CHENNAI DESK
A PUBLIC GRIEVANCE APPLICATION

IT5613 SOCIALLY RELEVANT PROJECT LABORATORY

A PROJECT REPORT

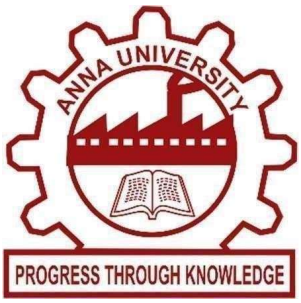
Submitted by

SHRISH R (2020506088)

THARUN C D (2020506103)

HEMANTH KUMAR V (2020506033)

SURENDRA N (2020506099)



DEPARTMENT OF INFORMATION TECHNOLOGY

MADRAS INSTITUTE OF TECHNOLOGY CAMPUS

ANNA UNIVERSITY, CHENNAI-600 044.

FEBRUARY 2023 - JUNE 2023

BONAFIDE CERTIFICATE

Certified that this project report titled **"Chennai Desk – A Public Grievance Application"** is the bonafide work of Shrish R (2020506033), Tharun C D (2020506103) and Hemanth Kumar V (2020506033), Surendra N (2020506099) who carried out the project work under my supervision.

Signature

Dr. Sumalatha

HEAD OF THE DEPARTMENT

Professor

Department of Information Technology

MIT Campus, Anna University

Chennai – 600044

Signature

Dr. Sumalatha

SUPERVISOR

Professor

Department of Information Technology

MIT Campus, Anna University

Chennai – 600044

ACKNOWLEDGEMENT

It is essential to mention the names of the people, whose guidance and encouragementmade us accomplish this project.

We express our thankfulness to our project supervisor **Dr. M. R. Sumalatha**, Professor,MIT Campus, for providing project guidance and encouragement to do the project.

We express our gratitude and sincere thanks to our respected Dean of MIT Campus, **Dr. J Prakash**, for providing computing facilities

Shrish R	**2020506088**
Tharun C D	**2020506103**
Hemanth Kumar V	**2020506033**
Surendra N	**2020506099**

CHAPTER 1
INTRODUCTION

1.1 OVERVIEW

In their day-to-day lives, people often encounter a range of public issues that can impact their well-being and daily routines. These issues may include inadequate public transportation, congested roads, water-related problems, insufficient parking facilities, pollution, limited access to healthcare or education, inadequate waste management, and unreliable utility services. These challenges can cause inconvenience, stress, and hinder productivity for individuals. Moreover, public issues can have broader societal implications, affecting the overall quality of life within a community. To address these concerns, it is crucial for individuals to voice their grievances, engage with relevant authorities, and work collaboratively to find sustainable solutions. Public Grievances play a crucial role in a democratic society and have several important implications such as citizen empowerment, improved public services and responsible governance. Chennai, the state's primate city both in area and population needs a public grievance redressal system where the complaints are addressed quickly and efficiently. Moreover, in a highly populated city like Chennai public grievances tend to be filed multiple times and this leads to a high wastage of system and human resources. The proposed system suggests an user-friendly mobile application where the users can provide detailed information about the nature of their grievance, attach supporting documents, and track the progress of their complaint. The user is able to view previously registered complaints, interact with the registered complaints and check its status as well. The user is also able to identify public complaints that have been registered nearby and refrain from making repeated complaints regarding the same issue. In the following section existing works and similar applications have been discussed.

1.2 OBJECTIVE

The objective of this project is to develop an interactive, user-friendly mobile application through which users are able to register their public grievances. The users need to be able to view the complaint and track its status discreetly. The secondary objective of this project is to allow users to view the complaints already registered, interact with the complaints and view it as well. A proximity based and a interaction based complaint retrieval is desired to be constructed. The project aims to implement an easy to store and retrieve data storage, which is highly scalable and replicable. Continued user interaction, conservation of system and human resources, public empowerment and quick response addressal are the end goals that the project looks to accomplish.

1.3 SCOPE OF THE PROJECT

The proposed system prioritizes conserving system and human resources by reducing repeated complaint grievance registration. Admin is able to monitor data retrieval, backend processing and data storage. A scalable database is also implemented ensuring safety and availability. The system enables easy user registration and validated login into the application by enforcing request-reply protocols to the backend. The system works on the whole of Chennai Corporation enabling real-time area tracking and location categorization. Different types of grievances have been identified and structured allowing users to freely register complaints without any restrictions. The system allows consistency, availability and scalability, which enables the system to be expanded to a wider area and implement more features in the future.

CHAPTER 2
LITERATURE SURVEY

In [1], the Greater Chennai Corporation authorities responsible for governing the city of Chennai, have implemented a public grievance registration system to address and resolve citizen complaints effectively. The system aims to provide a platform for citizens to register their grievances related to various public services and infrastructure issues. It ensures accessibility for a diverse range of users with different preferences and technological capabilities. Once a grievance is registered, it is typically categorized based on the nature of the issue and routed to the relevant department or authority for investigation and resolution. The system employs a tracking mechanism that assigns a unique identifier to each complaint, allowing citizens to monitor the progress of their individual grievances. Simple database retrieval is performed enabling easy routing of grievances to departments but local public is not made aware of the complaint made as the user who registered it, is the only person able to access it.

[2] is a popular social media platform and online community where users can engage in discussions, share content, and participate in various communities known as "subreddits." It allows individuals to post text, links, images, and videos and interact with others through comments, upvotes, and downvotes. Reddit covers a wide range of topics, including news, entertainment, technology, science, sports, and more. Users can subscribe to specific subreddits based on their interests to customize their Reddit experience. The platform's voting system determines the visibility of posts and comments, with popular content rising to the top. Reddit encourages anonymity, as users can create accounts without revealing personal information. It fosters vibrant discussions, provides a platform for sharing knowledge and experiences, and serves as a hub for communities to connect and exchange ideas.

[3] is a geocoding system that aims to simplify location referencing and improve address accuracy worldwide. Traditional addressing systems can be complex and prone to errors, especially in areas with inadequate infrastructure or ambiguous addressing schemes. what3words divides the world into a grid of 3x3 meter squares and assigns a unique combination of three words to each square. This means that every precise location on Earth has a specific three-word address. The system utilizes a combination of words from a curated dictionary to create easily memorable and communicable addresses. It provides a user-friendly interface and supports multiple languages, making it accessible to a global audience. The three-word addresses can be used in various scenarios, such as navigation, emergency services, package delivery, or sharing specific locations with friends and family. what3words has gained traction in diverse sectors, including travel, logistics, and humanitarian efforts. It offers a reliable and user-friendly alternative to traditional address systems, especially in areas where accurate addressing is challenging. By simplifying and standardizing location referencing, what3words aims to improve efficiency, reduce errors, and enhance accessibility to precise locations worldwide.

