# SFIA_project

Web application with CRUD functionalities,

# TITLE: Employee Management System

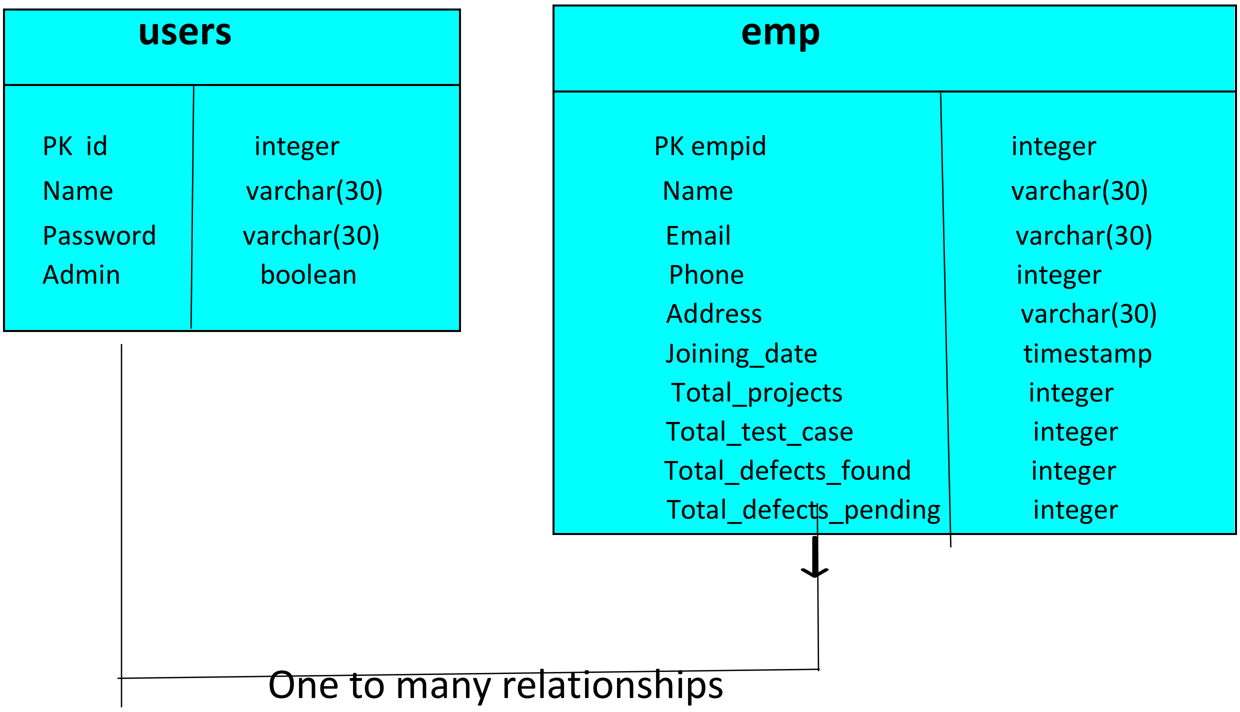
## Description:

Employee Management System is a web application that is used to keep track of employees’ personal details. This application enables us to perform CRUD functionalities where we can create/read/update/delete the employee’s records as per requirements. This is a web application with a python backend. Adopted flask web framework, and SQLite database to deliver CRUD functionality. This monolithic Flask application serves both the front and back end of the application.

The frontend aspect of the app will use HTML templates to serve the web pages, allowing the user to perform CRUD functionality with information from the SQL database.

The backend aspect of the application will use SQLite to model and integrate with the database. In the database, there is one table(entity) called emp and another table(entity) called users. Users table maintaining/creating users for login having fields(attributes) like id, username, and password for authentication, allowing access to employee’s details. Emp table has some fields(attributes) like Emp_id, Full name, phone, Address, Email_id, joining date, number of projects handled, etc. We can view all employee records and can also view specific full profile details for a single employee. We can add the new employee’s records to the table and can also delete or modify the records as per requirements.
This application is continuously built, tested, and deployed using Jenkins pipelines and will be hosted in a container to allow it to be deployed to a Docker Swarm.

SQLite SFIA_project.db ERD



users

PK id	integer
Name	varchar(30)
Password	varchar(30)
Admin	boolean

emp

PK empid	integer
Name	varchar(30)
Email	varchar(30)
Phone	integer
Address	varchar(30)
Joining_date	timestamp
Total_projects	integer
Total_test_case	integer
Total_defects_found	integer
Total_defects_pending	integer

One to many relationships

# THE OBJECTIVES OF THIS PROJECT TO ACHIEVE:

- To create a simple web application that maintains employees' personal details and integrates with a database to demonstrate CRUD functionality.
- To utilize containers to host and deploy this web application.
- To create a continuous integration (CI)/continuous deployment (CD) pipeline that will automatically test, build and deploy the application.

# FEATURES:

- Create new employee details.
- Review employee details.
- Update employee details.
- Delete employee details.

# BUILT WITH:

- Major framework/libraries and software used for this project:
- Python 3 – The programming language used.
- Flask- The web framework used
- SQLite3
- Docker
- Jenkins

# GETTING STARTED:

PREREQUISITES:

We need to install the following software for this project.

- Python IDE
- Flask framework
- SQLite3 for database
- Jenkins
- Docker
- Azure cloud service.

Minimum requirement：

Python 3.7 or above

Need to run this command from project directory to set python environment:

```
Python3 -m venv venv
```

Install all required python modules using requirements.txt :

```
Pip install -r requirements.txt
```

# USAGE:

To run this application (venv)->

```
Python3 app.py
```

# **HOW TO USE THE APP**:

This application uses to maintain employees' records of their personal details like name, id, email, phone, address, joining date, and total projects handled.

There is a Login page for the users who have access to employee records. If you are already a registered user, then you can directly log in with your credentials (username and password), if you are a new user then you first have to register yourself to access the employee's records. and then log in to the account.

After login, the authentication process is done and you will be at the dashboard page where You can find the 'add new employee' button. you can create/add new employee records here. you can also read/find the 'list of all employees. You can select one specific employee and can 'view full profile'/personal details including how many projects he/she handling etc. You can also update/modify the records for any employee by clicking on the 'update' button. You can delete the employee profile as well by clicking on the 'delete' button. So basically you can perform CRUD functionality here.

After this, you can log out from your account by clicking on the logout tab.

This is a basic web app displaying how to perform CRUD functionalities with SQLite database integration with the flask framework and also continuously build, test, and deploy through Jenkins pipelines via docker containerization and docker swarm hosted in the cloud.

#Deployment: Copy the IP address of VM:5000 (port) into the browser to view the application running on the server.

The work progress and the code can be tracked on

GitHub (anugoyal13/SFIA_project: web application with CRUD functionalities (github.com)).