Project Report: Kanban App

Name: Anushka Krishna
Roll: 21f1003017
Email: 21f1003017@student.onlinedegree.iitm.ac.in

I am Anushka, a final year electronics and communication engineering student, from Bihar.

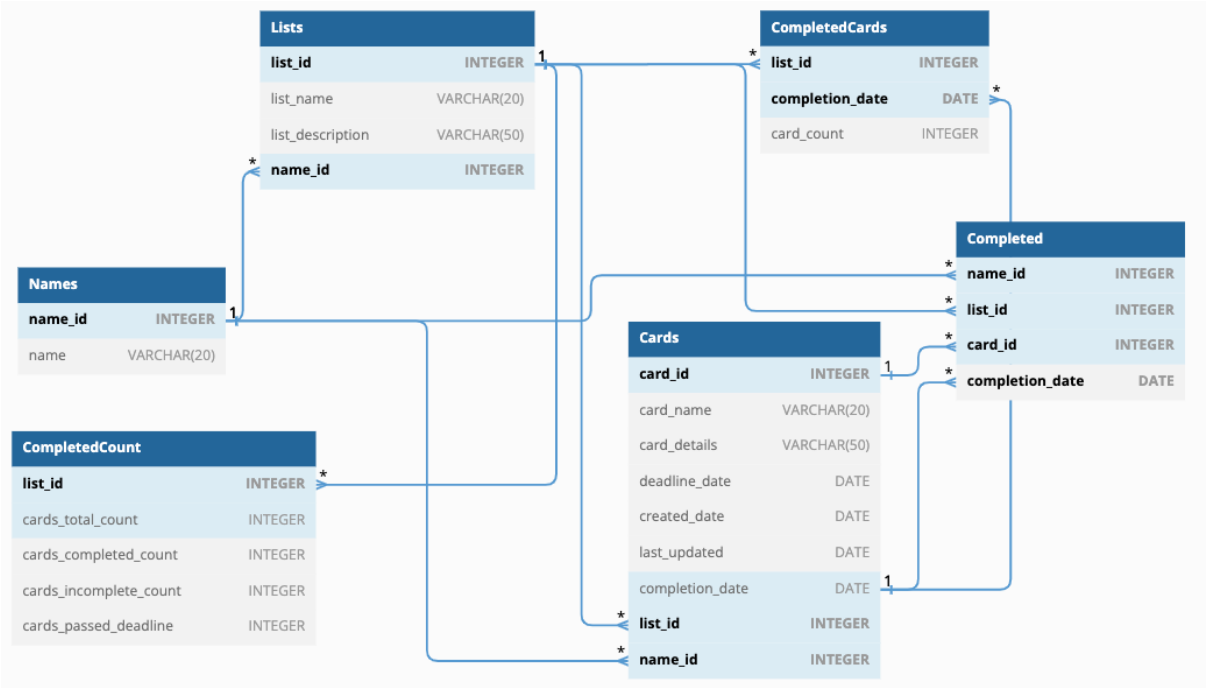
Description

Having used the Flask framework to create the Kanban app, in which different routes for different purposes and renders different pages, developed using HTML, based on the user's click. It seems to present the performance graph of the user in completing the tasks before the deadline, generated using matplotlib library of python. All different rendering is based on the Jinja2 library of python.

Technologies used

- **os**: For giving path of database and folders containing images
- **flask**: **Flask**: For creating the application
  render_template: For rendering the HTML pages on different clicks
  request: To get data and methods from forms
- **flask_sqlalchemy**- **SQLAlchemy**: For creating models in database
- **HTML**: For creating web pages
- **CSS**: (Inline and Internal) To add styling effects
- **Bootstrap**: To align web pages as required and create selections
- **matplotlib**: For creating graphs
- **datetime**: To use dates as data

DB Schema Design



It is an app-based project for public use so multiple users can login to create their own lists and tasks. The project requirement is to have a number of lists for a user, with multiple users being able to login so kept list_id unique with name_id as a foreign key. Further, then the addition of cards in every list is required, so card_id with list_id and name_id as a foreign key is used. list_id gets updated further on moving it to a different list. If the mark as complete is checked in the form, it gives the current date as the completion date. Those cards which have a deadline after the completion date are considered complete, one whose deadline is before the present date is considered a passed deadline if not completed or if completed after the deadline. Further, the cards completed before the deadline are collected with their name_id, list_id and card_id as primary key and completion_date in the Completed table to get a count of cards completed in a day for a list of a user. Further for graph creation CompletedCards table is created, storing how many cards were completed on a date for a list. Also, we get statistics of completion before the deadline, incomplete and passed deadline cards.

Architecture and Features

The code for the Kanban app is organized based on the usage and update required. The code for the app can be viewed inside the project folder with the name app.py python file. The app named Simple Kanban, on starting gives us login and sign-in page, which takes us to the home page where there are no lists for new users or an option to add them, and for existing users, there are existing lists and cards and also a possibility to add new lists and cards. For new users, adding cards in those options is also available once lists are added. On the home page, there are links for visually identifying complete and incomplete cards, the summary page and a link to log out. The user has the ability to update and delete the lists and cards using the buttons available at the right corner of lists and cards which onclick opens up the update page in case the update button is clicked and in case the delete button is clicked, confirmation with the user is made for the delete action. All the web pages which are rendered on different clicks are saved under the templates folder inside the Project folder. The images are saved inside the IMG folder which is inside the static folder. The python file models.py develops the database having 6 tables: Names for user details, Lists for lists details, Cards for card details and 3 secondary tables for presenting data of use on screen. The app.py consists of all controllers used to access the model and modify the view of the app.

Flowchart of Kanban App:
https://drive.google.com/file/d/1aJosoAmEOgxpCy5eUOf6jebJg6zEqpTD/view?usp=sharing

- @app.before_first_request
- @app.route("/", methods = ["GET", "POST"])
- @app.route("/name/addlist/<username>/<int:id>", methods = ["GET", "POST"])
- @app.route("/addcard/<int:ID>/<username>/<listname>/<int:listid>", methods = ["GET", "POST"])
- @app.route('/<username>/<int:id>/<int:list_id>/list_update', methods=["GET", "POST"])
- @app.route('/<username>/<int:id>/<int:list_id>/list_delete', methods=['GET', 'POST'])
- @app.route('/<username>/<int:id>/<int:list_id>/<listname>/<card_id>/card_update', methods=["GET", "POST"])
- @app.route('/<username>/<int:id>/<int:card_id>/card_delete', methods=['GET', 'POST'])
- @app.route('/<username>/<int:id>/summary', methods=['GET'])
- @app.route('/<username>/<int:id>/completed', methods = ['GET'])
- @app.route('/<username>/<id>/home', methods = ['GET'])

Video

https://drive.google.com/file/d/1xzHIShGUGQgaSjInNMfSnQ5d2m-kXqA-/view?usp=sharing