

Project Document

Author

Name: Shekhar Mohanty Hemant

Roll number: 21f1004056

Email: 21f1004056@student.onlinedegree.iitm.ac.in

Age: 22 years

About: Working professional at Tata Steel as Electrical technician, pursuing BS in Data Science and Applications from IITM.

Description

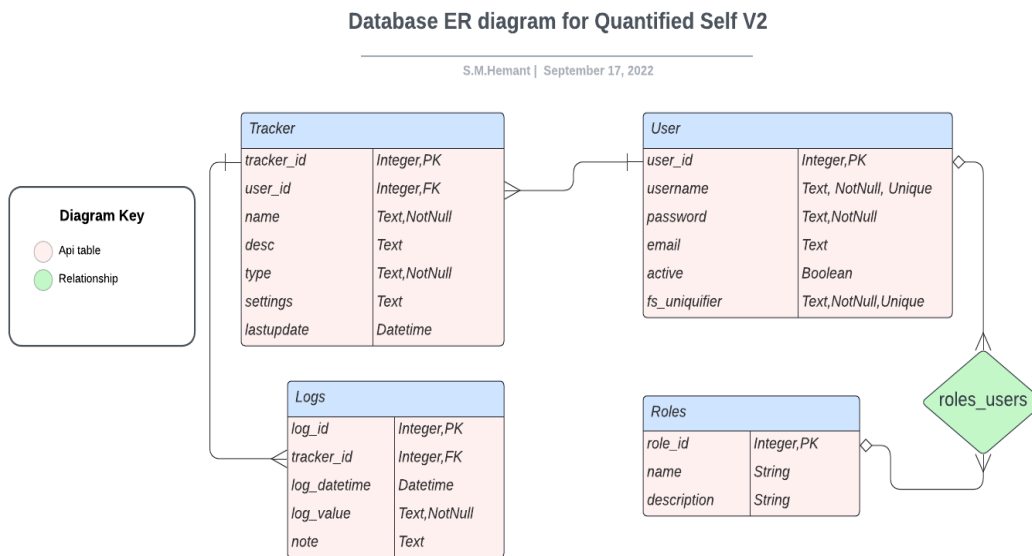
An application to allow users to log data of specific type and track it over any period of time along with features like exporting data, importing data, scheduling alerts and scheduling report generation automatically.

Technologies used

S.No	Technology name	version	Description
1	Flask	2.0.3	Framework for python application
2	SQLite	-	Database engine
3	Flask-SQLAlchemy	2.5.1	Flask extension providing sql db operations
4	Flask-RESTful	0.3.9	Flask extension for building REST API.
5	Flask-Security-Too	4.1.5	Flask extension for adding authentication in app.
6	Flask-Cors	3.0.10	For handling Cross origin resource sharing
7	Flask-Caching	2.0.1	For basic caching of application responses.
8	matplotlib	3.5.1	For creating charts for reports.
9	numpy	1.22.3	Handling chart data.
10	celery	5.2.7	Provides workers to execute task asynchronously
11	celery-redbeat	2.0.0	Provides ability to schedule tasks during runtime.
12	gevent	21.12.0	Works with gunicorn to provide async workers.
13	gunicorn	20.1.0	WSGI server for Flask application
14	redis	4.3.4	For a memory based database.
15	smtplib	-	For sending emails
16	httplib2	0.20.4	To send webhook responses
17	vue	2.7.8	Framework for Vue application

18	vue-cookies	1.8.1	To easily access browser cookies in vue script.
19	vue-router	3.5.1	Provides routes for navigation in the vue app.
20	vuex	3.6.2	A state management library for vue app.
21	vue/cli	5.0.8	Tool for handling vue application configurations.
22	echarts	5.2.0	For visualising data through dynamic charts.
23	bootstrap	5.2.0	Framework for developing responsive sites.

DB Schema Design



API Design

Four API classes:-

- Three of them has been designed for CRUD operations of all three resources present in application i.e, user, tracker and log
- A separate api has been designed for user authentication. [API document : yaml file.](#)

Architecture and Features

Application source code is divided into two parts

(i)Backend-for Flask application

(ii)Frontend- for vue application.

```

├── application (controllers,api,configuration)
├── exported_files (exported charts and pdf)
├── static (site assets like images,scripts)
└── templates (jinja templates for local site)

```

```

├── public (index.html, images)
├── src
│   ├── assets (svg, external scripts)
│   ├── components (spa components)
│   ├── router (vue router js file)
│   ├── store (vuex store js file)
│   └── views (dashboard,login view)

```

Video

[Presentation video link.](#)