

## Author

Devathi Sai Mani Kumar

21f3003246

21f3003246@student.onlinedegree.iitm.ac.in

I am a technology enthusiast and an autodidact person.

## Description

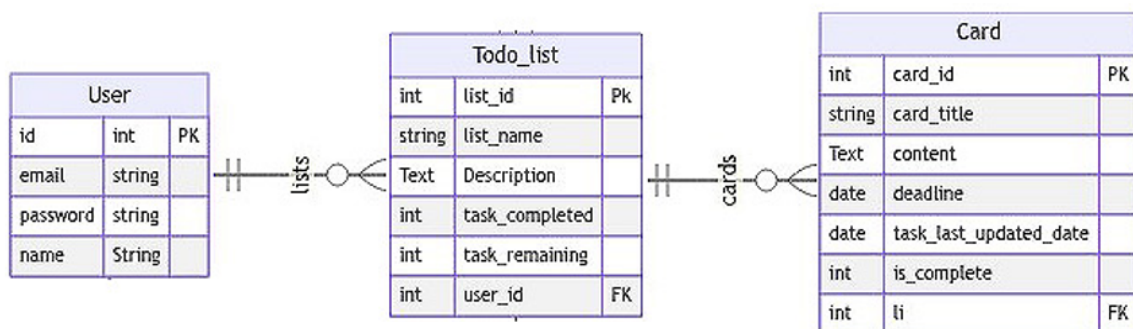
Kanban task management where users can manage multiple tasks and create a list of tasks of similar interest which can be deleted and updated further. Users can perform CRUD operations on lists, tasks and a summary page where users can view their tasks in the form of graphs.

## Technologies used

Flask, Bootstrap, Flask-sqlalchemy, Flask-login, Matplotlib, Flask-cors, Werkzeug, HTML, CSS, Flask-restful

Flask is used to design the web application, flask-sqlalchemy is used to store the data in the database, flask-login is used for authentication, matplotlib is used for plotting the data, werkzeug is used to hash the password that is it is used to get the website secure, Bootstrap is used to for styling the web pages. Html and css are used for creating the web pages. Flask-cors is used for cross origin resource sharing. Flask-restful is used for API design.

## DB Schema Design



I took one to many relationship named lists between User, Todo\_list and one to many relationship named cards between Todo\_list, Card using backref, user\_id and li as foreign keys respectively. Because of these relationships we can access all the lists from the User and user\_id from the Todo\_list and all cards from Todo\_list and list\_id from Card. And I added cascade as all delete on the relationships so that if we delete user or list respective lists or cards will be deleted successfully.

## API Design

API's are created using flask-restful. UserAPI, ListAPI, CardAPI, SummaryAPI classes are created where get, put, post operations can be done on the user. Get, put, post, delete operations for todo\_list and card api classes and get operations for summary api are created. These components are created for paths /api/user, /api/list, /api/card and /api/summary.

## Architecture and Features

The main directory where all the files are stored is KANBAN. Inside this directory two subdirectories db\_directory, kanban\_app and a main.py are present. kanban\_app acts as a python package, inside it \_\_init\_\_.py, api.py, auth.py, config.py, model.py, views.py files are present. In \_\_init\_\_.py file, the app and api are created. I used flask blueprints and created auth and view blueprints. The app controllers are present in auth and view blueprints and api resources are placed in main.py for making the work simpler. auth blueprint is used for user authentication and view blueprint is used for lists, cards and summary page controllers. Inside the model.py the database design is present where User, Todo\_list, Card relations are created. Inside the config.py the database directory path and secret key are initialised. Inside the api.py file the get, post, put, delete operations on user, list, card and summary api's are present. The html files for the web pages are placed under templates folder which are rendered using a jinja template engine. static folder consists of css and graph folders where css files and graph images are placed respectively.

Features of kanban application starts from user authentication where users can register and login from the home page. I hashed and salted the password using werkzeug. After successful login they are redirected to their respective kanban board where they can create Todo\_lists and can add tasks by giving the name and description. The deadline and the amount of percentage is given to a task. Amount of percentage completed is set to 0 and can update the percentage which can be seen in the progress bar. The styling for the web pages are made using the Bootstrap framework. Users can update and delete the existing tasks. A task can be moved from one task to another using the update option. In the summary page users can track the number of tasks completed and no of tasks remaining and the last updated date and the information about no of tasks passed the deadline without completion.

## Video

[https://drive.google.com/file/d/1e1ZfDgPVmVjZxfBZzWm\\_55K8RjYvK67/view?usp=drivesdk](https://drive.google.com/file/d/1e1ZfDgPVmVjZxfBZzWm_55K8RjYvK67/view?usp=drivesdk)