

Short Documentation of The Technical Issues Manager Web Application

This project is made by **OURABI Safouen**
during the internship at
SAMM Test and Automation

This documentation serves as a brief showcase for the portfolio. It is intended for demonstration purposes only and is not the official documentation.

2022--2023

The Goal of the project :

The goal of the web project is to create a web application that serves as a link between the support team and employees. Its primary purpose is to efficiently handle and manage software and hardware issues faced by the employees.

Technology Used

Front-end:

1. HTML: To structure the content of the web app.
2. CSS: To style the HTML elements and control the presentation of the web app.
3. JavaScript: To add interactivity and dynamic behavior to the web app.

Back-end:

1.Django:

To handle the server-side logic, manage the application's data, and facilitate communication between the front-end and the database.

It is understood that the project does not require the Bootstrap framework, and it is not meant to be responsive as it is intended to run in full-screen mode only. This means that the user interface will be designed to fit a specific screen size without adjusting to different devices or resolutions.

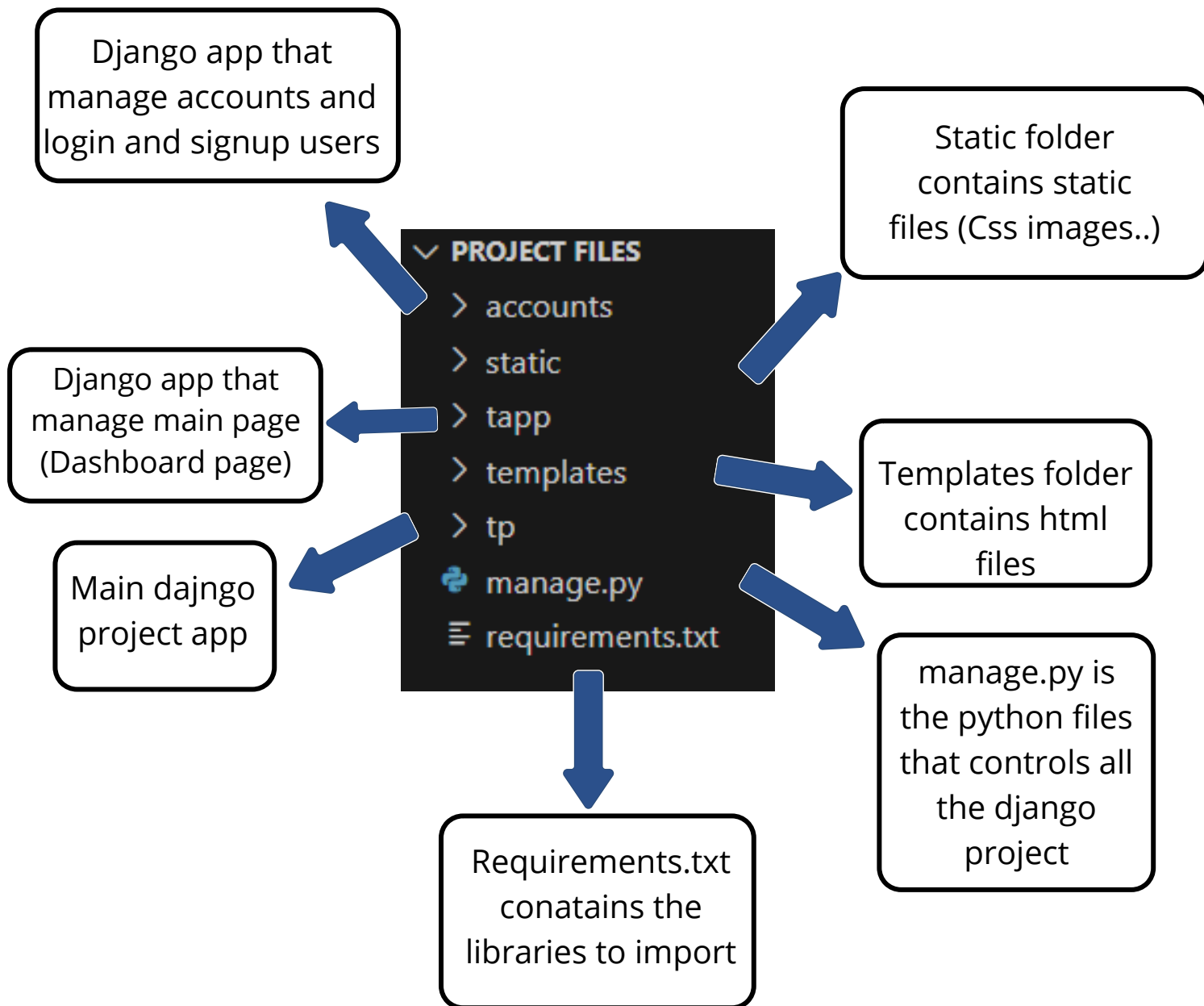
With this technology stack, the project can create a web app that efficiently links the support team and employees to manage software and hardware issues. Django's robust features and capabilities will enable smooth data handling and server-side operations, while HTML, CSS, and JavaScript will provide a functional and interactive user interface for the web application.

