IJNRD.ORG ISSN : 2456-4184
International Research Journal IJNRD Research Through Innovation
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

ENCOMAPP (FOR MUNICIPAL CORPORATION – COMPLAINT REGISTRATION APPLICATION)

JAICE LEEMA B, PORKODI S
M.E CSE FINAL YEAR STUDENT, ASSISTANT PROFESSOR CSE
GRACE COLLEGE OF ENGINEERING

ABSTRACT

This is an application to resolve the queries of the public. This app will be used by people to complaint about problems in their local area. Problems like garbage, water-related problems, etc will be taken to the municipal corporations and they will see the problem and will resolve the problems.

There will be 2 panels in this app. The first is a user panel and the second is a government panel.

Problem Statement

A complaint management app is basically an app to manage and resolve the queries of the people. It can be used by companies to resolve the queries of their customers or it can be used by the government to resolve the queries of the public/people.

Over all Introduction

- Government Panel
- User Panel

Government Panel

In this panel, the government will see all the complaints added by the public and will resolve them. This will also have a login and signup screen. After that, there will be a dashboard. The government can see all the problems placed by people on the dashboard. They can click on any problem and solve the issue and change its status to work in progress or complete so that the user who posted the complaint can see it.

User Panel

In this panel, the user or the public will put their complaints about problems in their local area. This will have a login and signup screen. After that, there will be a dashboard. The users can see their complaints on the dashboard and the status of the complaint. The user can also add new complaints and add images to state their problems more clearly.

ADVANTAGES:

It saves the time and the user spending cost. People need not to go for the corporation office directly,

People need not to follow any formalities.

There is no chance to miss any forms in any way.

People can know complaint status often.

Features of Complaint Management app

There are two panels in the app.

- The first panel is the user and the second panel is the government panel.
- Login and signup for both the panels.
- Users can add new complaints.
- Users can add images of complaints.
- Users can see all of their complaints and their status.
- Government can see all the complaints posted by people.
- Category-wise complaints.
- Government can open any complaint to resolve it.

Software Requirements

You should be familiar with the following tech to understand the project.

Java – Our application logic is written in Java.

XML – The application is designed using the XML markup language.

Firebase – Firebase will be used as the backend in this project.

Android Studio – It's a platform that allows us to create apps for Android devices.

Hardware Requirements

Hardware configuration to build this project:

* 64 Bit Windows 8/10/11/ubuntu
* Minimum 4GB ram for android studio
* Minimum 12GB of disk space for SDK, java, and IDE
* 1280 x 800 minimum screen resolution.

LITERATURE SURVEY

How squandering the executives is a significant piece of the metropolitan framework as it guarantees the security of the climate and human wellbeing. It isn't just a specialized natural issue yet, in addition, an exceptionally political one. The administration of waste is firmly connected to a scope of variables including metropolitan ways

of life, examples of utilization of assets, levels of work, and pay levels, among others [1].

How arranging various sorts of trash is a dreary assignment. Despite what is generally expected, it is the best method of disposing of trash. We ought to just sort our dustbins into two groupings, biodegradable and non-biodegradable, and dump the loss depending on the situation. We can help others around to concoct this also. By doing this we can reuse specific non-biodegradable waste or discard it securely to forestall soil contamination [4].

Literature Review

initiatives in India for Waste Management. The administration of strong waste has as of late drawn in extensive consideration from the Local and State Legislatures just as nearby (city) experts in India. Various partnerships and alliances are found in the region of solid waste management in India. These coalitions are public-private, local area public, and private-private game plans. To distinguish the situation from existing coalitions in the review region, it is first important to recognize

the different entertainers working in the field of waste administration [6].

The Green City Concept is one of the most recent reactions to a variety of activities and studies aimed at addressing the challenges produced by the dispersed model of city growth and assisting cities in becoming more sustainable (greener), less dispersed, and liveable [7].

Civic bodies, P ublic and the Government made joint efforts towards the cleanliness of the city .The infrastructure was Prepared with a strategy planning for improvement in sanititation. It is not just an effort of the municipal corporation but also the continuous support from people[8].

Floating cities will be the future as sea level is increasing and less land mass for the citizens. It will be self-sufficient regarding energy and other resources. The future smart cities will have smart citizens who will also improve the quality of living. Major goal of the paper is to discuss the condition of smart cities now and in the future.[9]

SOFTWARE DESCRIPTION

USER MODULE

- User can register and login to complaint their problems using name mobile number, address and other relevant details...

- Here, user can view all the types of problem available in corporation like water problem, road problem, etc...

PREVIEW MODULE

- In this Module User can know the status about their complaints.
- Preview module also helps the user to know other relevant problems in corporation,
- The user is going to view our compliant status report.

DOWNLOAD MODULE

- User can download Complaint form and Birth death certificate forms from the website...

- And also user can download other releumt complaint forms such as drainage, and etc..

SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product.It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product it is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce ullid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application wit is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were 'individually satisfaction, as shown by successfully unit testing, the combination of components is con-ect and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify

Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value ofcurrent tests is determined.

SYSTEM TESTING

System testing ensures that the entire integrated software system meets is requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

WHITE BOX TESTING

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

BLACK BOX TESTING

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. The test provides inputs and responds to outputs without considering how the software works.

ACCEPTANCE TESTING

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

SYSTEM IMPLEMENTATION

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new systewill work and be effective.
The implementation stage involves careful planning, investigation of the existing system and it's constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

INTERACTION MODEL

An interaction model is a design model that binds an application together in a way that supports the conceptual models of its target users. It is the glue that holds an application together. It defines how all of the objects and actions that are part of an application interrelate, in ways that mirror and support real-life user interactions.
It ensures that users always stay oriented and understand how to move from place to place to find information or perform tasks. It provides a common vision for an application. It enables designers, developers, and stakeholders to understand and explain how users move from objects to actions within a system.

CLIENT-DRIVEN INTERVENTIONS

Client-driven interventions are the means to protect customers from unreliable services. For example, services that miss deadlines or do not respond at all for a longer time are replaced by other more reliable services in future discovery operations.

PROVIDER-DRIVEN INTERVENTION'S

Provider-driven interventions are desired and initiated by the service owners to shield themselves from malicious clients. For instance, requests of clients performing a denial of service attack by sending multiple requests in relatively short intervals are blocked (instead of processed) by the service.

UML DIAGRAMS

Née prepare UML diagrams to understand a system in better and simple way. A single diagram is not enough to cover all aspects of the system. So UML defines various kinds of diagrams to cover most of the aspects of a system.You can also create your own set of diagrams to meet your requirements. Diagrams are generally made in an incremental and iterative way.There are two broad categories of diagrams and then are again divided into sub-categories:

- Structural Diagrams

- Behavioral Diagrams groups depending upon their relationship. Now these groups are known as components. Finally, component diagrams are used to visualize the implementation.

DEPLOYMENT DIAGRAM:

Deployment diagrams are a set of nodes and their relationships. These nodes are physical entities where the components are deployed. Deployment diagrams are used for visualizing deployment view of a system. This is generally used by the deployment team.

BEHAVIORAL

Any system can have two aspects, static and dynamic. So a model is considered as complete when both the aspects are covered fully.Behavioral diagrams basically capture the dynamic aspect of a system. Dynamic aspect can be further described as the changing/moving parts of a system.

UML has the following five types of behavioral diagrams:

Use case diagram   Sequence diagram   Activity diagram

STRUCTURAL DIAGRAMS:

The structural diagrams represent the static aspect of the system. These static aspects represent those parts of a diagram which forms the main

structure and therefore stable. These static parts are represents by classes, interfaces, objects, components and nodes. The four structural diagrams are:

Class diagram

Object diagram

Component diagram

Deployment diagram

CLASS

Class diagrams are the most common diagrams used in UML Class diagram consists or classes, interfaces, associations and collaboration. Class diagrams basically represent the object oriented view of a system which is static in nature. Active class is used in a class diagram to represent the concurrency of the system. Class diagram represents the object orientation of a system. So it is generally used for development purpose. This is the most widely used diagram at the time of system construction.

OBJECT

Object diagrams can be described as an instance of class diagram. So these diagrams are more close to 'real life scenarios where we implement a system. Object diagrams are a set of objects and their relationships just like class diagrams and also represent the static view of the system. The usage of object diagrams is similar to class diagrams but they are used to build prototype of a system from practical perspective.

COMPONENT DIAGRAM:

Component diagrams represent a set of components and their relationships. These components consist of classes, interfaces or collaborations. So Component diagrams represent the implementation view of a system. During design phase software artifacts (classes, interfaces etc) of a system are arranged in different

USE CASE

Use case diagrams are a set of use cases, actors and their relationships. They represent the use case view of a systetn. A use case represents a particular functionality of a system. So use case diagram is used to describe the relationships among the functionalities and their internal/external controllers. These controllers are known as actors.

SEQUENCE

A sequence diagram is an interaction diagram. From the name it is clear that the diagram deals with some sequences, which are the sequence of messages flowing from one object to another. Interaction among the
components of a system is very important from implementation and execution perspective. So Sequence diagram is used to visualize the sequence of calls in a system to perform a specific functionality.

PROPOSED ARCHITECTURE

The proposed architecture is developed using two different parts: 1. User view and 2. Admin View. More details about these two parts are follows:

User View: In this system, new users need to register themselves by signing up using their name, e-mail address, and password. After Registration, they can log in using the registered username/email and password. After Login, they can access the user dashboard and can view/raise complaints about waste management in their area.

Admin View: In this system, the admin can log in using their username/email and password. After login, they can access the admin dashboard where they will be able to view the complaints raised by users and can resolve them as soon as possible.

Firebase

Google Firebase Firebase is a mobile application development platform developed by Firebase, Inc. in 2011, then acquired by Google in 2014.

Firebase provides a real-time database and backend as a service. The service provides application developers an API that allows application data to be synchronized across clients and stored on Firebase's cloud.

ACTIVITY

Activity diagram describes the flow of control in a system. So it consists of activities and links. The flow can be sequential, concurrent or branched. Activities are nothing but the functions of a system. Numbers of activity diagrams are prepared to capture the entire flow in a system. Activity diagrams are used to visualize the flow of controls in a system. This is prepared to have an idea of how the system will work when executed.

USE CASE DIAGRAM

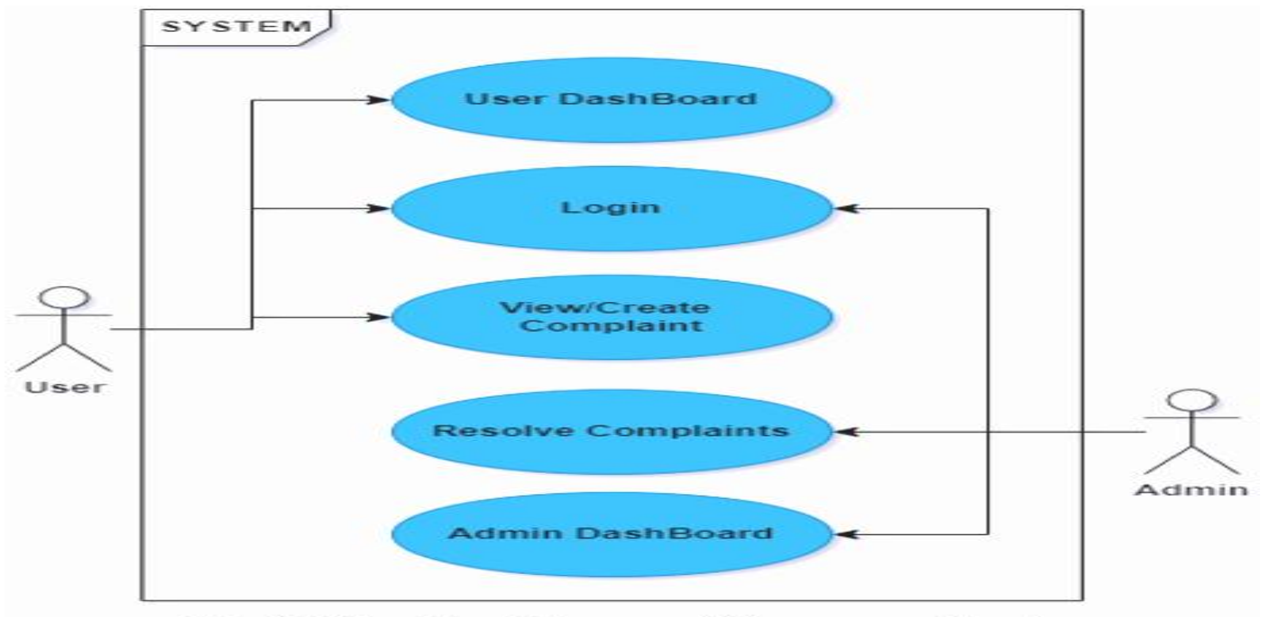


Fig 1.1 Use Case Diagram of the proposed system

System Architecture

The user firstly signs up him or herself and then every time he or she canjust log in as a customer. An admin corporation will be having a fixed username and password as the user cannot enter into the app after the attempts.The user when he or she logins, the User Dashboard page will be opened. Then the user must capture the image of the issue and fill the complaint form. When the Admin logins into the app, he or she should first go through the issues that are posted. The priority-based problems are solved one by one. Our app has two dashboards that are user and admin, the user can log in to post an issue and the admin is the one who belongs to the government can resolve the issues.If the user is a new user he/she can register by giving the details, if he/she is already a user then simply can give a username and password, if it is correct then can directly enter into the user dashboard and the by clicking on the camera icon one can move to the image selection page. Here he/she can add the image either by using the camera or by using the gallery. After that he/she has to provide details about the issue and then submit the issue. After this, by going to the complaint page he/she can verify the issue being recorded. Now the issue will be reflected in the firebase database and can be viewed by the admin.

