Advanced Database Systems Design

Homework 2

Sarika Sadineni (ssadinen@kent.edu, 811178010)

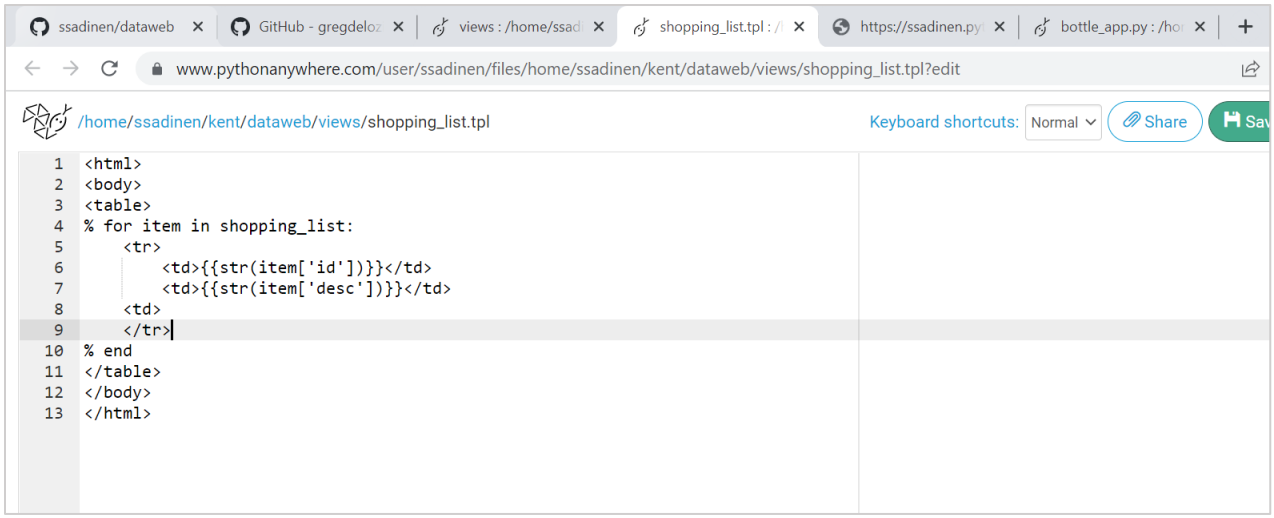
Web app - https://ssadinen.pythonanywhere.com/

Git repositories –

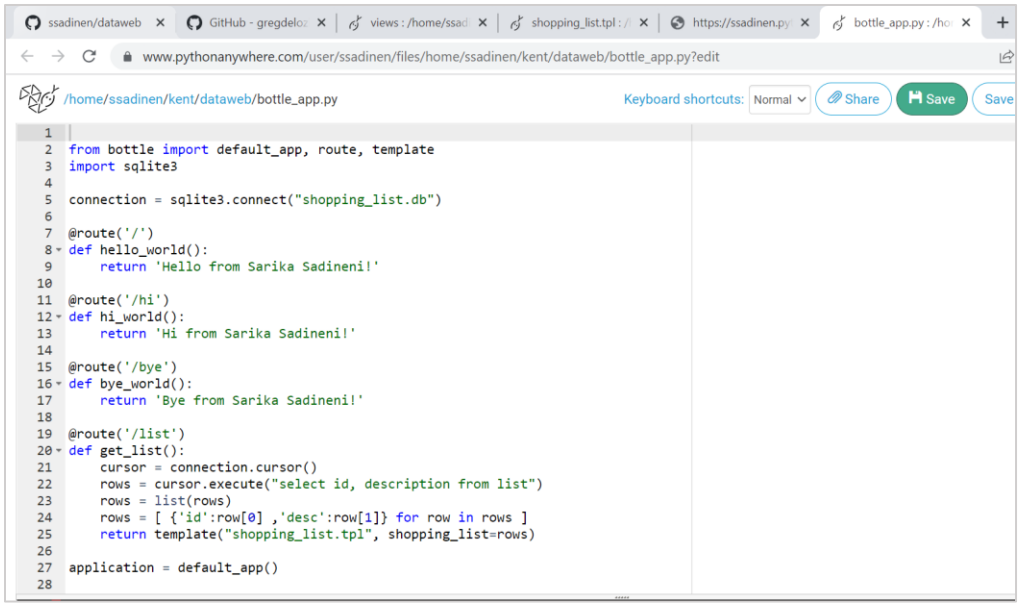
https://github.com/ssadinen/advanced-database-ssadinen

https://github.com/ssadinen/dataweb

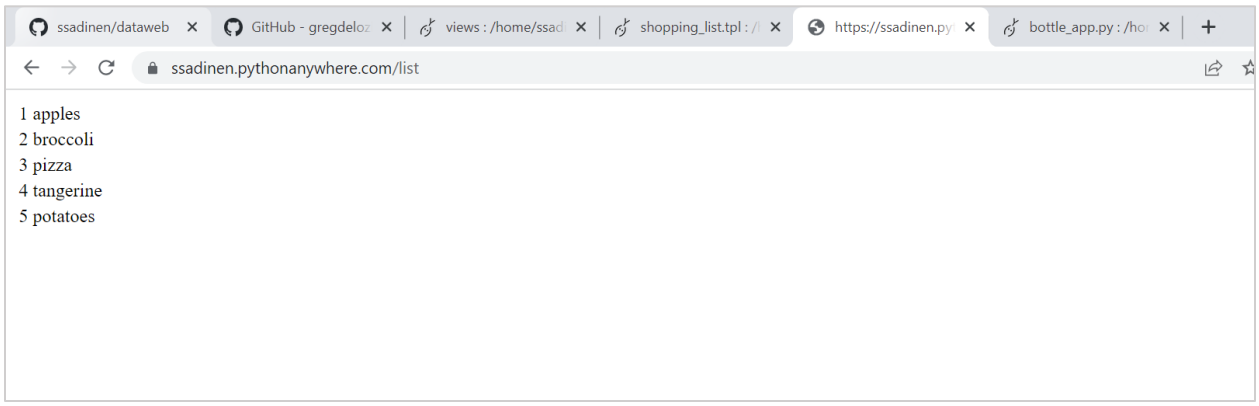
- Created a template for Shopping list as below and added table data items – id, desc



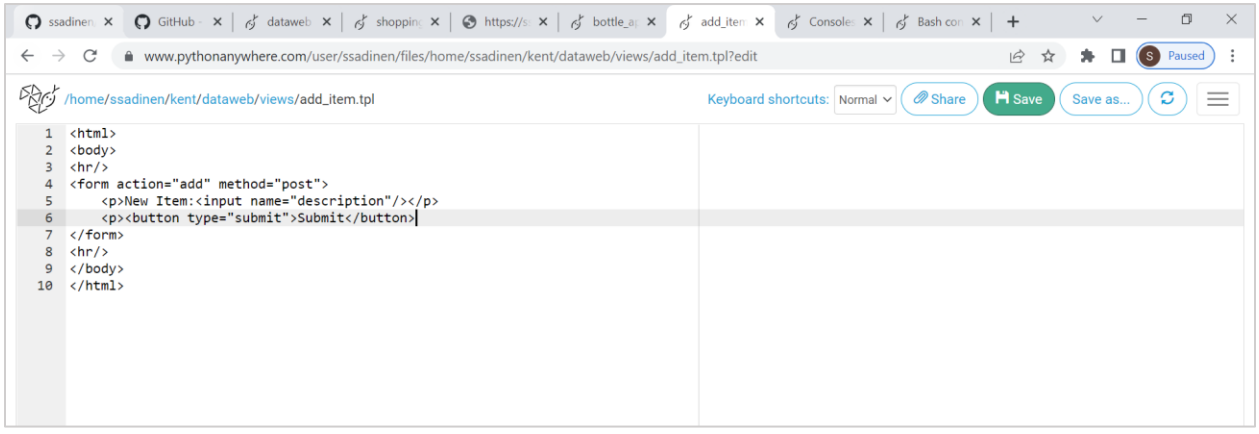
- Created bottle app.py in dataweb directory and added routes hi, bye for testing and route list for the shopping list application



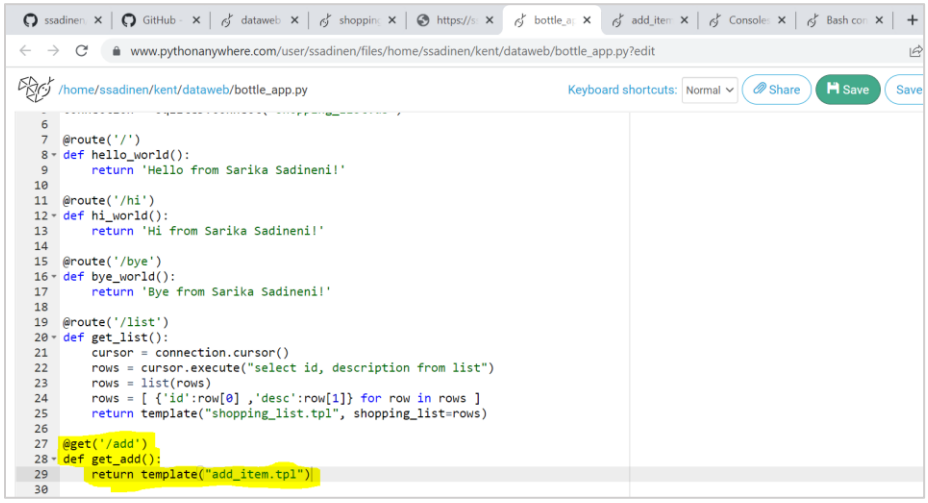
- Below is the 'list' route of the app with the item description and id selected from shipping_list.db



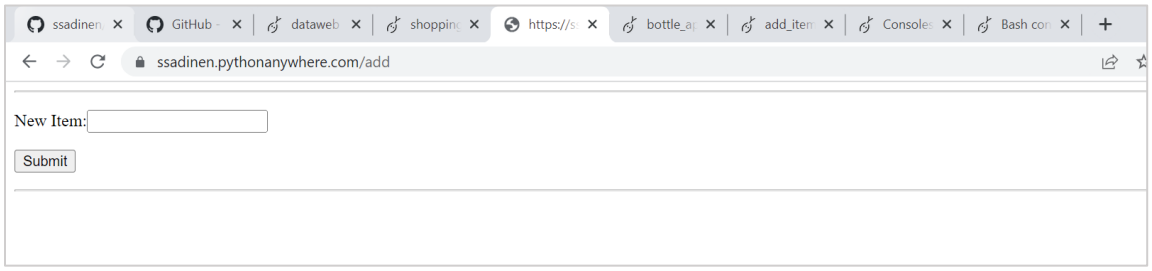
- Created add_item.tpl in web app views to add new items to the shopping list



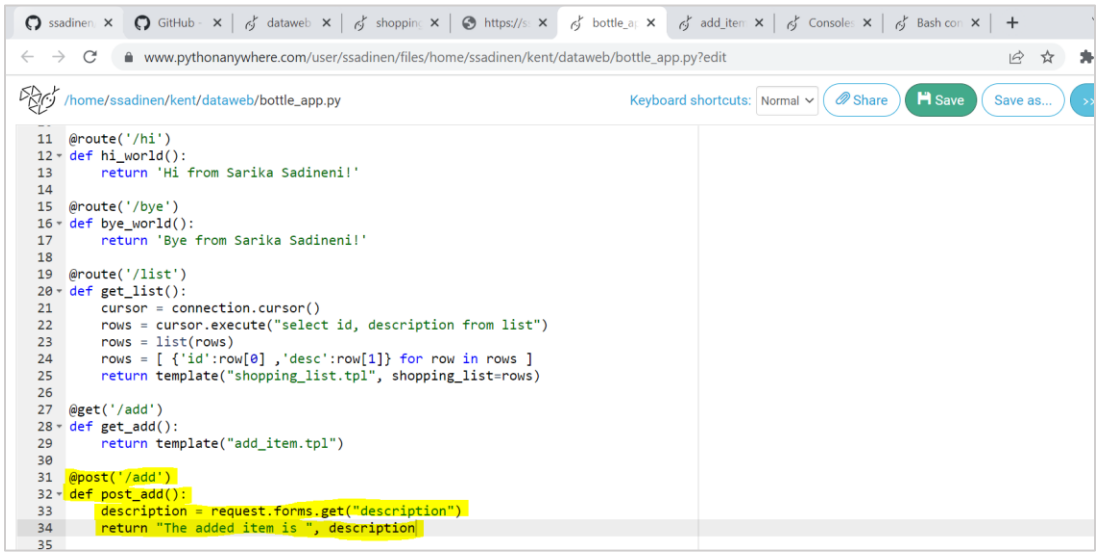
- Also added '/add' get route to the app so when user select on add item they will be taken to new route page /add of web page