2.PostgreSQL :

PostgreSQL is a popular open-source relational database management system known for its robustness, extensibility, and support for advanced features like JSON data, full-text search, and geospatial data. It integrates well with Django, and Django provides an Object-Relational Mapping (ORM) layer, making it easier to interact with the PostgreSQL database using Python code.

By using PostgreSQL, the web application can handle and store data efficiently, ensuring data integrity, scalability, and performance. Django's ORM abstracts the database operations, allowing developers to work with database records and relationships through Python classes and objects, reducing the need for writing raw SQL queries and enhancing development productivity.

Project Structure:

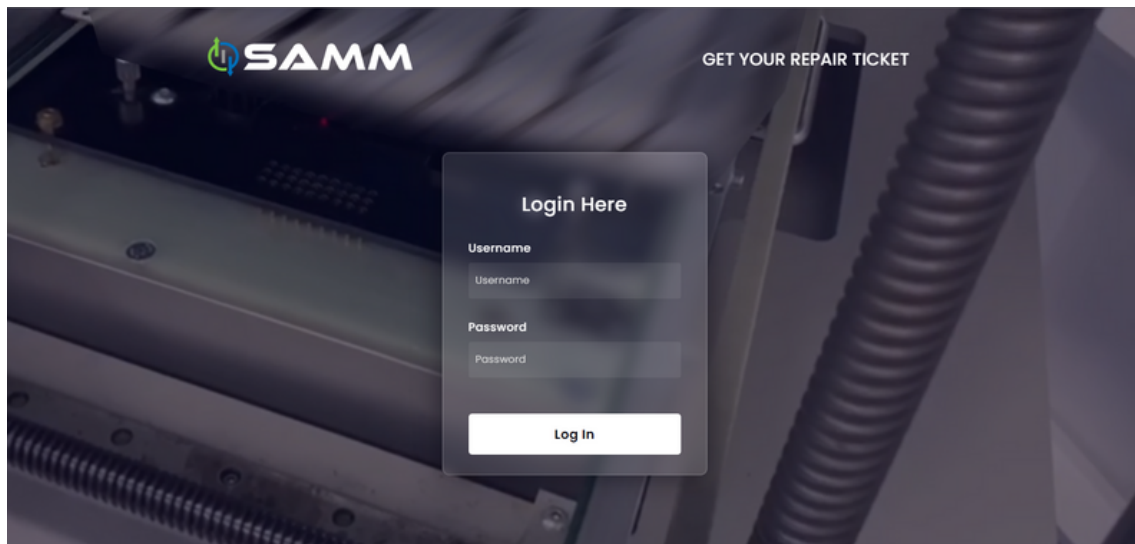


Features and Functionality

Note : In This Project The Technical Support Request From The Employees is Considered as a Ticket