Now when logged in with a fixed username and password provided to the admin the admin Dashboard appears with a list of issues posted by
the user. By tapping on the image, the admin will be transferred to googlemaps with the location where the issue was generated. With the help of the Resolve button, the admin can resolve the issue.

<u>SYSTEM ARCHITECTURE</u>

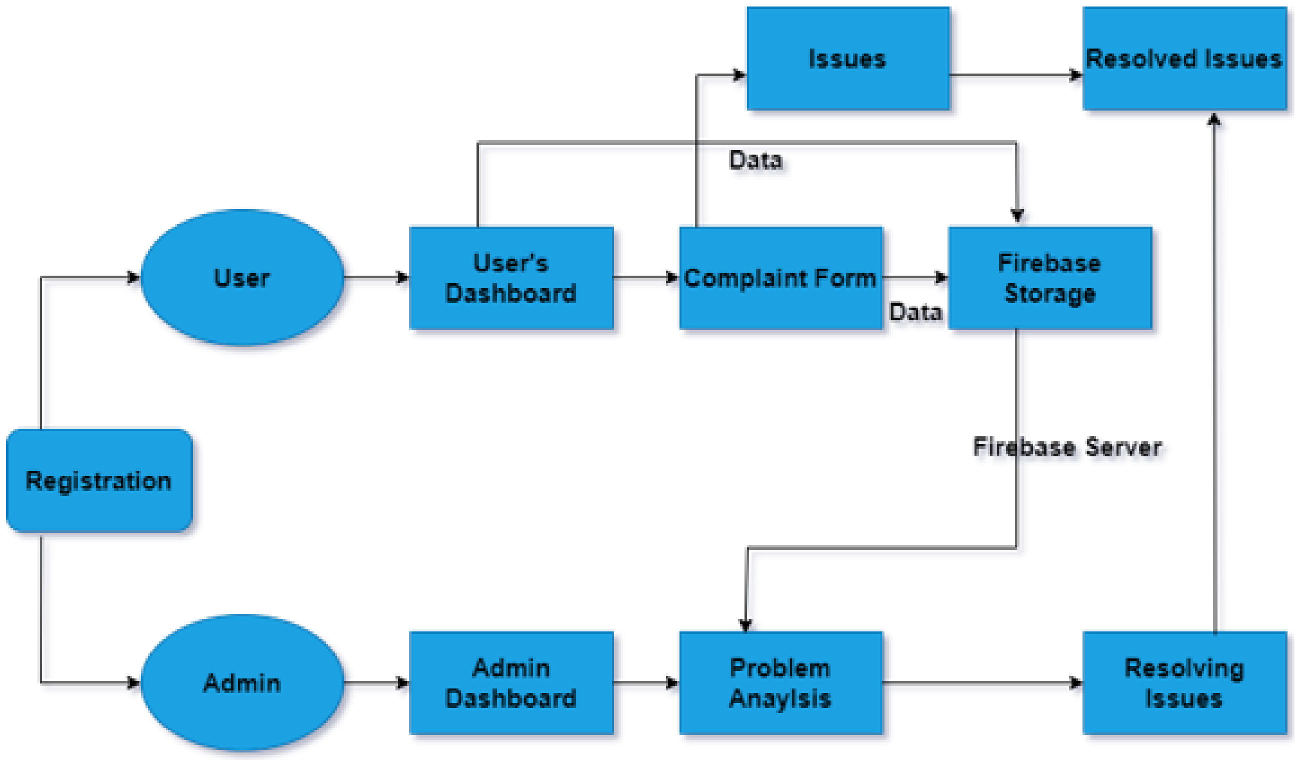
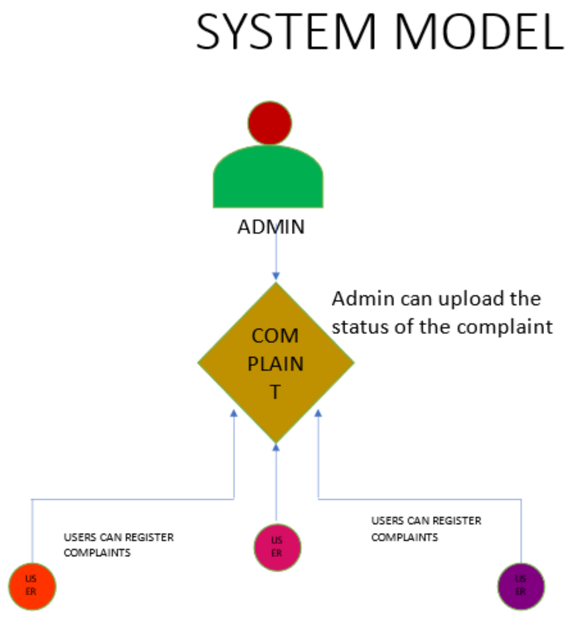


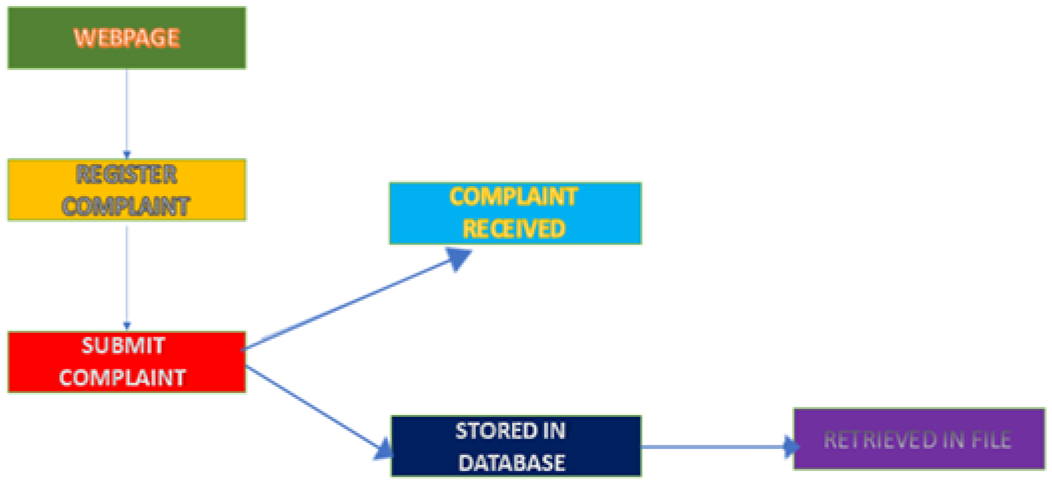
Fig 1.2 Overview of System Architecture


SYSTEM MODEL
ADMIN
COM PLAIN T
Admin can upload the status of the complaint
USERS CAN REGISTER COMPLAINTS
USERS CAN REGISTER COMPLAINTS
US ER
US ER
US ER

The advantages of thissystem model are:

- Human error is greatly reduced
- On board development becomes easy.
- Everyonecan issue their complaints easier.
- Admin side responses are efficient.
- Transparency in the governing processes.
- Saving of time and cost due to provision of services to the citizens through single window.
- Better decision making.
- Simplified office procedures.
- Checking corruption
- Better office and record management.

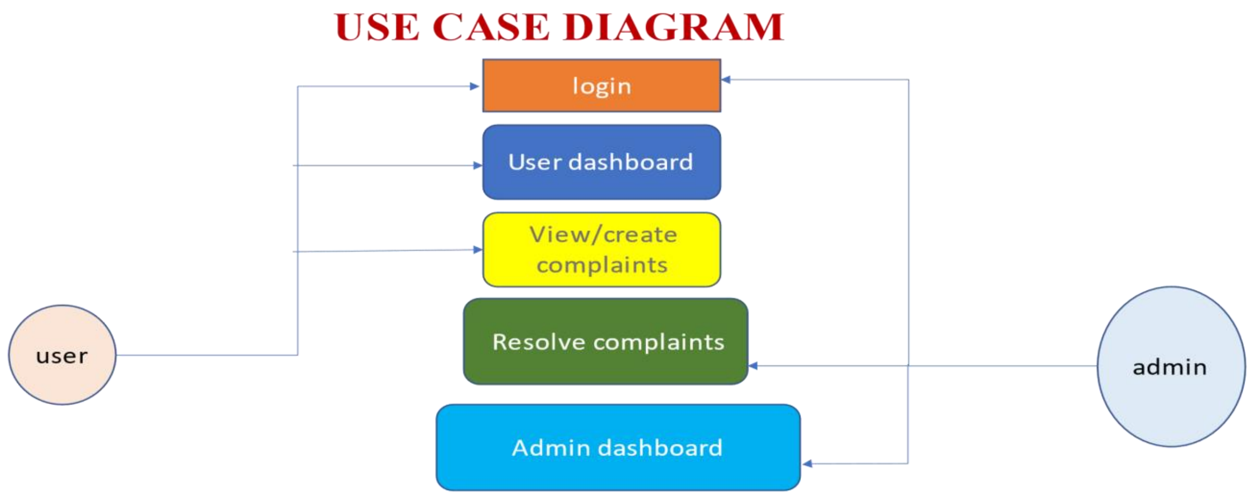
PROPOSED METHODOLOGY :



TECHNOLOGIES USED AES

(Advance Encryption standard) AES (Advanced Encryption System) [7]isone of the most powerful symmetric key encryption algorithm. Thisappuses AES to store information on the cloud for better security. Itis basedon a design principle known as a substitution– permutation network, a combination of both substitution and permutation, and is fast in both software and hardware. Unlike its predecessor DES, AES does not use a Feistel network. AES is a variant of Rijndael which has a fixed block size of 128 bits, and a key size of 128, 192, or 256 bits. By contrast, the Rijndael specification per se is specified with block and key sizes that may be any multiple of 32 bits, with a minimum of 128 and a maximum of 256 bits. Android Studio It is the officialintegrated development environment (IDE) for Google's Android operating system, built on Jet Brains' IntelliJ IDEA software and designed specifically for Android development. The project has been developed in Android Studio. Google Maps API Google Maps [4] is a web mapping service developed by Google. It offers satellite imagery, street maps, 360° panoramic views of streets (Street View), real-time traffic conditions (Google Traffic), and route planning for traveling by foot, car, bicycle, or public transportation. This API is used by the app to track location of the complaint.

