Author

- Rishikesh Vankayala
- 21f1000394
- 21f1000394@student.onlinedegree.iitm.ac.in
- I am 19 years old and currently studying B. tech in Computer Science Department in Anurag University, Hyderabad.

Description

This project is a GUI interface which helps users complete their tasks on time. Users can login into their accounts and do various tasks like creating, updating and deleting lists and as well as cards. Users can also get a summary of their completion on the tasks.

Technologies used

- Flask
- Jinja2
- Flask_sqlalchemy
- Flask_restful
- SQLite

Each technology has its own use.

- Flask is used for app routing,rendering the templates and for redirecting to other URLs.
- JInja2 is used in html to pass variables from python code to html file
- Flask_sqlalchemy is used to do CRUD operations with the data in the database
- Flask_restful is used to implement CRUD operations with flask.
- SQLite is a software used to alter with the database(creating database,creating tables,inserting,updating,deleting of data)

DB Schema Design

- User table

COLUMN NAME	COLUMN TYPE	CONSTRAINTS
username	STRING	PRIMARY KEY
password	STRING	-

- List Table

COLUMN NAME	COLUMN TYPE	CONSTRAINTS
lusername	STRING	PRIMARY KEY
lname	STRING	PRIMARY KEY
ldesc	STRING	-
lupdate	STRING	-

- Card Table

COLUMN NAME	COLUMN TYPE	CONSTRAINTS
username	STRING	PRIMARY KEY
l_name	STRING	PRIMARY KEY
c_name	STRING	PRIMARY KEY
c_content	STRING	-
c_deadline	STRING	-
c_toggle	STRING	-
c_update	STRING	-

- Using multiple primary keys in tables in order to eliminate duplicate rows.
- Primary key for users so as to prevent duplicate users.

API Design

In my API design I have created 4 elements : GET, PUT, POST, DELETE for CRUD operations on both lists and cards.

- POST is used to CREATE, either a list or a card depending on the parameters specified
- GET is used to READ,, either a list or a card depending on the parameters specified
- PUT is used to UPDATE, either a list or a card depending on the parameters specified
- DELETE is used to DELETE, either a list or a card depending on the parameters specified.

Architecture and Features

There are files in the root directory; kanban1.sqlite3, local_run, locat_setup, main.py, readme.md, requirements, and the templates folder which contains all the templates.

This application allows a certain user to create and manage lists of the works to be done, and also the managing of the cards that belong to some list. The user can also get a summary of the total no of works done and the pending tasks.

Features:

- Login page with password, sign in option to create the user
- CRUD operations on both lists and cards through web application and through API
- Lists display according to ASCII ordering
- Summary page which shows graphs
- Visual representation of which tasks are done or not done

Video

https://drive.google.com/drive/folders/1aG8BH34qOHturXfhf0c301d_c9x-VLhm?usp=sharing