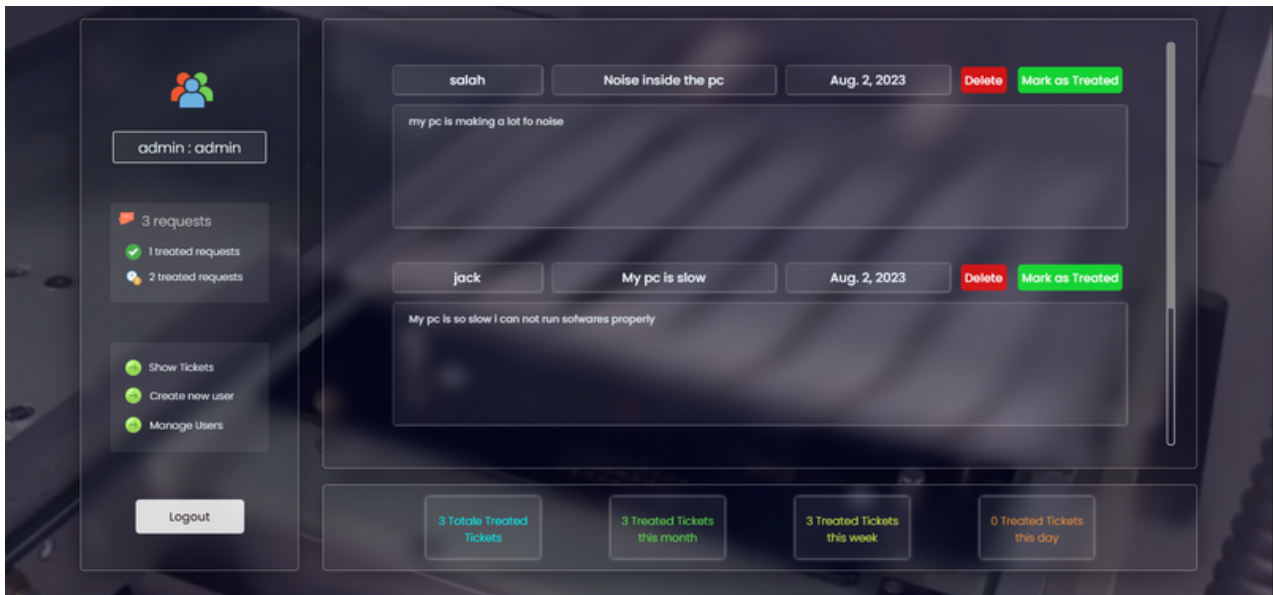
Technical Support Request = Repair Ticket

This web app has an admin which is a member of IT support Team and regular users which are the employees. You can login as an admin or a normal user.



