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

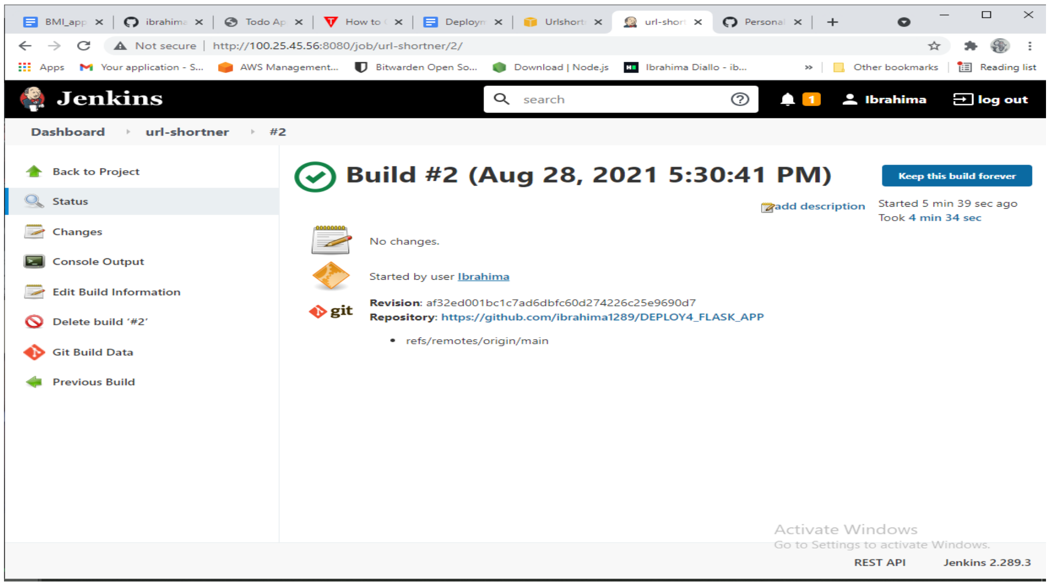
Step 2: Removed the README.md and Deployment4.pdf instructions from your forked repo before I start the build in Jenkins.

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.



http://100.25.45.56:8080/job/url-shortner/2/

Dashboard › url-shortner › #2

Build #2 (Aug 28, 2021 5:30:41 PM)

Started 5 min 39 sec ago
Took 4 min 34 sec

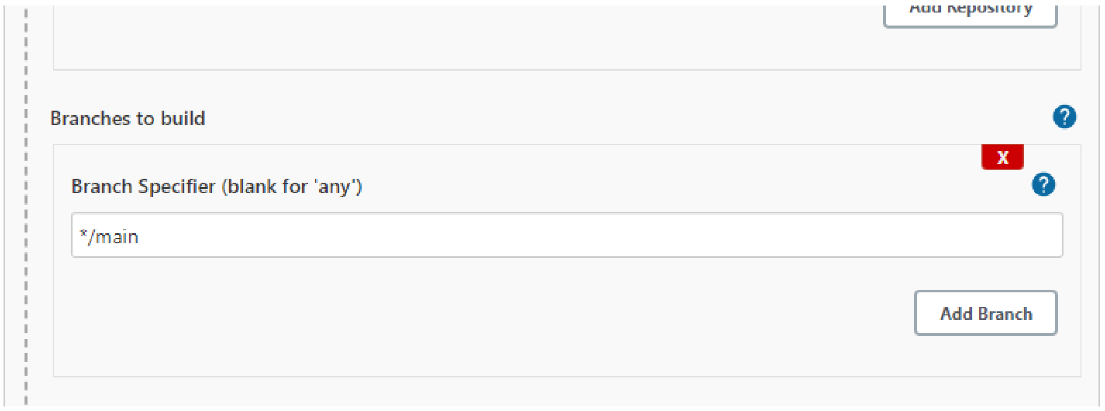
No changes.

