

# Project Report

## Kanban App

### Author

Name: Laxminandan Handique

Roll no.: 21f3003015

Email: [21f3003015@student.onlinedegree.iitm.ac.in](mailto:21f3003015@student.onlinedegree.iitm.ac.in)

I graduated in 2018 with a bachelor's degree in chemistry. I always had a desire to learn coding and I'm especially interested in the UI/UX part of application development.

### Description

App name: MyKanban

Description: A basic kanban app developed and submitted for the final project of the MAD-I course (May 2022 term)

### Technologies used

- Flask for backend
- Flask-SQLAlchemy for interacting with the SQLite database
- Flask-Login for authentication of users
- Flask-Bcrypt for hashing passwords
- Flask-RESTful for creating a REST API
- Matplotlib for graphs and visualizations
- Jinja2 for HTML generation
- CSS for styling (mainly flexbox and grid) + Bootstrap for icons

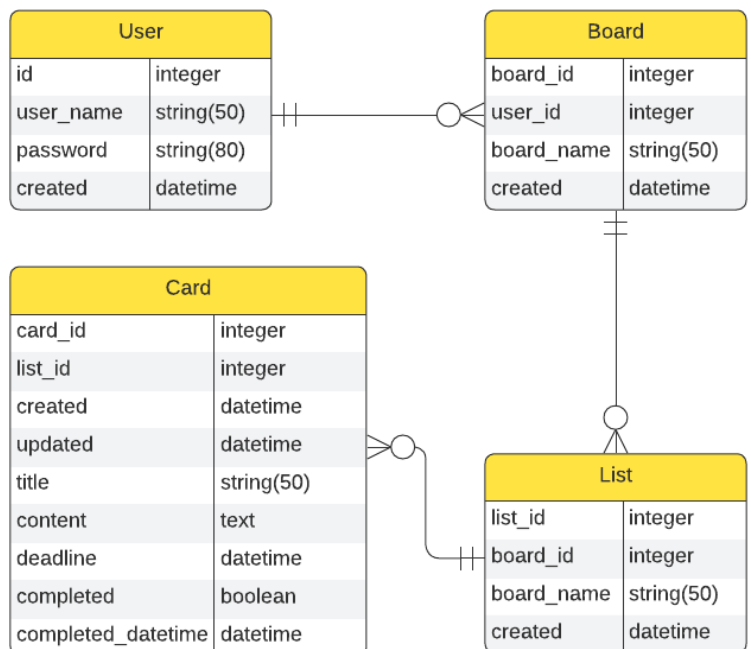
### DB Model:

The schema of the database consists of four relations for user, board, list and card. All of them have one-to-many relationships with the other as can be inferred from the graph.

I have added the boards relation to allow a user to maintain multiple boards for different projects.

The user\_name, board\_name, list\_name and title fields have a maximum length of 50. Additional constraints on the minimum length are defined in the application controllers.

The user\_name field must be unique and non-empty. The maximum length of a password is 20 characters in the controller but set as 80 in the schema so as to accommodate for hashing.



Each entity (user/board/list/card) upon creation gets its creation time captured by the application and stored in its created field.

## API Design

A REST API for the kanban app was made with the Flask-RESTful module. It has:

- Endpoints for basic CRUD operations on boards / lists / cards.
- Endpoint for getting cards completed per day
- Endpoint for getting the no. of complete / pending / overdue cards

## Architecture and Features

root/

```
├── application/
│   ├── api/
│   │   ├── api_validation.py
│   │   ├── board_api.py
│   │   ├── card_api.py
│   │   ├── list_api.py
│   │   └── stats_api.py
│   ├── config.py
│   ├── controllers_board.py
│   ├── controllers_card.py
│   ├── controllers_error.py
│   ├── controllers_list.py
│   ├── controllers_login.py
│   ├── controllers_reports.py
│   ├── controllers_stats.py
│   ├── controllers_user.py
│   ├── controllers_view.py
│   ├── database.py
│   ├── models.py
│   ├── stats.py
│   └── validations.py
├── db_directory
├── reports
├── static/
│   ├── css
│   ├── img
│   └── js
├── templates
└── main.py
```

On the left is the directory structure of the flask application.

### Features at a glance:

#### User

- Create / edit / delete an account for a user

#### Boards

- Create / edit / delete multiple boards for different projects

#### Lists

- Create / edit / delete multiple lists within each board
- Delete a list while moving the cards in it to a different list

#### Cards

- Create / edit / delete multiple cards within each list
- Move a card from one list to another (given that the lists are in the same board)
- Set a deadline for a card
- Mark a card as complete

#### Summaries

- No. of cards / lists / boards that you have (overall or within a board or a list)
- Cardwise breakdown of completed / pending / overdue cards
- Graph showing the no. of cards completed each day.
- Plot showing the percentage of completed / pending / overdue cards

#### Export

- Export a list as a (.csv) file
- Export (.csv) files for all lists within a board as a single (.zip) file

## Video

<https://drive.google.com/file/d/1CniOd0tF0XjnLKUEjUj118WMIImWJtbeD/view?usp=sharing>