PROJECT REPORT

On

FoundIt

Submitted in partial fulfilment of the requirement for the
Course
FULL STACK ENGINEERING (22CS037)
of

COMPUTER SCIENCE AND ENGINEERING
B.E. Batch-2022

Under the guidance of	**Submitted By:**
Mr. Rahul.	**Arshia 2210991368**
	Ishika 2210991689
	Mandeep 2210991879

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CHITKARA UNIVERSITY

Abstract

FoundIt is an innovative, user-centric web-based platform designed to simplify and streamline the process of reporting and recovering lost and found items. Built using the MERN stack (MongoDB, Express.js, React.js, Node.js), the platform offers an efficient and transparent solution to the common challenges faced in lost-and-found reporting, such as lack of visibility, ineffective communication, and delays in item recovery.

Traditional lost-and-found processes are often scattered across multiple channels, including offline notice boards, social media groups, and unorganized online platforms. These methods result in inefficiencies, making it difficult for users to track or recover their lost belongings. FoundIt eliminates these challenges by providing a centralized platform where users can report lost or found items, ensuring better visibility, faster resolutions, and improved communication.

The platform offers a seamless experience for users looking to report or search for lost and found items. Users can create detailed reports, specifying the item description, location, and timestamp of the lost or found event. An advanced search feature with customizable filters allows users to browse active and resolved cases effortlessly. Real-time updates and notifications ensure that users remain informed about the status of their reports and any potential matches.

Security and trust are key priorities for FoundIt. The platform includes user authentication via OTP verification, ensuring that all interactions are secure. Email notifications are sent to new users upon registration and for important updates related to lost-and-found reports. This verification system enhances trust within the community and reduces the risk of fraudulent claims.

To further improve efficiency, FoundIt plans to integrate advanced technologies in future updates. A chatbot assistant will be introduced to help users navigate the platform, provide search assistance, and improve the overall user experience. AI-driven matching algorithms will enhance search results, increasing the chances of successful recoveries. Additionally, a mobile application is in development to make the platform more accessible on the go.

By offering a structured and transparent system, FoundIt aims to create a community-driven approach to lost-and-found reporting, ensuring a faster and more reliable way for people to reclaim their belongings.

1.Introduction to the project

1.1 Background

Losing or finding valuable items is a common yet frustrating experience for individuals, whether in public places, workplaces, or educational institutions. The traditional methods of reporting and retrieving lost-and-found items are often ineffective, fragmented, and time-consuming. People rely on notice boards, social media posts, or word-of-mouth, which lack a structured approach and often result in unsuccessful recoveries.

For individuals who lose items, the challenge lies in identifying a centralized platform where they can report their loss and increase the chances of retrieval. Searching for lost belongings typically involves visiting multiple locations, checking with authorities, and browsing unorganized online listings. This lack of accessibility and coordination makes the process inefficient and frustrating.

Similarly, individuals who find lost items struggle with finding the rightful owners. Without a proper reporting system, many items remain unclaimed or go unnoticed. Public lost-and-found departments, while helpful, often lack proper digital record-keeping, leading to misplaced reports and unorganized management.

Another significant issue is the lack of transparency. Many lost-and-found reports remain unnoticed due to poor visibility, and there is often no way to verify whether a claim is genuine. This gap in the system creates uncertainty for both those who lose items and those who find them.

FoundIt is designed to address these challenges by offering a user-friendly digital platform dedicated to reporting and managing lost-and-found items efficiently. By leveraging modern web technologies such as the MERN stack (MongoDB, Express.js, React.js, Node.js), FoundIt provides a centralized and structured solution where users can report, search, and communicate regarding lost and found items.

The platform's features include detailed item reporting, advanced search filters, real-time status updates, verified claims, and secure communication channels. Users can describe the lost or found item, specify the location and time of the event, and receive timely notifications regarding updates on their reports.

Security and transparency are key priorities for FoundIt. The platform implements OTP verification to authenticate users and prevent fraudulent claims. Email notifications are sent to registered users to keep them updated on the status of their reports.

