

# [ Indian Institute Of Technology, Madras]

Bachelor of Science (BS) Degree in Data Science and Applications

## [Project Report On Kanban Application]

### [Modern Application Development - 01]

course code : CS2003P

[ Student Name:- Pawan Kumar Pathak ]

[ Roll No:- 21f2000999 ]

[ Student Email:- 21f2000999@student.onlinedegree.iitm.ac.in ]

“ My name is Pawan and I am currently persuing Data Science Course by IITMadras.

I am currently on diploma level of Programming and I have completed MAD\_1 course and This is the Project Report of KANBAN APPLICATION done by me.”

#### **Description**

*Kanban Application Used for tracking tasks, User can have multiple lists and Each list will have ID, Name, User can add one or more card to a list. Each card will have Title, Content, Deadline. System will capture the card created datetime. System can track Progress over time and show graphs trend lines etc. As Summary*

#### **Technologies Used**

- \* Flask for application Code
- \* Jinja2 Templates
- \* Bootstrap for HTML and CSS
- \* Matplotlib and Numpy for Graph
- \* SQLite for Database Storing
- \* Flask SQLAlchemy for db
- \* Flask Rest for API

#### **Documentation URLs**

- \* <https://flask-login.readthedocs.io/en/latest/> for Flask Login System
- \* <https://flask-restful.readthedocs.io/en/latest/> for Flask restful APIs
- \* <https://flask-sqlalchemy.palletsprojects.com/en/latest/> for Flask SQLALchemy DatabaseDesign

#### **API Design**

- \* APIs for interaction with lists and cards
  - CRUD on Lists ○ CRUD on Cards
- \* Validation
  - All form inputs fields - text, numbers, dates etc. with suitable messages ○ Backend validation before storing / selecting from database

#### **Video**

- \* <https://drive.google.com/file/d/1AbTZohpKfDwC7PwtZHAOikdB0TnBeYtA/view?usp=sharing>

