Kanban Project

Author

GANESH L
21f1002245
21f1002245@student.onlinedegree.iitm.ac.in

I'm from Bangalore, working as a Business Intelligence analyst. Although I'm very new to front end development, I find it very interesting.

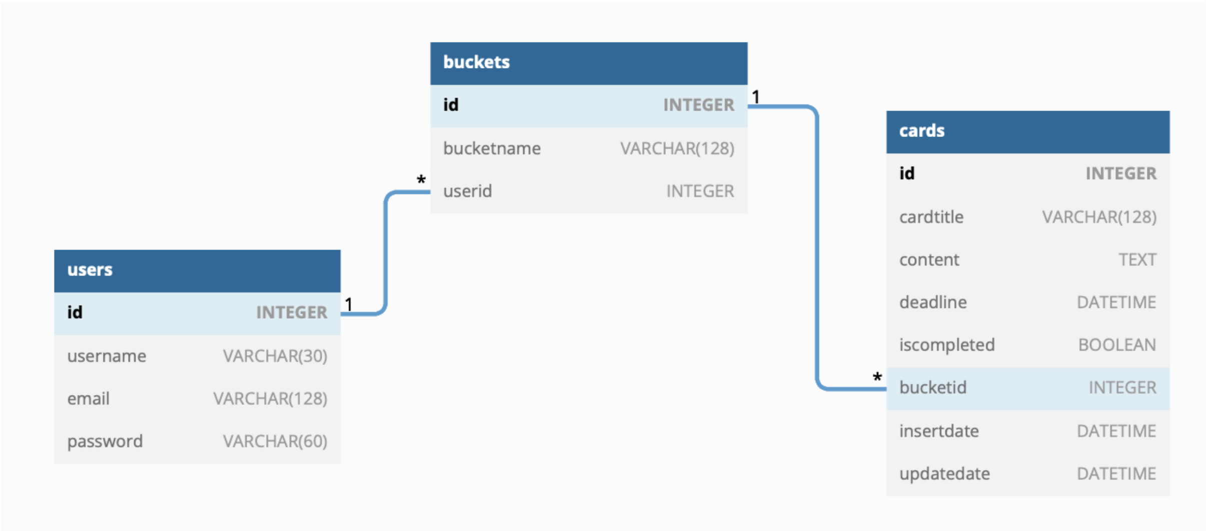
Description

Build a Kanban web application where in multiple users can create an account, login and make lists of tasks and track their performance with respect to the completion of the tasks based on their deadline

Technologies used

Flask – Main web framework
Flask-Login – to handle the user authentication
Flask-SQLAlchemy – ORM to communicate with sqlite db
Flask-WTF – To create forms for user input
Pandas – To analyse data for summary/reports
Matplotlib – For Data visulization

DB Schema Design



users	
id	INTEGER
username	VARCHAR(30)
email	VARCHAR(128)
password	VARCHAR(60)

buckets	
id	INTEGER
bucketname	VARCHAR(128)
userid	INTEGER

cards	
id	INTEGER
cardtitle	VARCHAR(128)
content	TEXT
deadline	DATETIME
iscompleted	BOOLEAN
bucketid	INTEGER
insertdate	DATETIME
updatedate	DATETIME

```
-- users definition
CREATE TABLE users (
      id INTEGER NOT NULL,
      username VARCHAR(30) NOT NULL,
      email VARCHAR(128) NOT NULL,
      password VARCHAR(60) NOT NULL,
      PRIMARY KEY (id),
      UNIQUE (username), -- username cannot repeat
      UNIQUE (email) -- email cannot repeat
   );

-- buckets definition
-- this is table to store the name of the lists, its called as buckets to
avoid naming conflicts with python list

CREATE TABLE buckets (
      id INTEGER NOT NULL,
      bucketname VARCHAR(128) NOT NULL,
      userid INTEGER NOT NULL, -- user who created the bucket
      PRIMARY KEY (id),
      -- one user cannot have more than one bucket with the same name
      UNIQUE (bucketname, userid),
      -- user associated with the bucket
      FOREIGN KEY(userid) REFERENCES users (id)
   );

-- cards definition
-- these are the tasks associated with each of the lists/buckets

CREATE TABLE cards (
      id INTEGER NOT NULL,
      cardtitle VARCHAR(128) NOT NULL,
      content TEXT NOT NULL,
      deadline DATETIME NOT NULL,
      iscompleted BOOLEAN NOT NULL,
      bucketid INTEGER,
      insertdate DATETIME NOT NULL,
      updatedate DATETIME NOT NULL,
      PRIMARY KEY (id),
      -- one bucket cannot have more than one card/task with the same name
      UNIQUE (cardtitle, bucketid),
      FOREIGN KEY(bucketid) REFERENCES buckets (id)
   );
```