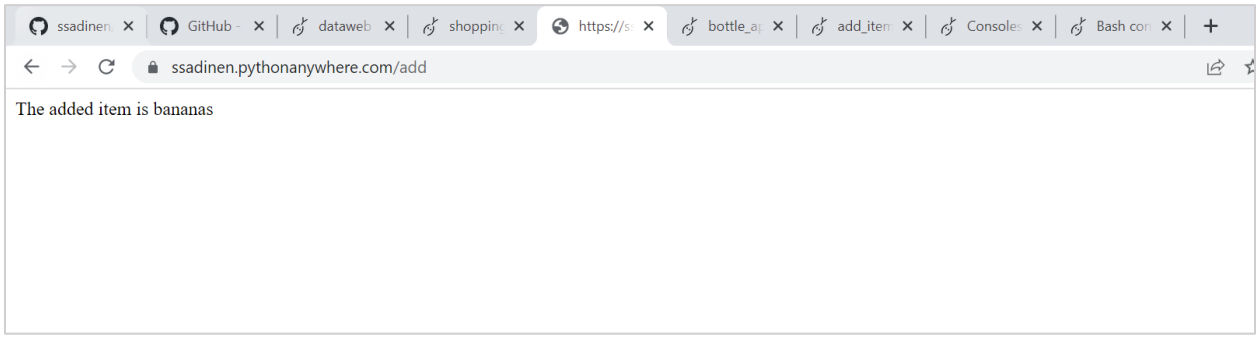
- Below is the route to /add a new item to the shopping list



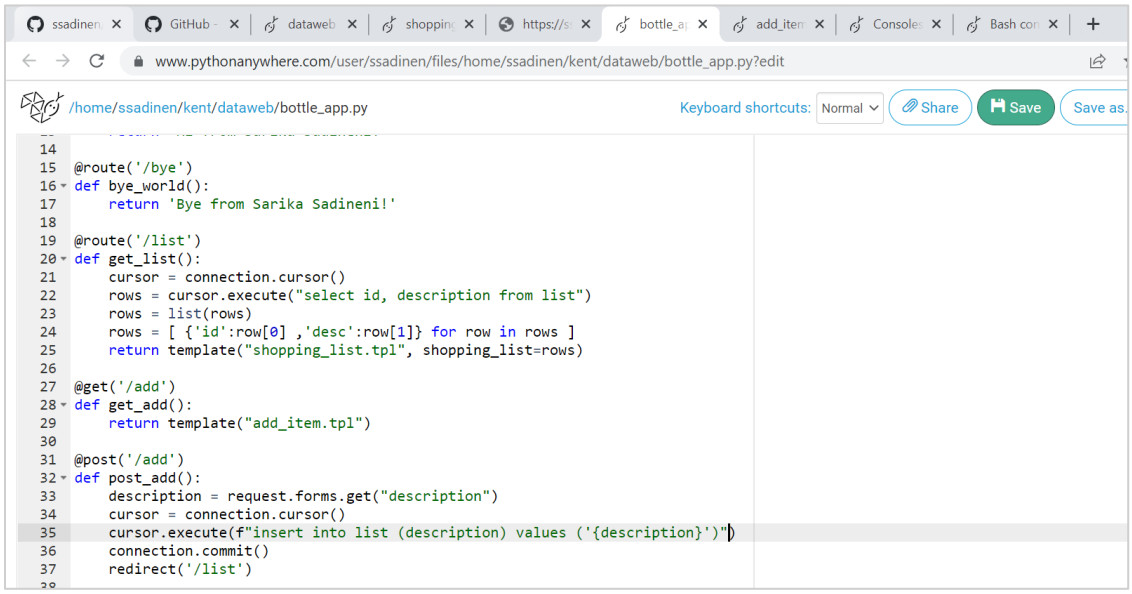
- With just get route we cannot add item to the list we need to add post route as well so the item will get added to the list as soon as user click on submit. So added below lines to /add a new item with post.



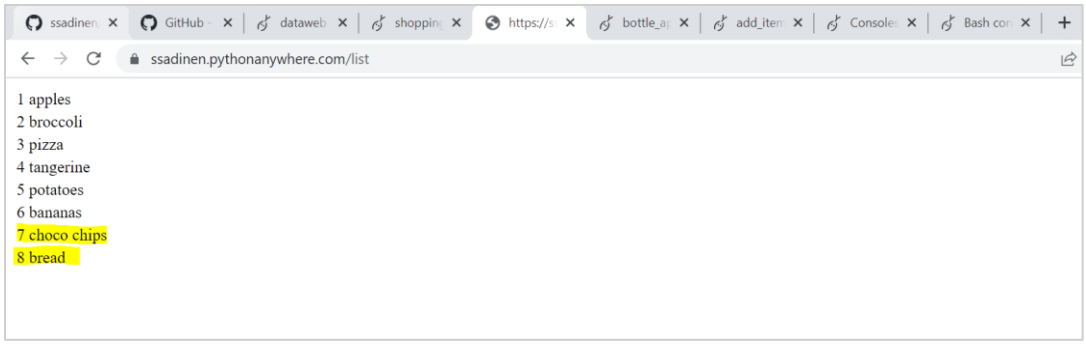
- After saving above lines and clicking on run, I've opened webapp and added new item bananas to the list.



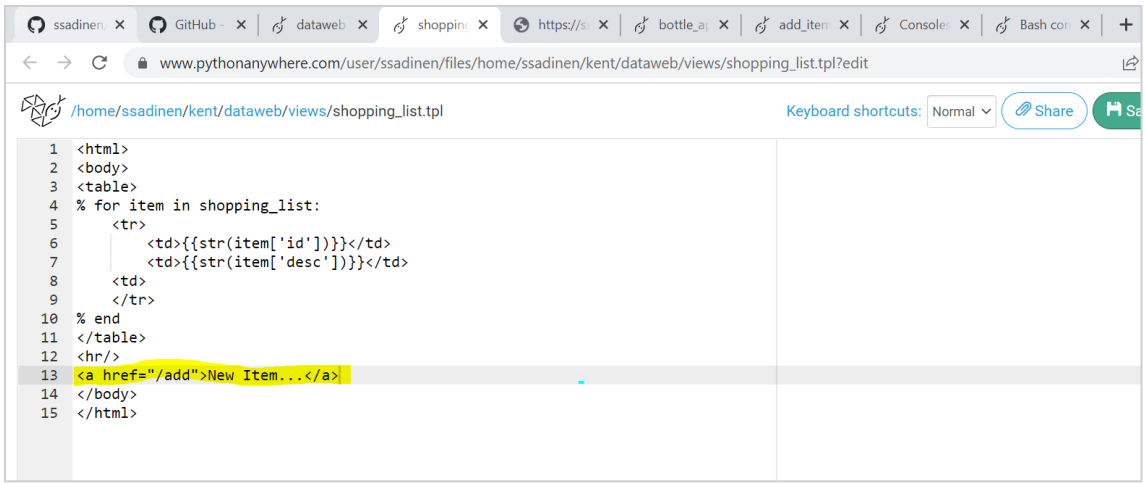
- Added below cursor.execute to insert the new items to the shopping list. Also added redirect, so when we open the webpage without any route we will be default redirected to the /list route



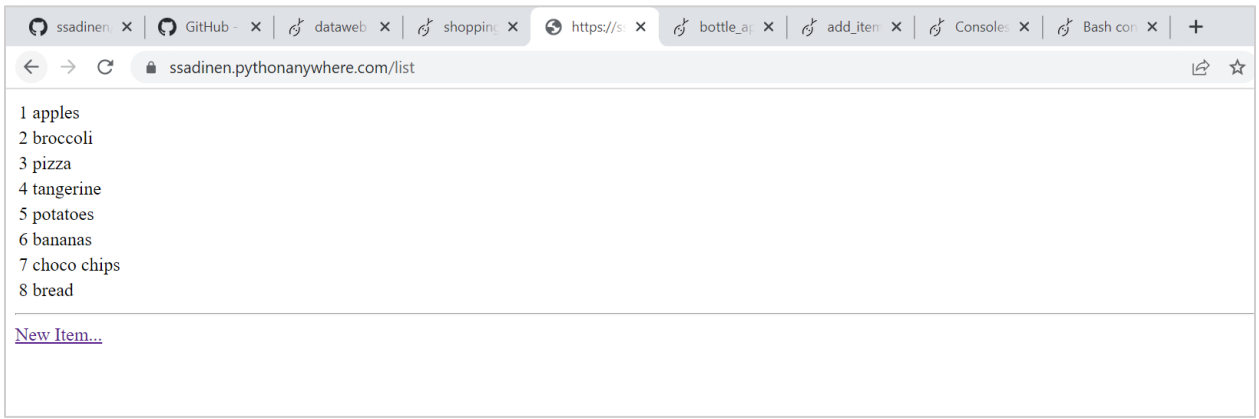
- After saving the above lines of code in bottle_app.py and reloading the webpage I have added chocochips and bread to the list as highlighted in below screenshot.



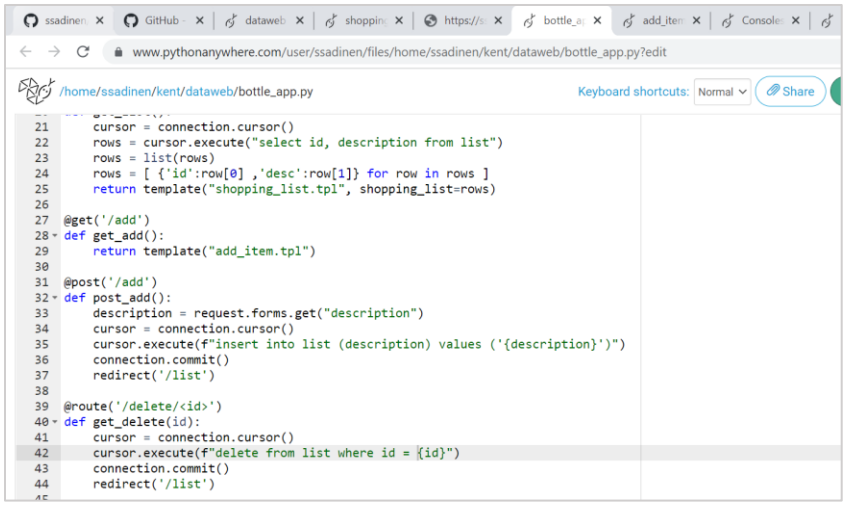
- In the previous step we got to go to /add route page in order to add the new item. In order to avoid going to new page by changing the route and add new item with a link in the same web page as list, I've added href in shopping_list.tpl for new item.



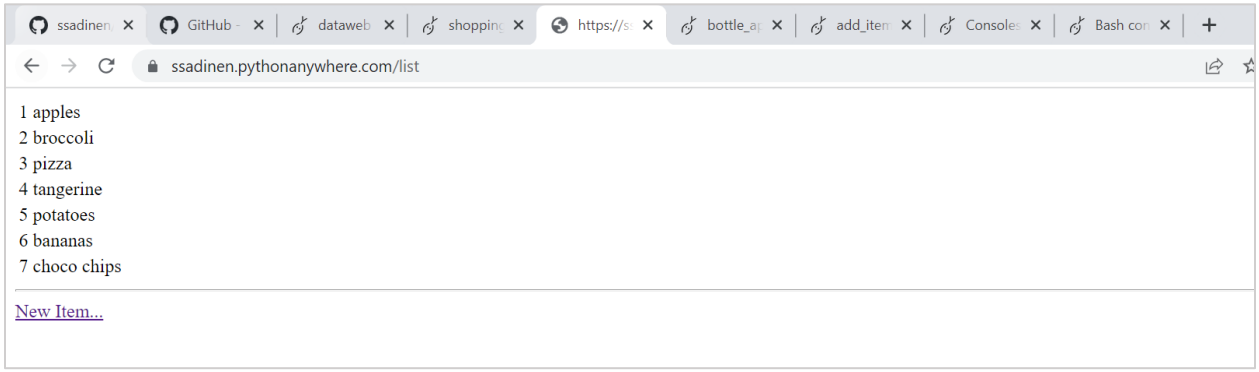
- After saving above template and reloading the bottle_app.py, New Item appears in the Shopping list page itself as shown in below screenshot.



