Author:

Noufal Rahman
21f1005287
21f1005287@studebt.onlinedegree.iitm.ac.in

I'm a 3rd year student pursuing BE CSE at Government College of Technology,Coimbatore, Tamilnadu. I love to code and to build and develop solutions that could be helpful for a community.

Description:

Kanban App is a sort of to-do list app with features like organizing tasks under a list, creating tasks with deadlines, description etc., There should be a signup/login page for creating an account and logging into their account. Once after the login, the lists page should be containing all the lists created by the user and under every list, there should be cards created under the list. The summary page should contain the overall summary for the user including the overview of lists and tasks

Technologies Used:

- HTML - Template
- Bootstrap - Stylesheet
- Flask
- SQLite - Database
- Flask-SQLAlchemy - Connection with database to the app
- Jinja2 - Template rendering
- Matplotlib - Graph plotting for summary

- **DB Schema Design:**

1. **User**

id	INTEGER	"id" INTEGER NOT NULL UNIQUE
user_id	TEXT	"user_id" TEXT NOT NULL UNIQUE
pwd	TEXT	"pwd" TEXT NOT NULL
name	TEXT	"name" TEXT NOT NULL
created_on	TEXT	"created_on" TEXT NOT NULL
last_login	TEXT	"last_login" TEXT

2. **List**

id	INTEGER	"id" INTEGER NOT NULL UNIQUE
name	TEXT	"name" TEXT NOT NULL
desc	TEXT	"desc" TEXT
created_on	TEXT	"created_on" TEXT NOT NULL

3. **Card**

id	INTEGER	"id" INTEGER NOT NULL UNIQUE
name	TEXT	"name" TEXT NOT NULL
description	TEXT	"description" TEXT
created_date	TEXT	"created_date" TEXT
edited_date	TEXT	"edited_date" TEXT
completed_date	TEXT	"completed_date" TEXT
deadline	TEXT	"deadline" TEXT
done	INTEGER	"done" INTEGER

4. **UserList - Relation between user and list**

id	INTEGER	"id" INTEGER NOT NULL UNIQUE
user_id	INTEGER	"user_id" INTEGER NOT NULL
list_id	INTEGER	"list_id" INTEGER NOT NULL UNIQUE

5. **ListCard - Relation between list and card**

id	INTEGER	"id" INTEGER NOT NULL UNIQUE
card_id	INTEGER	"card_id" INTEGER NOT NULL UNIQUE
list_id	INTEGER	"list_id" INTEGER NOT NULL

API Design:

APIs are designed logically to support the rest of the application. The API part is different from the actual application.

1. **User:**
   CRUD operations on User to get, delete, update and add user
2. **List:**
   a. With list_id: get, put, delete
   b. With user_id: get (view all the lists created by the user), post (create a list under the user since the list can't be standalone)
3. **Card:**
   a. With card_id: get, put. Delete
   b. With list_id: get (view all cards of a list), post (create a card under a list since a card can't be standalone)

Architecture and Features:

- APIs are located in the apis folder.
- The app routes are located in routes folder
- The app configuration is written in app.py and the database configurations are in application/db.py
- Database schemas are written in application/models.py
- The templates are in templates folder
- All media content are in static folder
- Features:
  1. Implementation of Sign In and remembering the logged in user for the session.
  2. Implementation of Drag and Drop of cards from one list to another
  3. Attractive UI with Bootstrap and CSS.
  4. Usage of Modals for Add List, Edit List, Add Card, Edit Card.
  5. Summary page with user summary and summary for each list along with a graphical representation with a bar graph.

Video Link:

https://drive.google.com/file/d/133s0CWlGhgB99Dfh2z4UBJFbN9w23jEb/view?usp=sharing