Looking ahead, FoundIt aims to incorporate AI-driven search recommendations to match lost-and-found reports more efficiently. Additionally, a chatbot assistant is planned to help users navigate the platform and provide real-time assistance. With future scalability in mind, mobile application development will further enhance accessibility, ensuring users can report and track lost-and-found items conveniently from anywhere.

FoundIt envisions a structured, transparent, and community-driven approach to lost-and-found reporting, ensuring a higher success rate for item recoveries while reducing the frustration and inefficiencies of traditional methods.

1.2 Objective:

The primary objective of **FoundIt** is to create a structured, efficient, and transparent platform for reporting and retrieving lost-and-found items. By centralizing the process and enabling direct communication between individuals who lose or find items, the platform eliminates inefficiencies in traditional lost-and-found systems. The specific goals of FoundIt are as follows:

- **Simplify the Lost-and-Found Process:** FoundIt provides an intuitive platform where users can easily report lost or found items with details such as description, location, and time. By offering a centralized system, it eliminates the need for scattered reports across social media, notice boards, or offline records, improving efficiency and success rates.

- **Enhance Transparency and Trust:** The platform ensures that every lost or found report is visible to all users in real-time. Verified user authentication through OTP verification prevents fraudulent claims and ensures that only genuine reports are submitted. Users can also update the status of their reports, marking them as resolved when an item is successfully retrieved.

- **Streamline Communication:** FoundIt integrates a secure messaging system that allows users to directly communicate with the person who reported an item as lost or found. This eliminates the need for intermediaries and speeds up the resolution process.

- **Ensure Security:** Security is a top priority for FoundIt. By incorporating user authentication via OTP and email verification, the platform prevents spam and fraudulent claims. Data encryption and verified listings ensure the integrity of lost-and-found reports, reducing the chances of false claims.

- **Advanced Search and Filters:** The platform includes an intelligent search system that allows users to filter lost or found items based on category, location, and date. This feature enables users to quickly locate relevant reports and improve their chances of finding their belongings.

- **Improve Report Management:** Users have access to a personalized dashboard where they can track the status of their lost or found reports, update details, and communicate with potential claimants. This structured approach makes it easier to manage and monitor ongoing cases.

Through these objectives, **FoundIt** aims to streamline the lost-and-found process, increase accessibility, foster trust, and improve the chances of reuniting individuals with their lost belongings efficiently and transparently.

1.3 Significance:

FoundIt is designed to revolutionize the way lost-and-found cases are handled by addressing inefficiencies in traditional systems. Its significance lies in the following key areas:

- **Increased Accessibility:** FoundIt provides a centralized digital platform where users can report lost or found items easily. Unlike traditional methods such as posters, social media posts, or word-of-mouth communication, FoundIt ensures that reports are accessible to a wider audience, increasing the chances of item recovery.

- **Enhanced Transparency:** Trust is a major concern in lost-and-found cases. Individuals often face difficulties verifying the authenticity of found item reports or ensuring rightful ownership. FoundIt addresses these issues by implementing user authentication, verified listings, and status updates, ensuring that all interactions on the platform are genuine and traceable.

- **Efficiency and Cost Savings:** Traditional lost-and-found methods require manual efforts, time, and sometimes third-party assistance (such as local authorities or security offices). By eliminating these inefficiencies, FoundIt speeds up the process, reduces unnecessary efforts, and provides a cost-effective solution to locating lost belongings.

- **Improved User Experience:** FoundIt offers an intuitive and user-friendly interface, enabling users to report lost or found items with minimal effort. Its search filters, real-time updates, and direct messaging system ensure a seamless and efficient experience for all users.

- **Scalability and Growth Potential:** Built with modern web technologies, FoundIt is highly scalable and can accommodate an increasing number of users and reports. Future enhancements such as AI-powered matching, image recognition for lost items, and chatbot-assisted searches will further improve the platform's efficiency and usability.

- **Global Impact:** Although initially designed for local communities and institutions, FoundIt has the potential to expand globally. With features like multi-language support, integration with local lost-and-found services, and mobile app accessibility, FoundIt can become a universal platform for reuniting people with their lost belongings worldwide.