- Created a new route to delete the item from list where id = {id}

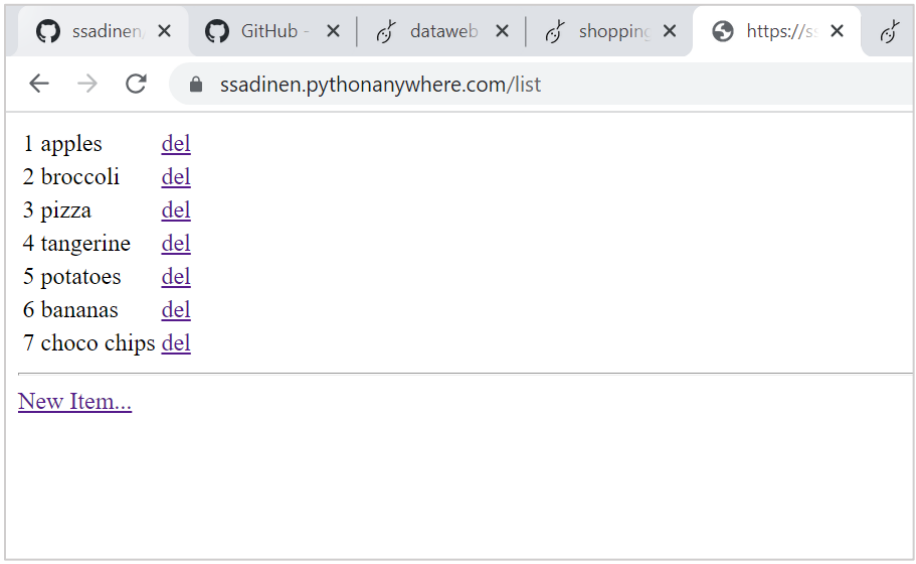


- Now I went to route /delete/8 to delete item with id 8. As per below screenshot I have deleted item 8 which was bread.

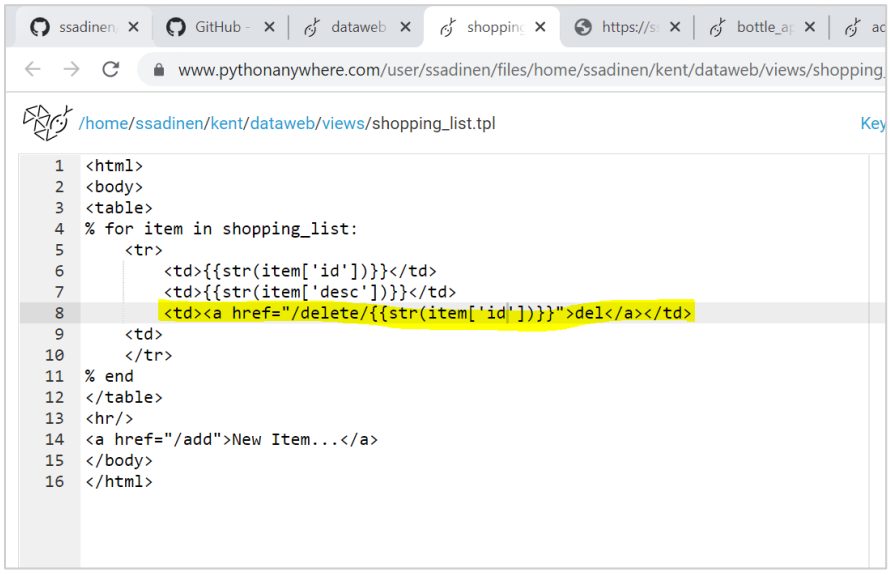


- Now rather than deleting an item by changing the route in web url, I have added href shopping list template to add delete in the shopping list web page itself

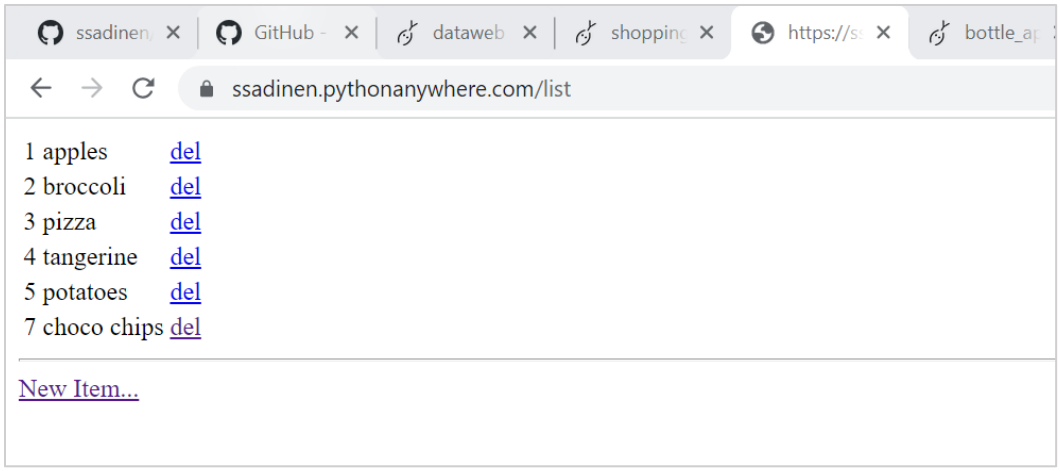
- After saving the above template and reloading the page I ran the web app again and could see delete option across each of the items however this delete is not associated with respective id's of items.



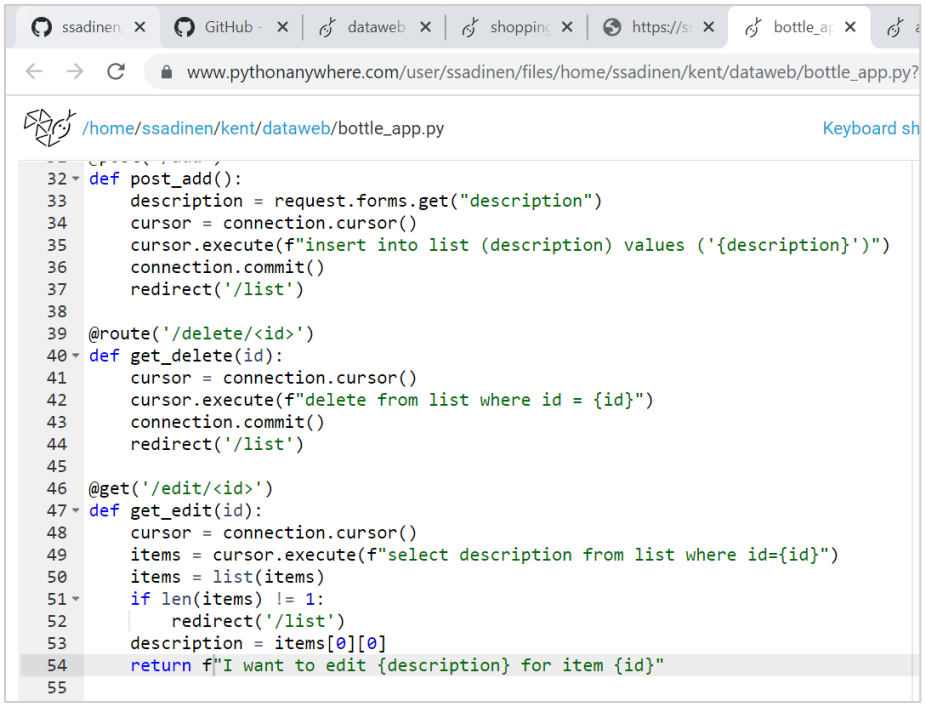
- To associate the respective id's of items in delete, I've modified the href of delete in shopping list template as below –



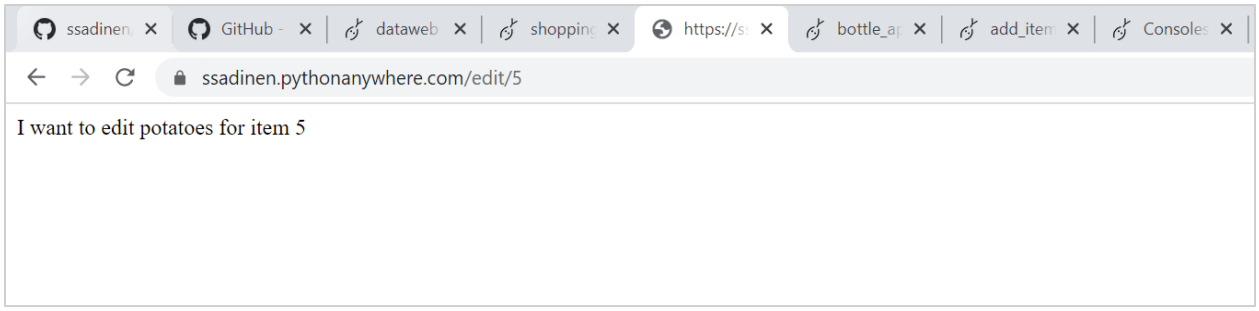
- I then saved the above template and reloaded the web page. After which I could see the id's associated for each of the items in the shopping list. I then deleted bananas which is of id 6.



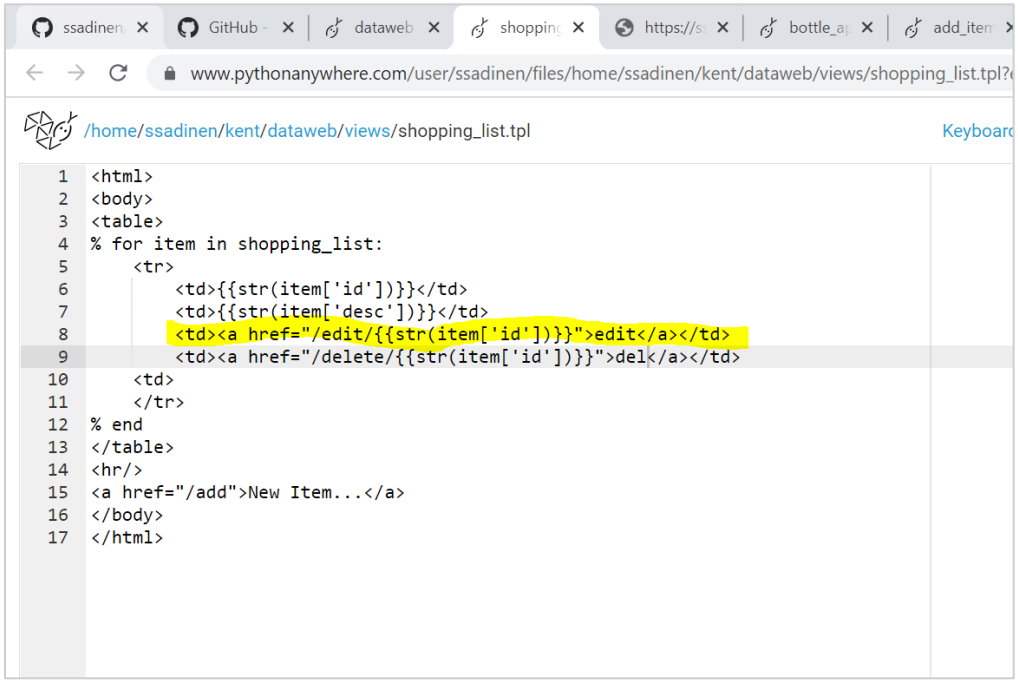
- In bottle_app.py, I have then added a /get route for edit in order to edit the items description in shopping list.



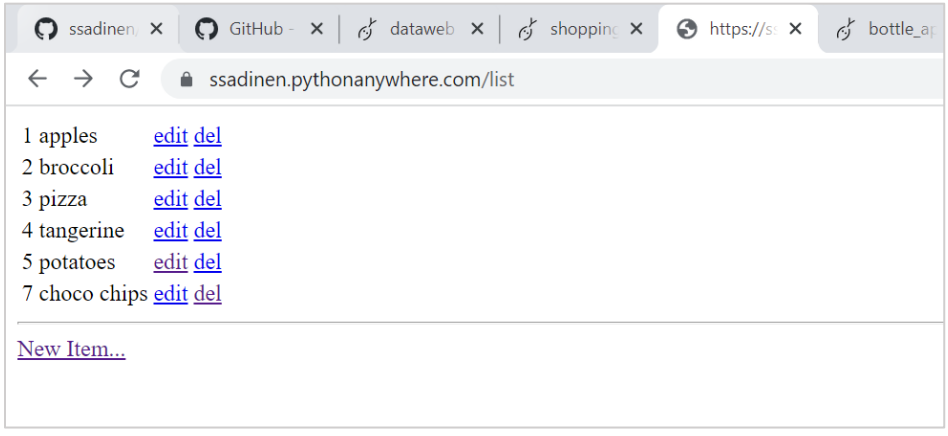
- After saving the above and reloading the web app, I accessed the web page with route edit/5 in order to edit item 5.