[4] is a popular social media platform known for its focus on visual content. Users can share photos and videos on their profiles, which are referred to as "posts." These posts allow individuals to express themselves, showcase their creativity, and engage with their followers. Instagram posts can be edited with filters, captions, and location tags to enhance their visual appeal and provide context. Users can also tag other accounts in their posts and add hashtags to increase discoverability. Additionally, Instagram offers various interactive features such as likes, comments, and shares, allowing users to engage with posts and connect with others. Posts on Instagram are displayed in users' feeds, where they can scroll through and view content from accounts they follow. The platform's algorithm determines the order in which posts appear in a user's feed, taking into account factors such as relevancy, engagement, and recency.

CHAPTER 3
SYSTEM ARCHITECTURE AND DESIGN

3.1 ARCHITECTURE

The system architecture comprises of two main parts, the server side and the client side. The client-side interface prompts the user with two choices account registration and log in option. Upon validation of the user account the user is directed to the dashboard where complaint retrieval is performed and the user is enabled to view and interact with the complaints displayed. The server side consists of database querying and API calls. The admin plays a major role overseeing the retrieval of data and facilitating the requests.

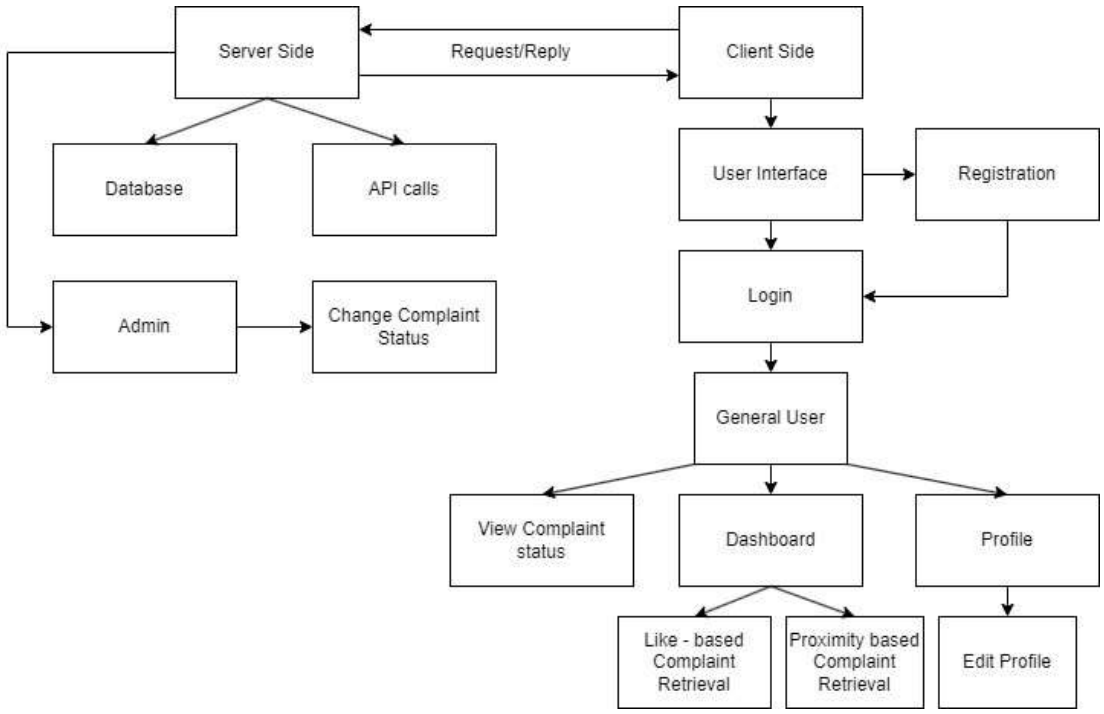


Fig 3.1 – System Architecture

3.2 USER - INTERFACE

The user interface is inaugurated with the registration module which collects user details and creates a user account by sending requests to the server and adding the respective user into the database. The login module accepts the email and password as input, validates it and gives the user access to their dashboard. The dashboard module retrieves public complaints registered by all users by sending a request to the server. The server retrieves the data in two ways. One based on the number of likes each complaint has acquired. The other way by making distance calculations from the user's location to every complaint and sends a payload of data where the complaints are structured based on closest proximity. The user is enabled to view the status each complaint displayed in their dashboard and like/unlike it as well. The profile module displays all the user details, and allows editing of those details using the edit profile module.

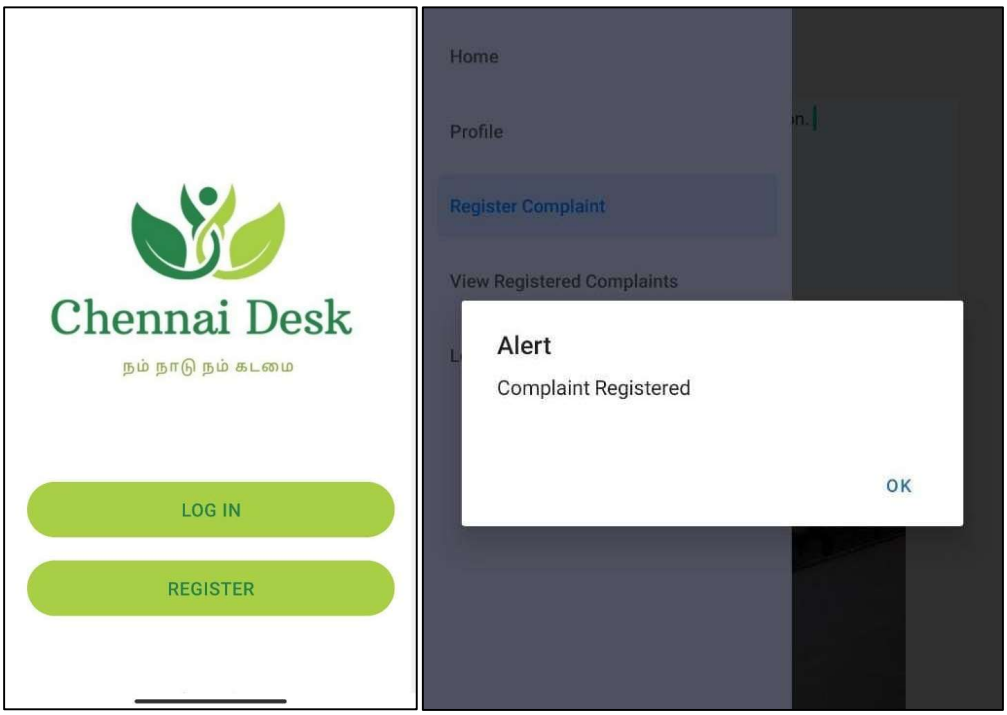


Fig 3.2 User-Interface

3.3 ADMIN

The admin oversees the working of the server and ensures that the connection to the database is open. The admin also has to make sure that the API calls are happening accurately. The admin also has the option to change status of a particular complaint. When a complaint is first registered, the status of the complaint is marked as 'open'. When the complaint is assigned to an authority, the admin can change the status of the complaint as 'In progress'. Once the complaint has been completely addressed and the issue has been resolved, the admin changes the status of the complaint to 'closed'. The admin also is able to debug the application for future errors as well.