Through its focus on efficiency, security, and accessibility, **FoundIt** aims to redefine the lost-and-found process, making it more structured, transparent, and effective for everyone.

2. Problem Definition and Requirements

2.1 Problem Statement

In today's fast-paced environment, reporting and recovering lost items remains a frustrating and inefficient process. Traditional methods such as posting flyers, making announcements, or relying on word-of-mouth communication often result in delays, limited reach, and a low success rate. Many lost items go unreported, and found items remain unclaimed due to the absence of a structured and reliable system.

Several key challenges contribute to the inefficiency of the current lost-and-found system:

- **Limited Accessibility:** People who lose or find items often struggle to reach a wide audience. Notices posted on bulletin boards, social media groups, or word-of-mouth communication limit visibility, reducing the chances of recovering lost items.

- **Inefficient Communication:** Individuals who find lost items may have difficulty identifying the rightful owner, leading to delays or disputes. Conversely, those searching for lost items may struggle to track updates or contact the person who found them. Without a dedicated platform, communication is often fragmented and unorganised.

- **Lack of Centralisation:** Lost-and-found reports are scattered across various sources such as social media, community groups, or institutional offices, making it difficult for users to track and cross-reference information. The absence of a unified platform results in inefficiencies and missed connections between those who lose and find items.

- **Trust Issues:** There is often no way to verify the legitimacy of a lost-and-found report. People may hesitate to return found items due to concerns about fraud, while those searching for their belongings may find it challenging to confirm whether an item truly belongs to them.

FoundIt is designed to tackle these challenges by providing a **centralised, transparent, and user-friendly** platform where individuals can report and search for lost items efficiently. With features like verified user authentication, a structured search and listing system, and direct communication channels, FoundIt streamlines the lost-and-found process, increasing the chances of item recovery and fostering trust within communities.

2.2 Software and Hardware Requirements

Frontend Technologies

The frontend of **FoundIt** is designed using modern web technologies to ensure an intuitive, interactive, and responsive user experience across various devices.

- **React.js** – The application is built using React.js, a powerful JavaScript library for developing dynamic and reusable UI components. It efficiently manages state using the virtual DOM and promotes a seamless user experience. React Router is used for client-side navigation, ensuring smooth transitions between pages.

- **HTML5** – Provides the structural foundation of the web application, ensuring semantic and accessible content organization.

- **CSS3** – Used for styling the application, ensuring responsiveness and a visually appealing interface. Tailwind CSS or Bootstrap may be employed to streamline the design process with pre-built components.

- **JavaScript (ES6+)** – Implements interactive features and handles client-side logic. Asynchronous features like Promises and `async/await` improve data handling for a smooth user experience.

Backend Technologies

The backend of **FoundIt** is built to handle authentication, data management, and secure communication efficiently.

- **Node.js** – A lightweight and scalable JavaScript runtime for handling asynchronous server-side operations efficiently.

- **Express.js** – A fast and flexible Node.js framework used for creating APIs, handling authentication, and managing routing.

- **MongoDB** – A NoSQL database that efficiently stores and retrieves structured and unstructured data such as user details, lost/found item reports, and images. Its flexible schema design enables easy scalability.

- **Mongoose** – An ODM (Object Data Modelling) library for MongoDB that simplifies database interactions, enforces schema validation, and ensures consistency in stored data.

- **JSON Web Tokens (JWT)** – Used for secure authentication, ensuring that only verified users can report, claim, or manage lost/found items.