- I then modified the shopping list template to directly edit items from shopping list oage rather than going to new route.



- After saving above template and reloading the web app, I could see the edit option across each of the shopping list items in the web app.

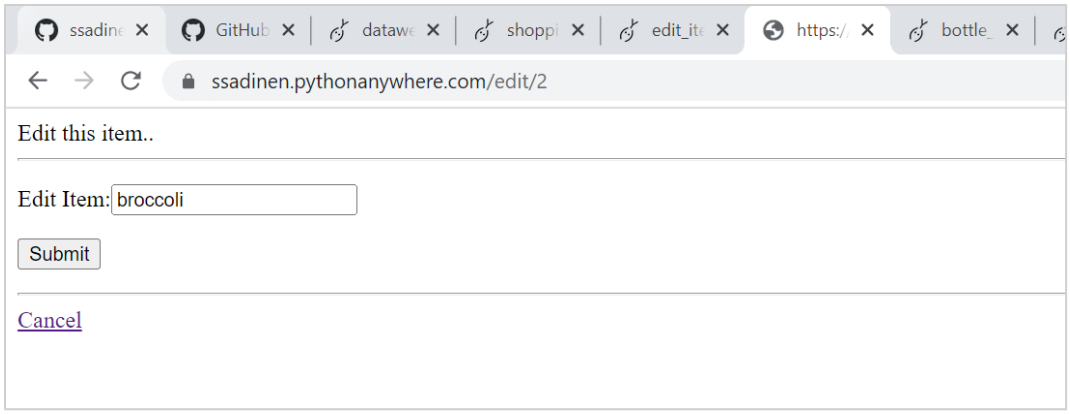


- In order to edit item in the list completely we need to add post method. Hence created a new template of edit_item and included the form action.

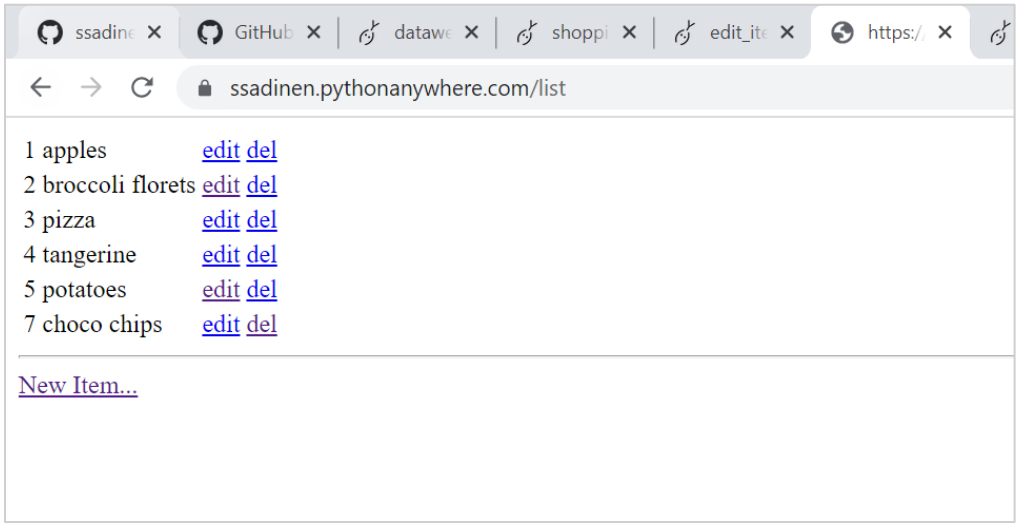
- After saving the edit_item template, I have then added post route in bottle_app.py inoder to post the items edited in the shopping list web app.

```
@route('/delete/<id>')
def get_delete(id):
    cursor = connection.cursor()
    cursor.execute(f"delete from list where id = {id}")
    connection.commit()
    redirect('/list')

@get('/edit/<id>')
def get_edit(id):
    cursor = connection.cursor()
    items = cursor.execute(f"select description from list where id={id}")
    items = list(items)
    if len(items) != 1:
        redirect('/list')
    description = items[0][0]
    return template("edit_item.tpl", id=id, description=description)

@post('/edit/<id>')
def post_edit(id):
    description = request.forms.get("description")
    cursor = connection.cursor()
    cursor.execute(f"update list set description = '{description}' where id={id}")
    connection.commit()
    redirect('/list')
```

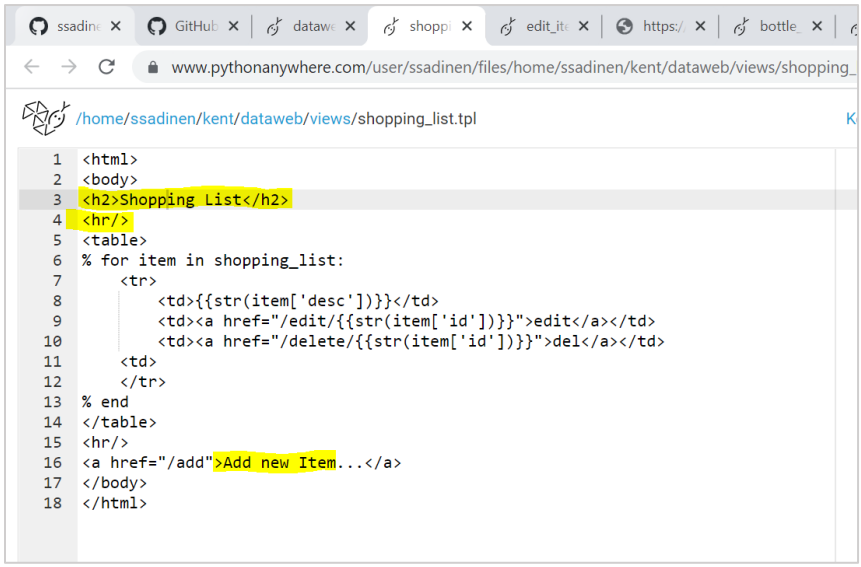
- After saving the bottle_app.py, I have reloaded the web app and then edited the item broccoli to broccoli florets and clicked on submit



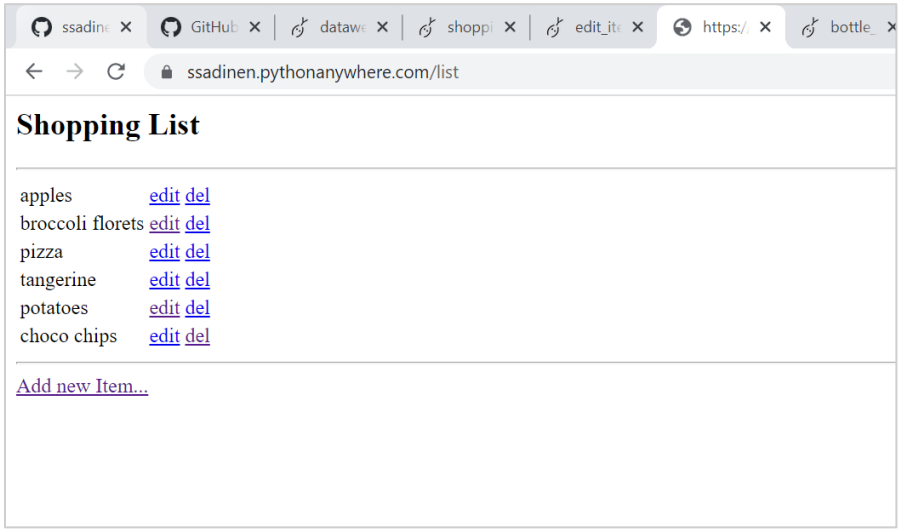
- As we can see below I can edit the items description in the shopping list.



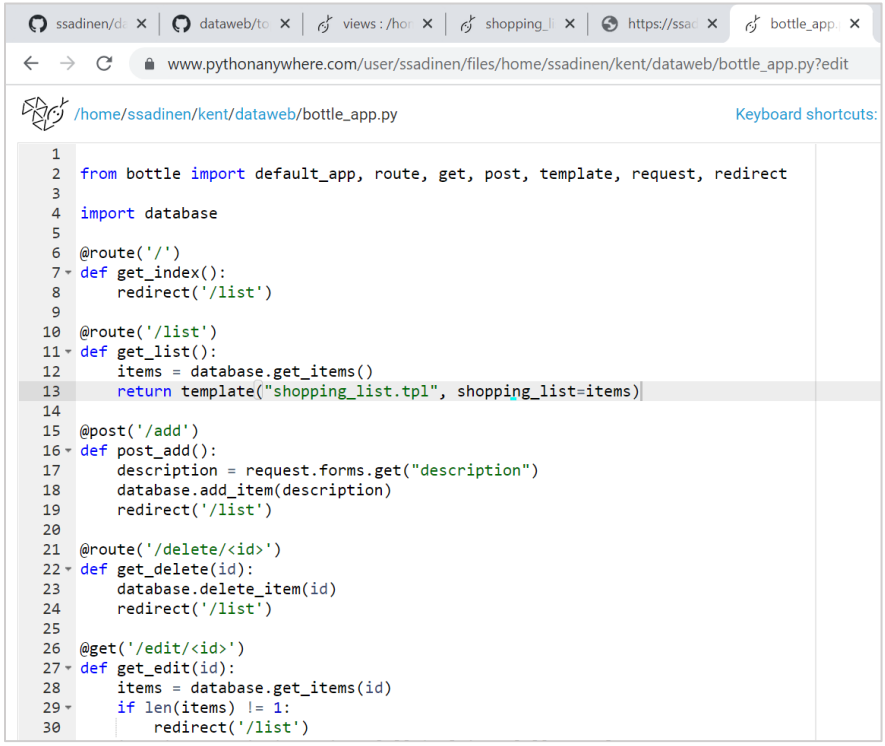
- I then added heading Shopping List and change Add item to Add new item in shopping list template



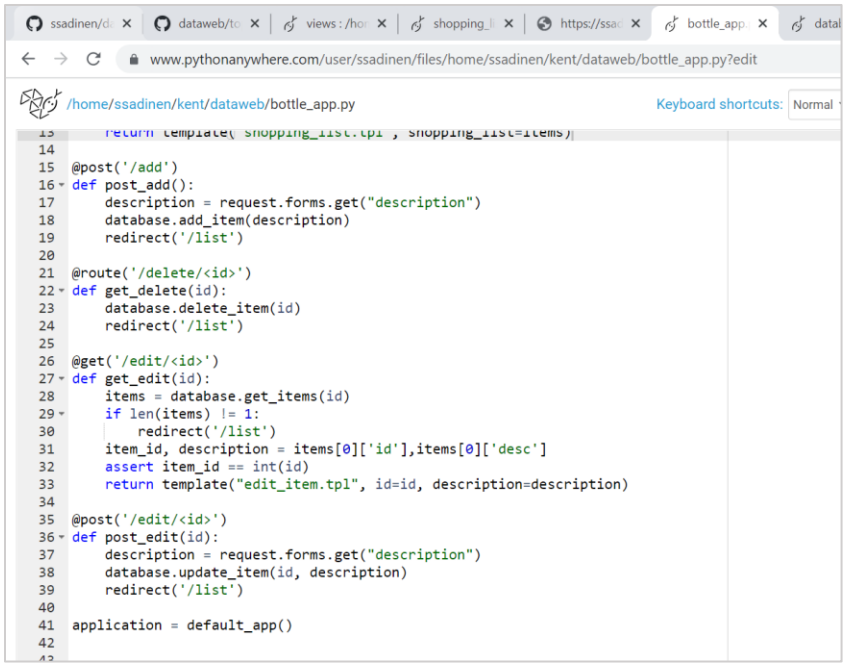
- After saving the above template and reloading the web page, I can see the heading of and the change to Add New item in the web page.