Started by user Ibrahima

Revision: af32ed001bc1c7ad6dbfc60d274226c25e9690d7
Repository: https://github.com/ibrahima1289/DEPLOY4_FLASK_APP

- refs/remotes/origin/main

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



Add Repository

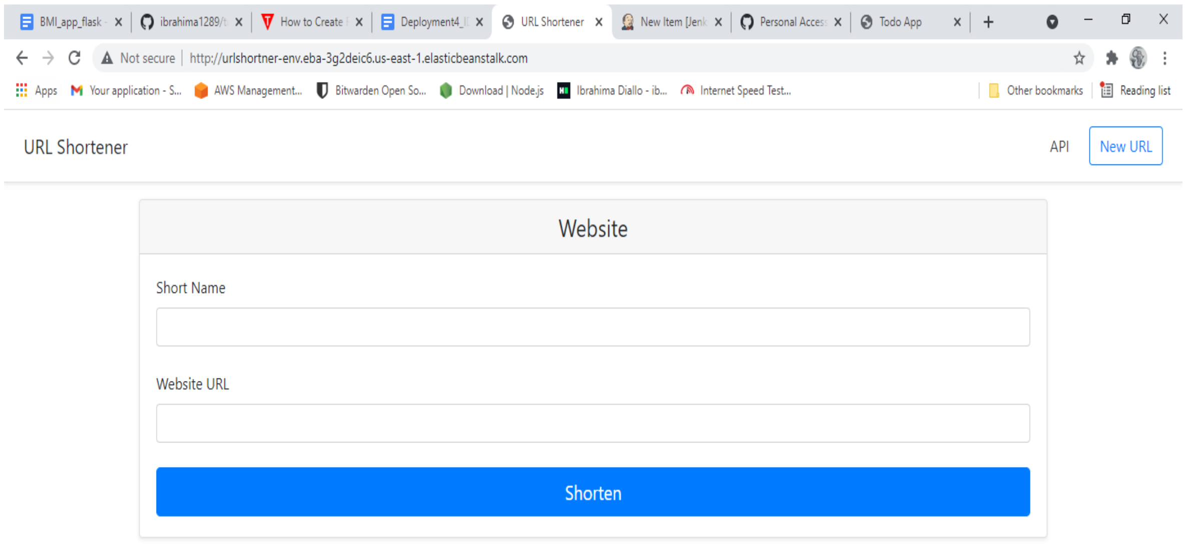
Branches to build

Branch Specifier (blank for 'any')

*/main

Add Branch

The ***url-shortner*** home page



Not secure | http://urlshortner-env.eba-3g2deic6.us-east-1.elasticbeanstalk.com

URL Shortener

API

New URL

Website

Short Name

Website URL

Shorten

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.