USE CASE DIAGRAM:



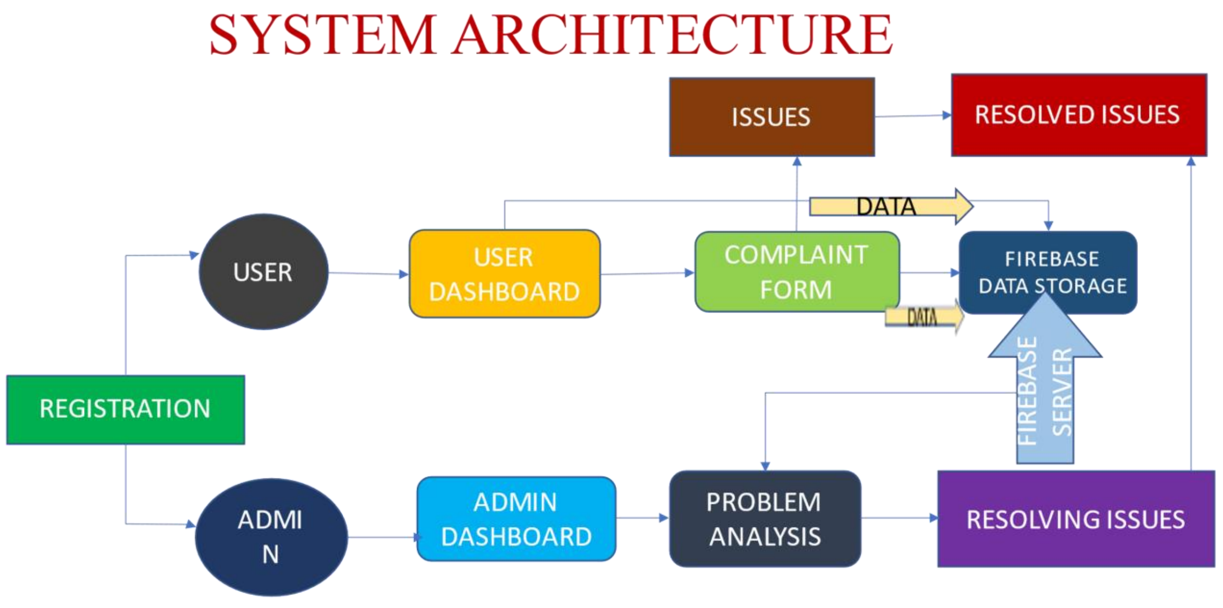


Fig:1.2 overview of system architecture

1.IMPLEMENTATION

A. Architectural Design Fig -4.1: Architecture and Modules 1. User Registration: This module will provide a form to the new user for creating an account. This module will be used to gather all necessary personal information of the user. Before registering the complaint user will have to provide his personal details like Name, Address, Contact Number, etc. This will be used for authenticating the user and the complaint registered by him. After collecting all the necessary information from the user a unique user ID will be assigned to that user.

2. Complaint Registration Module:

This module will provide a form to the user for registering their complaint. It will be mandatory for the user to fill all the fields provided in the form. This form will contain field regarding the complaint categories like complaint about road maintenance, electricity maintenance, water distribution system, garbage collection.

The form a button to upload the image related to the complaint sitelocation. It will also contain a description section where user can provide extra information about the complaint. At the end of the form there will be button which will be used to detect the location of the user. After filling all these details the user has to click the submit button to register his complaint.

Image Upload Module: This module will be used to upload the image related to complaint site. User can upload the image by capturing it from camera of the mobile phone or can upload it from gallery.

4. Database:

Firebase database will be used for storing the information received from all other modules. The database will contain different tables which will store all the information regarding the users, complaint data. It will automatically synchronize all devices as soon as some status updating is done for a complaint

5. Administration Module:

Administration module is used for monitor the entire system. It will take the information from the Database and will feed it to the complaint processing module. Administration module will monitor the entire process of complaint registration like, how many complaints are registered, how many notification has been sent, how many complaints are resolved.

6. Notification Module:

Task of the Notification module is to take the information from the Administration module and then forward it to concern staff. The notification to the staff will contain the information like the location of the complaint, type of work they have to perform, and description about the complaint given by the user. It also notify updates of complaints to users.

B. WORKING OF SYSTEM

1. Registration of User : Initially, User have to Register by filling a small form and Verifying his/her email, mobile number for further processing.

2. Complaint Form Filling : After Registration, User can login and fill up a form giving details about the complaint like the concerned department's name, location of the affected site, etc. The user will have to capture an image of the site of complaint and upload it on the app so as to provide an evidence for the complaint.

3. Complaint ID Generation: A unique complaint ID for each complaint will be generated for each complaint to distinguish between the complaints. The ID will be the complaint location co-ordinates. This will be very helpful for the case when several people are lodging the same complaint. When the same complaint gets registered twice the app will not accept the complaint and give a message saying already registered.

4. Deciding the Priority of the Complaint: Every complaint will be associated with a number of support counter. Anyone who is affected by the problem can support the complaint. The more the number of support the more is the severity of the complaint and the higher up it will be on the feed

5. Notification and Processing of the Complaint: Once the complaint has been lodged then according to its category a notification to the respective department will be sent. In the department the complaints will be ordered according to their severity and will be processed accordingly.

6. Processing of the Complaint: Processing is done in back end at Firebase database server hidden from the user. The staff member are periodically notified by the firebase server about the complaint.

7. Status Updating of the Complaint: Every complaint will be associated with a status which will show the status of the complaint like pending, serving, served or lodged. The status

will be marked according to the process done on the complaint. As soon as the staff members take care of the complaint they can update the status and everyone affected by it will be notified about the update.

8. Generation of the Alert Message: If any complaint is pending for more than the allocated time for processing, then an ALERT message will go to the Admin so that an immediate action can be taken for it. A timer will be set for every complaint on the expiration of which the ALERT message will be sent to the admin

9. Verification of Problem Resolution: The person who has registered complaint has taken out time from his busy schedule and did registered that complaint for the efficient working of public services, therefore it's the responsibility of the system to appreciate his effort and notify him about the work performed by that department in resolving the complaint.

In complaint resolve notification, the system will collect response from the workers who were appointed for solving that complaint. The response will contain all the information regarding the complaint which was resolved containing details about its type, date and time of resolve, the location of the place from where it was resolved.

This collected response will be stored in a centralized server. The system administrator can monitor this data to check the
working efficiency of its department and workers who were allocate the job.
The collect data will be forwarded to the respective user who has registered the complaint C. SPAM DETECTION There can be some users who might keep on registering the same complaint again and again.

Depending upon the data collected from user the system is generating a Complaint ID. All the ID generated by the system will have a unique format. The ID format has been designed in such a way that the registered complaint can be tracked for Spam.

For Example, if same user registers too many complaint again and again then system can detect such user from the allocated user ID. Such user could be blocked from registering complaints again on the system.
If someone tries to register a fake complaint by giving a fake image, the staff members can use their good judgment to see for image forgery being thoroughly familiar with the city.
Nonetheless, the support option helps here anyways. A fake complaint cannot get a lot of supports since no one would recognize it as a true complaint, pushing the complaint downwards in the priority queue and then eventually the complaint will be rejected.

5. RESULT AND ANALYSIS

It will help as an interface between the citizen and the various government authorities. The complaints can be lodged easily in a matter of minutes and status of the complaint can be checked with alert messages given to the staff on serious issues.

Hence, social problems will be reduced, making the city take its steps towards the digital India initiative. It can be applied to solve various government related grievance issues such as Water resource issues, sanitation issues, health security issues, food security issues, pothole issues and the likes.

6. CONCLUSION

We successfully presented an architecture of an Android app which uses Google Fire-base a Secure Real Time Database
to maintain Complaints of the Citizens. In big countries like India, where the cities are growing faster day by day, an application like this will be very useful to manage the cities properly, help out the people in need immediately and will reduce the load of filing complaints in offices. Active participation of citizens in keeping their cities clean, tidy and a better place to live.

This app will boost ”Digital India” and ”Smart City Campaign” and now people who registered their complaint can get real time status of it. It will bring transparency in working of public departments like Municipal Corporations, Nagar Palikas, and Gram Panchayats which will reduce Corruption. Citizens can give feedback about the Govt Work which will lead to better communication between Government and Citizens.

IMPLEMENTATION

Android Studio:
Android Studio is the official integrated development environment (IDE) for Google's Android working machine, in light of JetBrains' IntelliJ program and arranged basically for Android developers. It is to be accessible for download on Windows, macOS, and Linux-based thoroughly working constructions or as a membership-based in 2020. It's a substitute for the E-ADT as an essential IDE for Android application development.

Android Studio was announced on may additionally sixteen, 2013, at the Google I/O convention. It turned within the early get entry to preview degree beginning from version zero.1 in May 2013, then entered the beta stage starting from model zero. eight which turned into launches in June 2014. The primary solid build was launched in December 2014, starting from version 1. Zero.On May 7, 2019, Kotlin supplanted Java as

Google's inclined toward language for Android application advancement. Java keeps on being upheld, as is C++

Features

- Gradle-based form support
- Android-explicit refactoring and handy solutions

- Build up instruments to get execution, ease of use, adaptation similarity, and different issues
- ProGuard coordination and application marking capacities
- Format based wizards to make normal Android plans and parts
- A rich plan editor that grants customers to migrate UI parts, decision to survey designs on various screen courses of action
- Support for building Android Wear applications
- Innate assistance for Google Cloud Stage, engaging joining with FirebaseCloud Illuminating (Earlier 'Google Cloud Illuminating') andGoogle Application Engine
- Android Virtual Gadget (Emulator) to run and investigate applications in the Android studio.

FIREBASE

Firebase is a stage progressed via Google for developing portable and web programs. It became, toward the start,
an unprejudiced organisation established in 2011. In 2014, Google got the stage and its miles are currently their lead introducing for application improvement.It is utilized for cloud utility improvement and makes it less hard to introduce and scale the utility.
Firebase's first item turned into the Firebase Realtime Information base, a Programming interface that synchronises programming realities all through iOS, Android, and web devices, and stores it on Firebase's cloud. Programming software engineers can utilize the item to assemble ongoing, cooperative applications. Firebase brought $1.1 million up in seed financing in May 2012 from Flybridge Capital Accomplices, Greylock Accomplices, Originator Aggregate, and New Undertaking Associates.

In June 2013, Firebase brought $5.6 million up in Series A sponsoring from Affiliation Square Undertakings and Flybridge Capital Assistants.In 2014,Firebase shipped off two things. Firebase Hosting and Firebase Authentications. This arranged the association as a flexible backend as a help.

In October 2014, Firebase was acquired by Google. Following a year, in October 2015, Google acquired Divshot, an HTML5 web-hosting, to combine it with the Firebase.
In May moreover 2016, at Google I/O, the partnership's yearly engineer meeting, Firebase presented Firebase Analytics and declared that it was growing its

administrations to turn into a backend-as-a-administration (BaaS) for developers. Firebase fuses different Google services, including Google Cloud, AdMob, and GoogleAuth, to offer architects a more broad extent of things and scale.

Google Cloud Illuminating, the Google organization to send message pop-ups to Android, changed into outdated through Firebase, Firebase Cloud, which added the ability to supply spring-up messages to each io and web device.

In July 2016, Google declared that it was securing the versatile designer stage LaunchKit, which worked in application engineering publicizing, and would be collapsing it into the Firebase development. In January 2017, Google obtained texture and Crashlytics from Twitter to include the contributions to Firebase.

In October 2017, Firebase sent off Cloud Firestore, a real-time document data set as the replacement item to the exceptional Firebase Realtime.

RESULT

Our application is better than others in the market as it is easy to use,takes care of privacy of the user,reduces spam complaints by using location and images.

A. Login Page

Login with valid email id and password which was generated by users while signup can be done as shown in figure 1.1

B. SignUp Page

Signup the App to register by providing the user's details like Email id and Password and clicking on the signup button can be done as shown in figure

MAPS

After Login the app will open the home page. In our Home Page maps will open and it will automatically detect the location wherever you are as shown in figure.

Camera:

After clicking on camera you will have 2 options:-

Choose from Gallery

Take photos by using a Camera.

COMPLAINT FORM

Complaint Form

After Choosing a photo a complaint form will open and you want to enter the title and description after that you can submit it as shown in figure 1.6

SIGNOUT

SignOut Page

You can go to your profile and sign out from there as shown

Conclusion

These Clean City applications would help citizens and authorities to keep the city clean by making the communication between both parties quicker and also help resolve cleanliness-related issues in the city. It will also improve the
hygiene of citizens as overflowing garbage bins are the main cause of many harmful diseases.