Hardware Requirements

To ensure smooth operation and scalability, the following hardware configurations are recommended:

For Users (Client-Side)

- Minimum 4 GB RAM (Recommended: 8 GB or higher)
- Dual-Core Processor (Recommended: i5 or higher)
- Stable internet connection
- Modern web browser (Chrome, Firefox, Edge)

For Servers (Hosting the Platform)

- **Development Server:**
  - Minimum: 8 GB RAM, Quad-Core CPU, 50 GB SSD storage
  - Recommended: 16 GB RAM, 8-Core CPU, 100 GB SSD storage

- **Production Server (Cloud-Based, e.g., AWS, DigitalOcean, or Heroku):**
  - Load-balanced architecture with autoscaling
  - 32 GB+ RAM for high traffic handling
  - Database cluster for secure and scalable data storage

3. Proposed Design/Methodology

3.1 IDEs/Editors

- **Visual Studio Code (VS Code)** – A lightweight yet powerful IDE, widely used for JavaScript, React.js, and Node.js development. It provides essential extensions like Prettier for code formatting, ESLint for linting, and GitLens for version control integration.
- **IntelliJ IDEA (Optional)** – A feature-rich IDE that can be used for both frontend and backend development. It supports JavaScript, React.js, Node.js, and database management with integrated debugging and testing tools.

3.2 Version Control

- **Git** – Used for version control, allowing developers to track changes, manage branches, and collaborate effectively.
- **GitHub/GitLab** – Hosting platforms for the repository, enabling collaboration through features like issue tracking, pull requests, and continuous integration (CI/CD) for automated deployment.

3.3 Package Managers

- **npm (Node Package Manager)** – Manages JavaScript dependencies for both frontend (React.js, Tailwind CSS) and backend (Express.js, Mongoose).

3.4 Frameworks

- **React.js (Frontend)** – A component-based JavaScript library for building interactive and responsive user interfaces.
- **Express.js (Backend)** – A minimal and flexible web framework for Node.js, used to build RESTful APIs and manage HTTP requests efficiently.
-

3.5 Requirements

Server Requirements

1. Operating System

- **Linux (Ubuntu)** – Preferred for its stability, security, and scalability in production environments.
- **Windows Server** – Alternative for Windows-based development, suitable for enterprises needing Windows-specific services.

2. Web Server

- **Nginx** – Used as a reverse proxy, handling static files, SSL termination, and load balancing for improved performance.

3. Application Server

- **Node.js** – Powers the backend services of **Founit**, handling API requests, database interactions, and authentication in a fast, event-driven manner.

4. Database

- **MongoDB** – A NoSQL database used to store lost/found reports, user details, and transaction logs. Its flexible schema makes it ideal for scalable applications.
- **Mongoose** – An ODM (Object Data Modeling) library used to enforce schema validation and simplify database interactions.

5. Cloud Infrastructure

- **Firebase** – Used for handling real-time features like notifications, chat systems, and OTP-based authentication. Firebase ensures instant updates for lost/found item reports.
- **Amazon S3 (Optional)** – Used for secure image storage, enabling users to upload and retrieve images of lost and found items efficiently.