3.4 GEOCODING

For every user and complaint registration geocoding of their location is performed. Geocoding is the process of converting a location given in text into its respective latitude and longitude. The geocoding process in the proposed system is performed using the Nominatim API. It uses OpenStreetMap data to process the queries gives and return its respective coordinates. In the system, when a user registers their account, their location data is obtained and further geocoded and stored into the MongoDB database. Similar processing is done for each complaint registered as well. When proximity based retrieval is performed, the distance between the user location and each of the registered complaint locations is calculated. The Haversine distance which is the angular distance between any two points on the surface of a sphere, is calculated between the user and the complaints. Then the complaints are returned in sorted order back to the client-side application.

CHAPTER 4
ABOUT OUR PROJECT

4.1 FReMP BASED APPROACH

The FReMP stack is a highly scalable full stack framework, which can be used to build robust and modern web apps using Flask, ReactJS, MongoDB and Python. Unlike other famous stacks such as MEAN and MERN, the FReMP stack uses Python to handle back-end operations. The proposed system uses the FReMP stack to implement its methodology. React Native has been used to contruct a mobile application instead of ReactJS which is used for web development.

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools. It is a very small and well documented framework, which uses Werkzeug for routing, debugging and WSGI support and Jinja2 for templating. Advantages of Flask include lightweight and easy-to-understand framework, and a built-in development server. The proposed system has implemented the flask framework for database retrieval, API calls and processing data as well. It is highly useful in implementing the proximity-based and like-based data retrieval. It also enables faster processing and availability for the proposed application.

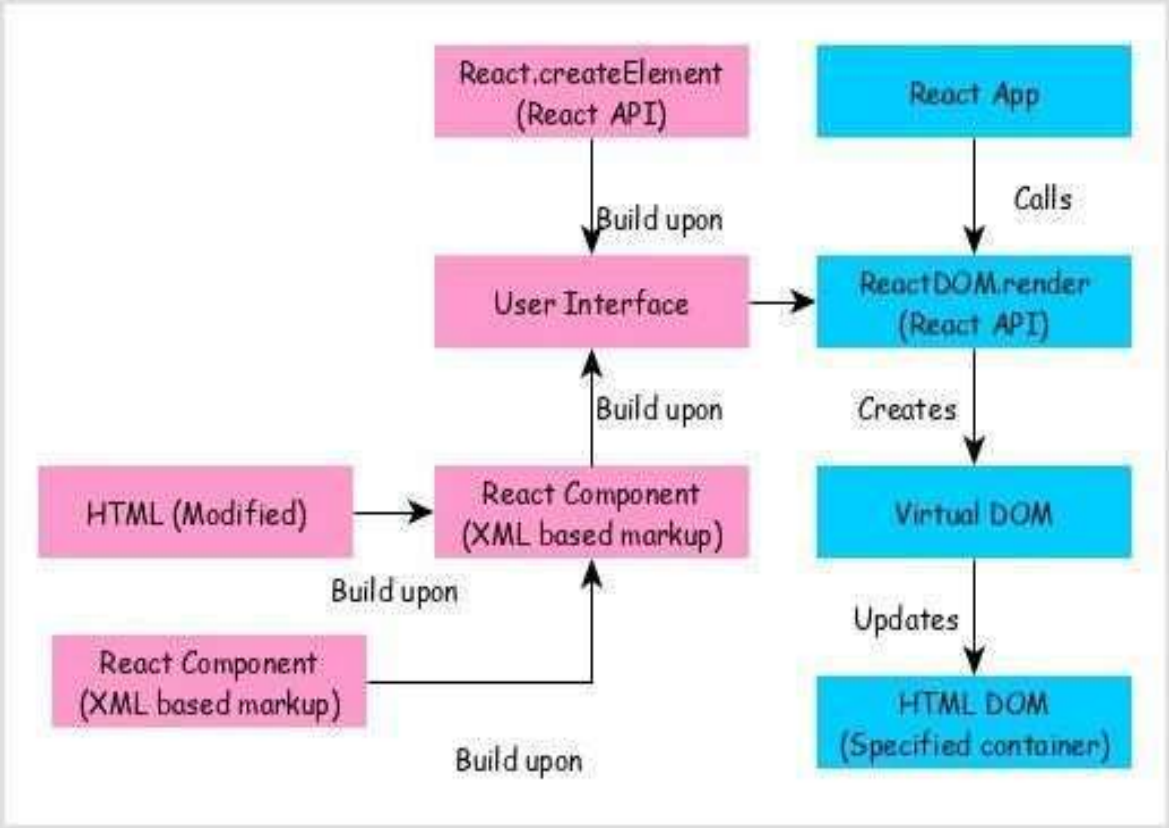


Fig 4.1: Architecture of React JS

Thirdly, the JavaScript library React.js is widely used for creating user interfaces with the help of the material UI for styling. It makes it possible to design UI elements that are adaptable, responsive, and interactive. The front end of an application that provides performance reviews might be built with React.js. Employee profiles, Performance indicators, performance reviews, and interactive graphs and charts can all be built as separate UI pieces. React.js's virtual DOM rendering and streamlined state management contribute to the framework's overall speed boost. React Native is a JavaScript framework for building native mobile apps. It uses the React framework and offers large amount of inbuilt components and APIs. The system uses the React Native framework to build a interactive user-interface which enables the user to register public grievances easily.

MongoDB, a NoSQL database that uses the versatile document-like JSON (JavaScript Object Notation) format to store data. An E-complaint system stores user data like user details and user location details in a database. Complaint details like complaint type, description, complaint address details and proof details like images needs to be stored as well. Because of its flexible and dynamic schema, MongoDB is well-suited for use in dynamic applications. SQL injection is not possible in MongoDB because MongoDB does not use Structured Query Language (SQL) for querying and manipulating data. Unlike traditional relational databases that rely on SQL, MongoDB is a NoSQL database that uses its own query language called the MongoDB Query Language (MQL).

