

Author

Aniket Dash

21f1006102

21f1006102@student.onlinedegree.iitm.ac.in

DESCRIPTION

The Kanban Board required to have certain functionalities which included:

- having relevant 'cards' or divisions to organize the tasks of an individual or a group of individuals.
- Present a summary of progress and efficiency in completing given tasks before deadline

TECHNOLOGIES USED

The technologies used are:

- Flask – For server-side operations.
- SQLite3 – For database.
- Python os module – For using operating system dependent functionality.
- Flask SQLAlchemy - Adding SQLAlchemy support for the Flask application.
- Flask Login – For ease of logging and registering related validation.
- Python datetime module
- HTML/CSS/JS - For client-side operations and user interface

- Chart.js - For constructing a chart for summary.

DATABASE SCHEMA DESIGN

The database is a SQLite database. The schema for this database outlines the structure of two simple tables:

- Table 1:
 - Title: User
 - Fields: ID, Username, Password
 - Constraints:
 - ID – Primary Key
 - Username – Not Null
 - Password – Not Null

This table is the parent class in the model.

- Table 2:
 - Title: Task
 - Fields: Title, Status, Deadline, user_id
 - Constraints:
 - Title – Primary Key, Unique, Not Null
 - Status – Not Null
 - User_id – Foreign Key(user.id)

This table is the child to the table titled as 'user'.

- One user can have one or many tasks. So, there is a one-to-many relationship between the above-mentioned tables.

- Without the task ID, the addition of tasks into cards using 'status' becomes convenient. Here, there is no need to increment or keep track of the task ID to add a new task.
- Addition of task into the selected card can be done with Flask from the form responses.

ARCHITECTURE AND FEATURES

This application could've been a single file but it becomes overwhelming to keep all the code in one file.

The organization of the app is similar to that of the structure of basic Flask application. To mention the directories we can do a nested list:

- /:
 - Main.py- contains an application factory function called 'create_app'
 - Requirements.txt- A txt file of all the dependencies required to run the app
- /application:
 - Config.py- Stores configurations for the app
 - Controllers.py:
 - decorators '@app.route' creates a connection between the URL and the function that returns a response
 - Current_app is a special object that points to the Flask application handling the request.
 - Database.py
 - Models.py-Models are defined in this file
- /Db_directory:
 - Appdb.db - Database file for the app
- /Static:

The stylesheets for the templates

- Home.css
- Login.css
- Register.css
- Summary.css
- /js
 - Home.js
- /Templates:
 - Base.html - The parent template
 - Home.html
 - Login.html
 - Register.html
 - Summary.html
- Docs
 - Project report.pdf
 - Requirements.txt
 - Checklist.txt

Features of the app

- Flask is a micro web framework written in python.
- This app uses Jinja templating – Special placeholders in the template allow writing code similar to python syntax.
- Flask 'g' is used. 'g' is a good place to store resources during a request. Technically it's a namespace object that can store data during an application context.

- For the visualization of the summary of an user 'chart.js' has been used. It is a JavaScript visualization library.
- For session management Flask-Login has been used. It handles common tasks of logging in, logging out and remembering user's session over extended period of time.
 - LoginManager – This class contains the code that lets your application and Flask-Login work together such as, how to load an user from an ID.
 - @Login_required – When a view is decorated with this, it will ensure that the user is logged in and authenticated before actually calling the view.
- Usermixin – Flask-Login requires a User model with the following properties:
 - Is_authenticated()
 - Is_active()
 - Is_anonymous()
 - Get_id()

Usermixin provides default implementations of the above-mentioned methods, which eliminates the necessity to implement them manually.

VIDEO

Here's the link to my project presentation video.

- https://drive.google.com/file/d/1eJYHJmL51ilOvHN5Zo6qoz_vDPpWWho1/view?usp=sharing