Client Requirements

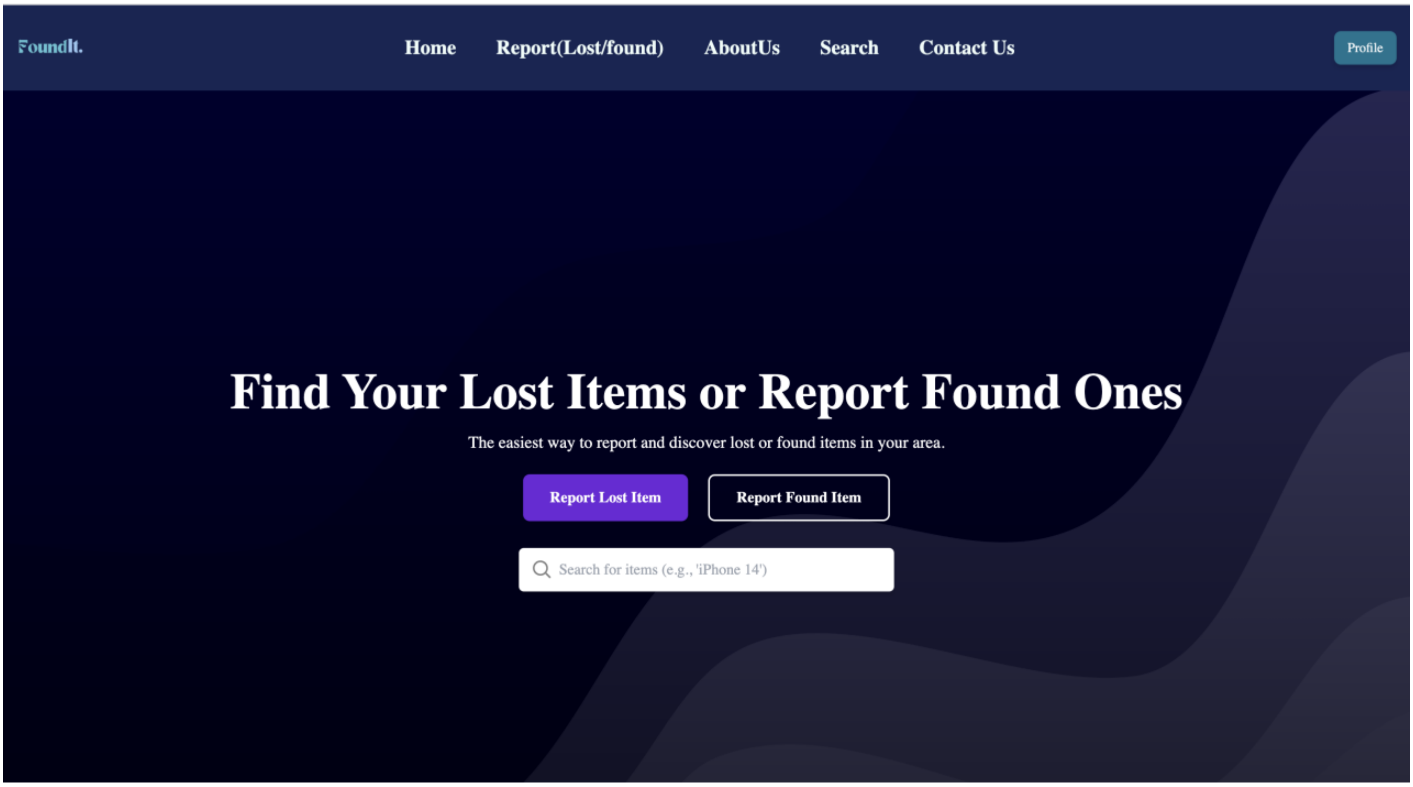
1.Internet Connection

- A stable and high-speed internet connection (e.g., broadband, 4G, or 5G) is required for accessing the platform smoothly. A reliable connection ensures that the platform's interactive features, such as property searches and messaging, work efficiently.

2. Screen Resolution

- The platform is designed to be **responsive** and works seamlessly across various devices with different screen resolutions. Whether on **desktop**, **tablet**, or **mobile**, the design will automatically adapt to provide an optimal user experience.