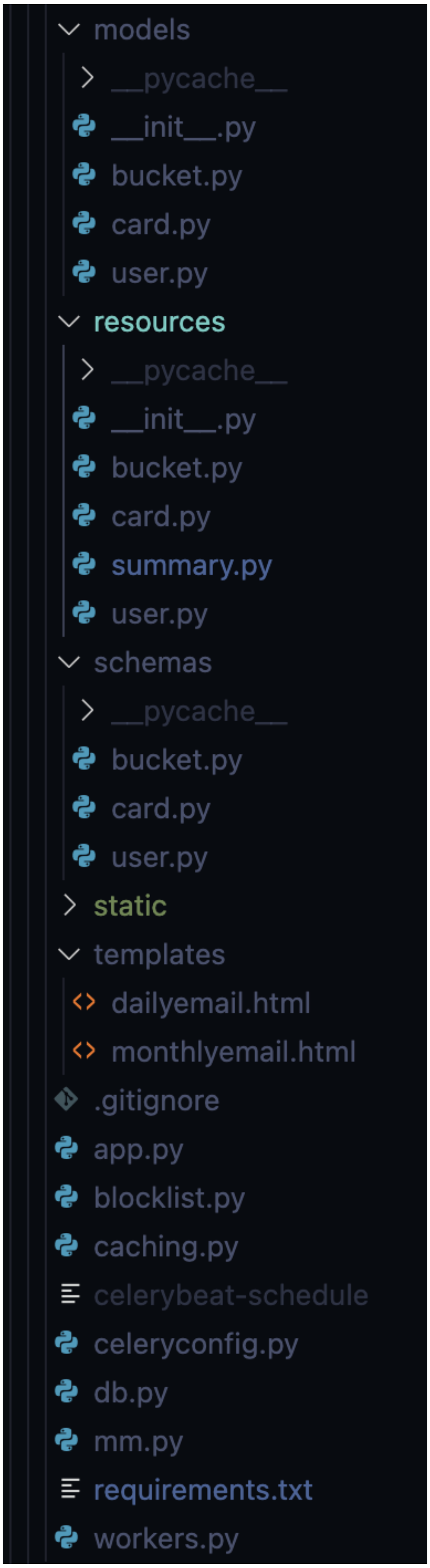
Architecture and Features

Folder and files structure

- **db_directory** has the sqlite DB. It can be anywhere on the machine. Adjust the path in application/config.py.
- **application** is where our application code is
- **env** folder sets up the virtualenv.
- **static** - default static files folder. It serves at '/static' path.
- **templates** - Default flask templates folder

