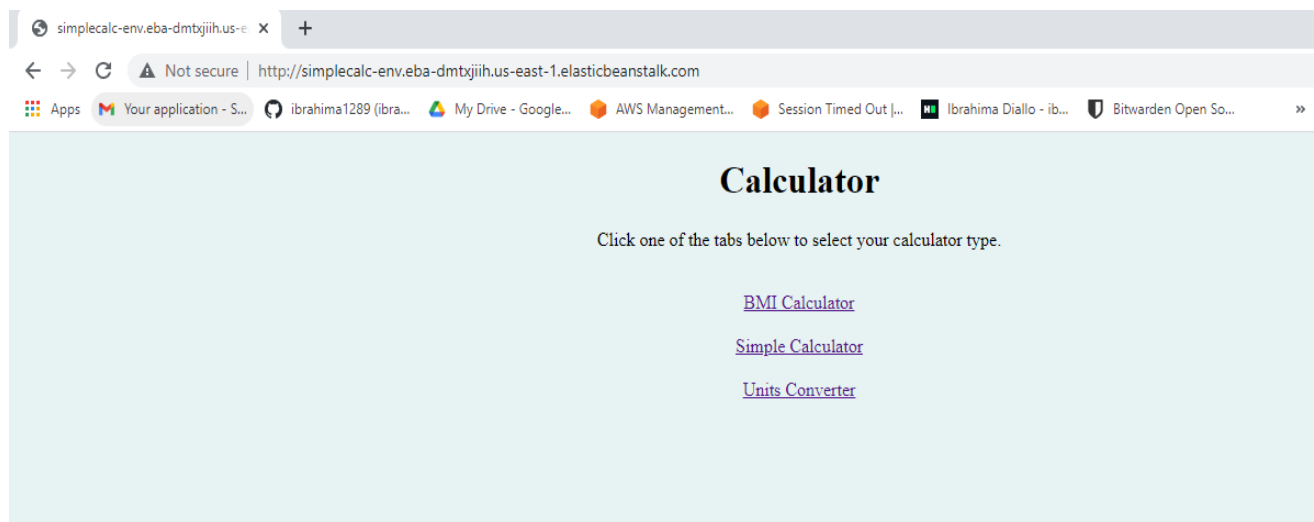
In Jenkins, the build was successful as we can see below.



S	W	Name ↓	Last Success	Last Failure	Last Duration	
✓	☁	bmiapp-flask	6 min 41 sec - #2	22 min - #1	4 min 32 sec	🔍
✓	☁	todoapp-flask	2 hr 44 min - #4	2 hr 49 min - #3	4 min 32 sec	🔍
✓	☁	url-shortner	5 days 4 hr - #2	5 days 4 hr - #1	4 min 34 sec	🔍

Icon: S M L Legend Atom feed for all Atom feed for failures Atom feed for just latest builds

The calculator web page:



Sources:

1. <https://flask.palletsprojects.com/en/2.0.x/installation/>
2. <https://www.youtube.com/watch?v=1k3cNPWVpcY>
3. <https://stackabuse.com/building-a-todo-app-with-flask-in-python>
4. <https://help.pythonanywhere.com/pages/Flask/>
5. <https://github.com/sqlalchemy/sqlalchemy>
6. <https://github.com/vr025/Calculator-Application-Using-Python-Flask>