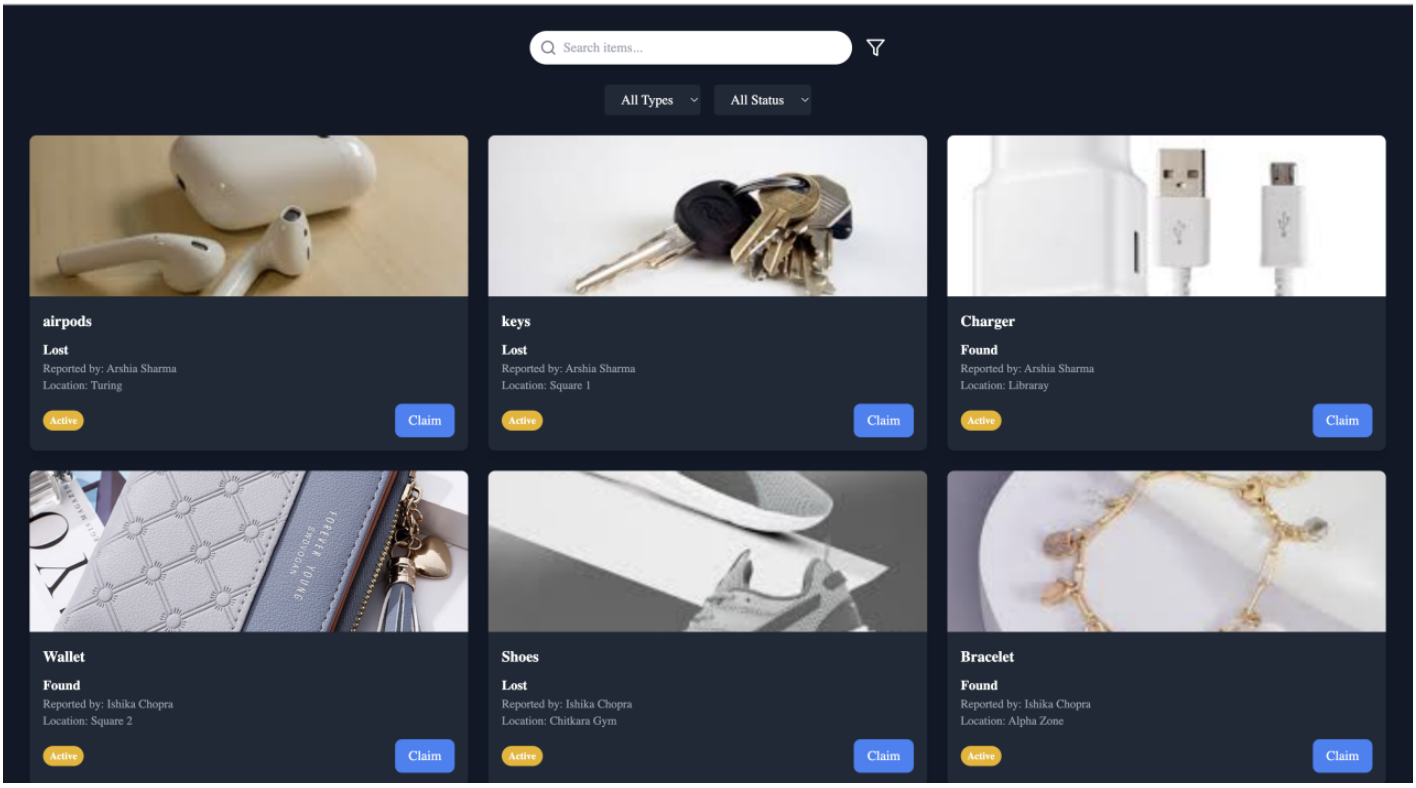
4.Result



Find Your Lost Items or Report Found Ones

The easiest way to report and discover lost or found items in your area.