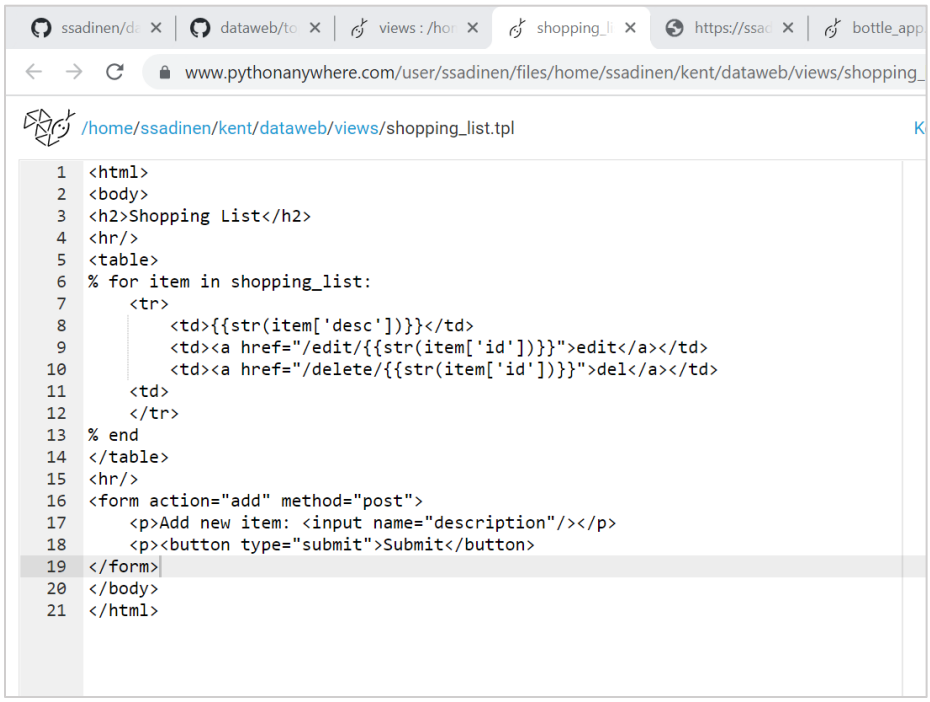
- Deleted the routes for hi, bye. Below shows the lines in bottle_app.py for routes - /list, /add, /delete



- Below screenshot shows lines in bottle_app.py for /update.



- I then updated shopping_list.tpl to include add new item in the same page as the shopping list.



- I've created database.py to separate the database related code from app related code. To test if each of get, add, delete, update are working fine, I've included below test_get_items() to test if get items is working fine.

/home/ssadinen/kent/dataweb/database.py Keyboard shortcuts: Normal

```python
import sqlite3

connection = sqlite3.connect("shopping_list.db")

def get_items(id=None):
    cursor = connection.cursor()
    if id:
        items = cursor.execute(f"select id, description from list where id={id}")
    else:
        items = cursor.execute("select id, description from list")
    items = list(items)
    items = [ {'id':item[0] ,'desc':item[1]} for item in items ]
    return items

def test_get_items():
    print("testing get_items....")
    items = get_items()
    assert type(items) is list
    assert len(items) > 0
    assert type(items[0]) is dict
    assert 'id' in items[0].keys()
    assert 'desc' in items[0].keys()
    assert type(items[0]['id']) is int
    assert type(items[0]['desc']) is str
    pass

def add_item(description):
    cursor = connection.cursor()
    cursor.execute(f"insert into list (description) values ('{description}')")
    connection.commit()
```

- I've included below test_add_item() to test if add item is working fine.

www.pythonanywhere.com/user/ssadinen/files/home/ssadinen/kent/dataweb/database.py?edit

/home/ssadinen/kent/dataweb/database.py Keyboard shortcuts: Normal

```python
def add_item(description):
    cursor = connection.cursor()
    cursor.execute(f"insert into list (description) values ('{description}')")
    connection.commit()

import time

def random_string():
    return str(time.time())

def test_add_item():
    print("testing add_item....")
    description = random_string()
    add_item(description)
    items = get_items()
    item = items[-1]
    assert description == item['desc']

def delete_item(id):
    cursor = connection.cursor()
    cursor.execute(f"delete from list where id = {id}")
    connection.commit()

def test_delete_item():
    print("testing delete_item....")
    description = random_string()
    add_item(description)
    items = get_items()
    item = items[-1]
```

- I've included below test_delete_item() to test if delete item is working fine

/home/ssadinen/kent/dataweb/database.py Keyboard shortcuts: Normal

```
def test_delete_item():
    print("testing delete_item....")
    description = random_string()
    add_item(description)
    items = get_items()
    item = items[-1]
    assert description == item['desc']
    delete_item(item['id'])
    items = get_items()
    for item in items:
        assert description != item['desc']

def update_item(id, description):
    cursor = connection.cursor()
    cursor.execute(f"update list set description = '{description}' where id={id}")
    connection.commit()

def test_update_item():
    print("testing update_item....")
    description = random_string()
    add_item(description)
    items = get_items()
    item = items[-1]
    id = str(item['id'])
    description = item['desc']
    new_description = description.replace("1","9").replace(".",":")
    update_item(id, new_description)
    items = get_items()
    new_found = False
```

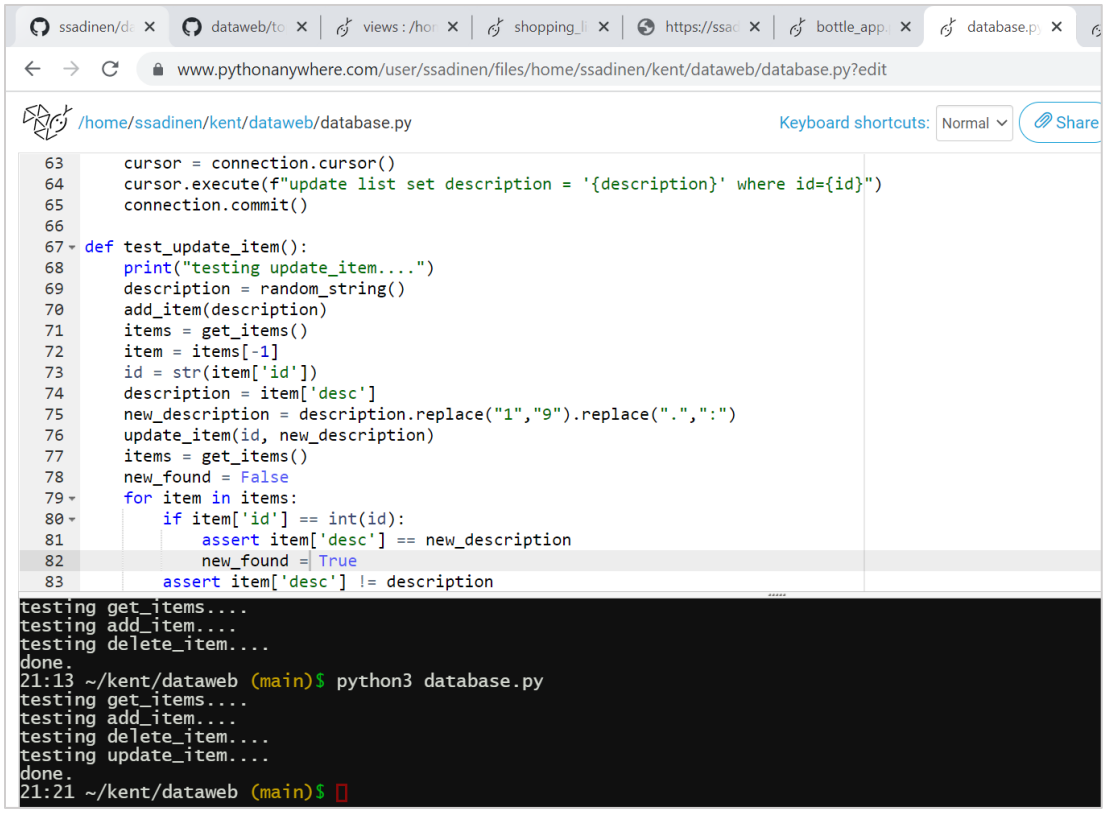
- I've included test_update_item() to test if update item is working fine.

www.pythonanywhere.com/user/ssadinen/files/home/ssadinen/kent/dataweb/database.py?edit

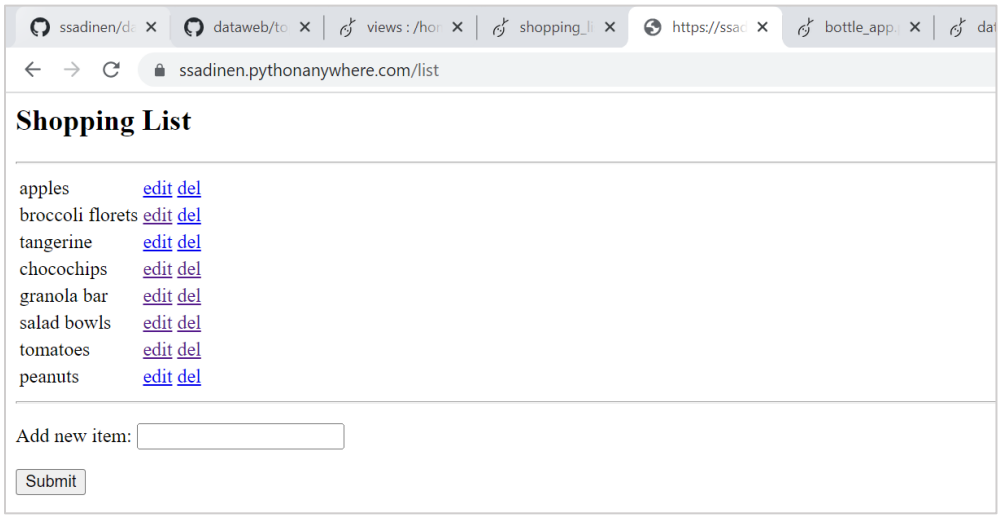
/home/ssadinen/kent/dataweb/database.py Keyboard shortcuts: Normal

```
    cursor = connection.cursor()
    cursor.execute(f"update list set description = '{description}' where id={id}")
    connection.commit()

def test_update_item():
    print("testing update_item....")
    description = random_string()
    add_item(description)
    items = get_items()
    item = items[-1]
    id = str(item['id'])
    description = item['desc']
    new_description = description.replace("1","9").replace(".",":")
    update_item(id, new_description)
    items = get_items()
    new_found = False
    for item in items:
        if item['id'] == int(id):
            assert item['desc'] == new_description
            new_found = True
        assert item['desc'] != description
    assert new_found


if __name__ == "__main__":
    test_get_items()
    test_add_item()
    test_delete_item()
    test_update_item()
    print("done.")
```

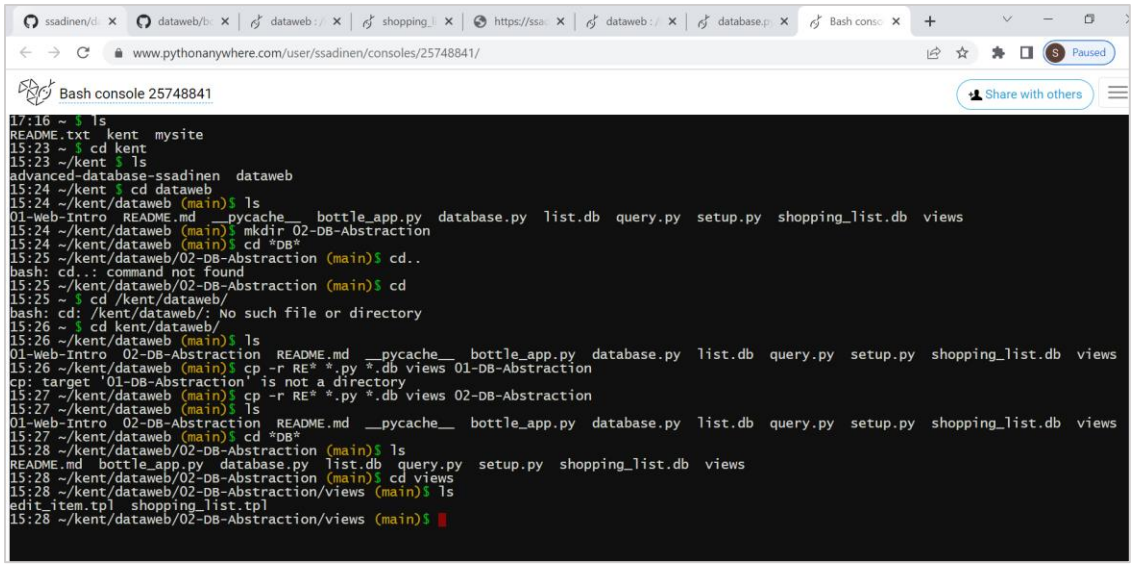
- I then opened the bash console in database.py and ran python3 database.py to check if the testing rows are printed.



