Low Fidelity Prototype

Project Title: **Final Year Project management system using theory of shame**

Student Name	**:**	**Muhammad Haikal**
Matric Number	**:**	**283950**
Supervisor	**:**	**Ahmad Hanis Bin Mohd Sabli**

1. PROBLEM AND SOLUTION OVERVIEW

1.1. Problem

This initiative endeavors to address the challenges prevalent in education project management by conceptualizing a user-centric Final Year Project Management System, which integrates practical elements alongside the application of the theory of shame. The primary objective revolves around the development of this system to align with the anticipated requirements, coupled with its meticulous testing utilizing black-box methods while concurrently assessing its user-friendliness. Ultimately, the overarching aim is to augment the management of final year projects within educational institutions by establishing an efficient and user-focused framework.

1.2. Objective

The objectives of the development of this application is as follows:

- Designing the architecture, user interface, and system flow of the Final Year Project system with the applied Theory of Shame to enhance its accessibility, usability, and security.
- Developing the Final Year Project management system by incorporating the theory of shame, utilizing essential tools, and conducting independent testing.
- Conducting a user-friendliness evaluation and black-box testing on selected users of this application.

1.3. Proposed Solution

The proposed solution entails the development of a web application designed to facilitate the tracking, reminders, and visibility of tasks for final year project stakeholders, including students, supervisors, and coordinators. This application aims to provide a centralized platform where users can access their current tasks and receive reminders for upcoming deadlines. Additionally, the system will highlight users who have not submitted their tasks, allowing for efficient monitoring and intervention as necessary. Through these features, the web application seeks to streamline project management processes.

2. DESCRIPTION OF LOW-FIDELITY PROTOTYPE

2.1 Programming Tools

Category	Tool
Front-End	❖ React ❖ React Router ❖ TypeScript ❖ React-unicon ❖ esLint ❖ styled-components ❖ Vite ❖ Motion Framer ❖ Material UI (MUI)
Back-End	❖ ExpressJS ❖ express-router ❖ bcrypt ❖ modemon ❖ postgres pg ❖ postman
Database	❖ PostgreSQL
Utility	❖ Docker ❖ Git ❖ Github

2.2 General Functionalities

SPARES, an acronym for Shame-Prompted Action for Responsible Evaluation and Submission, introduces a unique environment tailored for academic staff and students to engage in event and task tracking within the educational setting. At the heart of SPARES lies the concept of "shaming," whereby individuals face the prospect of having their names listed in a designated "Hall of Shame" should they fail to meet established expectations or deadlines. This innovative approach leverages psychological pressure as a means of fostering accountability and promoting timely action among users. By integrating the element of shame into the system's framework, SPARES creates an environment wherein users are motivated to fulfill their responsibilities and adhere to project timelines, thereby enhancing overall productivity and accountability within the educational context.

2.3 Limitations

A notable limitation of the SPARES system is its reliance on modern web browsers supporting the latest version of JavaScript for accessing the website, thus potentially excluding users with outdated or incompatible browsers. Moreover, internet connectivity is essential for accessing the SPARES website, posing a barrier to users in areas with limited or unreliable internet access. These constraints may restrict the accessibility of the SPARES system, particularly for individuals in regions with technological or connectivity challenges.

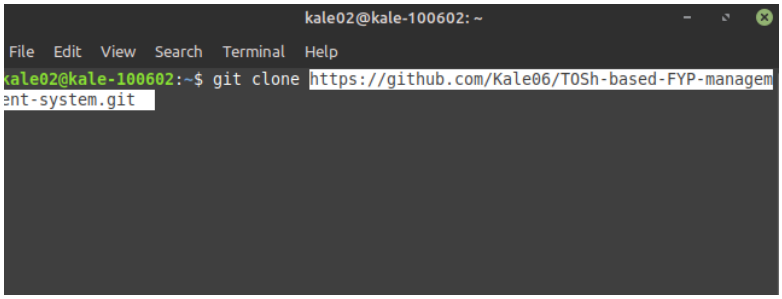
3. MAIN FUNCTIONALITIES

3.1 Running and Stopping the System

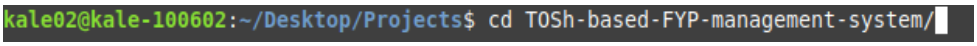
Listed below is the manual to run the prototype