Landing page



airpods
Lost
Reported by: Arshia Sharma
Location: Turing

keys
Lost
Reported by: Arshia Sharma
Location: Square 1

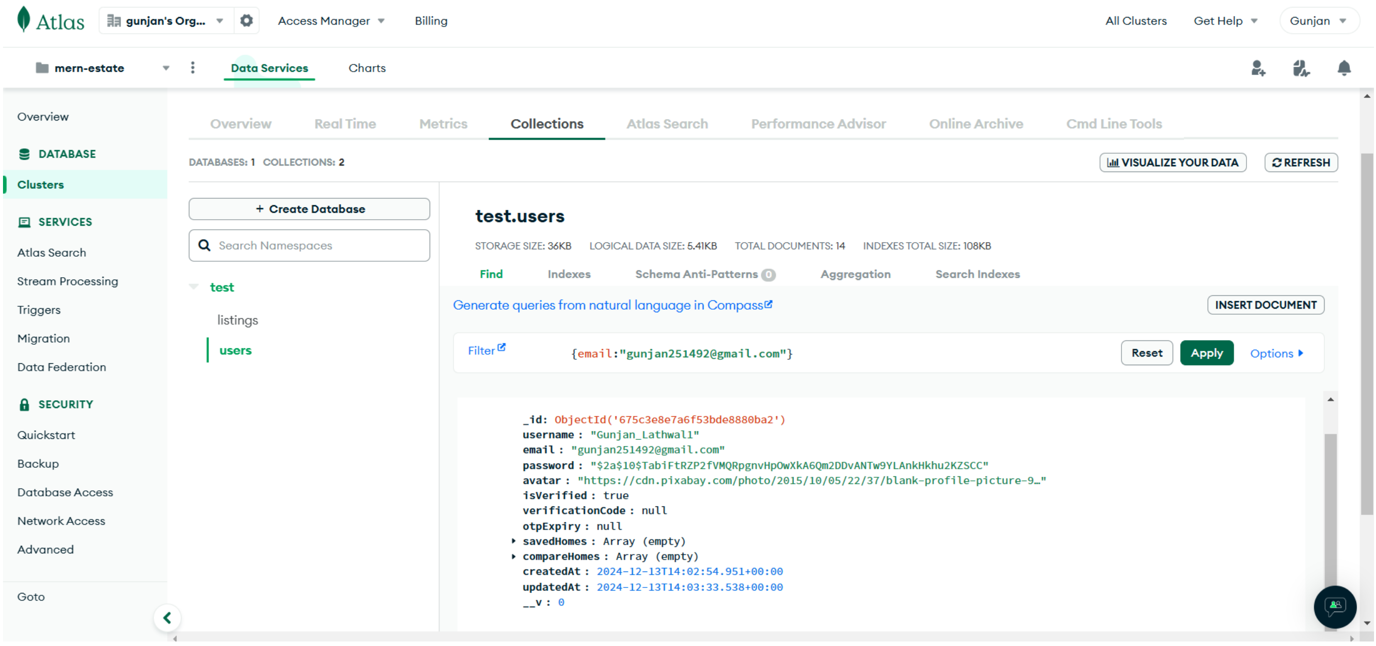
Charger
Found
Reported by: Arshia Sharma
Location: Libraray