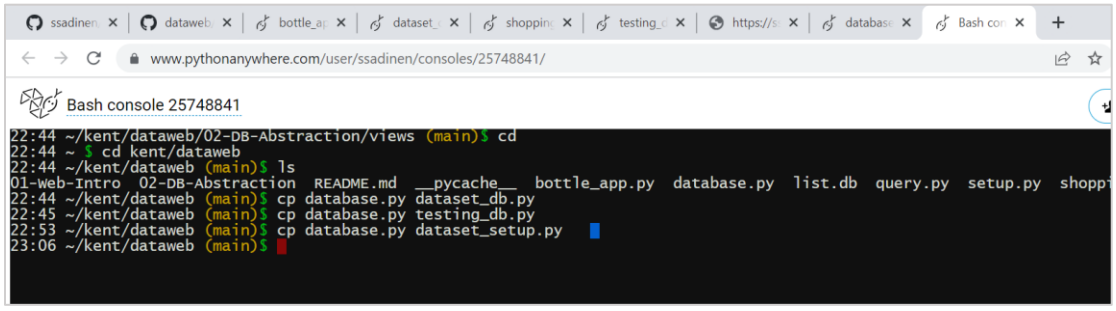
- Now I've saved the above database.py, bottle_app.py and the reloaded the webpage to check if the web app is displaying all the shopping list items, form correctly. I've added granola bar, salad bowls, tomatoes, peanuts for testing at various levels changes of get, add, update, delete functions.



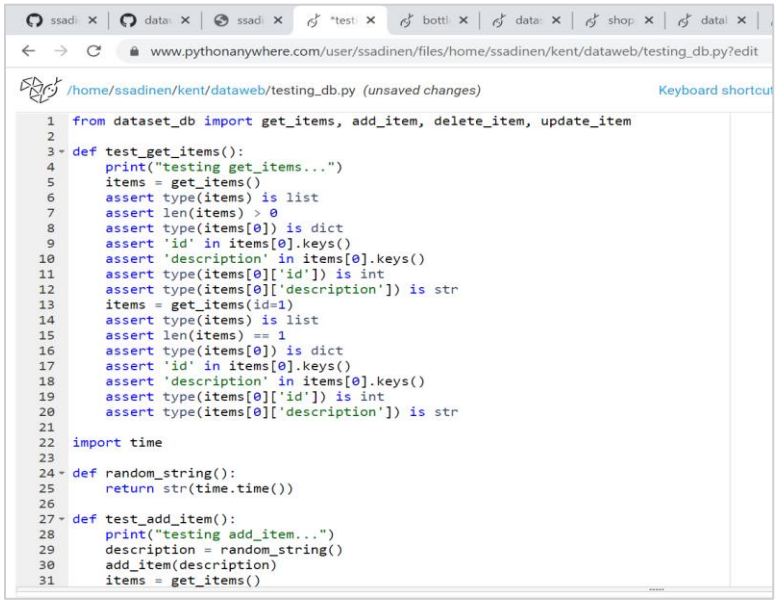
- I then copied all the files to a new directory 02-DB-Extraction.



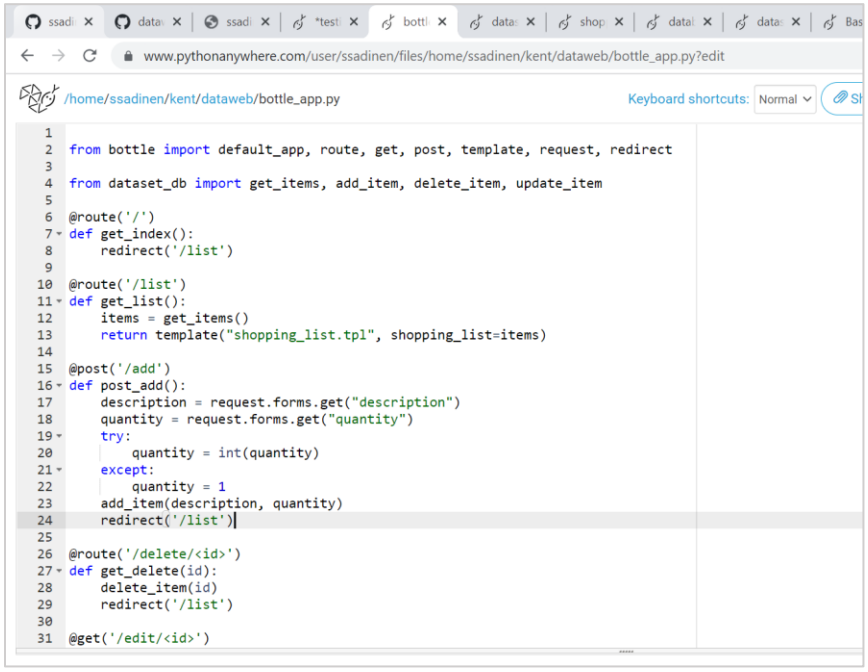
- I then copied database.py into 3 different files as shown in below screenshot.



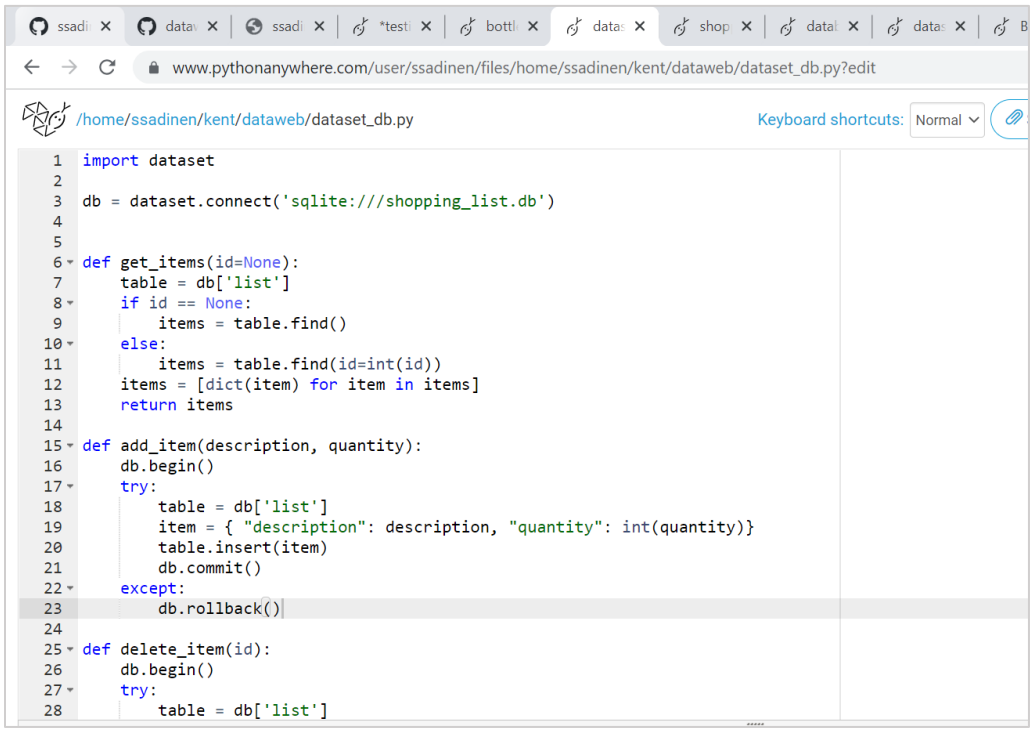
- I then moved all the testing re. code from bottle_app.py into testing_db.py file as shown in below screenshot by importing get_items, add_item, update_item, delete_item from dataset_db



- In bottle_app.py I've then imported get_items, add_item, update_item, delete_item from dataset_db.



- In dataset_db.py, I've added dataset connection and modified old references.



- In shopping list template I've added the data item and form action for quantity.

www.pythonanywhere.com/user/ssadinen/files/home/ssadinen/kent/dataweb/views/shopping_list.tpl?edit

/home/ssadinen/kent/dataweb/views/shopping_list.tpl

```
<html>
<body>
<h2>Shopping List</h2>
<hr/>
<table>
% for item in shopping_list:
    <tr>
        <td>{{str(item['description'])}}</td>
        <td>{{str(item['quantity'])}}</td>
        <td><a href="/edit/{{str(item['id'])}}">edit</a></td>
        <td><a href="/delete/{{str(item['id'])}}">del</a></td>
    <td>
    </tr>
% end
</table>
<hr/>
<form action="add" method="post">
    <p>Add new item: <input name="description"/></p>
    <p>Quantity: <input name="quantity"/></p>
    <p><button type="submit">Submit</button>
</form>
</body>
</html>
```

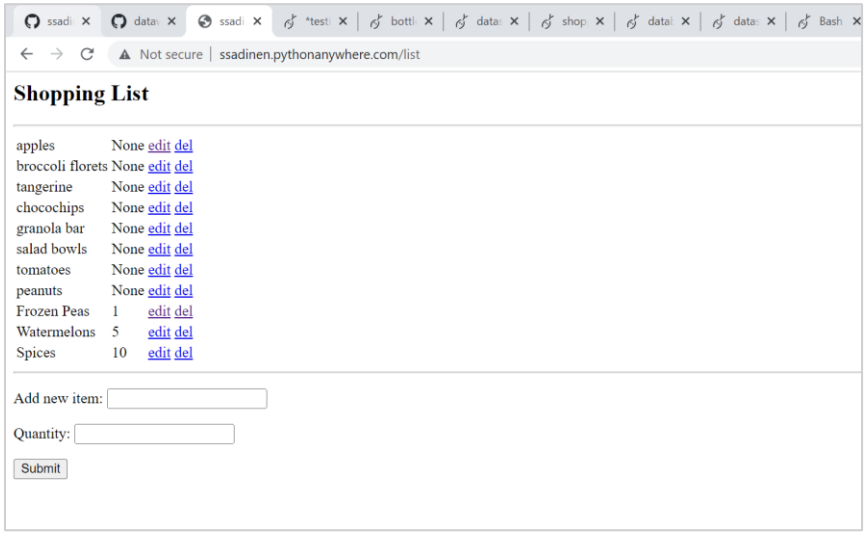
- In dataset_setup.py, I've imported the dataset, added dataset connection.

www.pythonanywhere.com/user/ssadinen/files/home/ssadinen/kent/dataweb/dataset_setup.py?edit

/home/ssadinen/kent/dataweb/dataset_setup.py

```
import dataset

db = dataset.connect('sqlite:///shopping_list.db')

try:
    db['list'].drop()
except:
    pass

db.begin()
try:
    table = db['list']
    items = [
        { "description": 'apples' },
        { "description": 'broccoli' },
        { "description": 'pizza' },
        { "description": 'tangerine' },
        { "description": 'potatoes' }
        ]
    table.insert_many(items)
    db.commit()
except:
    db.rollback()
```

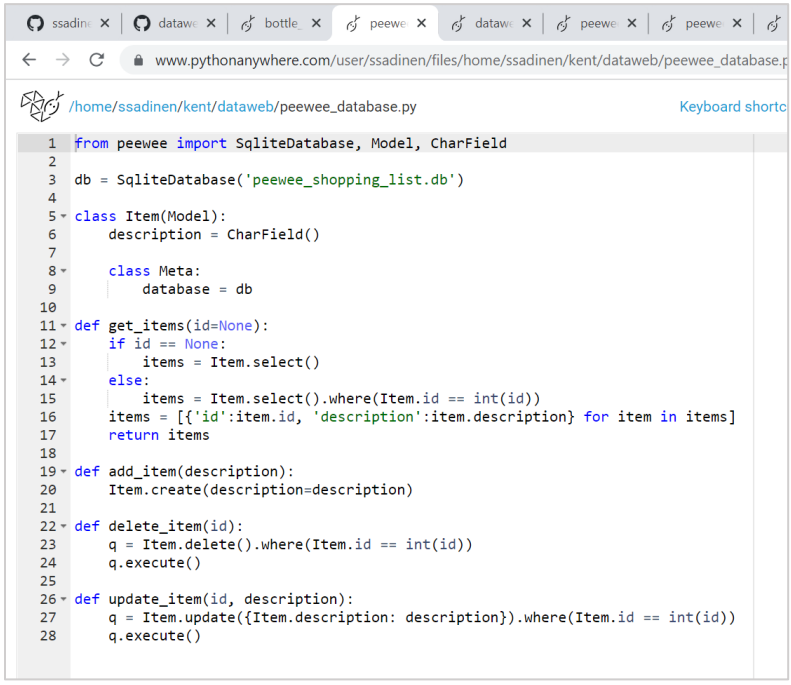
- After saving the above files, I've reloaded the webpage and I could add the quantity for new items as shown in below screenshot.



