Author

SATYAM AWASTHI
22dp1000096
22dp1000096@student.onlinedegree.iitm.ac.in
I’m a DAD student and currently pursuing my diploma in programming from IIT Madras.
This is the Project Documentation for MAD-1 project.

Description

We have to create a simple Kanban App depicting the core functionality of managing tasks. User have to create lists and then have to add cards to them. The list could be edited and Cards could be moved from one list to other. There should be a summary page to view the Progress of tasks.

Technologies used

- flask
- Flask
- flask_sqlalchemy
- flask_login
- flask_bcrypt
- matplotlib
- datetime
- Bootstrap
- Jinja2

Flask : for creating web Framework at backend for Kanban Application.

flask_sqlalchemy : for interaction with database.

flask_login : for proper login framework for security purpose and current user tracking.

flask_bcrypt : for hashing the passwords generated by user for user privacy and security.

datetime : in order to use Datetime objects for date and time calculations.

matplotlib : to generate the bar graph for showing it in summary page.

Jinja2 : for rendering templates and to have a little bit of front end logic at html.

Bootstrap : for improving the aesthetics of HTML pages for pleasing user interface.

DB Schema Design

Table Name : **user**

Table : user		
Column	**Type**	**Constraints**
id	INTEGER	PRIMARY KEY
username	VARCHAR(20)	NOT NULL, UNIQUE
password	VARCHAR	NOT NULL
full_name	VARCHAR(20)	NOT NULL
email	VARCHAR	NOT NULL, UNIQUE
created_at	DATETIME	NOT NULL

Table Name : **list**

Table : list		
Column	**Type**	**Constraints**
list_id	INTEGER	PRIMARY KEY
name	VARCHAR	NOT NULL
description	VARCHAR	NOT NULL
user_id	INTEGER	FOREIGN KEY references user.id, NOT NULL

Table Name : **card**

Table : user		
Column	**Type**	**Constraints**
card_id	INTEGER	PRIMARY KEY
title	VARCHAR	NOT NULL
content	VARCHAR	NOT NULL
deadline	DATETIME	NOT NULL
completed_switch	BOOLEAN	NOT NULL
created_date	DATETIME	NOT NULL
last_updated	DATETIME	NOT NULL
completed_at	DATETIME	----
lists_id	INTEGER	FOREIGN KEY references list.list_id, NOT NULL

Relationships:

- One to many relationship from user to list with user as parent and list as child because a single user can create many lists but the lists are associated with only that user who has created them.
- One to many relationship from list to card with list as parent and card as child because a single list may have many cards but cards are associated with only that list in which they have been put.

Reason:

The structure of the database of user, list and card table is build based on the requirements of the data to be used for displaying the value in user interface and for calculation in generating the bar graph.

Architecture and Features

Architecture:

The business logic and all the coding is done in a single python file named app.py. This file is placed in the project folder which will be in the root directory.

All the HTML templates are placed in the templates folder which will be in the project folder.

All the images which are generated by matplotlib will be put in the static folder which will be in the project folder.

The ProjectReport.pdf will be in the Docs folder which will be in the project folder.

Features:

- This Simple Kanban app have a proper login framework which makes use of flask_login for logging in the user after user provides valid username and password.
- The password entered by user is hashed by using flask_bcrypt for user's privacy.
- If the user doesn't have an account he/she can sign up and register their account.
- User can create lists and add cards to them. The list and cards could be edited and cards could be moved from one list to other. Cards have a button to mark it as complete.
- At the bottom of each card the status of the card is shown whether it is completed or incomplete so that just by looking at the card the user can know its status.
- User can view the summary page which shows the progress of tasks as per individual lists and also shows the bar graph based on that which includes completed tasks, In progress tasks and tasks whose deadline have been crossed.
- Validation is provided at backend so that the name of the lists should be different for the Same user and the title of cards should not be same within the same list meanwhile the titles could be same when cards are in different lists.
- Validation is also provided at frontend so that to check that every field in the form is filled, for checking the email, and to check the deadline marked by user in creating the card should be on or after the date of making the card it should not be before that.
- A dropdown named Account Actions is provided in the navigation bar so that user can look for it's details, User manual and can delete the account after providing valid credentials.
- Bootstrap is used in html pages to make the user interface more pleasing.

Video

https://drive.google.com/file/d/1Kz0lAu0sSMBL1hG1wi7bQtYLLJ4dOIj9/view?usp=sharing