REFERENCES

1. Christoph Scharff, ISWA An Web-Based Application for waste management to ensure sustainable development components necessary to attain sustainable waste management. Waste Management Report (Pages 75-80), January 2002.http://www.sustentabilidad.uai.edu.ar/pdf/ing/waste_management.pdf.

2. Andrei Borozdukhin, Olga Dolinina, Vitaly Pechenkin Built-in system in vehicles and garbage containers to develop an optimal route for garbage collectionOctober 2016 DOI: 10.1109/CIST.2016.7805019 Conference: 2016 4th IEEE International Colloquium on Information Science and Technology (CIST).

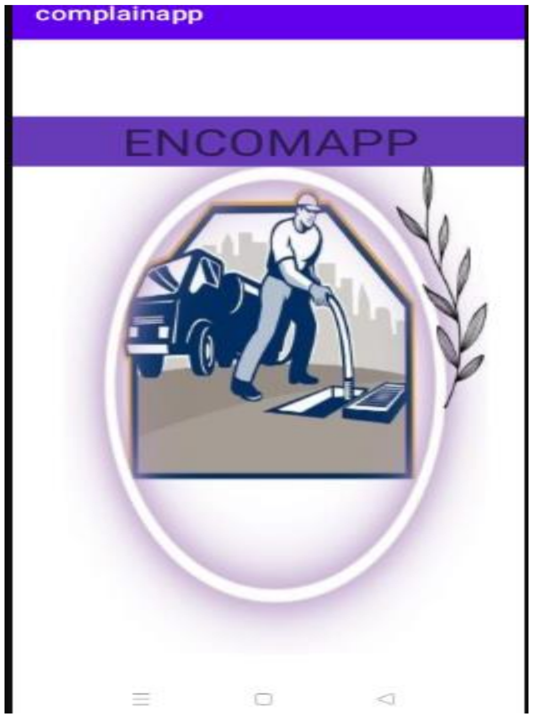
3. Wenrui Li, Bharat Bhushan, Jerry Gao, School of Information Engineering, Nanjing XiaoZhuang University, Nanjing, Jiangsu, 211171, China" Smart Clean: Smart City Street Cleanliness System Using Multiple-Level Assessment Model", Article ·November (2018).

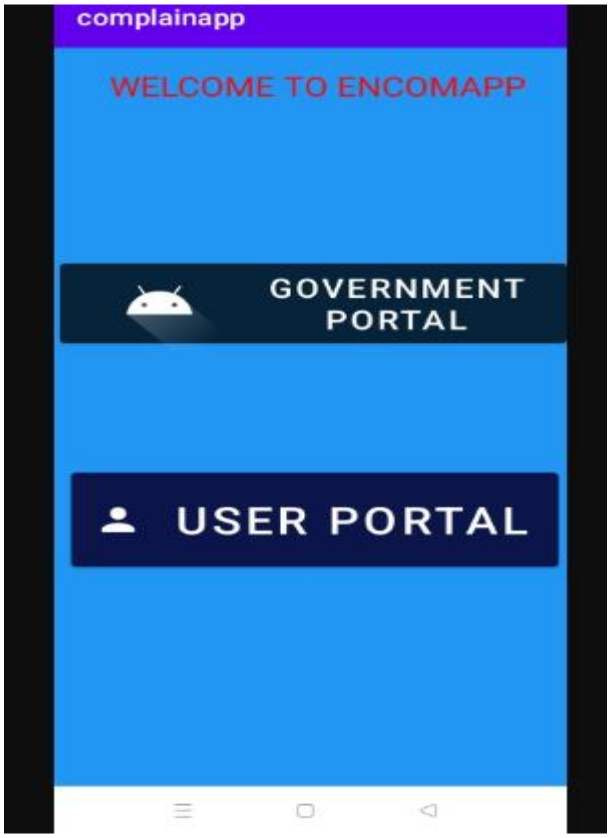
4. Udhaya Mohan Babu, Kalaiyarasan Ganesan to tidy up the roads, streets, and framework of India's urban communities, more modest towns, and country regions. My

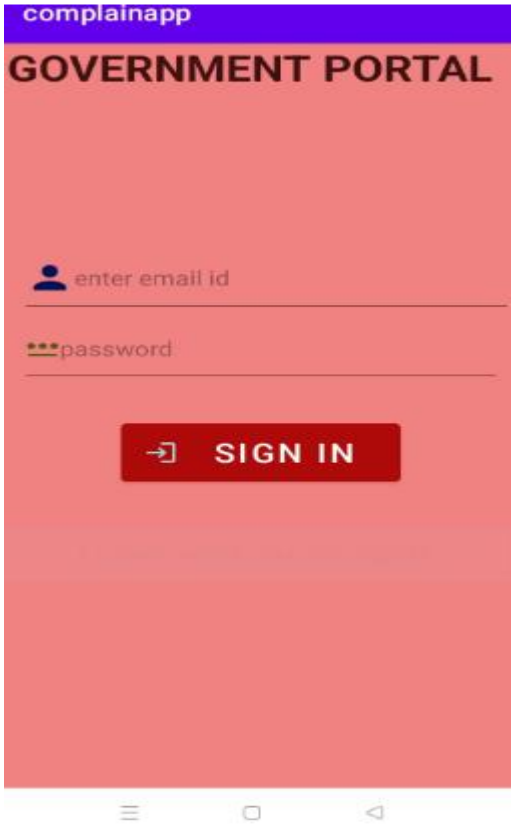
Clean City (MC3) October 2017 Conference: Teacher Education Enhancing Clean Indian and Green India in 21st Century.
5. A. Kirimtat, O. Krejcar, A. Kertesz and M. F. Tasgetiren, "Future Trends and Current State of Smart City Concepts: A Survey," in IEEE Access, vol. 8, pp. 86448-86467, 2020, doi: 10.1109/ACCESS.2020.2992441.
6. Chung-Ming Huang, Chia-Ching Yang, ChunYu Tseng, Chih-Hsun Chou. ASandip Kendre Centralized Traffic Control Mechanism for Evacuation of Emergency Vehicles Using the DSRC Protocol.Laboratory of Multimedia Mobile Networking Department of Computer Science and Information Engineering National Cheng Kung University, Tainan, Taiwan, R.O.C

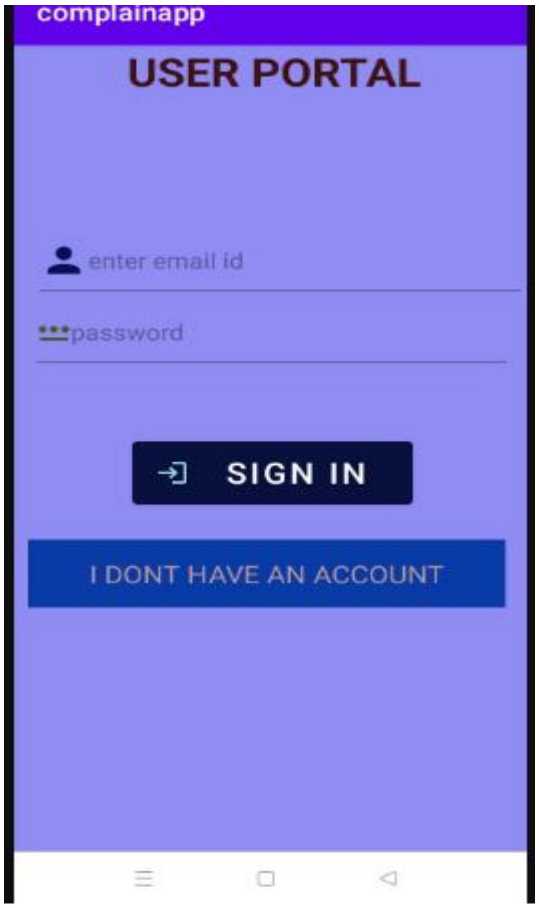
7. Sunil Kumar Kopparapu.(2008). Natural Language Mobile Interface to Manage Citizen Complaints.TCS Innovation Lab – MumbaiVoice Enabled Android Application for Tracking complaint and Pothole Notification System Using GPS and .
8. Google Maps Javascript API v3. internet:developers.google.com/map/web/I.
9.Google Firebase Internet: https://firebase.google.com/
10Pimpri Chinchwad Municipal Corperation SMS and WebBased Complaint Monitoring System July 2012.

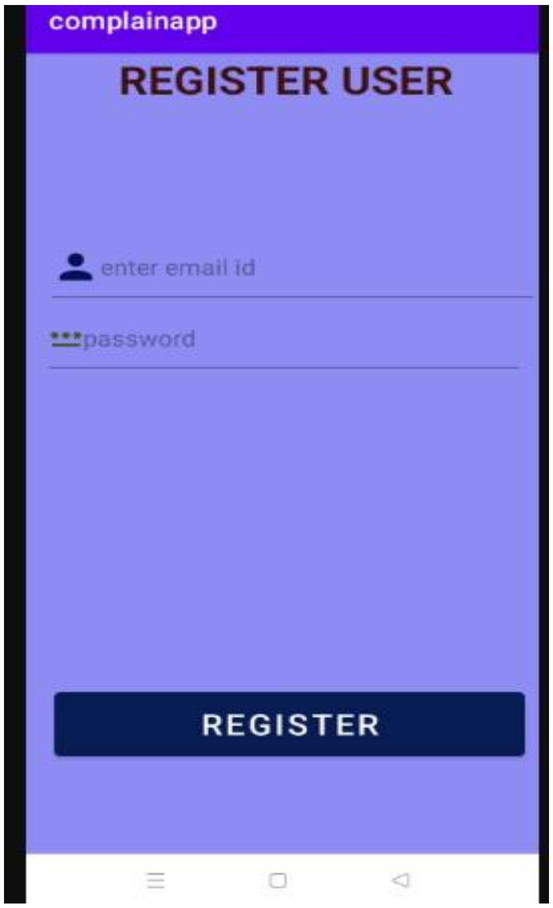
OUTPUT:

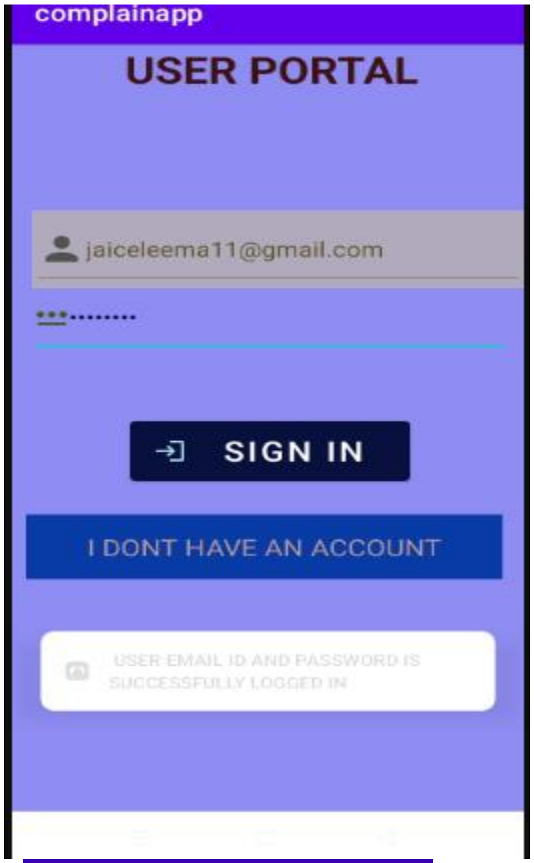


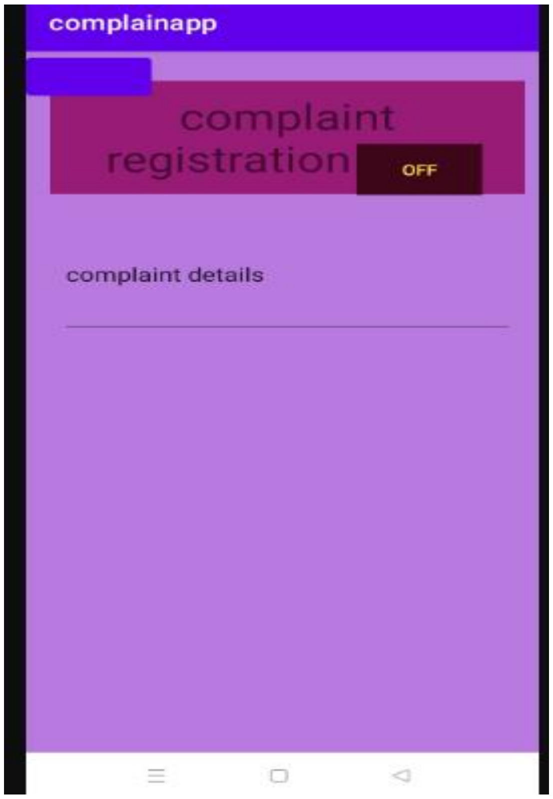

complainapp
WELCOME TO ENCOMAPP
GOVERNMENT PORTAL
USER PORTAL

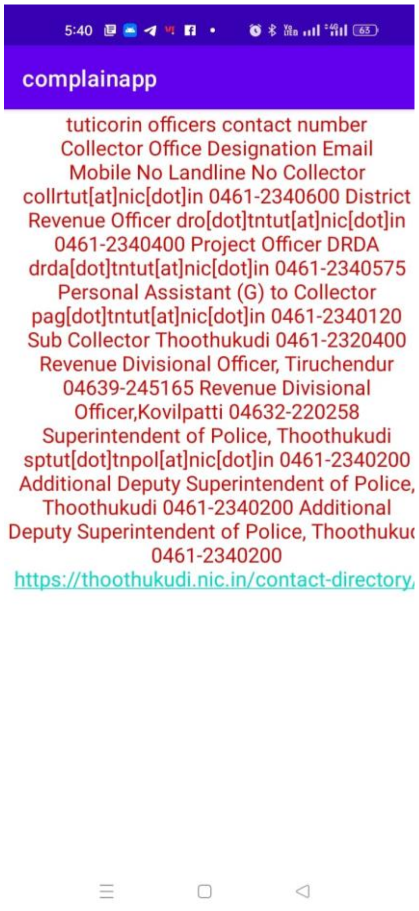

complainapp
GOVERNMENT PORTAL
enter email id
password
SIGN IN


complainapp
USER PORTAL
enter email id
password
SIGN IN
I DONT HAVE AN ACCOUNT

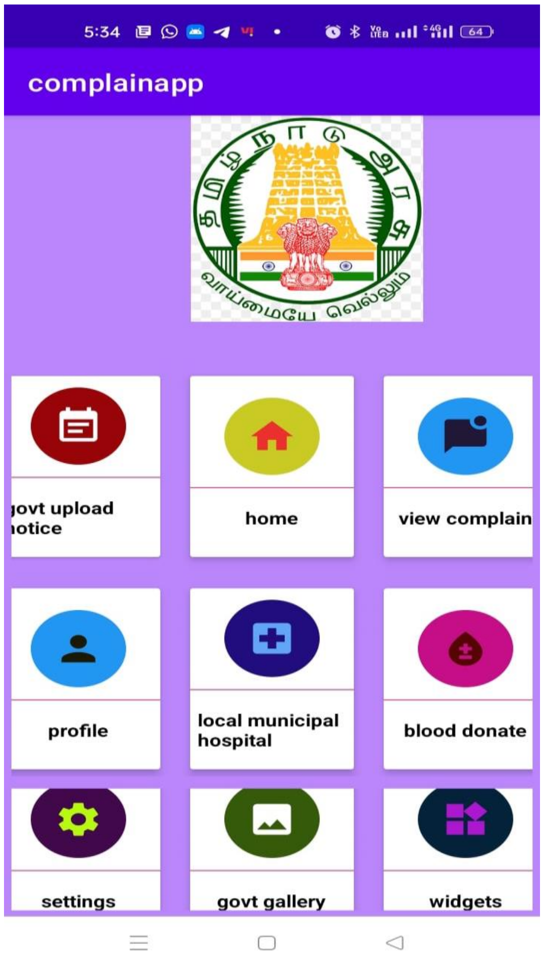

complainapp
REGISTER USER
enter email id
password
REGISTER

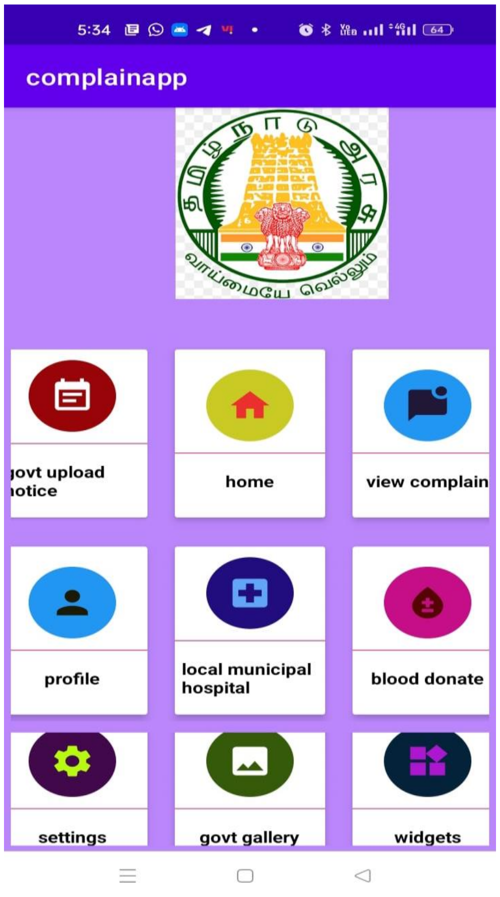


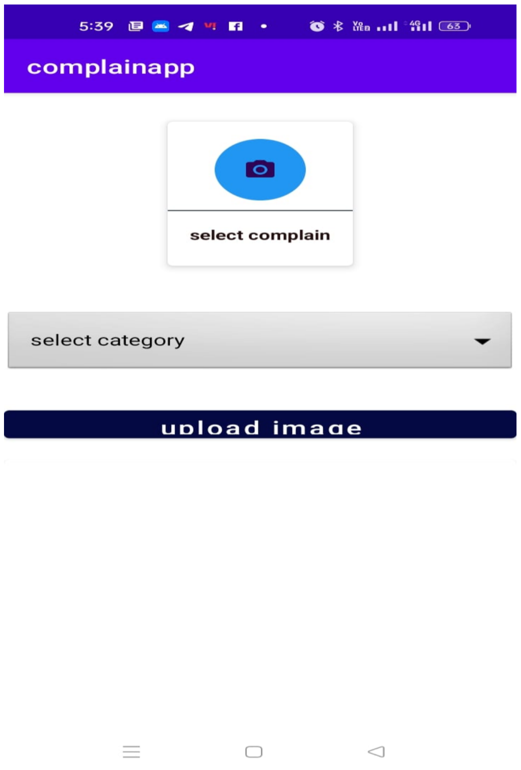


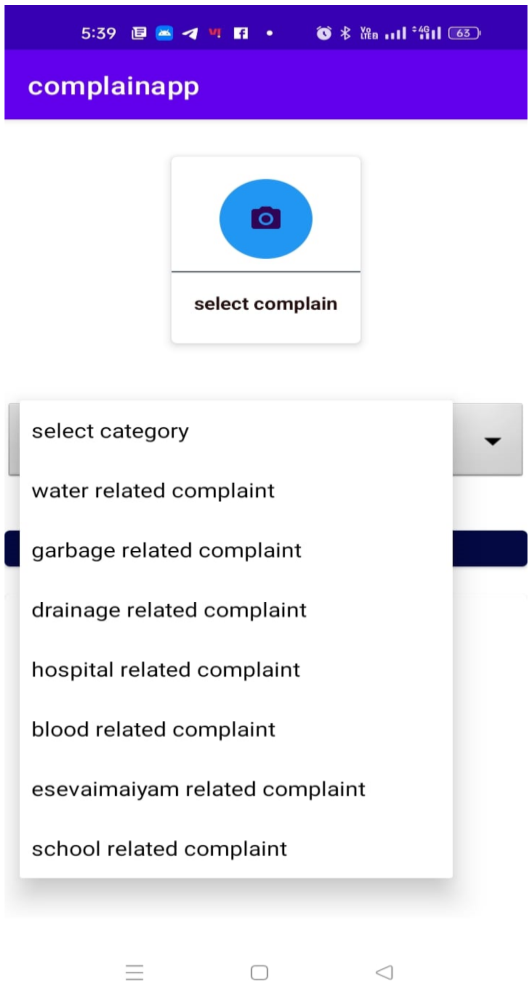


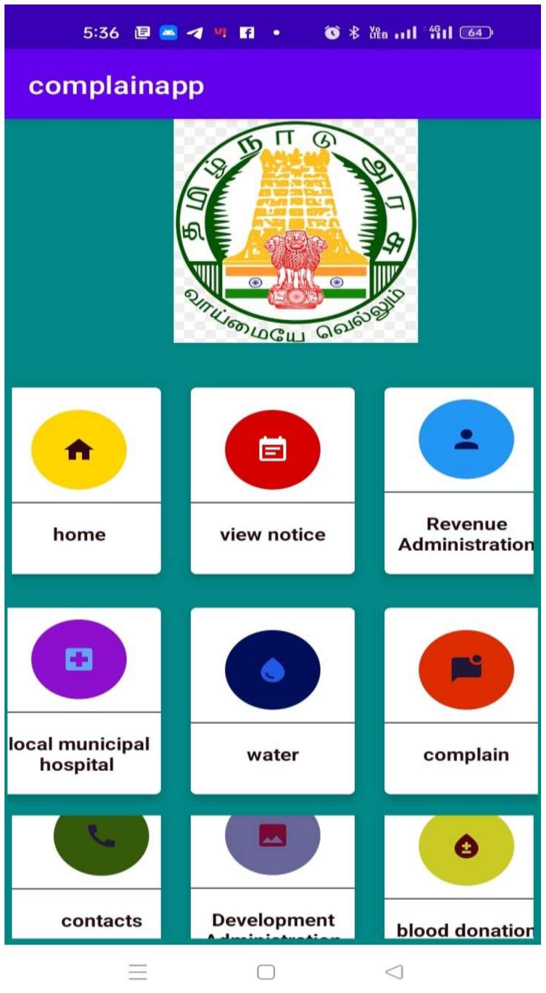
tuticorin officers contact number Collector Office Designation Email Mobile No Landline No Collector collrtut[at]nic[dot]in 0461-2340600 District Revenue Officer dro[dot]tntut[at]nic[dot]in 0461-2340400 Project Officer DRDA drda[dot]tntut[at]nic[dot]in 0461-2340575 Personal Assistant (G) to Collector pag[dot]tntut[at]nic[dot]in 0461-2340120 Sub Collector Thoothukudi 0461-2320400 Revenue Divisional Officer, Tiruchendur 04639-245165 Revenue Divisional Officer,Kovilpatti 04632-220258 Superintendent of Police, Thoothukudi sptut[dot]tnpol[at]nic[dot]in 0461-2340200 Additional Deputy Superintendent of Police, Thoothukudi 0461-2340200 Additional Deputy Superintendent of Police, Thoothuku 0461-2340200 https://thoothukudi.nic.in/contact-directory,

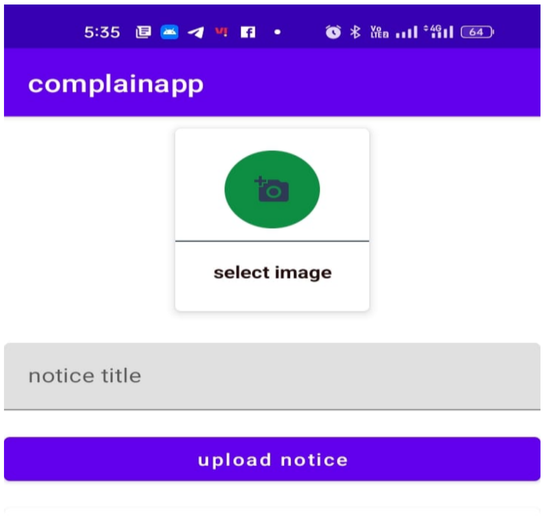


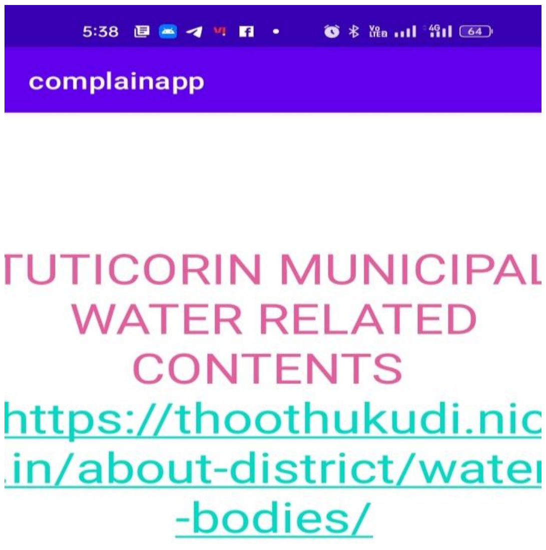

5:34
complainapp
govt upload notice
home
view complain
profile
local municipal hospital
blood donate
settings
govt gallery
widgets