- I've copied all the files edited to a new directory 03-Datasets

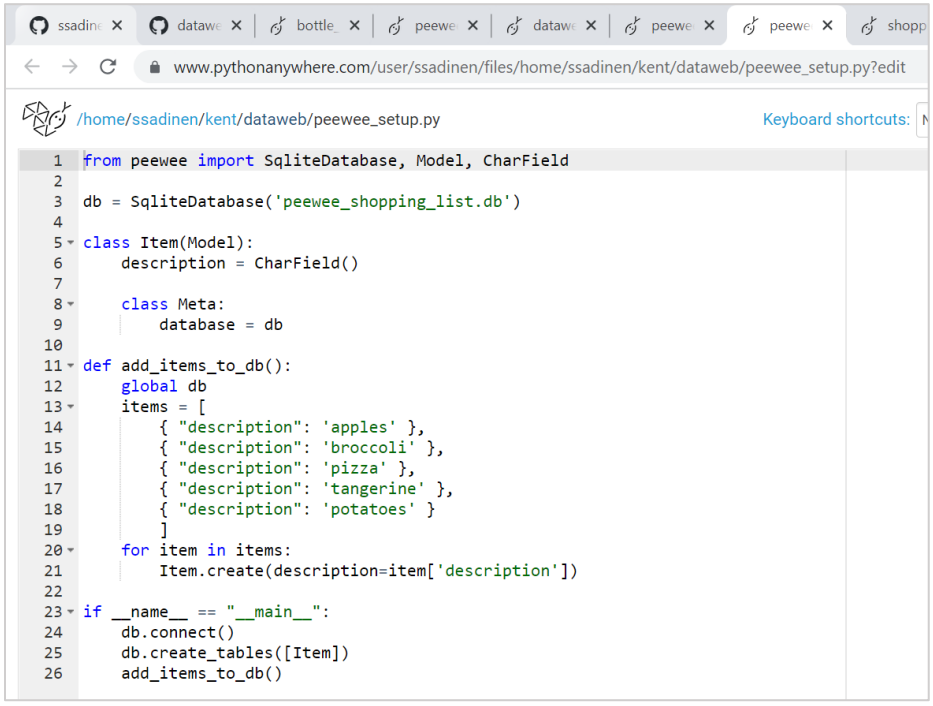
```
03:13 ~/kent/dataweb (main)$ cp -r *.md *.py sho* views 03-Datasets
03:13 ~/kent/dataweb (main)$ cd 03*
03:14 ~/kent/dataweb/03-Datasets (main)$ ls
README.md      database.py    dataset_setup.py  setup.py          shopping_list.db-shm  testing_db.py
bottle_app.py  dataset_db.py  query.py          shopping_list.db  shopping_list.db-wal  views
03:14 ~/kent/dataweb/03-Datasets (main)$
```

- I then created peewee_database.py file and modified the code as shown below –

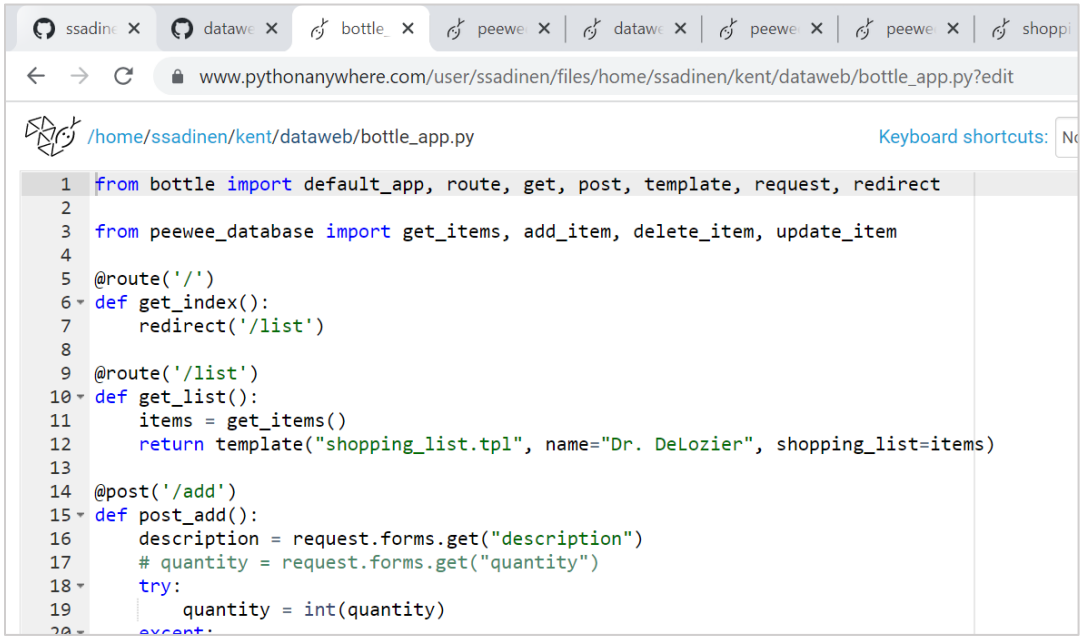



```python
from peewee import SqliteDatabase, Model, CharField

db = SqliteDatabase('peewee_shopping_list.db')

class Item(Model):
    description = CharField()

    class Meta:
        database = db

def get_items(id=None):
    if id == None:
        items = Item.select()
    else:
        items = Item.select().where(Item.id == int(id))
    items = [{'id':item.id, 'description':item.description} for item in items]
    return items

def add_item(description):
    Item.create(description=description)

def delete_item(id):
    q = Item.delete().where(Item.id == int(id))
    q.execute()

def update_item(id, description):
    q = Item.update({Item.description: description}).where(Item.id == int(id))
    q.execute()
```

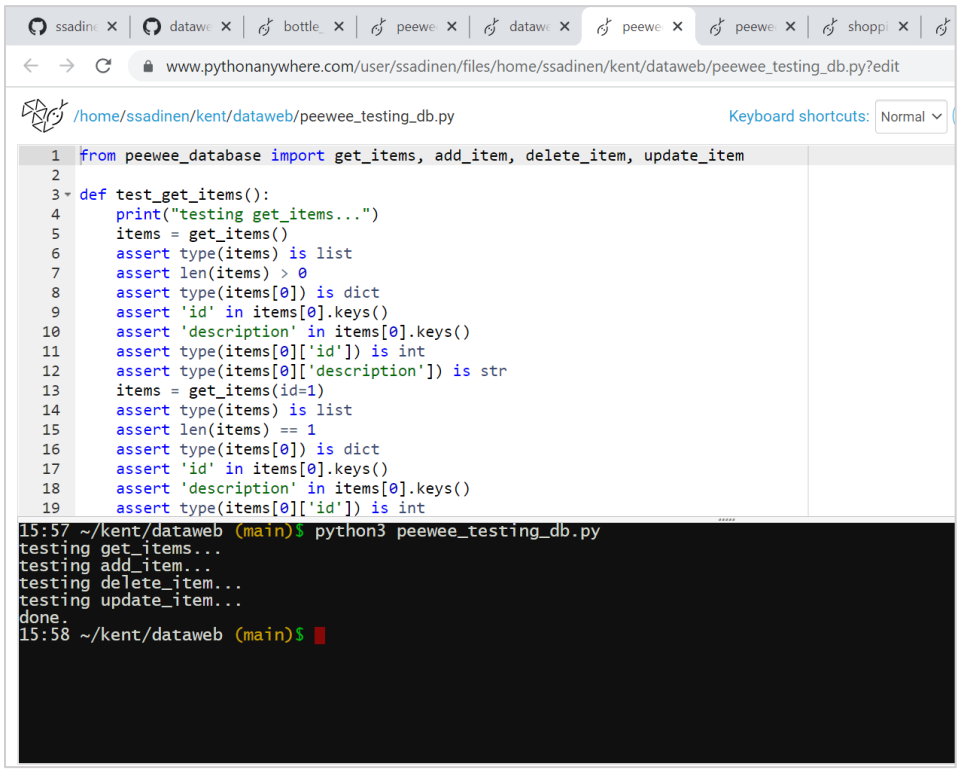
- I then created peewee_setup.py file and modified the code by importing required contents from peewee.



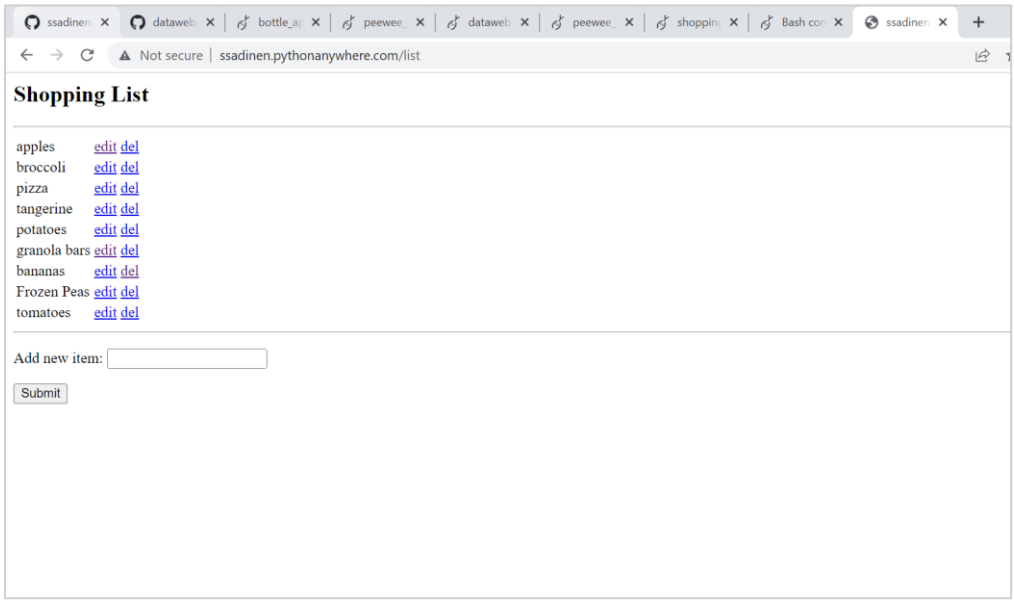
- I then updated bottle_app.py and imported get_items, add_item, delete_item, update_item from peewee_database.



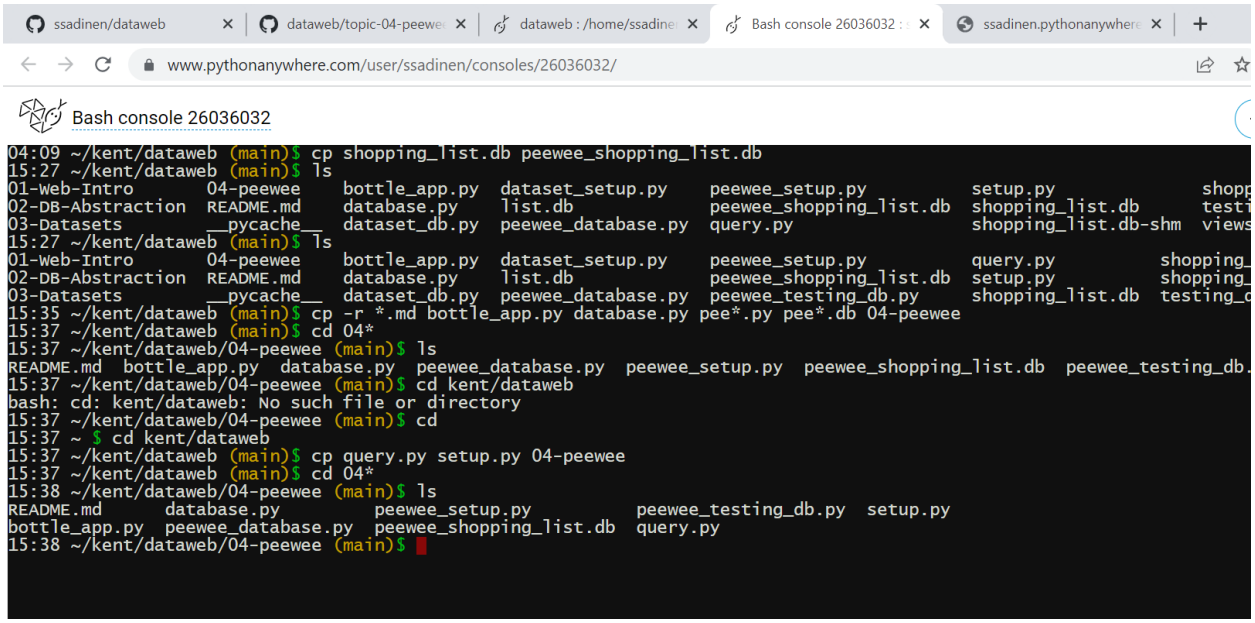
- Copied testing_db.py to peewee_testing_db.py and modified to import get_items, add_item, delete_item, update_item from peewee_database.
  I then saved the file after making required changes and opened bash console. I then ran python3 peewee_testing.db command to test get_items, add_item, delete_item, update_item.



