## Author

**Name**: Arpit Jain
**Roll No**: 21f1006008
**Email**: 21f1006008@student.onlinedegree.iitm.ac.in

## Description

An application to manage tasks (cards). There are lists with cards in them. Need to track task competition and provide stats for progress.
Create a UI for all operations
Create a REST API for CRUD

## Technologies used

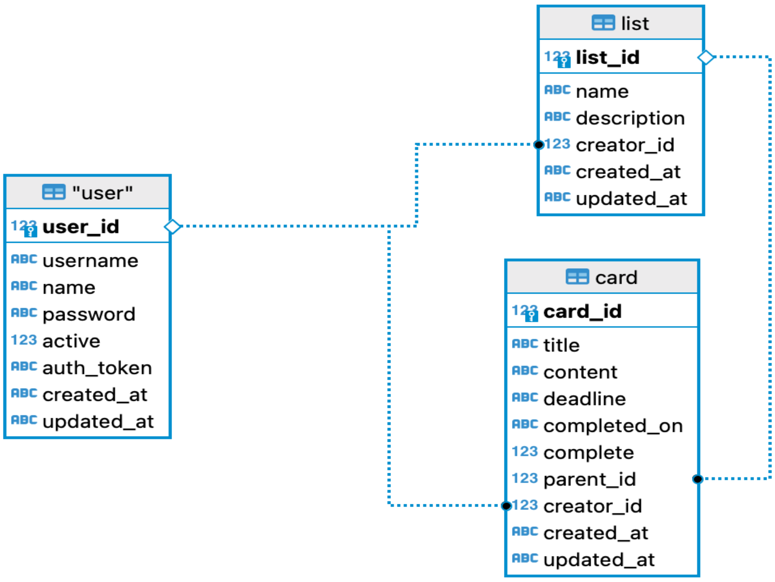
- DB: SQLite [Easy to use, Lightweight]
- Web Framework: Python Flask [Easy to use, Lightweight, Extensible]
- Styling: Bootstrap CSS [Easy to use, -Heavy]
- Python packages used:

| **Name** | **Purpose** |
|---|---|
| flask | Web framework that forms the base of our application |
| flask-sqlalchemy | sqlalchemy is an ORM for Python<br>flask-sqlalchemy enables us to use sqlalchemy easily in our app including configuration and database session management |
| flask-restful | Helps to create REST APIs by providing automatic route generation, request body parsing and validation, response marshalling |
| flask-wtf | Helps in form validation. Used to reuse code for edit and create forms |
| email_validator | For email validation |
| bcrypt | For password hashing |
| flask-Migrate, alembic | Helps to generate migration files and perform migration |
| Matplotlib, pandas | Used to plot the timeline |

## API Design

**Main Board**: /
**JSON Rest API**: /api/[card/list]/<[card_id/list_id]>
**Manage Cards**: /cards/new /card/<card_id>/[edit/delete] move_card
**Manage Lists**: /lists/new /list/<list_id>/[edit/delete]
**Stats**: /stats
**Users**: /login /signup /logout /generate_token

## DB Schema Design



A user can create many lists [creator_id in list]
A user can create many cards [creator_id in card]
A card belongs to one list [parent_id]

## Architecture and Features

Top Folders: application, db_directory, migrations, static, templates.
Application: Has controllers, models, database
Models: Has individual files for each model
Controller: controllers are grouped on the basis of their functionality
user / list / card / stats - Each contain
Index file with all the controllers
Form file with the definition of different forms used: [create/edit - list, card, user , move - card, delete - list]
api - contains individual files for each resource (card, list)
There are also many utils files for authentication / error handling / loading card and list objects

Templates

- Stored hierarchically
- There is a base template, which includes the basic HTML tags and CSS and JS tags. There are blocks that are declared later containing the actual content
- Each view declares their own blocks for main content, header/title.

Features Implemented:

- User Signup, Login, Logout
- Can create lists [max 5], edit list, add cards to list, view a list, delete list -> move cards to another list or delete the cards
- Can create cards, edit cards, mark card as complete, delete card
- Move card between lists
- Can view basic stats

**Video** : https://drive.google.com/file/d/1duvqc9juPkWpuOPslr4hRuTG3tYeWyEB/view?usp=sharing