5:39
complainapp
select complain
select category
upload image

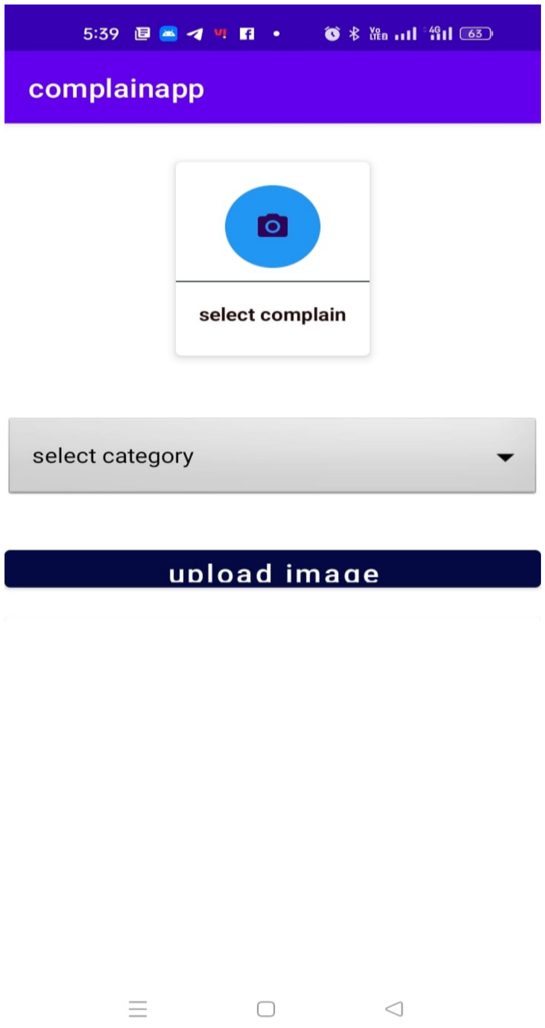

5:39
complainapp
select complain
select category
water related complaint
garbage related complaint
drainage related complaint
hospital related complaint
blood related complaint
esevaimaiyam related complaint
school related complaint

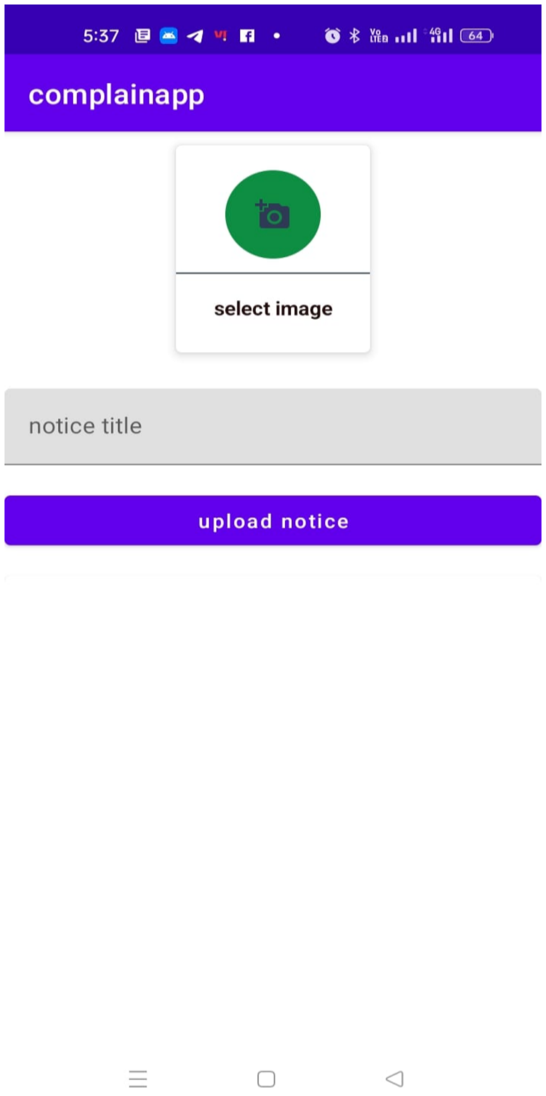

5:36
complainapp
home
view notice
Revenue Administration
local municipal hospital
water
complain
contacts
Development
blood donation

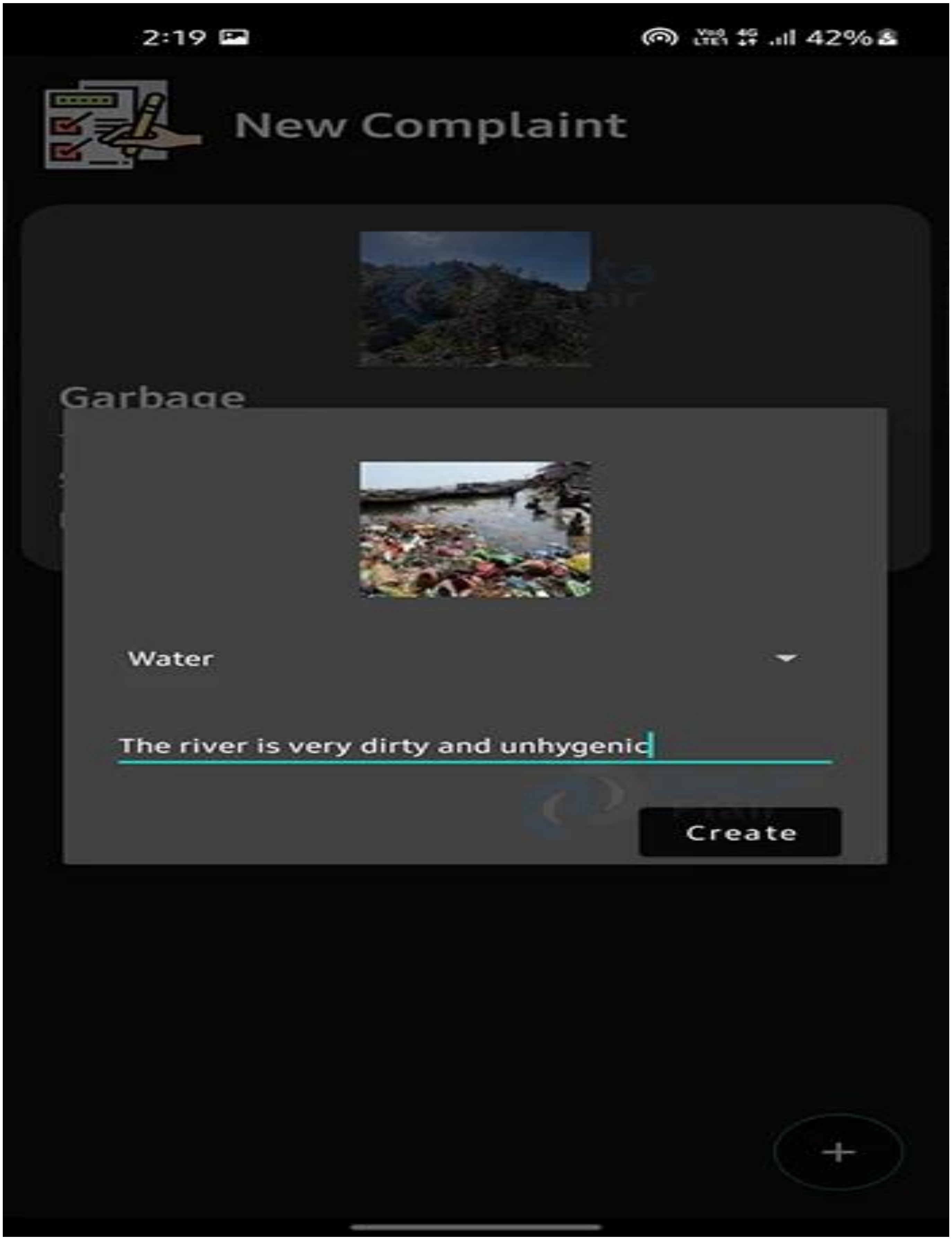




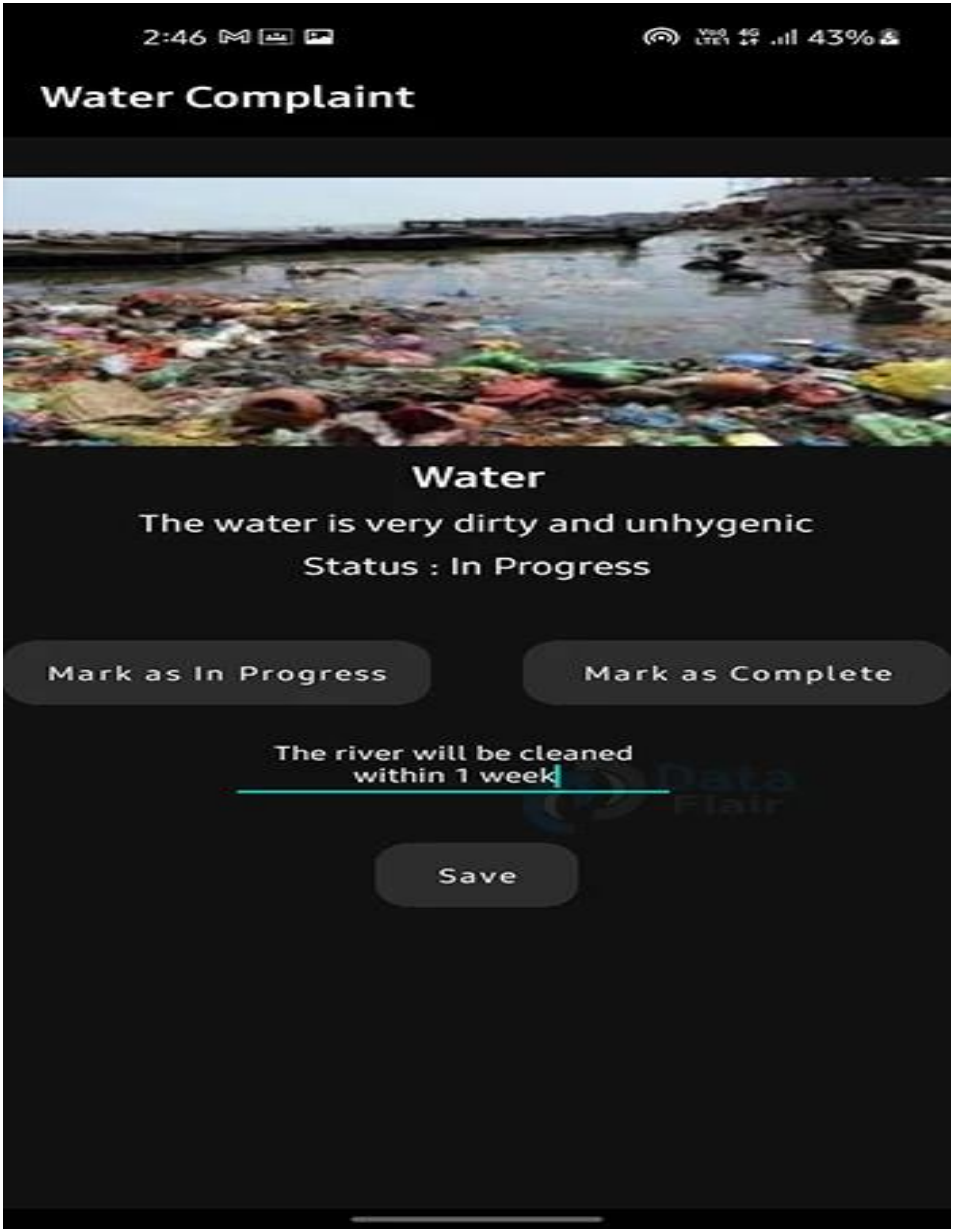
TUTICORIN MUNICIPAL WATER RELATED CONTENTS https://thoothukudi.nic.in/about-district/wate -bodies/







