PROJECT REPORT SYNOPSIS

ON

FoundIt

SUBMITTED TO

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

FOR

Full Stack Engineering(22CS037)

Submitted By:

Name(s): Ishika Chopra, Arshia Sharma, Mandeep Kaur

University Roll No(s).: 2210991689, 2210991368, 2210991879

Semester:6th

Session: 2022-2026

CHITKARA UNIVERSITY

Index

Problem Statement:

The goal is to develop an efficient lost and found platform, "FoundIt" , where users can report lost or found items, and administrators can manage these reports. Users can provide details such as location, time, description, and images of the lost or found items. The platform will help individuals recover lost items by offering a searchable, filterable database and marking issues as active or resolved.

Title of project:

FoundIt

Objective:

- Build a lost and found reporting system: Allow users to report lost or found items with detailed descriptions, images, and location details.
- Implement a user-friendly interface: Provide an intuitive platform for users to browse, search, and filter items based on criteria such as category, date, and status (active/resolved).
- Create an efficient management tool: Enable administrators to review, approve, and manage reports while ensuring the authenticity of submissions.

Key Learning's:

1. User Management:

- Secure authentication and authorization for users and administrators.
- Role-based access control to allow users to report items and admins to manage reports.

2. Item Reporting and Tracking:

- APIs for handling lost and found item submissions, updates, and resolutions.
- Real-time status tracking for reported items (active or resolved).

3. Communication and Notifications:

- APIs for sending notifications, alerts, and updates regarding item status.
- Tools for users to contact finders or rightful owners securely.

Options available to execute the project:

- Programming Languages:
  - React.js: A front-end JavaScript library for building interactive user interfaces.
  - Node.js: A JavaScript runtime enabling scalable backend operations.

- Backend Frameworks:
  - Express.js (Node.js): A minimal web application framework used for creating the API and routing.
  - Mongoose: An ODM (Object Data Modeling) tool for MongoDB to manage database schemas.

- APIs:
  - Used standard RESTful APIs for client-server communication.

- Version Control:
  - GitHub: A version control system for collaboration, code review, and project management.

- Database:
  - MongoDB: A NoSQL database known for its high performance and scalability, ideal for managing real estate listings and user data.

Advantages/ Disadvantages:

- **Advantages:**

- Efficient Lost & Found Management:

  - Users can quickly report and search for lost or found items, increasing the chances of recovery.
  - Administrators can manage and verify reports efficiently.

- Scalability and Flexibility:

  - MongoDB ensures the platform can handle large datasets and user traffic efficiently.
  - APIs allow easy integration of new features such as automated matching of lost and found items.

- User Engagement:
    - Interactive features like search filters and status updates enhance user experience.
    - Notifications keep users informed about their reported items.

- **Disadvantages:**

- Technical Complexity:
    - Implementing real-time updates for lost and found reports requires regular maintenance and debugging.
    - API integration for notifications and item tracking increases backend complexity.

- Security Concerns:
    - Protecting user data, including contact details and item descriptions, requires robust security measures.
    - Secure authentication methods must be implemented to prevent unauthorized access.

- Verification Challenges:
    - Ensuring the authenticity of lost and found reports can be difficult without proper verification mechanisms.
    - Users might misuse the platform, requiring moderation efforts.

CHITKARA UNIVERSITY

References:

1. Node.js. (n.d.). Introduction to Node.js. Retrieved from https://nodejs.org/en/learn/getting-started/introduction-to-nodejs
2. React. (n.d.). Getting Started. Retrieved from https://reactjs.org/docs/getting-started.html
3. Mongoose. (n.d.). Mongoose v6.1.0: Documentation. Retrieved from https://mongoosejs.com/docs/
4. Express. (n.d.). Installing. Retrieved from https://expressjs.com/en/starter/installing.html
5. MongoDB. (n.d.). MongoDB Atlas. Retrieved from https://www.mongodb.com/cloud/atlas
6. Mozilla. (n.d.). JavaScript. Retrieved from https://developer.mozilla.org/en-US/docs/Web/JavaScript
7. npm. (n.d.). npm. Retrieved from https://www.npmjs.com/