SQL injection is a vulnerability that occurs when untrusted user input is not properly sanitized and is directly concatenated into SQL queries. Attackers can exploit this vulnerability by injecting malicious SQL code, which can lead to unauthorized access, data manipulation, or even data loss. In MongoDB, data is stored in a JSON-like document format known as BSON (Binary JSON). Queries in MongoDB are performed using MQL, which is designed to work directly with these documents and does not involve the use of SQL syntax. MQL provides its own set of operators and methods for querying and manipulating data, eliminating the risk of SQL injection vulnerabilities. MongoDB is often described as a "schemaless" or "schema-flexible" database, which means that it does not enforce a strict schema or structure for the data stored within it. This is in contrast to traditional relational databases where a fixedschema must be defined upfront.

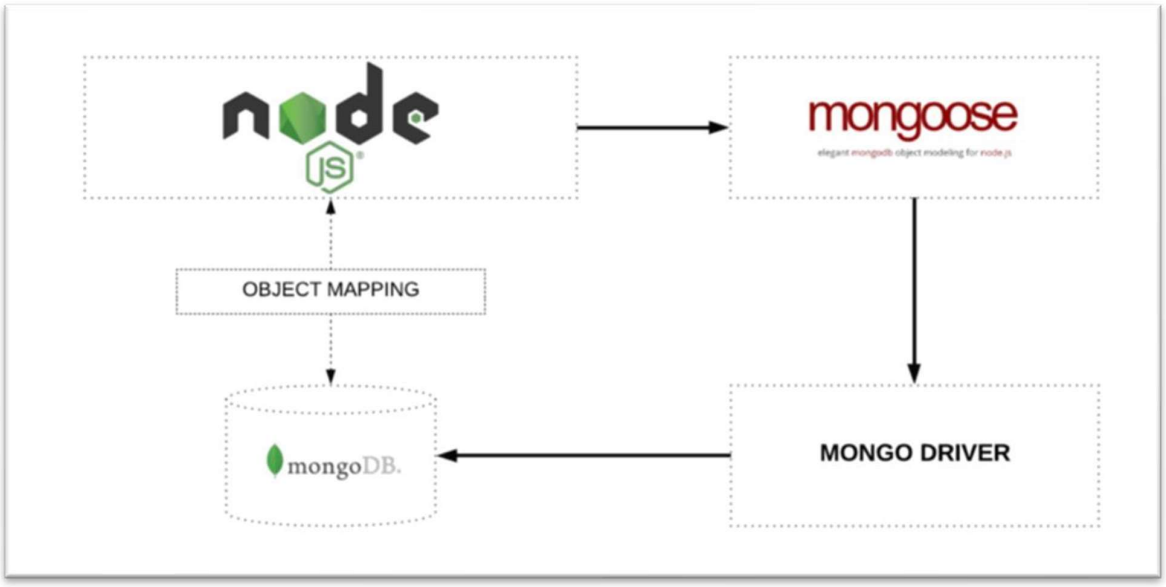


Fig 4.2 : Mapping of data into the database

Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation via the off-side rule. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library. NumPy is a fundamental library for scientific computing in Python. It provides a multidimensional array object called ndarray, which is efficient for storing and manipulating large datasets. Pandas is a high-level data manipulation library built on top of NumPy. It provides data structures and functions designed for working with structured or tabular data. pandas is widely used for tasks such as data cleaning, exploration, analysis, and preparation. In the proposed system, pandas has been used to process the data after obtaining it from database to find the distance between locations. Numpy library has been used to convert location coordinates into their respective radian values.

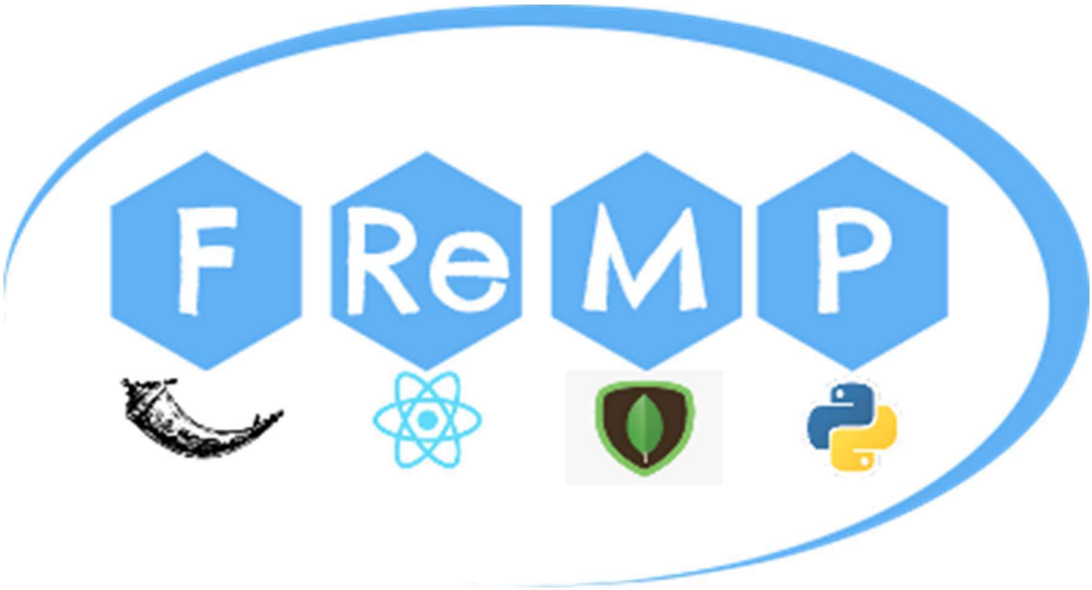


Fig 4.3 : Working of FReMP

Node.js is a server-side JavaScript runtime that facilitates the execution of JavaScript programs. It's ideal for developing scalable, high-performance web apps due to its event-driven, non-blocking I/O style. Node.js can be utilized as the server-side runtime to power the Flask backend of a Public Grievance Application. You may process data, manage numerous connections, and interact with the MongoDB database all at once.

Overall, the FReMP stack combines the strengths of FLASK, React, MongoDB, and Python to provide developers with a powerful, efficient, and flexible solution for building modern web applications. It offers high performance, productivity, scalability, and a vibrant ecosystem of tools and resources to support the development process.

4.2 LANDING SCREEN

A new user will be able to register their personal details and address details and create an account to log in to the app. The page will also have the option to log where the user will be prompted for the registered email and password. Upon entering the details, the it will be validated and the database would be queried at the backend. If the query fails, an alert would be sent to the user to recheck their details. Upon success the user would then be directed to the dashboard.

4.3 USER DASHBOARD

The user dashboard consists of 4 screens. The Home Screen displays a list of public complaints which have been registered in the application. The complaints are retrieved in a like-based and a proximity-based algorithm. The user is able to like/unlike each complaint, view the number of like on a complaint and view the complaint separately using the view option. The Profile screen displays the details of the user including their address details. The user is also given the edit profile option, clicking which the user is redirected to another screen where their details could be edited and updated as well. The Register Complaint Screen enables the user to register their public grievance. The user is prompted for details like complaint person's name and email, complaint address details , complaint type, title, description and complaint proof. The anonymous registration option is also provided to the user. The View Registered Complaint Screen allows the user to track all the complaints registered by them. Each Complaint is able to be viewed separately in chronological order. Finally the user is also provided a logout option clicking which they are logged out of the app.