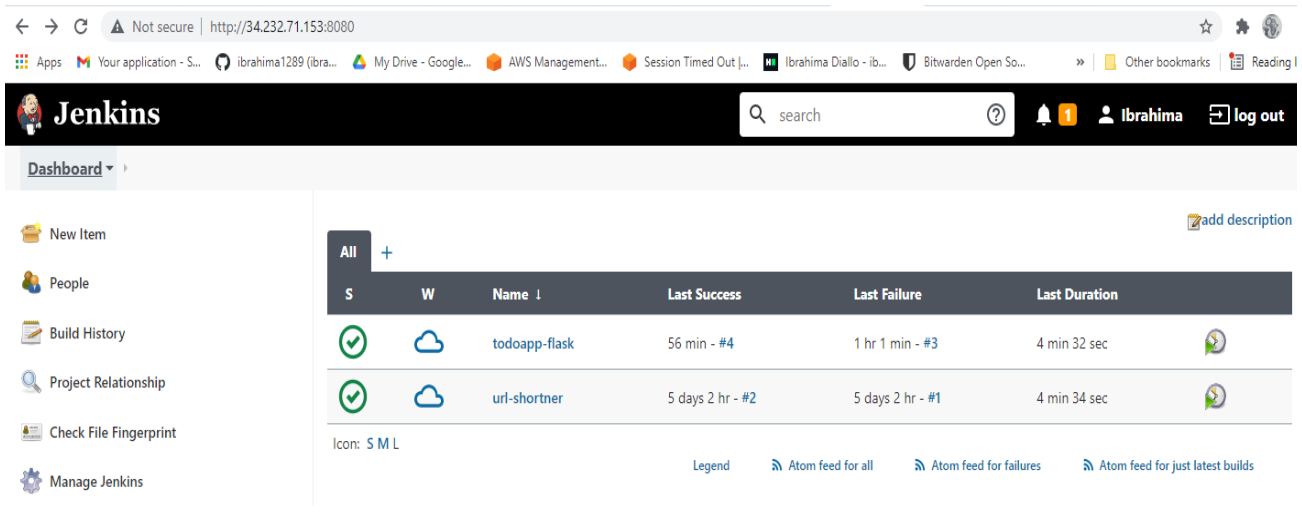
```
$ pip freeze > requirements.txt
```

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.



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:

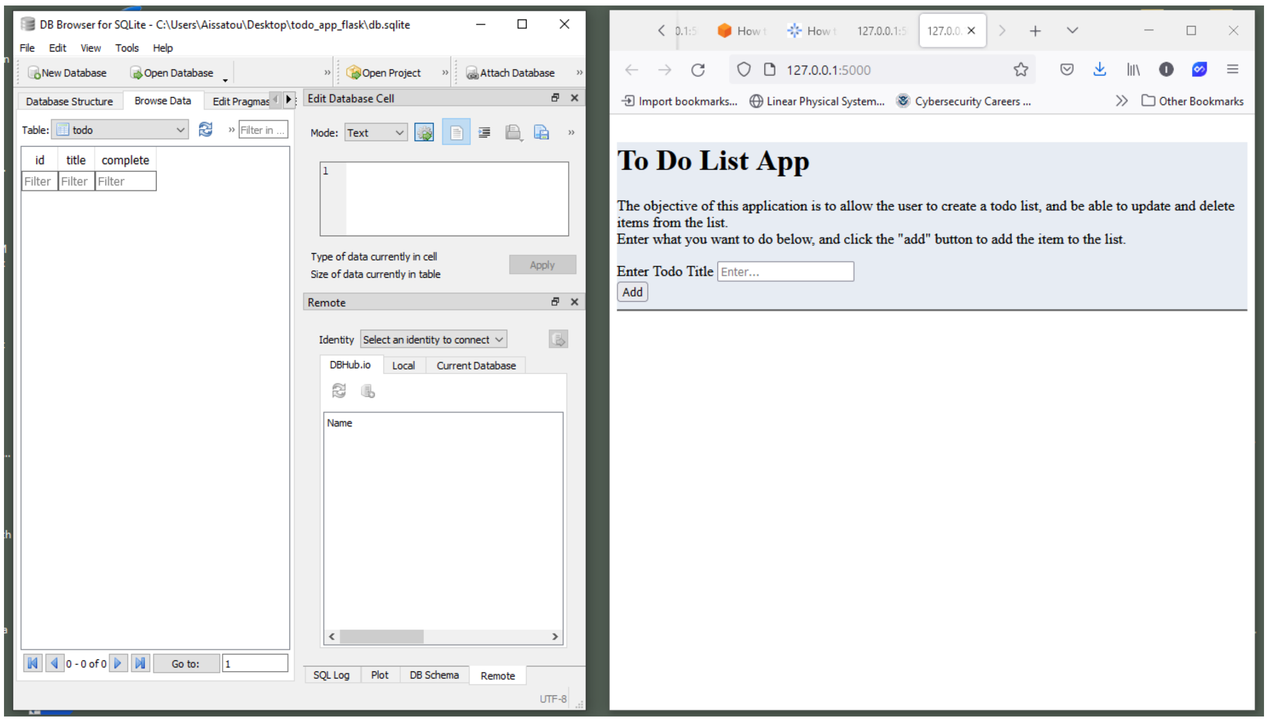
1. I ran into some errors when I was requesting a build in Jenkins. I found out that in the python file I was using the main clause which is not needed for the source file because python is not running directly to the local drive. (Sai)

```
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:

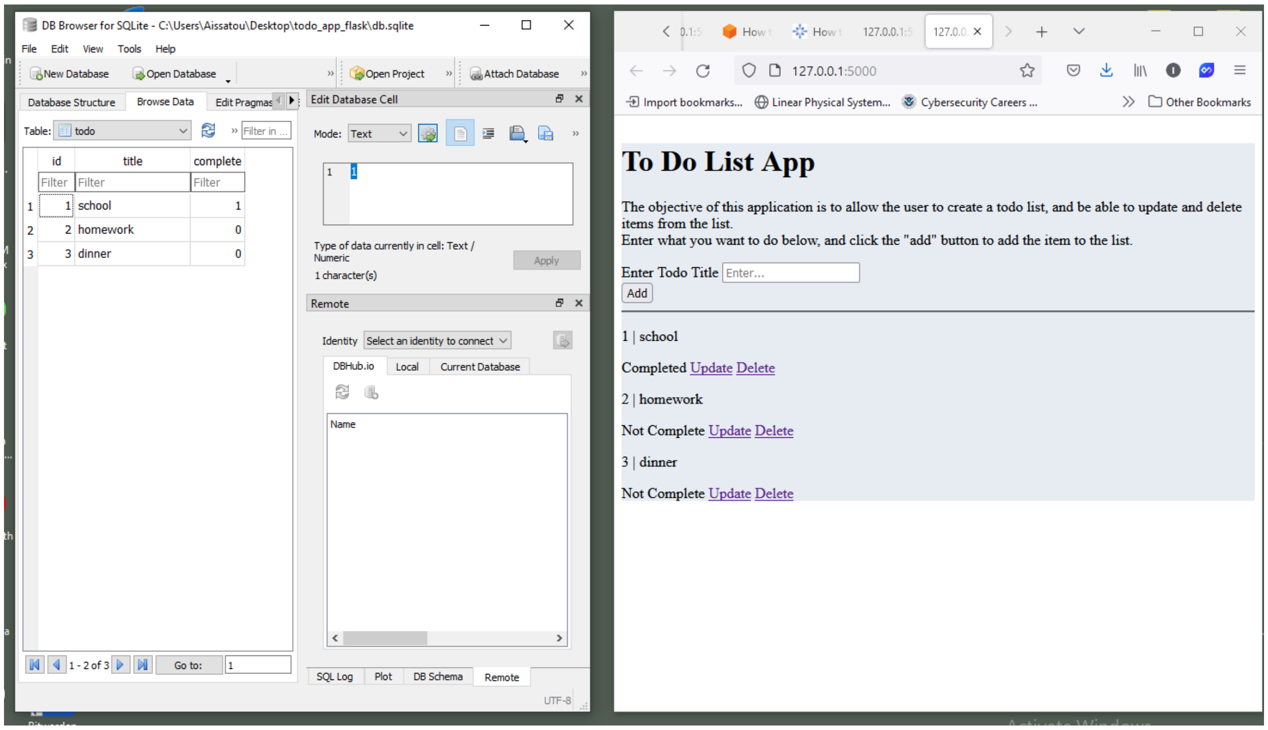

DB Browser for SQLite - C:\Users\Aissatou\Desktop\todo_app_flask\db.sqlite
File Edit View Tools Help
New Database
Open Database
Open Project
Attach Database
Database Structure
Browse Data
Edit Pragmas
Edit Database Cell
Table: todo
Mode: Text
id title complete
Filter Filter Filter
Type of data currently in cell
Size of data currently in table
Apply
Remote
Identity Select an identity to connect
DBHub.io Local Current Database
Name
0 - 0 of 0
Go to: 1
SQL Log Plot DB Schema Remote
UTF-8
127.0.0.1:5000
Import bookmarks...
Linear Physical System...
Cybersecurity Careers ...
Other Bookmarks

To Do List App

The objective of this application is to allow the user to create a todo list, and be able to update and delete items from the list.
