

HW2

<http://jayasrimaditai.pythonanywhere.com/> - [Web application link](#)

For this Homework I have created a student registration page. As part of demonstration, I have provided the screenshots only for listing the students from the database. Also tested CRUD operations. For further operations, please check the github link.

<https://github.com/JayasriMaditai/dataweb/tree/main/HW2>

Q1) 1. Direct access to the data using the sqlite3 SQL library with connections and cursors

Fig 1: Files in the Direct Access Folder

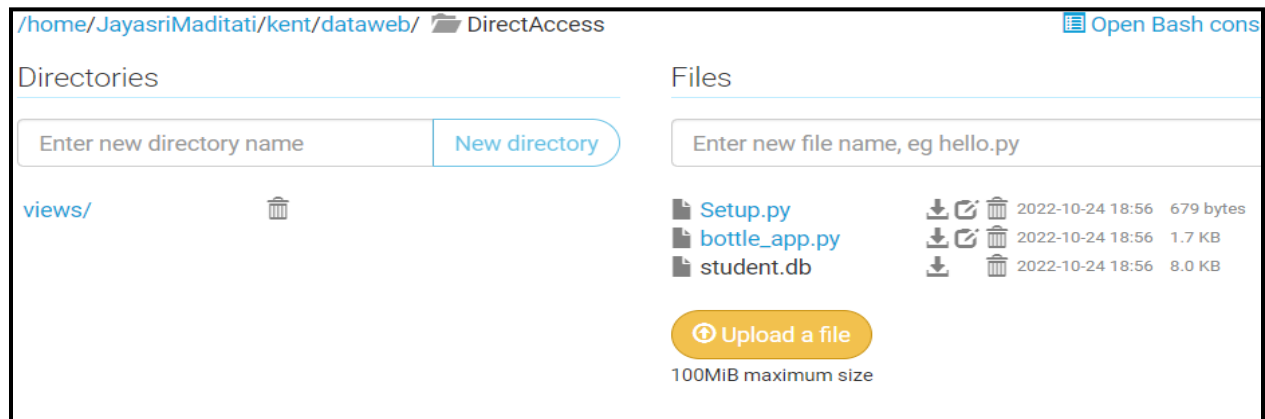


Fig 2: Below is a snippet of the code using SQLite3 SQL library to list all the students in the database for the web application

```
/home/JayasriMaditai/kent/dataweb/DirectAccess/bottle_app.py

# Web application for student database (Direct Access)
from bottle import default_app, route, template, get, post, request, redirect
import sqlite3

connection = sqlite3.connect("student.db")
@route('/')
def get_index():
    redirect('/list')

@route('/list')
def get_list():
    cursor = connection.cursor()
    rows = cursor.execute("select * from student")
    rows = list(rows)
    rows = [{"id":row[0], "Firstname":row[1], "Lastname":row[2], "City":row[3]} for row in rows]
    return template('views/student_list.tpl', items = rows )
```