Folder Tree

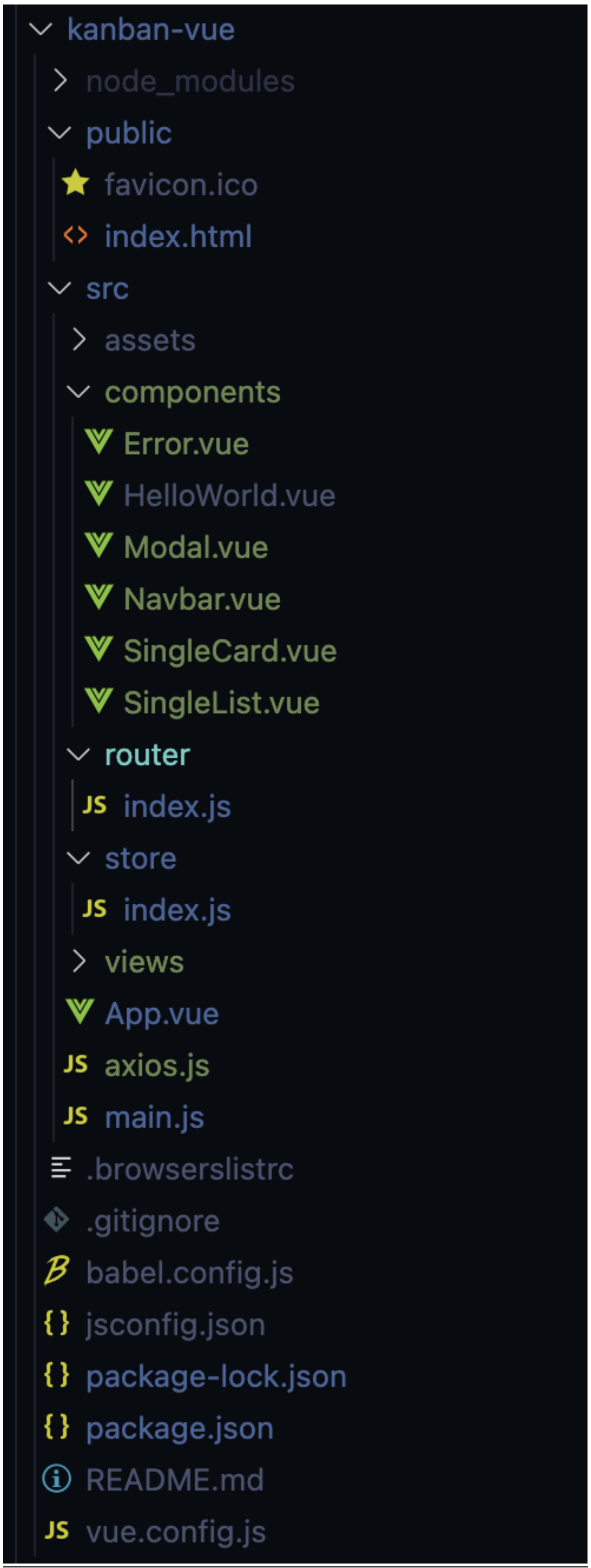
Flask backend API



```
models
  __pycache__
  __init__.py
  bucket.py
  card.py
  user.py
resources
  __pycache__
  __init__.py
  bucket.py
  card.py
  summary.py
  user.py
schemas
  __pycache__
  bucket.py
  card.py
  user.py
static
templates
  dailyemail.html
  monthlyemail.html
.gitignore
app.py
blocklist.py
caching.py
celerybeat-schedule
celeryconfig.py
db.py
mm.py
requirements.txt
workers.py
```

Folder Structure for Front end

VueJs Front end



```
∨ kanban-vue
  > node_modules
  ∨ public
      favicon.ico
      index.html
  ∨ src
    > assets
    ∨ components
        Error.vue
        HelloWorld.vue
        Modal.vue
        Navbar.vue
        SingleCard.vue
        SingleList.vue
    ∨ router
        index.js
    ∨ store
        index.js
    > views
      App.vue
      axios.js
      main.js
    .browserslistrc
    .gitignore
    babel.config.js
    jsconfig.json
    package-lock.json
    package.json
    README.md
    vue.config.js
```

Video

https://drive.google.com/file/d/1pSorclaPkWPna2BWa3P1-g2eCVdlctO7/view?usp=sharing