4.4 PROPOSED SYSTEM IMPLEMENTATION

The client-server approach is incorporated into the planned architecture for the system. A graphical representation of the client-server model, coupled with an explanation of its primary capabilities. It is a representation of the specific functions of both the client and the server, as well as the ways in which those functions link with one another. In this context, the candidates function as the client, and Admin acts as the server, controlling the capabilities of the client.

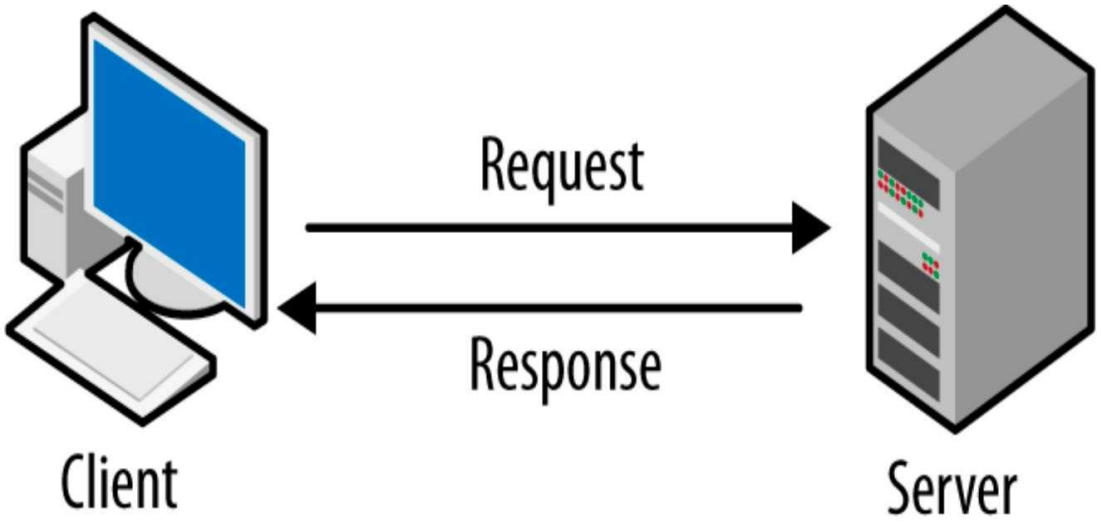


Fig 4.4: Client – Server model

4.4.1 Server

The server has the access to control all the activities, insert new users into the database upon registration and query their credentials upon login. Server has the access to insert, update, and view each complaint and retrieve it whenever needed. Server can check the status of the various users, retrieve their registered complaints. Server also retrieves the complaints and processes it return it based on proximity. The server returns complaints according to the number of likes by querying the mongodb database using the sort function. The server also makes sure to geocode the locations of each person and complaint and store it in the database.

4.4.2 Client

The client sends requests to the server every time a screen is changed. The first request to the server is made when the user register/login into the application. Once the user is validated and logged in two requests are sent to the server to obtain complaint details in a liked-based and proximity-based fashion. Each time the user wants to view a complaint separately, a request is made to the server to obtain its details. Two other screens where requests are made, are the edit profile screen and vie registered complaints screen. Every time the system waits for a request-reply protocol to end, a loading indicator is rendered to the user.

4.5 IMPLEMENTATION ENVIRONMENT

Several parts and technologies make up a performance appraisal site's implementation environment. Here's a rundown of the main points:

Incoming HTTP requests are processed by these servers, which then communicate with the system's backend to retrieve and distribute the data.

The backend of the Public Grievance Application is where all the action happens, it's where requests are processed, where data is stored and retrieved, and where the business logic is actually implemented. The components of the b ackend for a FReMP stack implementation are as follows:

1. Node.js is the server-side runtime environment for running JavaScript.
2. Flask is the backend of the application used for processing and developing web applications using python, implemented on Werkzeug and Jinja2

3. MongoDB is a NoSQL database suitable for storing information about employees, such as names, addresses, and performance indicators.

Thirdly, the frontend is the component of your performance review site that customers see and engage with. The components of the front end of a MERN stack implementation are as follows:

1. HTML and CSS are markup and styling languages used to organize and display content on the World Wide Web.

2. To create user interfaces using JavaScript, there is a library called React.js. It offers reusable parts and efficiently adjusts the user interface in response to actions taken by the user.

JavaScript is used for client-side logic implementation, backend API calls, and user interaction. Integration with other services and systems may require the use of application programming interfaces (APIs) on your performance review website. Application programming interfaces (APIs) facilitate communication between programs. To describe and implement the interaction between your front- and back-end components, you can utilize RESTful APIs or GraphQL which serves as the backend of the MERN project, providing data storage, retrieval, and manipulation functionalities. It handles HTTP requests and communicates with the MongoDB database to perform CRUD operations (Create, Read, Update, Delete) on the data.

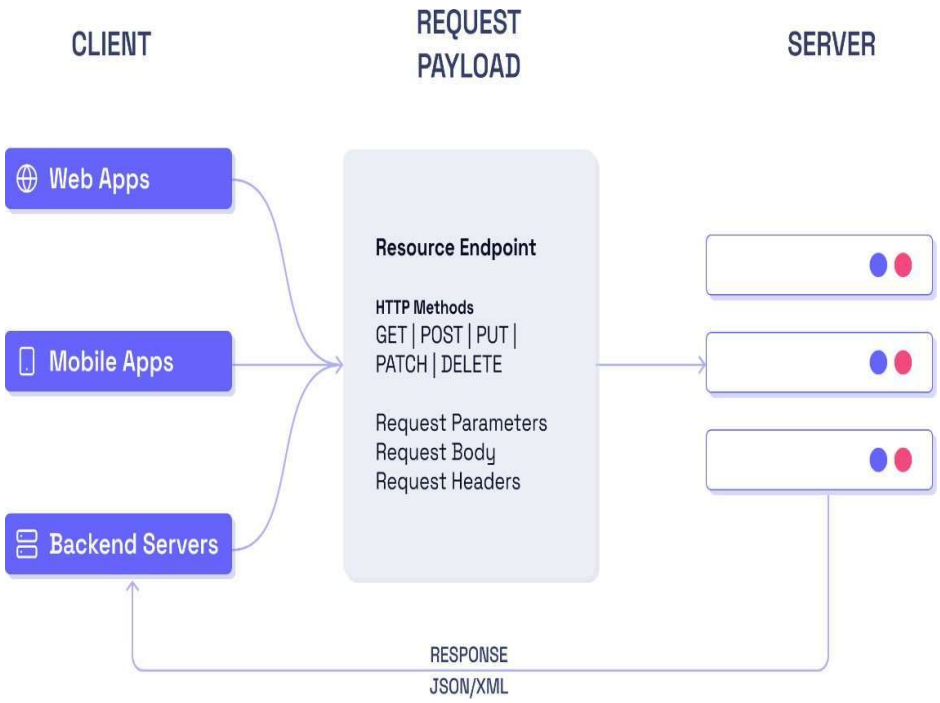


Fig 4.5 : Working of REST API

Constructing and maintaining your performance evaluation website is easier with the help of a wide range of development tools and frameworks. Tools for streamlining development processes include code editors (like Visual Studio Code and Sublime Text), version control systems (like Git) to maintain the consistent and updated information, package managers (like npm and Yarn) depending on the platform used, etc.