Q2) Indirect access to the data using the datasets library

Fig 3: The below screenshot lists all the files of the Indirect access folder

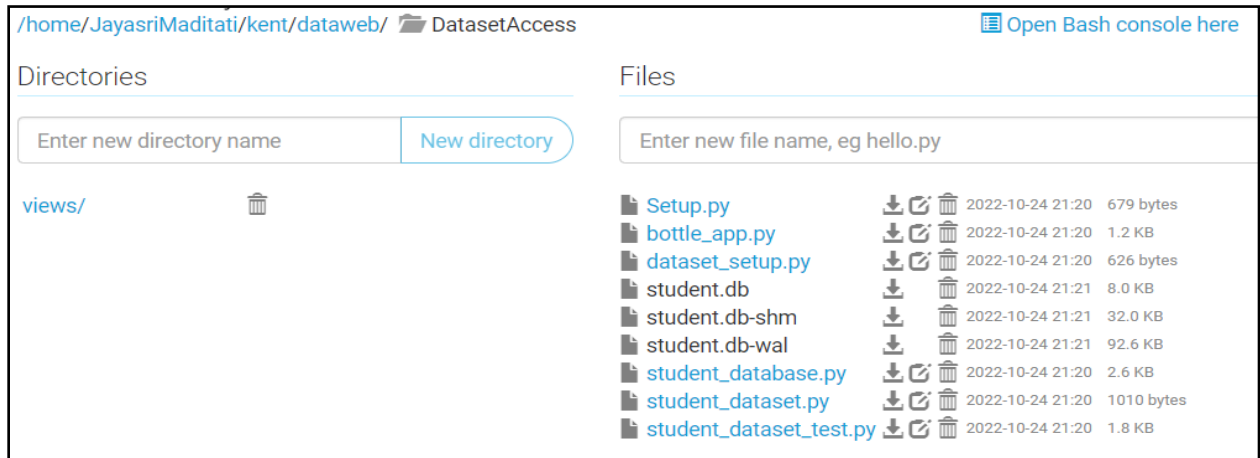


Fig 4: Below is a snippet of the code using Indirect access to the data using the datasets library to list all the students in the database

```
/home/JayasriMadiati/kent/dataweb/DatasetAccess/bottle_app.py

# Web application using DB abstraction layer - Dataset Database
from bottle import default_app, route, template, get, post, request, redirect

from student_dataset import get_list, add_student, delete_student, update_city

@route('/')

def get_index():
    redirect('/list')

@route('/list')
def get_student():
    rows = get_list()
    return template('views/student_list.tpl', items = rows )
```

Fig 5: The below snippet details the data abstraction layer using the datasets library. (Indirect access)

```
/home/JayasriMeditati/kent/dataweb/DatasetAccess/student_dataset.py


import dataset

db = dataset.connect("sqlite:///student.db")


def get_list(id=None):
    table = db['student']
    if id:
        rows = table.find(id = int(id))
    else:
        rows = table.find()
    rows = [dict(row) for row in rows]
    return rows
```

Q3) ORM access to the data using the Peewee ORM library




















Fig 6: Lists all the files of ORM access

/home/JayasriMeditati/kent/dataweb/HW2/  Peewee ORM [Open Bash console here](#)

Directories

[views/](#) 

Files

 bottle_app.py	  	2022-10-27 00:19	1.3 KB
 peewee_setup.py	  	2022-10-27 00:19	863 bytes
 peewee_student.db	 	2022-10-27 00:19	8.0 KB
 peewee_student_test.py	  	2022-10-27 01:18	1.8 KB
 student_orm.py	  	2022-10-27 00:19	820 bytes

100MiB maximum size

Fig 7: The below snippet details the data abstraction layer using the peewee ORM library. (Indirect access)

```
/home/JayasriMadiati/kent/dataweb/Peewee ORM/bottle_app.py

# Web application using Peewee ORM
from bottle import default_app, route, template, get, post, request, redirect

from student_orm import get_list, add_student, delete_student, update_city

@route('/')

def get_index():
    redirect('/list')

@route('/list')
def get_student():
    rows = get_list()
    return template('views/student_list.tpl', items = rows )
```

Fig 8: Below is a snippet of the web application code using ORM access to the data using the Peewee library to list all the students in the database

```
/home/JayasriMadiati/kent/dataweb/Peewee ORM/student_orm.py Keyboard

from peewee import SqliteDatabase, Model, CharField

db = SqliteDatabase('peewee_student.db')

class Student(Model):
    Firstname = CharField()
    Lastname = CharField()
    City = CharField()
    class Meta:
        database = db

def get_list(id=None):
    if id == None:
        rows = Student.select()
    else:
        rows = Student.select().where(Student.id==int(id))
    rows = [{ 'id':row.id, 'Firstname':row.Firstname, 'Lastname':row.Lastname, 'City':row.City} for row in rows]
    return rows
```

Web Application Demo:

List all the students

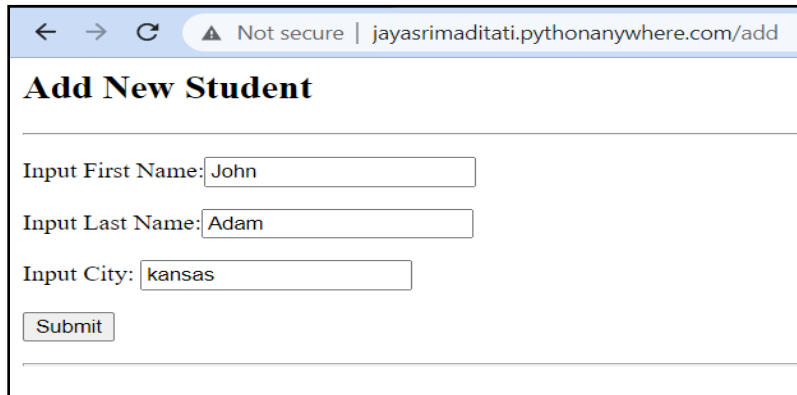


The screenshot shows a web browser window with the address bar displaying "Not secure | jayasrimaditai.pythonanywhere.com/list". The page title is "Student Registration". Below the title is a table with the following data:

FirstName	LastName	City	Action
Tom	keen	Dallas	Delete Edit
Kristin	Summer	Orlando	Delete Edit
Jayasri	Maditai	Cleveland	Delete Edit
John	Graff	Miami	Delete Edit
Pearl	Stacy	Moab	Delete Edit
Chandler	Bing	New York	Delete Edit

Below the table is a link: [Add New Student](#).

Add New Student



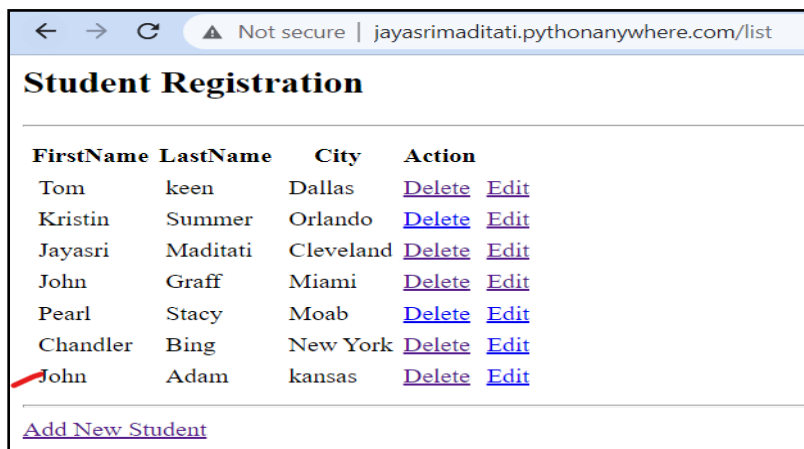
The screenshot shows a web browser window with the address bar displaying "Not secure | jayasrimaditai.pythonanywhere.com/add". The page title is "Add New Student". Below the title are three input fields:

Input First Name:

Input Last Name:

Input City:

Below the input fields is a "Submit" button.



The screenshot shows a web browser window with the address bar displaying "Not secure | jayasrimaditai.pythonanywhere.com/list". The page title is "Student Registration". Below the title is a table with the following data:

FirstName	LastName	City	Action
Tom	keen	Dallas	Delete Edit
Kristin	Summer	Orlando	Delete Edit
Jayasri	Maditai	Cleveland	Delete Edit
John	Graff	Miami	Delete Edit
Pearl	Stacy	Moab	Delete Edit
Chandler	Bing	New York	Delete Edit
John	Adam	kansas	Delete Edit

Below the table is a link: [Add New Student](#).

Edit the Student's City

← → ↻ ⚠ Not secure | jayasrimaditati.pythonanywhere.com/edit/2

Edit Student's City

First Name:

Last Name:

Edit City:

[Cancel](#)

← → ↻ ⚠ Not secure | jayasrimaditati.pythonanywhere.com/list

Student Registration

FirstName	LastName	City	Action
Tom	keen	<u>Omaho</u>	Delete Edit
Kristin	Summer	Orlando	Delete Edit
Jayasri	Maditati	Cleveland	Delete Edit
John	Graff	Miami	Delete Edit
Pearl	Stacy	Moab	Delete Edit
Chandler	Bing	New York	Delete Edit
John	Adam	kansas	Delete Edit

[Add New Student](#)

Delete Student (Pearl Stacy has been deleted)

← → ↻ ⚠ Not secure | jayasrimaditati.pythonanywhere.com/list

Student Registration

FirstName	LastName	City	Action
Tom	keen	Omaho	Delete Edit
Kristin	Summer	Orlando	Delete Edit
Jayasri	Maditati	Cleveland	Delete Edit
John	Graff	Miami	Delete Edit
Chandler	Bing	New York	Delete Edit
John	Adam	kansas	Delete Edit

[Add New Student](#)