Wallet
Found
Reported by: Ishika Chopra
Location: Square 2

Shoes
Lost
Reported by: Ishika Chopra
Location: Chitkara Gym

Bracelet
Found
Reported by: Ishika Chopra
Location: Alpha Zone

All Listings



DATABASES: 1 COLLECTIONS: 2

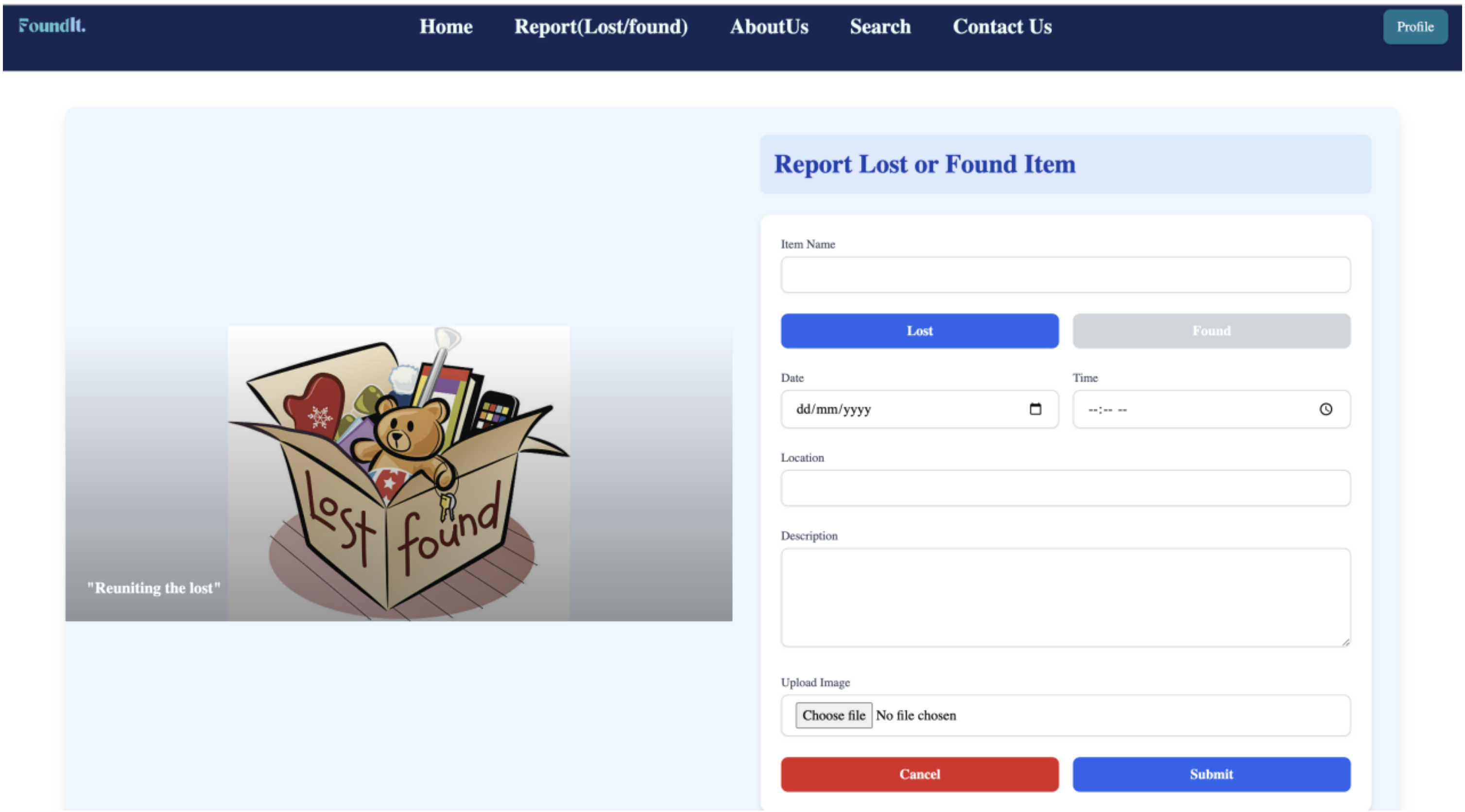
test.users

STORAGE SIZE: 36KB LOGICAL DATA SIZE: 5.41KB TOTAL DOCUMENTS: 14 INDEXES TOTAL SIZE: 108KB

Filter {email:"gunjan251492@gmail.com"}

```
_id: ObjectId('675c3e8e7a6f53bde8880ba2')
username : "Gunjan_Lathwal1"
email : "gunjan251492@gmail.com"
password : "$2a$10$TabiFtRZP2fVMQRpgnvHpOwXkA6Qm2DDvANTw9YLAnkHkhu2KZSCC"
avatar : "https://cdn.pixabay.com/photo/2015/10/05/22/37/blank-profile-picture-9…"
isVerified : true
verificationCode : null
otpExpiry : null
savedHomes : Array (empty)
compareHomes : Array (empty)
createdAt : 2024-12-13T14:02:54.951+00:00
updatedAt : 2024-12-13T14:03:33.538+00:00
__v : 0
```

MongoDB database



Report Lost or Found Item

Item Name

Date dd/mm/yyyy

Time --:-- --

Location

Description

Upload Image Choose file No file chosen

Create listings

Full Stack Engineering,22CS037

5.References

Books:

"Eloquent.JavaScript: A Modern Introduction to Programming"

Author: Marijn Haverbeke
Eloquent JavaScript
A comprehensive guide to modern JavaScript, covering essential concepts and advanced techniques.

"JavaScript: The Good Parts"

Author: Douglas Crockford
JavaScript: The Good Parts
A focused book on the best features of JavaScript and how to use them effectively.

Online Resources and Tutorials:

1.MDN Web Docs-
Comprehensive documentation and tutorials for HTML, CSS, JavaScript, and web development best practices.

2.W3Schools-
Tutorials and references on web technologies like HTML, CSS, JavaScript, and SQL.

3.freeCodeCamp-
Interactive coding lessons and projects on web development, including HTML, CSS, JavaScript, React, and more.

Tools and Frameworks Documentation:

1.React.js Documentation-
Official documentation for React, including guides, API references, and best practices.