It's important to remember that the implementation environment can change depending on the needs, technology, and infrastructure choices of the organization. The aforementioned parts make up the basic structure for a performance evaluation website; however, the precise configuration may vary depending on your specific requirements.

CHAPTER 5

RESULTS AND DISCUSSIONS

5.1 OUTPUT SCREENSHOTS

The landing screen for the user when the application is opened is shown below.

Fig 5.1 : Landing Screen

The dashboard of the user when the application is logged in is shown below. Complaints are displayed in order based on the number of likes.

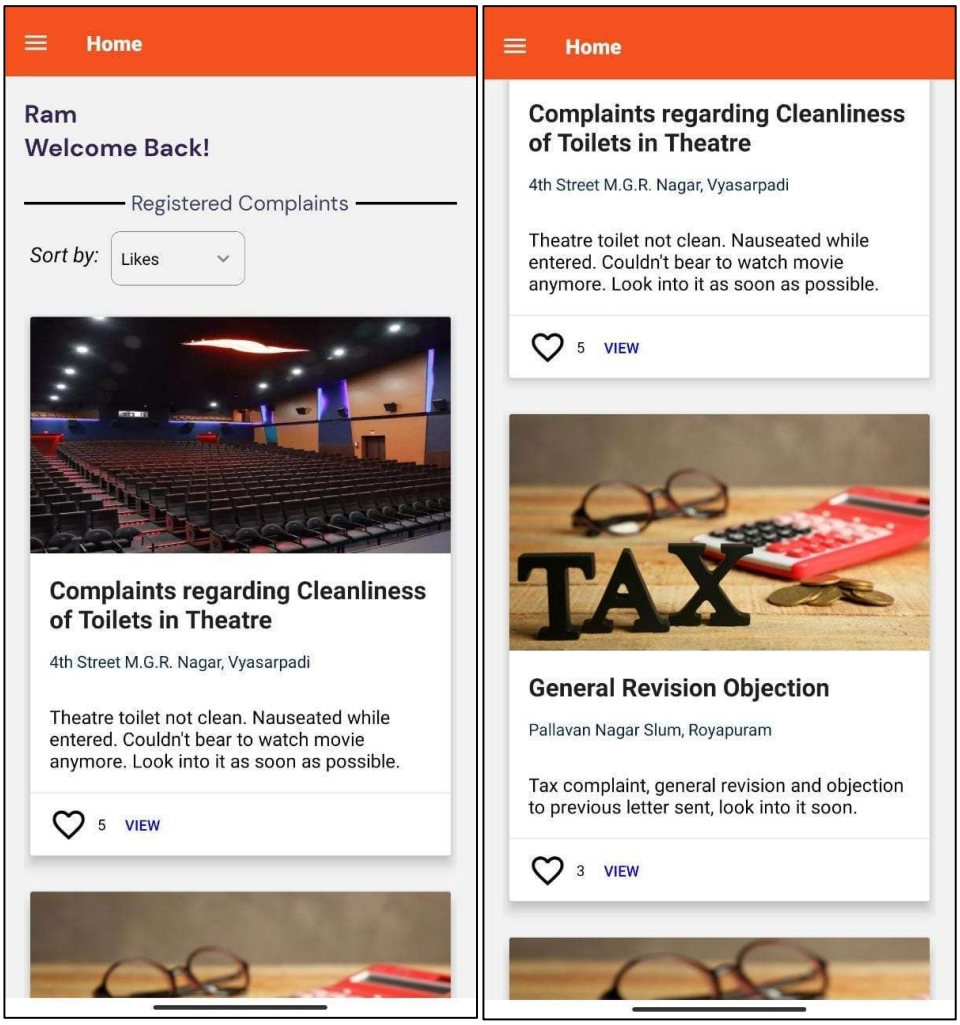


Fig 5.2 : User Dashboard

User is enabled to like complaints and view complete details and description.

