**TASKLY**

1. **SYSTEM IDENTIFICATION**

As humans, we are often faced with tasks to do daily, weekly or even yearly. While each task has consequences when left undone, some might be more severe than others. We are not computers and are prone to forgetting.

With the introduction of the web application taskly, it helps users create, view, update and delete tasks. It also prioritizes tasks based on importance and puts them in categories.

The users of this system are the administrators and all humans.

1. **SDLC APPLICATION**

**Planning Phase**

An opportunity to create a website that allows users perform the basic CRUD on tasks.

Scope: It includes all tasks inputted by the user and excludes data tracking.

Objectives: It aims to optimize the performance of the users.

Expected value: It has direct user benefits. If used properly, it can optimize and save users’ time. It also helps the user make plans on their daily tasks.

Feasibility Analysis:

Technical; Can we build it?

Front end- React

Backend- Python

Data storage- SQL lite

Economic; Will it provide business value?

Yes it would, if marketed and sold to the target audience which are working class youths.

Operational; Will it be used?

Yes, there is high user demand.

Project Constraints:

There is financial constraints due to it being a startup.

There are time constraints to finish it before the beginning of the next economic quarter.

The website has to be in accordance with the data protection laws of the country.

1. **Analysis Phase**

To get the requirements for the website, I combined a quantitative survey and a competitor’s analysis of a slightly similar website.

FUNCTIONAL REQUIREMENTS:

1. **User Authentication**
   * Users should be able to sign up, log in, and log out.
   * Admins should be able to manage users.
2. **Task Management (CRUD)**
   * Users can create new tasks.
   * Users can view their tasks.
   * Users can update/edit tasks.
   * Users can delete tasks.
3. **Task Prioritization**
   * Users should be able to set task importance (e.g., high, medium, low).
   * System should sort tasks by priority.
4. **Task Categorization**
   * Users should be able to group tasks into categories (e.g., Work, Personal, School).
   * Users can filter tasks by category.
5. **Reminders/Notifications**
   * System may notify users of upcoming or overdue tasks.
6. **Admin Features**
   * Admin can view overall system usage.
   * Admin can delete inappropriate data (if user support is needed).

NON-FUNCTIONAL REQUIREMENTS:

1. **Performance**
   * The system should load tasks within 2 seconds.
   * The database should handle up to 10,000 tasks without performance issues.
2. **Usability**
   * The interface should be simple and intuitive for all age groups.
   * Tasks should be manageable in less than 3 clicks.
3. **Security**
   * Passwords must be encrypted.
   * Data must comply with data protection laws.
4. **Scalability**
   * The system should allow expansion (adding more features like reminders, collaboration, mobile app).
5. **Reliability & Availability**
   * The system should be available 99% of the time.
   * Data should be backed up daily.
6. **Compatibility**
   * Accessible via major browsers (Chrome, Edge, Firefox, Safari).
7. **Maintainability**
   * Code should follow best practices for readability and debugging.
   * Future developers should be able to add features easily.

**UseCase Diagram**