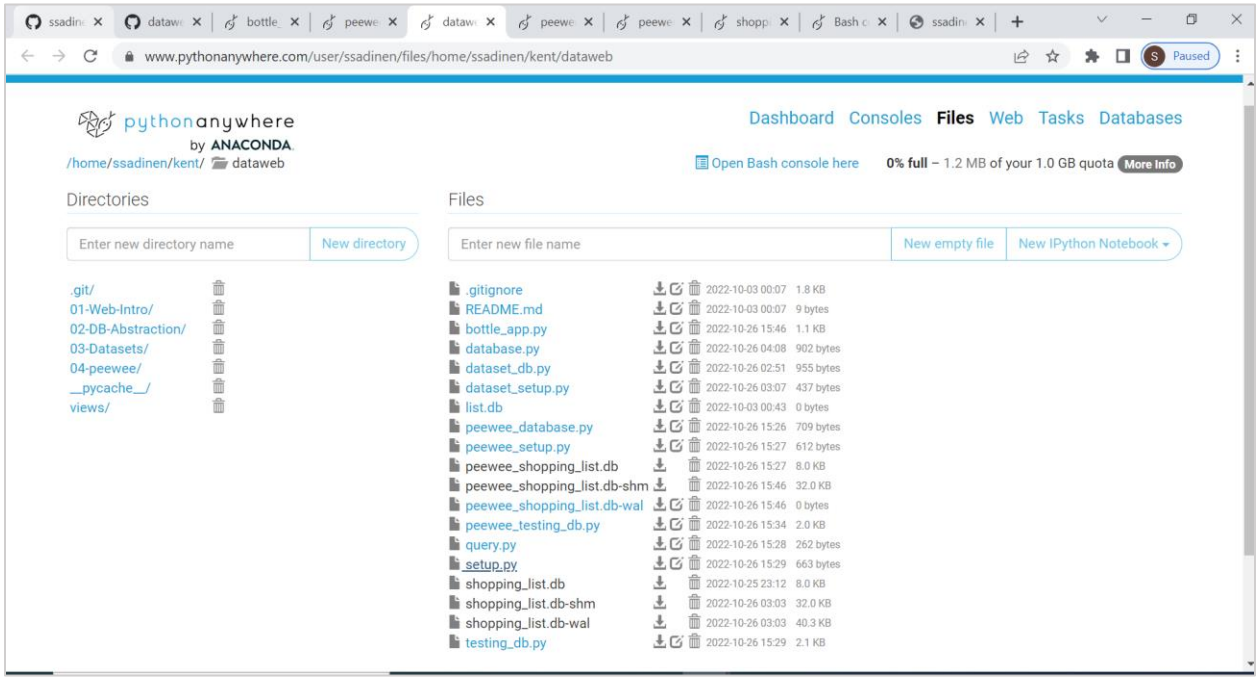
- I then saved all the above files and reloaded the website. I could see all the contents in web page as expected.



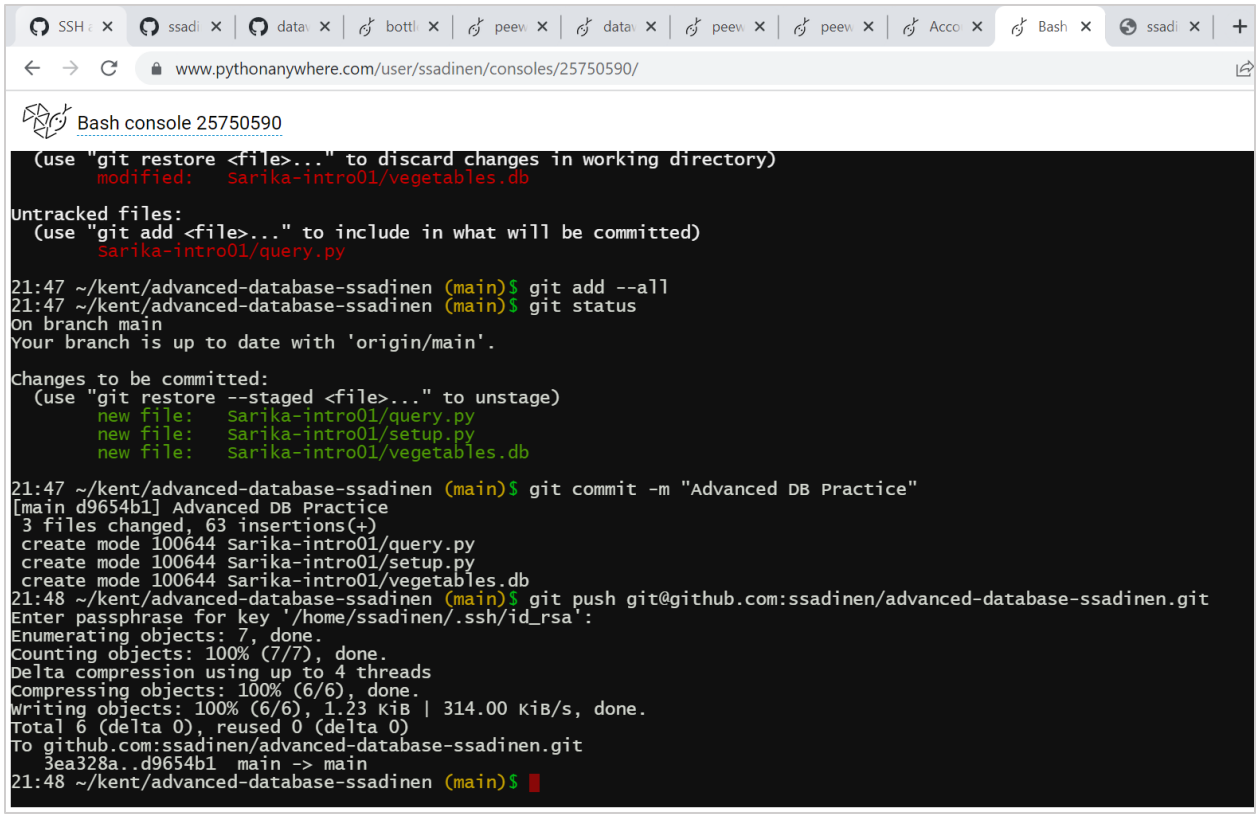
- I've then copied all the new files into a new directory 04-peewee

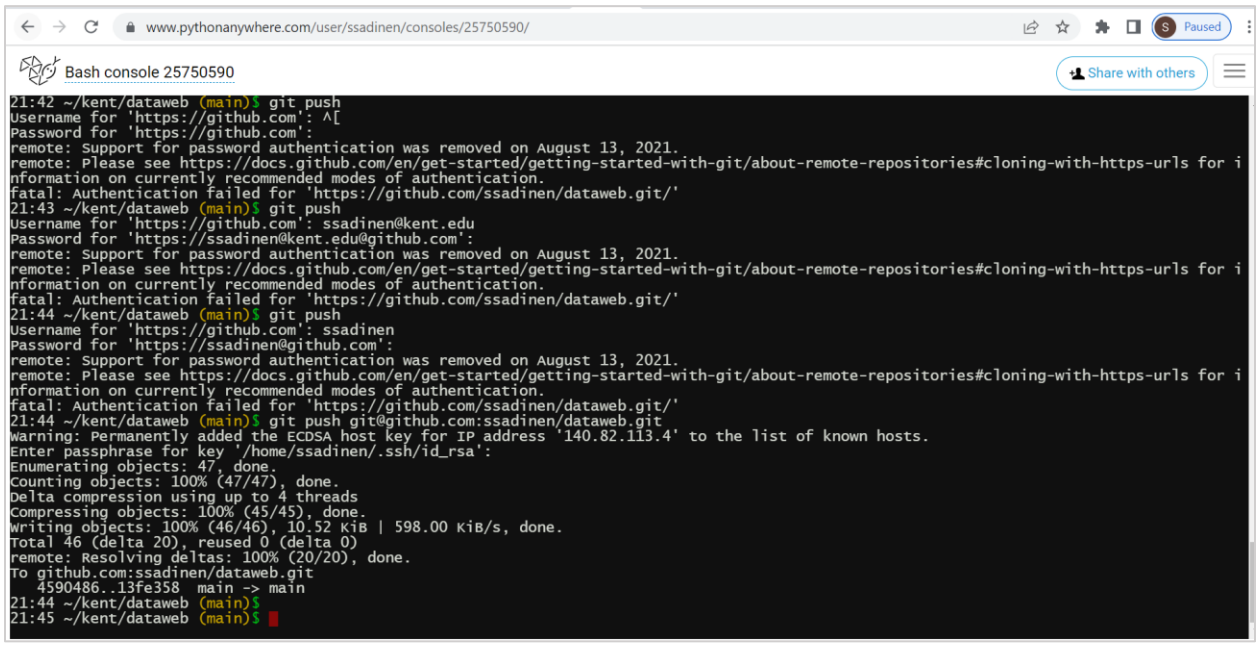


- Below are the files and directories I've created for my practice of first 4 topics of dataweb.



- I have then committed and pushed all contents to my github by generating ssh key and adding it into my git.


www.pythonanywhere.com/user/ssadinen/consoles/25750590/
Bash console 25750590
(use "git restore <file>..." to discard changes in working directory)
modified: Sarika-intro01/vegetables.db
Untracked files:
(use "git add <file>..." to include in what will be committed)
Sarika-intro01/query.py
21:47 ~/kent/advanced-database-ssadinen (main)$ git add --all
21:47 ~/kent/advanced-database-ssadinen (main)$ git status
On branch main
Your branch is up to date with 'origin/main'.
Changes to be committed:
(use "git restore --staged <file>..." to unstage)
new file: Sarika-intro01/query.py
new file: Sarika-intro01/setup.py
new file: Sarika-intro01/vegetables.db
21:47 ~/kent/advanced-database-ssadinen (main)$ git commit -m "Advanced DB Practice"
[main d9654b1] Advanced DB Practice
3 files changed, 63 insertions(+)
create mode 100644 Sarika-intro01/query.py
create mode 100644 Sarika-intro01/setup.py
create mode 100644 Sarika-intro01/vegetables.db
21:48 ~/kent/advanced-database-ssadinen (main)$ git push git@github.com:ssadinen/advanced-database-ssadinen.git
Enter passphrase for key '/home/ssadinen/.ssh/id_rsa':
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 1.23 KiB | 314.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0)
To github.com:ssadinen/advanced-database-ssadinen.git
3ea328a..d9654b1 main -> main
21:48 ~/kent/advanced-database-ssadinen (main)$


www.pythonanywhere.com/user/ssadinen/consoles/25750590/
Paused
Bash console 25750590
Share with others
21:42 ~/kent/dataweb (main)$ git push
Username for 'https://github.com': ^[
Password for 'https://github.com':
remote: Support for password authentication was removed on August 13, 2021.
remote: Please see https://docs.github.com/en/get-started/getting-started-with-git/about-remote-repositories#cloning-with-https-urls for information on currently recommended modes of authentication.
fatal: Authentication failed for 'https://github.com/ssadinen/dataweb.git/'
21:43 ~/kent/dataweb (main)$ git push
Username for 'https://github.com': ssadinen@kent.edu
Password for 'https://ssadinen@kent.edu@github.com':
remote: Support for password authentication was removed on August 13, 2021.
remote: Please see https://docs.github.com/en/get-started/getting-started-with-git/about-remote-repositories#cloning-with-https-urls for information on currently recommended modes of authentication.
fatal: Authentication failed for 'https://github.com/ssadinen/dataweb.git/'
21:44 ~/kent/dataweb (main)$ git push
Username for 'https://github.com': ssadinen
Password for 'https://ssadinen@github.com':
remote: Support for password authentication was removed on August 13, 2021.
remote: Please see https://docs.github.com/en/get-started/getting-started-with-git/about-remote-repositories#cloning-with-https-urls for information on currently recommended modes of authentication.
fatal: Authentication failed for 'https://github.com/ssadinen/dataweb.git/'
21:44 ~/kent/dataweb (main)$ git push git@github.com:ssadinen/dataweb.git
Warning: Permanently added the ECDSA host key for IP address '140.82.113.4' to the list of known hosts.
Enter passphrase for key '/home/ssadinen/.ssh/id_rsa':
Enumerating objects: 47, done.
Counting objects: 100% (47/47), done.
Delta compression using up to 4 threads
Compressing objects: 100% (45/45), done.
Writing objects: 100% (46/46), 10.52 KiB | 598.00 KiB/s, done.
Total 46 (delta 20), reused 0 (delta 0)
remote: Resolving deltas: 100% (20/20), done.
To github.com:ssadinen/dataweb.git
4590486..13fe358 main -> main
21:44 ~/kent/dataweb (main)$
21:45 ~/kent/dataweb (main)$