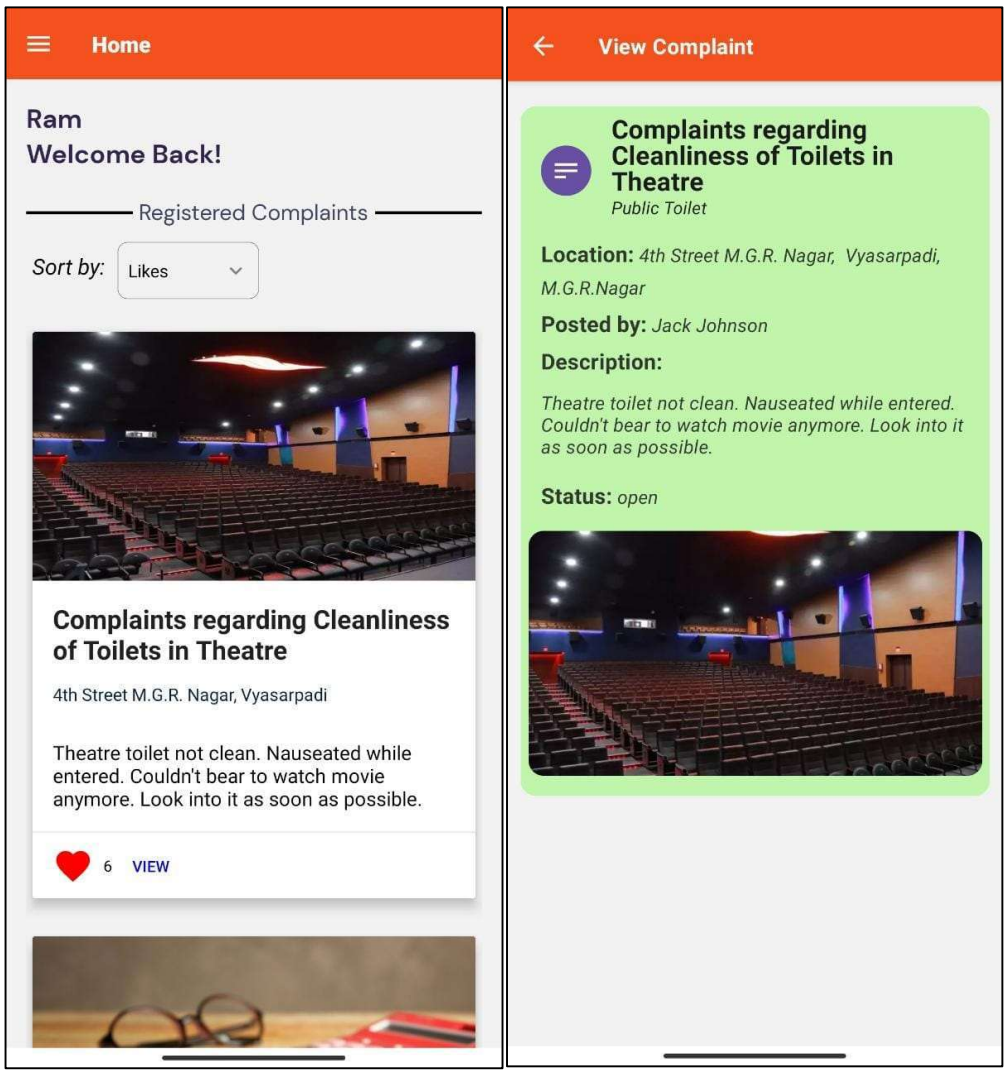


Fig 5.3 : Interaction of registered complaints

Complaints are displayed based on proximity from the user. The current user ‘Ram’ is from Nesapakkam and the closest complaints to him have been displayed first. It can be verified that Nesapakkam is closer to Egmore than Sowcarpet.

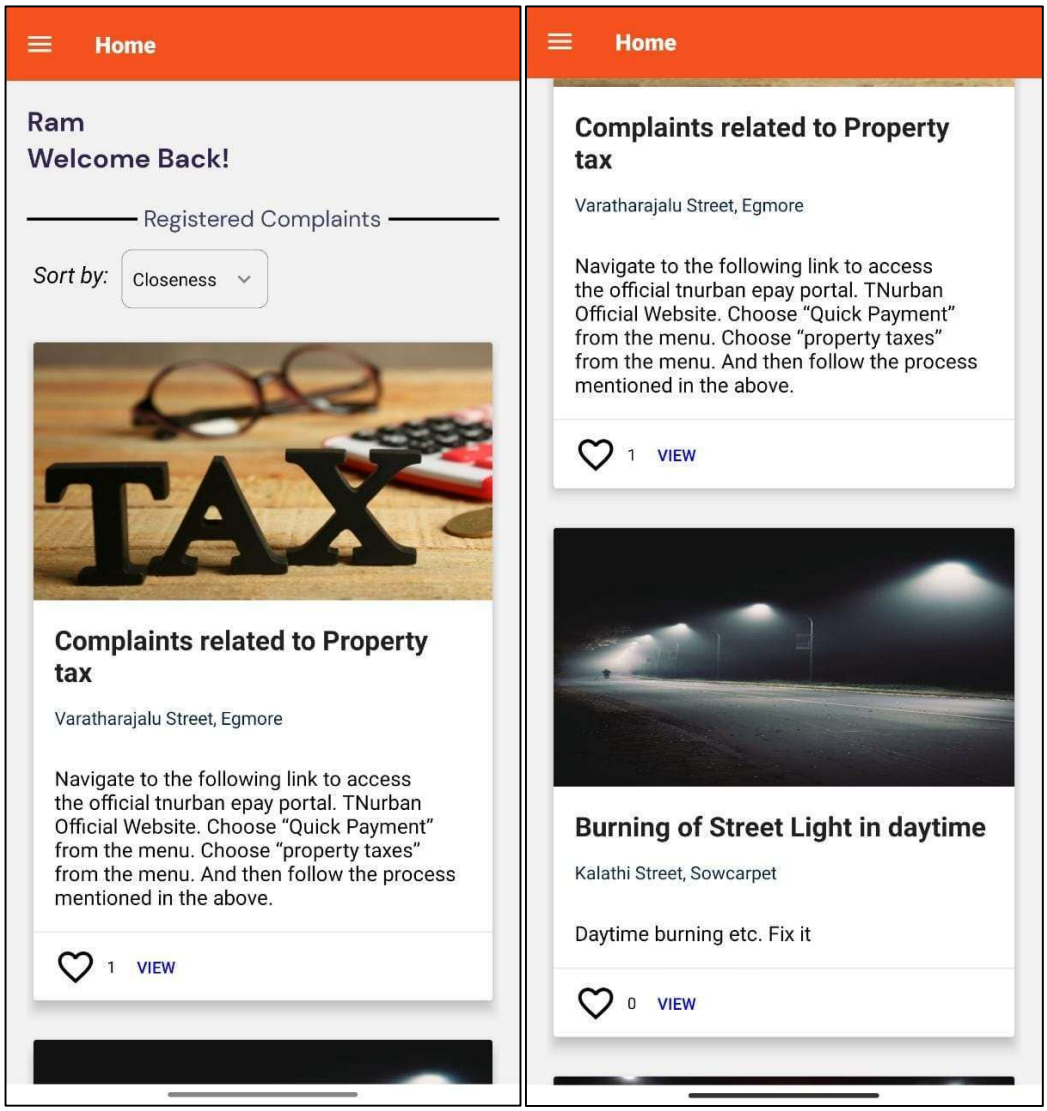


Fig 5.4 : Proximity based retrieval

Complaint Registration Module is shown below.

