

# Author

Sri Krishna Teja Aradhyula

21f2001389

[21f2001389@student.onlinedegree.iitm.ac.in](mailto:21f2001389@student.onlinedegree.iitm.ac.in)

I am pursuing an online degree at IIT Madras and also currently, I am in b tech 3<sup>rd</sup> year at vignan's foundation for science technology and research. I am passionate about programming and problem solving

## Description

In the Kanban application, task management done by users can manage multiple tasks and create a list of tasks that can be deleted and updated further. Users can perform CRUD operations on lists, and cards, also there is a summary page where users can view their tasks in the form of graphs

## Technologies used

- Flask
- Bootstrap
- HTML
- CSS
- Flask-sqlalchemy
- Flask-login
- Matplotlib
- Werkzeug
- Flask-restful
- Numpy
- SQLite

## DB Schema Design



The database consists of users, lists, and cards relations. Constraints in the users table are id is the primary key and username is unique. Constraints in lists tables are list\_id as a primary key and list\_name as unique user\_id is a foreign key relationship with users and include on cascade on update, delete. As well as constraints in the cards table are card\_id is a primary key and card\_title is unique and list ID is a foreign key which is related with list be in lists and include on cascade on update, delete. all relations one to many

## API Design

API is created using flask-restful. Users, Lists, and Cards classes are created where get, put, post operations can be done on the users. Get, put, post, and delete operations for lists and cards are done. These components are created for paths

- /api/users, /api/users/{username} – for Users
- /api/lists, /api/lists/{list\_name} - for Lists
- /api/cards, /api/cards/{card\_title} – for cards

## Architecture and Features

In the project folder, you can see the app.py file its main file which contains all app routes. api.py file contains flask-restful code which performs CRUD operation on users, lists, and cards. this api.py is connected to app.py so, no need to run it. database.db file contains a database of project which was created using SQLite. All the Html files are in the templates folder. The requirements.txt file specifies all the requirements, extensions, etc that are required for the app.

In this project in starting page contains two buttons one is for login another one is for signing up. if your first time, you can sign up, or else you can login using your username and password if you are an authorized user you will redirect to the home page where you can create multiple numbers list and inside a list, you can create multiple numbers of cards with card title, content of card, deadline to complete the task and also you need to mention how much it completed you can see five radio buttons of 0%, 25%, 50%, 75%, and 100% by choosing one you can maintain your progress and also you can edit the card as well as list and also delete the card and list if your delete any list which consists some cards then the cards will also delete. And also, you can see the cards are differentiated with different colours based on how much the task was completed on top of the page you can see the name summary by clicking on that name you can go to the summary page on that page the all-task progress represented in the form of bar graphs and also you can see which tasks pass the deadline without completion

## Video

<https://drive.google.com/file/d/1Nm8L-hkOTsp5rirp9Gc25aKXCPFDvuD8/view?usp=sharing>