The river is very dirty and unhygenic


2:46 43%

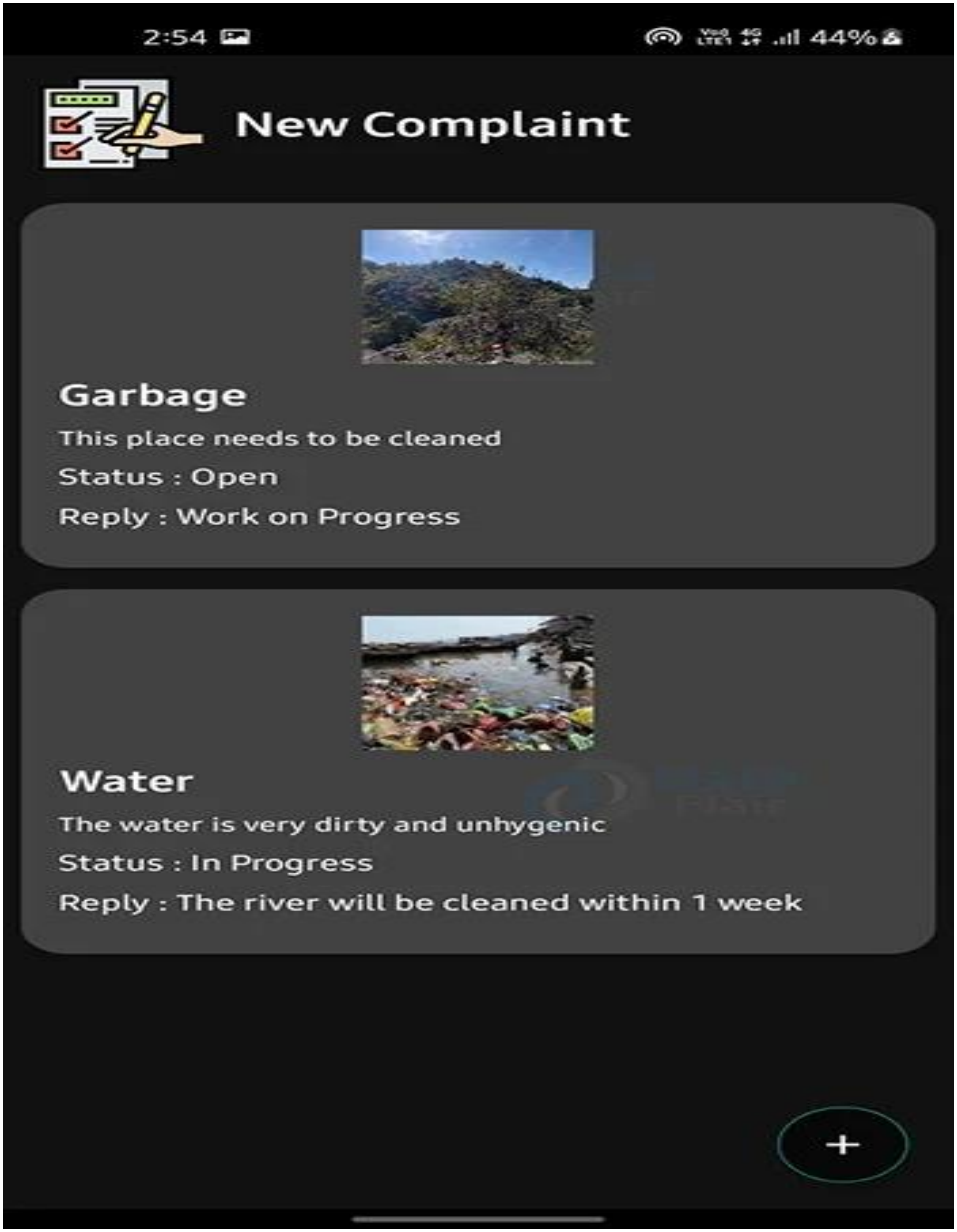
Water Complaint

Water

The water is very dirty and unhygenic

Status : In Progress

The river will be cleaned within 1 week



2:54 44%

New Complaint

Garbage

This place needs to be cleaned

Status : Open

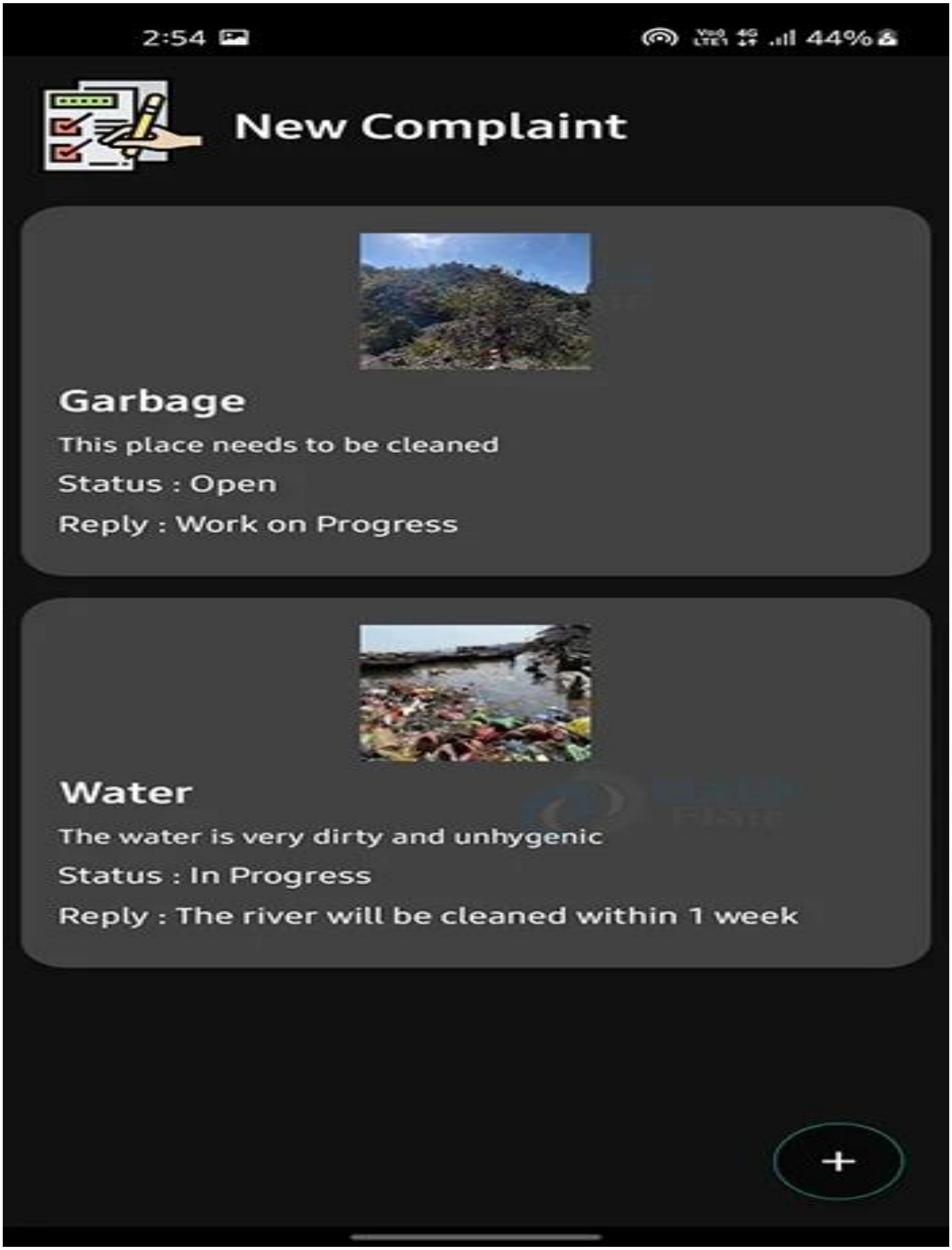
Reply : Work on Progress

Water

The water is very dirty and unhygenic

Status : In Progress

Reply : The river will be cleaned within 1 week



New Complaint

Garbage

This place needs to be cleaned

Status : Open

Reply : Work on Progress

Water

The water is very dirty and unhygenic

Status : In Progress

Reply : The river will be cleaned within 1 week

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:background="@drawable/lee2"
   android:layout_height="match_parent"
   tools:context=".MainActivity">
<TextView
      android:id="@+id/textView"
      android:layout_width="match_parent"
      android:layout_height="153dp"
      android:layout_alignParentStart="true"
      android:layout_alignParentLeft="true"
      android:layout_alignParentTop="true"
      android:layout_alignParentEnd="true"

android:layout_marginStart="17dp"
      android:layout_marginLeft="17dp"
      android:layout_marginTop="234dp"
      android:layout_marginEnd="0dp"
      android:padding="10dp"
      android:text="JAI APP"
      android:textColor="#448AFF"
      android:textColorHighlight="@color/black"
      android:textSize="90sp" />
</RelativeLayout>
```

```
android:layout_marginStart="17dp"
      android:layout_marginLeft="17dp"
      android:layout_marginTop="234dp"
      android:layout_marginEnd="0dp"
```

```
    android:padding="10dp"
    android:text="JAI APP"
    android:textColor="#448AFF"
    android:textColorHighlight="@color/black"
    android:textSize="90sp" />
```

```
package com.example.mathes;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;
public class MainActivity extends AppCompatActivity
{
  @Override
  protected void onCreate(Bundle savedInstanceState)
  {
    super.onCreate(savedInstanceState);

setContentView(R.layout.activity_main);
    Thread thread=new Thread()
    {
    public void run()
    {
      try
      {
        sleep(10000);
      }
      catch(Exception e)
      {
        e.printStackTrace();
      }
finally
            {
                Intent welcomeIntent = new Intent(MainActivity.this,
WelcomeActivity.class);
                startActivity(welcomeIntent);
        }
        };
        thread.start();
    }
    @Override
    protected void onPause()
    {
        super.onPause();
        finish();
    }
}

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@drawable/lee3"
    tools:context=".WelcomeActivity">
<Button
        android:id="@+id/button"
        android:layout_width="match_parent"
        android:layout_height="103dp"
        android:layout_alignParentStart="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginStart="0dp"



android:layout_marginLeft="0dp"
        android:layout_marginTop="132dp"
        android:background="@color/teal_200"
        android:text="DRIVER LOGIN"
        android:textColor="#546E7A"
        android:textSize="45sp"
        app:backgroundTint="#BC22D6" />
    <Button
        android:id="@+id/buttn"
        android:layout_width="432dp"
        android:layout_height="117dp"
        android:layout_alignBottom="@+id/button"
        android:layout_alignParentStart="true"

android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_marginStart="0dp"
        android:layout_marginEnd="-21dp"
        android:layout_marginRight="-21dp"
        android:layout_marginBottom="-214dp"
        android:text="CUSTOMER LOGIN"
        android:textSize="38sp“
        app:backgroundTint="#3F1047" />
</RelativeLayout>
Welcome.java
 package com.example.mathes;
 import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.widget.Button;
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".DriverLoginRegisterActivity">
    <TextView
        android:id="@+id/govt_Status"
        android:layout_width="406dp"
        android:layout_height="50dp"
        android:layout_marginTop="16dp"
        android:text="GOVT LOGIN REGISTER"
        android:textColor="#070737"
        android:textSize="35sp"
        android:textStyle="bold"
        app:layout_constraintTop_toTopOf="parent"
        tools:layout_editor_absoluteX="4dp" />
<EditText
        android:id="@+id/email_driver"
        android:layout_width="match_parent"
        android:layout_height="75dp"
        android:layout_alignParentTop="true"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_marginStart="16dp"
        android:layout_marginLeft="16dp"
        android:layout_marginTop="98dp"
        android:layout_marginEnd="0dp"
        android:layout_marginRight="0dp"
        android:ems="10"
        android:hint="ENTER EMAIL ADDRESS"
        android:inputType="textEmailAddress"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        tools:layout_editor_absoluteY="100dp" />
<EditText
        android:id="@+id/password_driver"
        android:layout_width="414dp"
        android:layout_height="85dp"
        android:layout_alignParentTop="true"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_marginStart="4dp"
        android:layout_marginLeft="4dp"
        android:layout_marginTop="201dp"
        android:layout_marginEnd="-3dp"
        android:layout_marginRight="-3dp"
```