# DB Schema Design

## Architecture and Features

### Tables (7)

Name	Type	Schema
<b>association</b>		CREATE TABLE association ( todo_id INTEGER NOT NULL, list_id INTEGER NOT NULL, PRIMARY KEY (todo_id, list_id), FOREIGN KEY(todo_id) REFERENCES todo (todo_id), FOREIGN KEY(list_id) REFERENCES listform (list_id) )
todo_id	INTEGER	"todo_id" INTEGER NOT NULL
list_id	INTEGER	"list_id" INTEGER NOT NULL
<b>card</b>		CREATE TABLE card ( card_id INTEGER NOT NULL, list VARCHAR NOT NULL, title VARCHAR NOT NULL, content VARCHAR(100) NOT NULL, deadline DATETIME NOT NULL, time_created DATETIME DEFAULT (CURRENT_TIMESTAMP), time_updated DATETIME, PRIMARY KEY (card_id), UNIQUE (title) )
card_id	INTEGER	"card_id" INTEGER NOT NULL
list	VARCHAR	"list" VARCHAR NOT NULL
title	VARCHAR	"title" VARCHAR NOT NULL
content	VARCHAR(100)	"content" VARCHAR(100) NOT NULL
deadline	DATETIME	"deadline" DATETIME NOT NULL
time_created	DATETIME	"time_created" DATETIME DEFAULT (CURRENT_TIMESTAMP)
time_updated	DATETIME	"time_updated" DATETIME
<b>card_stat</b>		CREATE TABLE card_stat ( cardstat_id INTEGER NOT NULL, cardstat_name VARCHAR(30) NOT NULL, cardstat_complete INTEGER NOT NULL, cardstat_daillyhours INTEGER NOT NULL, cardstat_approx INTEGER NOT NULL, cardstat_rate INTEGER NOT NULL, time_created DATETIME DEFAULT (CURRENT_TIMESTAMP), time_updated DATETIME, PRIMARY KEY (cardstat_id) )
cardstat_id	INTEGER	"cardstat_id" INTEGER NOT NULL
cardstat_name	VARCHAR(30)	"cardstat_name" VARCHAR(30) NOT NULL
cardstat_complete	INTEGER	"cardstat_complete" INTEGER NOT NULL
cardstat_daillyhours	INTEGER	"cardstat_daillyhours" INTEGER NOT NULL
cardstat_approx	INTEGER	"cardstat_approx" INTEGER NOT NULL
cardstat_rate	INTEGER	"cardstat_rate" INTEGER NOT NULL
time_created	DATETIME	"time_created" DATETIME DEFAULT (CURRENT_TIMESTAMP)
time_updated	DATETIME	"time_updated" DATETIME
<b>listform</b>		CREATE TABLE listform ( list_id INTEGER NOT NULL, "listName" VARCHAR(30), "Description" VARCHAR(100) NOT NULL, time_created DATETIME DEFAULT (CURRENT_TIMESTAMP), time_updated DATETIME, PRIMARY KEY (list_id), UNIQUE ("listName") )
list_id	INTEGER	"list_id" INTEGER NOT NULL
listName	VARCHAR(30)	"listName" VARCHAR(30)
Description	VARCHAR(100)	"Description" VARCHAR(100) NOT NULL
time_created	DATETIME	"time_created" DATETIME DEFAULT (CURRENT_TIMESTAMP)
time_updated	DATETIME	"time_updated" DATETIME
<b>status</b>		CREATE TABLE status ( card VARCHAR NOT NULL, "Pending" INTEGER, "Complete" INTEGER, "Incomplete" INTEGER, "Less_than_50" INTEGER, "More_than_50" INTEGER, PRIMARY KEY (card) )
card	VARCHAR	"card" VARCHAR NOT NULL
Pending	INTEGER	"Pending" INTEGER
Complete	INTEGER	"Complete" INTEGER
Incomplete	INTEGER	"Incomplete" INTEGER
Less_than_50	INTEGER	"Less_than_50" INTEGER
More_than_50	INTEGER	"More_than_50" INTEGER
<b>todo</b>		CREATE TABLE todo ( todo_id INTEGER NOT NULL, todo_user VARCHAR(20) NOT NULL, todo_name VARCHAR(20) NOT NULL, PRIMARY KEY (todo_id) )
todo_id	INTEGER	"todo_id" INTEGER NOT NULL
todo_user	VARCHAR(20)	"todo_user" VARCHAR(20) NOT NULL
todo_name	VARCHAR(20)	"todo_name" VARCHAR(20) NOT NULL

Name	Type	Schema
<b>user</b>		CREATE TABLE user ( user_id INTEGER NOT NULL, username VARCHAR(20) NOT NULL, password VARCHAR(80) NOT NULL, PRIMARY KEY (user_id), UNIQUE (username) )
user_id	INTEGER	"user_id" INTEGER NOT NULL
username	VARCHAR(20)	"username" VARCHAR(20) NOT NULL
password	VARCHAR(80)	"password" VARCHAR(80) NOT NULL

### Indices (0)

Name Type Schema

### Views (0)

Name Type Schema

### Triggers (0)

Name Type Schema

## Features Details

All codes are written in a single python file named as app.py inside ProjectFolder. Proper login system implemented from flask login document where user name and password saved in data file after login and password saved in Encrypted form. After login Kanbanboard will open where all the list and card where shown after creating it cards can be moved from one list to another and all the cards and list can be update as well as delete. In every list there is bottom section to track your progress where work details can be filled and statistical data chart will be shown on summary page. CRUD operation on list and card are also performed from backend APIs.

### |----ProjectFolder

#### |---docs

|--ProjectReport.pdf

|--video.txt

#### |---documentation

|--kanbanApi.yaml

|--kanbanApi\_raw.txt

#### |--templates

|--card.html

|--cardlist.html

|--home.html

|--kanbanboard.html

|--listform.html

|--login.html

|--register.html

|--summary.html

|--update.html

|--updatecard.html

#### |--app.py

|--requirements.txt

|--tododb.db

|--README.md