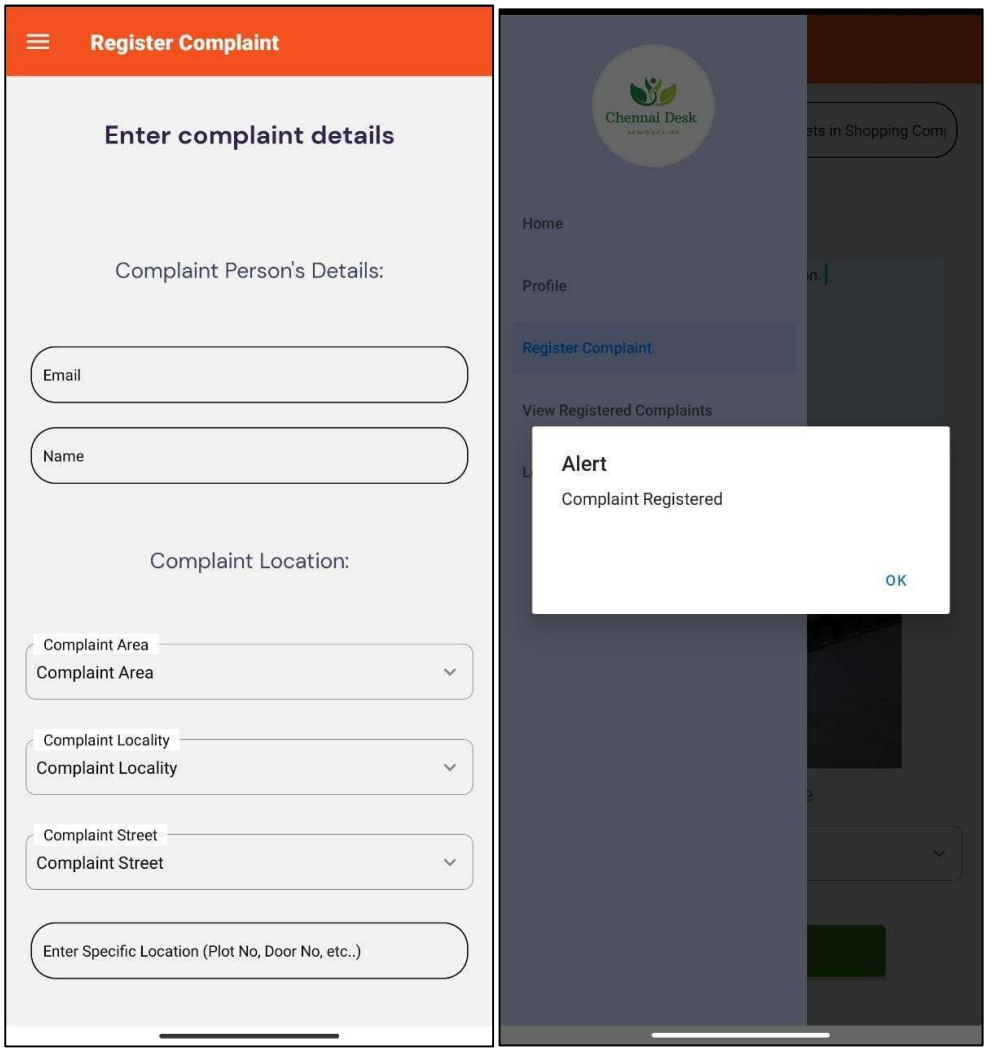


Fig 5.5 : Complaint registration module

The view registered complaints module and viewing a closed complaint is shown below.

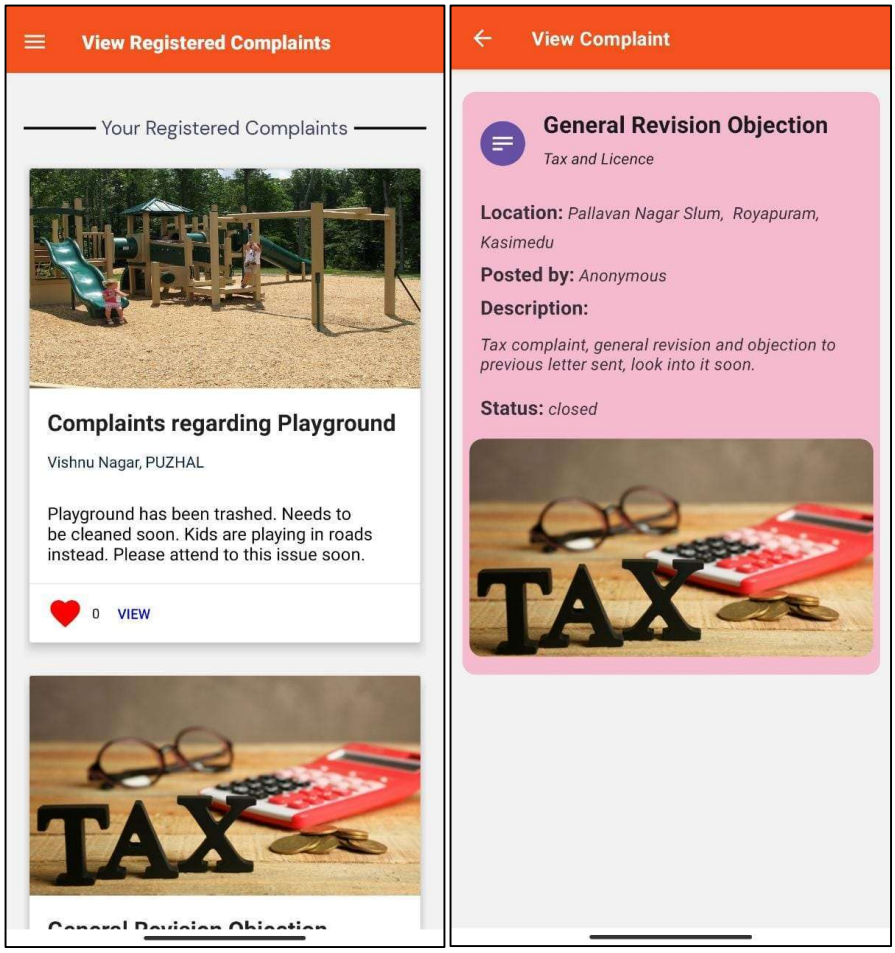


Fig 5.6: View Registered Complaints

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

Addressal of public grievances is a long and tiring task. The proposed system has enabled easier registering of public complaints by providing an easy-to-understand user-interface. The system has also achieved success in retrieving a proximity-based complaint retrieval, so that users will be aware of public grievances registered near to them. A highly scalable and reliable database has been implemented along with a responsive and a consistent backed server. The server takes some time to process the complaints as the database size increases. This can be reduced by consistent replication and deletion of closed complaint along with other practices like indexing.

6.2 FUTURE WORK

The proposed system has been implemented with minimal resources. With high amount of resources, processing speed can be reduced significantly. Moreover with more accurate geocoding API, exact location of users and complaints can be pinpointed as well. The system can also be expanded to a wider area possibly the whole state of Tamil Nadu. The system can be scaled to meet the resource requirement of a larger area of governance.

REFERENCES

[1] Namma Chennai App- https://www.tncitizenconnect.com/nammachennai

[2] Reddit- https://www.techtarget.com/searchcio/definition/Reddit#:~:text=Reddit%20is%20a%20social%20news,use%20the%20website's%20basic%20features.

[3] What3words - https://what3words.com/toddler.geologist.animated

[4] Instagram - https://en.wikipedia.org/wiki/Instagram https://www.instagram.com/

[5] FReMP Tutorial - https://fremp.github.io/

[6] MONGO DB - https://www.mongodb.com/docs/manual/tutorial/getting-started/

[7] React JS - https://react.dev/learn

[8] MongoDB with NodeJS - https://www.mongodb.com/languages/mongodb-with-nodejs

[9] npm - https://www.npmjs.com/