Enter what you want to do below, and click the "add" button to add the item to the list.

Enter Todo Title Enter...
Add

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



	id	title	complete
1	1	school	1
2	2	homework	0
3	3	dinner	0

1 - 2 of 3

Type of data currently in cell: Text / Numeric
1 character(s)

To Do List App

The objective of this application is to allow the user to create a todo list, and be able to update and delete items from the list.
Enter what you want to do below, and click the "add" button to add the item to the list.

Enter Todo Title Enter...
Add

1 | school

Completed Update Delete

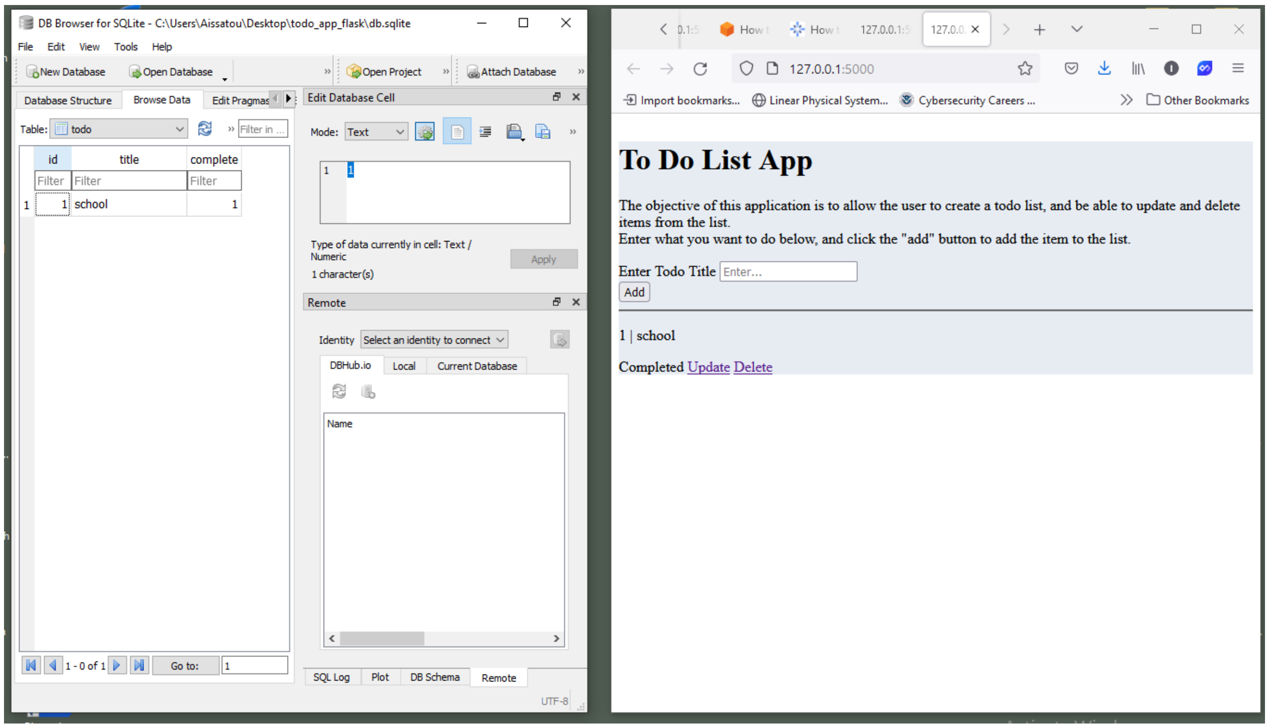
2 | homework

Not Complete Update Delete

3 | dinner

Not Complete Update Delete

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



	id	title	complete
1	1	school	1

1 - 0 of 1

Type of data currently in cell: Text / Numeric
1 character(s)

To Do List App

The objective of this application is to allow the user to create a todo list, and be able to update and delete items from the list.