Starting SPARES

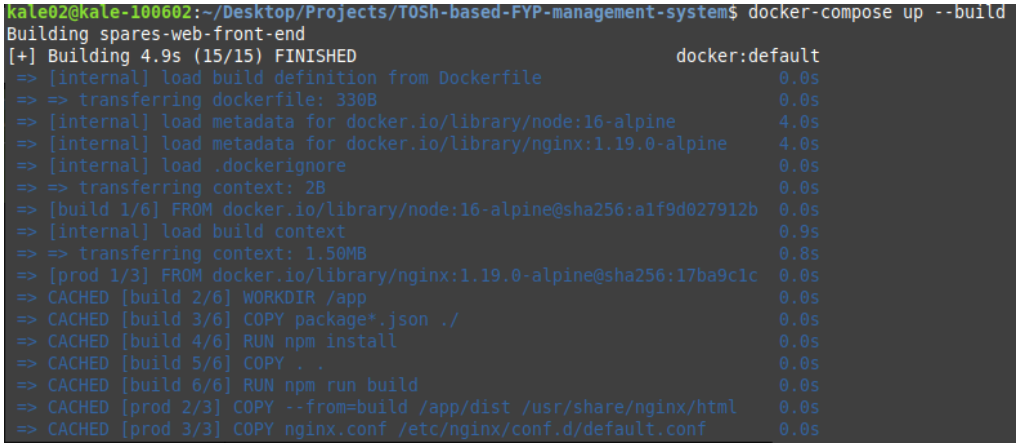
- Clone the github repository using git https://github.com/Kale06/TOSh-based-FYP-management-system.git



- Change the directory of the terminal into the project's directory

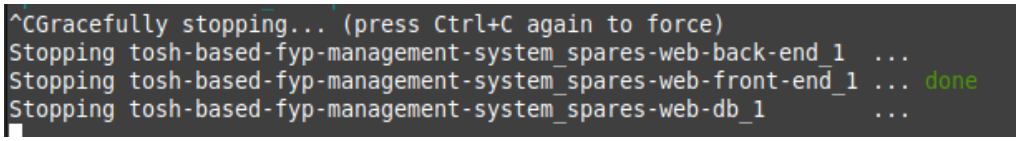


- Use docker's command docker-compose up –build to build the project and run it simultaneously



Stopping SPARES

- Press ctrl + c on the terminal



- Wait for the container to stop

Pre-requisites of Running the system

Listed below are the required software of running the on-going prototype :

- Docker
- Git

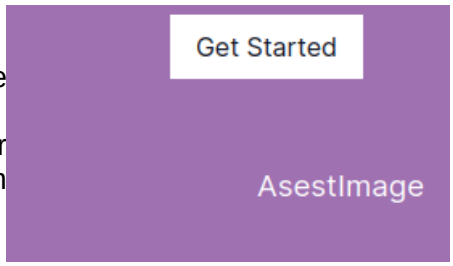
3.2 Specific Functionalities

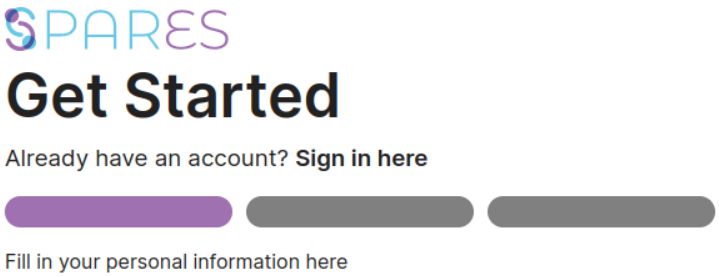
(description about your prototype's functionalities – provide explanation for each screenshot. The description must cover input, process and output perspectives)