```
        android:ems="10"
        android:hint="ENTER PASSWORD"
        android:inputType="textPassword"
        app:layout_constraintStart_toStartOf="parent"
        tools:layout_editor_absoluteY="217dp" />
    <Button
        android:id="@+id/Driver_Login_btn"
        android:layout_width="409dp"
        android:layout_height="wrap_content"
        android:layout_alignParentStart="true"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginStart="2dp"
        android:layout_marginEnd="0dp"
        android:layout_marginBottom="218dp"
        android:text="LOGIN"
        android:textSize="26sp"
        app:backgroundTint="#870D0D"
        tools:layout_editor_absoluteX="1dp"
        tools:layout_editor_absoluteY="305dp"
        android:layout_alignParentLeft="true"
        android:layout_alignParentRight="true"
        android:layout_marginLeft="2dp"
        android:layout_marginRight="0dp" />
    <TextView
        android:id="@+id/Driver_link"
        android:layout_width="406dp"
        android:layout_height="40dp"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:layout_alignParentEnd="true"
        android:layout_marginStart="7dp"
        android:layout_marginTop="441dp"
        android:layout_marginEnd="-2dp"
        android:text="DON'T HAVE AN ACCOUNT"
        android:textSize="30sp"
        tools:layout_editor_absoluteX="5dp"
        tools:layout_editor_absoluteY="367dp"
        android:layout_alignParentLeft="true"
```

```
android:layout_alignParentRight="true"
        android:layout_marginLeft="7dp"
        android:layout_marginRight="-2dp" />
    <Button
        android:id="@+id/driver_register_btn"
        android:layout_width="277dp"
        android:layout_height="72dp"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:layout_marginBottom="30dp"
        android:text="REGISTER"
        android:textSize="30sp"
        app:backgroundTint="#8B149F"
         />
</RelativeLayout>
package com.example.mathes;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import android.app.ProgressDialog;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
public class DriverLoginRegisterActivity extends AppCompatActivity {

    private Button dlo;
    private Button dreg;
    private TextView dst;
    private TextView dlink;
    private EditText demail;
    private EditText dpass;
    private ProgressDialog loadingbar;
    private FirebaseAuth mAuth;
   @Override
      protected void onCreate(Bundle savedInstanceState) {
       super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_driver_login_register);
       mAuth=FirebaseAuth.getInstance();
      dst=(TextView) findViewById(R.id.Driver_Status);
       dlink=(TextView) findViewById(R.id.Driver_link);
       dlo=(Button)findViewById(R.id.Driver_Login_btn);
       dreg=(Button) findViewById(R.id.driver_register_btn);
demail=(EditText) findViewById(R.id.email_driver);
       dpass=(EditText) findViewById(R.id.password_driver);
       loadingbar=new ProgressDialog(this);
       dreg.setVisibility(View.INVISIBLE);
       dreg.setEnabled(false);
       dlink.setOnClickListener(new View.OnClickListener() {
           @Override
           public void onClick(View view)
           {
               dlo.setVisibility(View .INVISIBLE);
               dlink.setVisibility(View.INVISIBLE);
               dst.setText("REGISTER DRIVER");
               dreg.setVisibility(View.VISIBLE);
               dreg.setEnabled(true);
           }
       });

dreg.setOnClickListener(new View.OnClickListener() {
               @Override
               public void onClick(View view)
               {   String driemail=demail.getText().toString();
                   String dripassword=dpass.getText().toString();
                 dreg(driemail,dripassword);
               }           });
           dlo.setOnClickListener(new View.OnClickListener() {
               @Override
               public void onClick(View view)
               {
                   String driemail=demail.getText().toString();
```

```
                String dripassword=dpass.getText().toString();
                driverlogin(driemail,dripassword);
            }           });    }
    private void driverlogin(String driemail, String dripassword)
    {
        if (TextUtils.isEmpty(driemail))
{
            Toast.makeText(DriverLoginRegisterActivity.this, "PLEASE ENTER EMAIL
ADDRESS ", Toast.LENGTH_SHORT).show();
        }
        if (TextUtils.isEmpty(dripassword))
{
            Toast.makeText(DriverLoginRegisterActivity.this, "PLEASE ENTER YOUR
PASSWORD", Toast.LENGTH_SHORT).show();
        }

        else
        {
            loadingbar.setTitle("DRIVER LOGIN DETAILS");
            loadingbar.setMessage("PLEASE WAIT WHILE WE ARE CHECKING
YOUR CREDENTIALS....");
            loadingbar.show();
mAuth.signInWithEmailAndPassword(driemail,dripassword).addOnCompleteListener
(new OnCompleteListener<AuthResult>() {
            @Override
public void onComplete(@NonNull Task<AuthResult> task)
            {
                if(task.isSuccessful())
                {
                    Toast.makeText(DriverLoginRegisterActivity.this, " DRIVER
LOGGED IN SUCCESSFULLY ", Toast.LENGTH_SHORT).show();
                    loadingbar.dismiss();
                }
                else
                {
                    Toast.makeText(DriverLoginRegisterActivity.this, "LOGIN
UNSUCCESSFUL....PLEASE TRY AGAIN ", Toast.LENGTH_SHORT).show();
                    loadingbar.dismiss();
                }
            }
          });
        }
    }
    private void dreg(String driemail, String dripassword)
    {
if (TextUtils.isEmpty(driemail))
        {
            Toast.makeText(DriverLoginRegisterActivity.this, "PLEASE ENTER EMAIL
ADDRESS ", Toast.LENGTH_SHORT).show();
        }
         if (TextUtils.isEmpty(dripassword))
         {
             Toast.makeText(DriverLoginRegisterActivity.this, "PLEASE ENTER YOUR
PASSWORD", Toast.LENGTH_SHORT).show();
         } dlo.setOnClickListener(new View.OnClickListener() {
              @Override
              public void onClick(View view)
              {
                 String driemail=demail.getText().toString();
                 String dripassword=dpass.getText().toString();
                 driverlogin(driemail,dripassword);
              }
          });
    }
private void driverlogin(String driemail, String dripassword)
    {
        if (TextUtils.isEmpty(driemail))
        {
            Toast.makeText(DriverLoginRegisterActivity.this, "PLEASE ENTER EMAIL
ADDRESS ", Toast.LENGTH_SHORT).show();
        }
        if (TextUtils.isEmpty(dripassword)) {
            Toast.makeText(DriverLoginRegisterActivity.this, "PLEASE ENTER YOUR
PASSWORD", Toast.LENGTH_SHORT).show();
        }
        else
        {
            loadingbar.setTitle("DRIVER LOGIN DETAILS");
        loadingbar.setMessage("PLEASE WAIT WHILE WE ARE CHECKING
YOUR CREDENTIALS....");
        loadingbar.show();
mAuth.signInWithEmailAndPassword(driemail,dripassword).addOnCompleteLi
stener(new OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task)
            {
                if(task.isSuccessful())
                {
                    Toast.makeText(DriverLoginRegisterActivity.this, " DRIVER
LOGGED IN SUCCESSFULLY ", Toast.LENGTH_SHORT).show();
                    loadingbar.dismiss();
                }
                else
                {                    Toast.makeText(DriverLoginRegisterActivity.this,
"LOGIN UNSUCCESSFUL....PLEASE TRY AGAIN ",
Toast.LENGTH_SHORT).show();
                    loadingbar.dismiss();
                }
            }
        });
    }
  }
private void dreg(String driemail, String dripassword)
  {
    if (TextUtils.isEmpty(driemail))
    {
        Toast.makeText(DriverLoginRegisterActivity.this, "PLEASE ENTER
EMAIL ADDRESS ", Toast.LENGTH_SHORT).show();
    }
     if (TextUtils.isEmpty(dripassword))
     {
         Toast.makeText(DriverLoginRegisterActivity.this, "PLEASE ENTER
YOUR PASSWORD", Toast.LENGTH_SHORT).show();
     }
        else
        {
            loadingbar.setTitle("DRIVER LOGIN REGISTER");
            loadingbar.setMessage("PLEASE WAIT WHILE WE ARE
REGISTER YOUR DATA....");
            loadingbar.show();
mAuth.createUserWithEmailAndPassword(driemail,dripassword).addOnCompl
eteListener(new OnCompleteListener<AuthResult>() {
                @Override
                public void onComplete(@NonNull Task<AuthResult> task)
                {
                    if(task.isSuccessful())
                    {
Toast.makeText(DriverLoginRegisterActivity.this, " DRIVER EMAIL ID AND
PASSWORD IS SUCCESSFULLY REGISTERED",
Toast.LENGTH_SHORT).show();
                        loadingbar.dismiss();
                    }                   else
                    {
                        Toast.makeText(DriverLoginRegisterActivity.this,
"REGISTRATION UNSUCCESSFUL....PLEASE TRY AGAIN ",
Toast.LENGTH_SHORT).show();
                        loadingbar.dismiss();
                    }
                }
            });
        }       }   }
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".CustomerLoginRegisterActivity">
    <TextView
        android:id="@+id/customer_status"
        android:layout_width="397dp"
        android:layout_height="56dp"
        android:layout_alignParentStart="true"
        android:layout_alignParentEnd="true"
        android:layout_marginStart="23dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="-9dp"
        android:text="CUSTOMER LOGIN REGISTER"
        android:textColor="#D30E69"
        android:textSize="28sp"

    android:textStyle="bold"
        app:layout_constraintTop_toTopOf="parent"
        tools:layout_editor_absoluteX="13dp" />
         <EditText
        android:id="@+id/email_customer"
        android:layout_width="409dp"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_alignParentEnd="true"
        android:layout_marginStart="12dp"
        android:layout_marginLeft="12dp"
        android:layout_marginTop="134dp"
        android:layout_marginEnd="-2dp"
        android:ems="10"
        android:inputType="textEmailAddress"
        app:layout_constraintStart_toStartOf="parent"
        tools:layout_editor_absoluteY="121dp" />



    <EditText
        android:id="@+id/password_customer"
        android:layout_width="409dp"
        android:layout_height="wrap_content"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:layout_marginStart="0dp"
        android:layout_marginLeft="12dp"
        android:layout_marginTop="248dp"
        android:ems="10"
        android:inputType="textPassword"
        app:layout_constraintStart_toStartOf="parent"
        tools:layout_editor_absoluteY="224dp" />
```

```
 <Button
        android:id="@+id/customer_login_btn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentStart="true"
android:layout_alignParentLeft="true"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:layout_marginStart="142dp"
        android:layout_marginLeft="142dp"
        android:layout_marginBottom="223dp"
        android:text="LOGIN"
        android:textSize="28sp"
        app:backgroundTint="#DD2685"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        tools:layout_editor_absoluteY="316dp" />
    <Button
        android:id="@+id/customer_register_btn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentStart="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentBottom="true"
        android:layout_marginStart="120dp"

        android:layout_marginLeft="120dp"
        android:layout_marginBottom="42dp"
        android:text="REGISTER"
        android:textSize="30sp"
        app:backgroundTint="#9A1E1E"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        tools:layout_editor_absoluteY="509dp" />
<TextView
        android:id="@+id/customer_link"
        android:layout_width="404dp"
        android:layout_height="41dp"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginBottom="149dp"
        android:backgroundTint="#15154A"
        android:text="DON'T HAVE AN ACCOUNT"

android:textSize="28sp"
        android:textStyle="italic"
        tools:layout_editor_absoluteX="6dp"
        tools:layout_editor_absoluteY="427dp" />
</RelativeLayout>
package com.example.mathes;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import android.app.ProgressDialog;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
public class CustomerLoginRegisterActivity extends AppCompatActivity {
    private Button clo;
    private Button creg;
    private TextView cst;
    private TextView clink;
    private EditText cemail;
    private EditText cpass;
    private ProgressDialog loadingbar;
    private FirebaseAuth mAuth;
@Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_customer_login_register);
        mAuth=FirebaseAuth.getInstance();
```