Enter what you want to do below, and click the "add" button to add the item to the list.

Enter Todo Title Enter...
Add

1 | school

Completed Update Delete

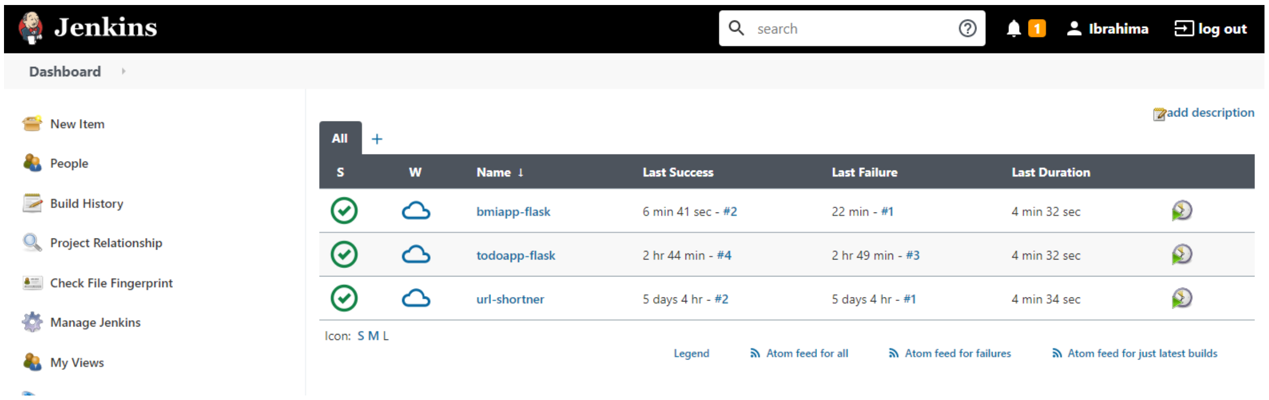
3 - Costum app: BMI Calculator

Goal for this deployment:

The BMI (Body Mass Index) calculator gives the user some advice whether or not to lose/gain weight to stay healthy.

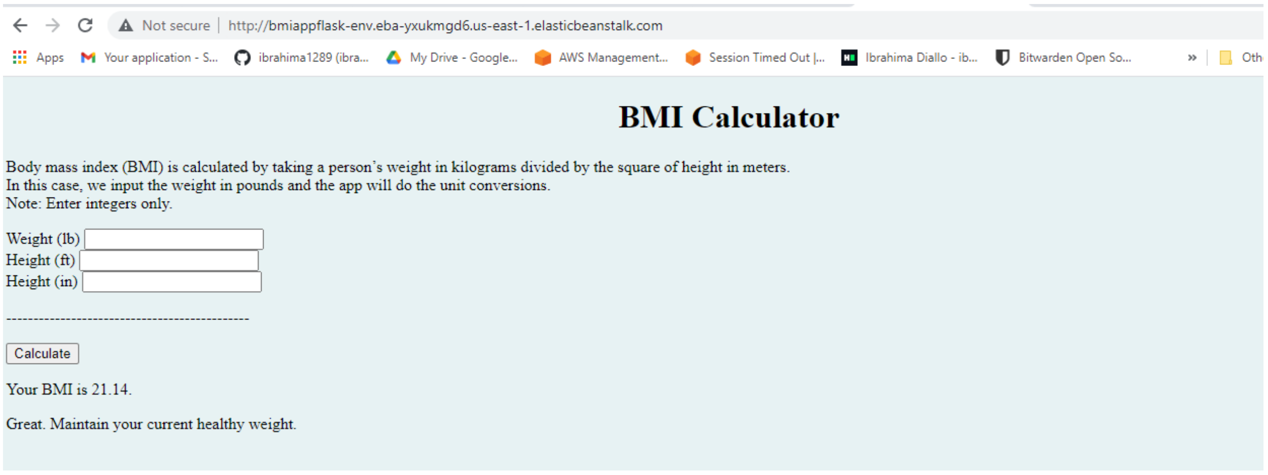
Setups: We will repeat the same steps as the previous two deployments except for the database setup.

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

The BMI calculator web page:



BMI Calculator

Body mass index (BMI) is calculated by taking a person's weight in kilograms divided by the square of height in meters.
In this case, we input the weight in pounds and the app will do the unit conversions.
Note: Enter integers only.

Weight (lb)
Height (ft)
Height (in)

Calculate

Your BMI is 21.14.

Great. Maintain your current healthy weight.

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