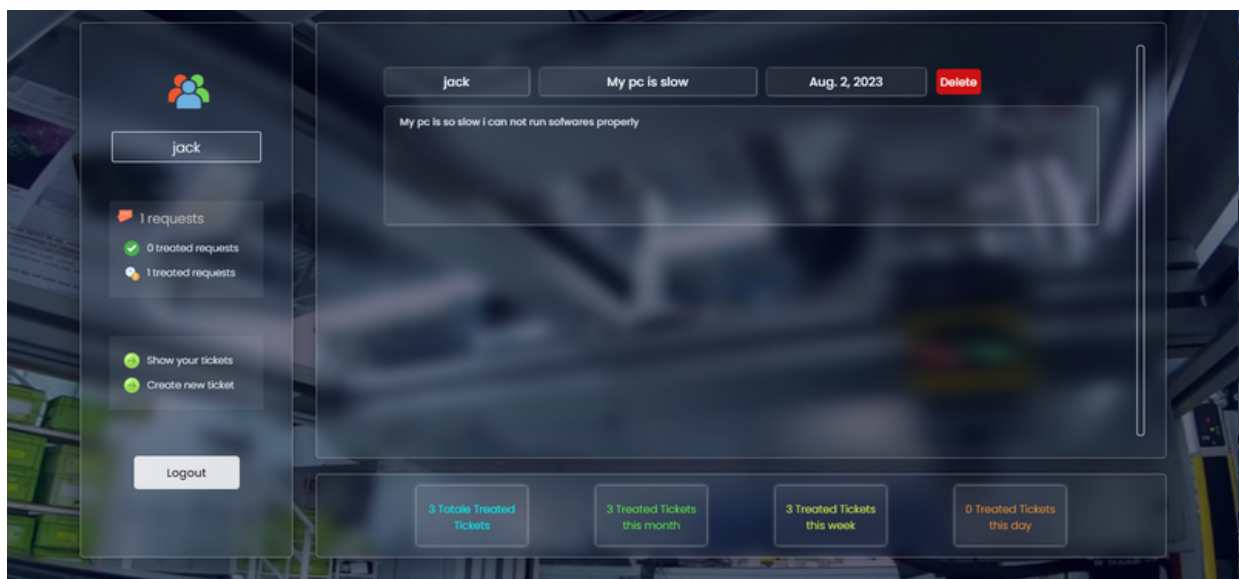
Admin user privileges

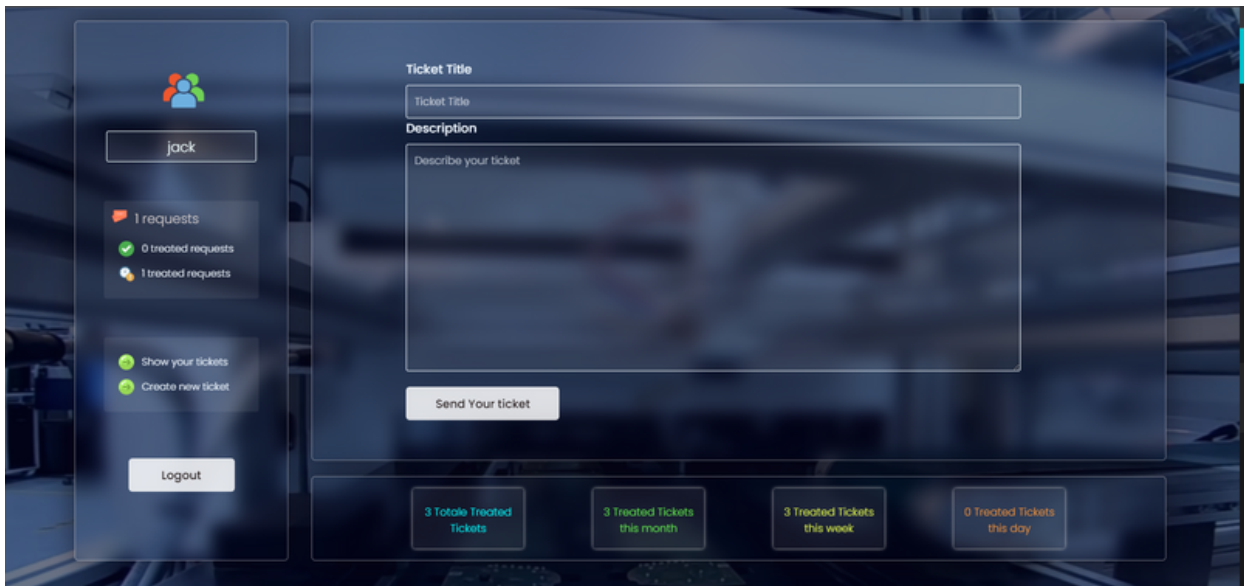
- Add new users
- Delete Users
- Modify users Data
- See all tickets cerated by all users
- Delete any tickets
- Mark tickets as treated tickets



Normal user privileges

- Create a new Repair Ticket
- See all Tickets created by him
- Delete his Tickets
- Mark Tickets as treated Tickets
- See Totale requests





Thank You

This documentation serves as a brief showcase for the portfolio. It is intended for demonstration purposes only and is not the official documentation.

Safouen Ourabi

ICT Engineering Student, Higher School of Communication of Tunis

Phone : +216 92739560

Email : safouen.ourabi@supcom.tn

SUP'COM

