Author

Trivikram Umanath
21f1005359
21f1005359@student.onlinedegree.iitm.ac.in
Endeavouring Fulfilment

Description

The aim of this project is to create a web application which helps the user to list down and keep a track of his/her activities and sub activities namely in the form of list and cards.Additionally the app also provides mechanisms to summarise and notify the summary to the user as well as remind the user of pending activities.

Technologies used

The application uses Flask, FlaskRESTful, SQLAlchemy and Vue JS to implement core functionalities. Vue JS is used for templating and reactivity in HTM.Vuejs is also used for scripting. Css is further implemented for styling. Redis has been used for Caching to ensure faster results at optimizable places. Celery has been used to implement background jobs for processes like monthly reports and daily reminders.Weasy print is used to create pdf reports.SMTP Lib is used to send successive emails.MailHog is used as a smtp server. Webhooks Plugins are also used.

DB Schema Design

TABLE USER

Column Name	Type	Constraints
id	Integer	Primary Key ,AutoIncrement
email	Text	Not Null
password	Text	Not Null
fs_uniquifier	Text	Unique
active	Boolean	Not Null

Every user is uniquely identifiable.
TABLE List

Column Name	Type	Constraints
List_Id	Integer	Primary Key ,AutoIncrement
User_Id	Integer	Foreign Key(“User.id”)

Name	Text	Default
Description	Text	Default

List Table has a primary key as List_Id and a foreign key called User_id referring to User.id from the user table .One to many relationships between User and List.A User can have multiple lists.

TABLE Card

Column Name	Type	Constraints
Card_Id	Integer	Primary Key ,AutoIncrement
List_Id	Integer	Foreign Key("List.List_Id")
Title	Text	Default
Deadline	Text	Default
Status	Text	Default
Content	Text	Default

List Table has a primary key as List_Id and a foreign key called User_id referring to User.id from the user table .One to many relationships between List and Card.A User can have multiple lists.

API Design

The application has a robust API with multiple endpoints protected by authorization with Flask Security Token auth for optimal security. Further the API allows for (CRUD) Creation , Reading ,Updation and Deletion of Lists and Cards.

Architecture and Features

The project is divided into frontend and backend code.The Backend code houses the main.py code and the controllers,api,tasks,workers,..etc are all housed in the application folder of the backend.The Frontend folder contains the Components and Views folder in the src section housing all the components and views implemented and used.The index.js is also present mapping the routes and the views.The application has a login system, Dashboard for list and cards(CRUD on decks and cards) along with an export,summarise using stats functionality available.The application also has the functionality to remind of pending tasks ,mail reports and messages on webhooks.

Video

https://drive.google.com/file/d/1y4lEEp19URowQkwNJy8KpcTgdsqyBw54/view?usp=sharing