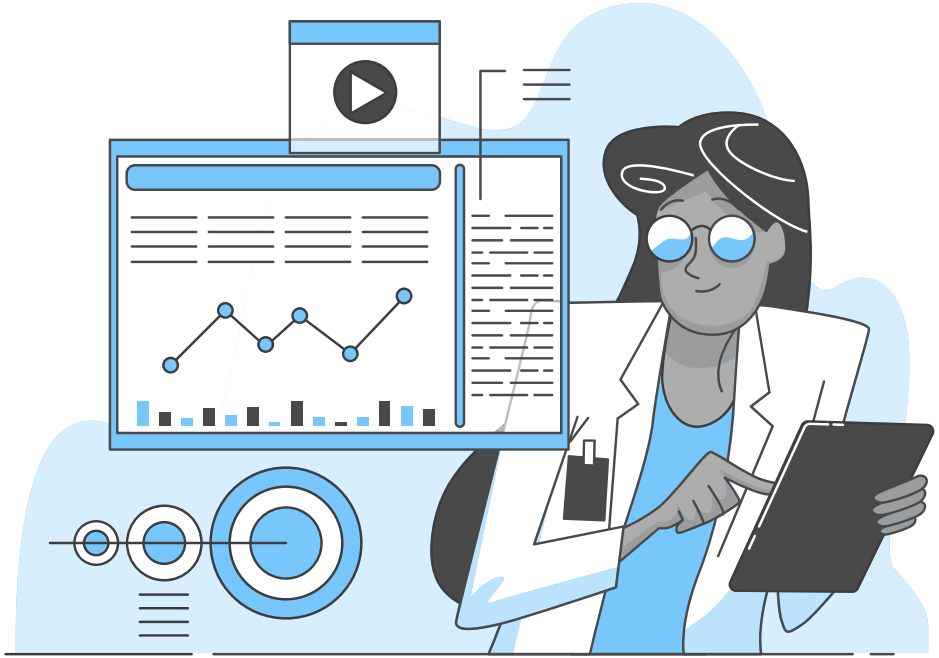
MORINGA LOST AND FOUND APPLICATION

Problem statement

In today’s fast-paced world, losing personal items such as keys, wallets, smartphones, and other valuables is a common and stressful occurrence. Traditional methods of finding lost items, such as relying on verbal announcements or physical notices, often prove to be inefficient and ineffective. Many individuals struggle to reunite with their lost belongings, leading to frustration and, in some cases, significant financial loss.
Current solutions for lost and found management are fragmented and inadequate. Users frequently face challenges in:

Solution

To address these challenges, the goal is to develop a comprehensive Lost and Found application that streamlines the process of reporting, tracking, and reuniting lost items with their owners. The application should provide a user-friendly interface for reporting and searching for lost items, facilitate effective communication between finders and seekers, and include robust verification mechanisms to ensure authenticity and reliability.



Meet the Team

01

James Kimanzi

UI / UX Designer

02

Moshood Abiola

Frontend Developer

03

Peter Ngujiri

Frontend Developer

04

Dinah Ngatia

Backend Developer

05

Kevin Moino

Backend Developer

Tools used

React

Developing the frontend

Flask

Developing the backend

Git Hub

For version control and collaborating

Figma

Designing of the UI /UX components

DB diagram

Database design and drawing ERD diagrams

…

…

…

Challenges

1. Integration - Integration between the React frontend and Flask backend works proved challenging as we encountered several errors.

2. Deployment complexity - Deploying both the React frontend and Flask backend involved setting up and maintaining separate deployment pipelines and we needed to handle issues related to deployment environments, such as setting up servers and managing dependencies.

3. Collaboration - Coordinating between frontend and backend developers and managing the integration process presented a significant challenge and resulted in components not working most of the time

...

Milestones

Frontend Development - Implement key user interfaces in React, such as item reporting forms, search functionality, and user authentication.
Backend Development - Develop core backend features, including data models, item tracking, and user management.
Frontend and Backend Development - We were able to host both our backend and front end in remote servers
Testing and Quality Assurance - Write and execute unit tests for both React components and Flask endpoints to ensure they function correctly in isolation.