a. Login

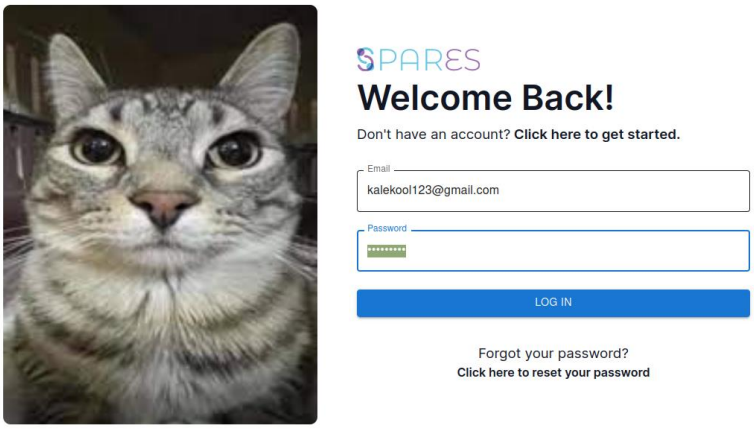
User clicks on the get started button then they will be navigated to the registration page.

In the registration page the user clicks on Sign in here.

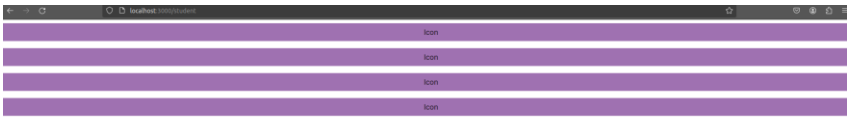




Using a created account from the database, user inputs their email and password respectively where then it will send to the back-end of the server to generate a token where the token then will be sent back to the client-side.

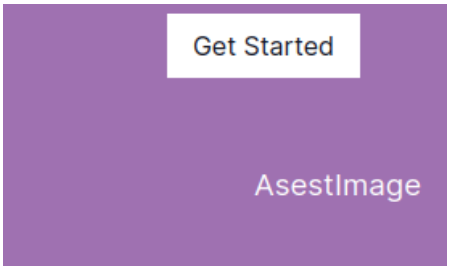


The token then will be stored in the cookies of the web for further validation and the user will be navigated to the dashboard.

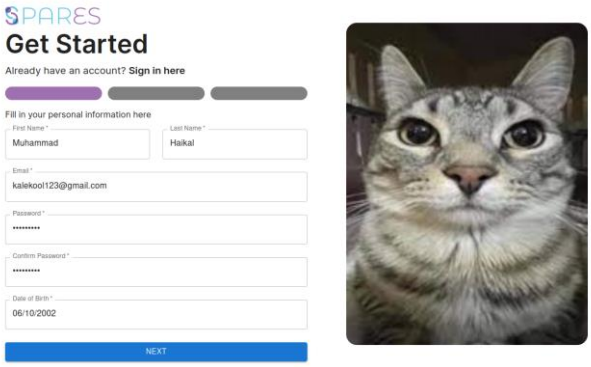


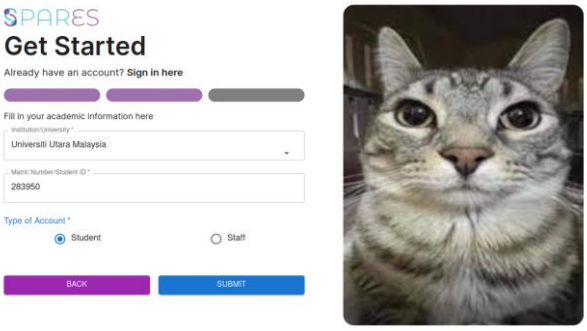
b. Register

User registers by clicking on the get started button.

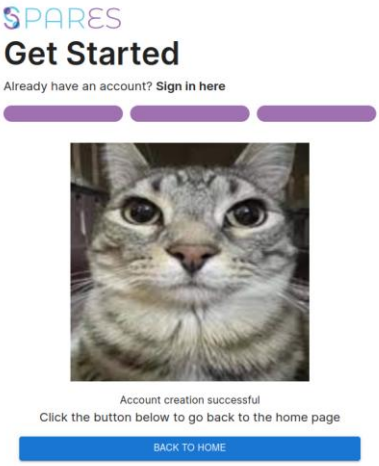


User then inputs the required datas.





User then clicks on submit where then the back-end will generate a salt and a hashed password for the user.


SPARES
Get Started
Already have an account? Sign in here
Account creation successful
Click the button below to go back to the home page
BACK TO HOME

Prepared by

Student's Signature

Date

Supervisor's Approval

Remark:

Project Supervisor's Signature & Stamp

Date