TESTING A BLOCKCHAIN DATABASE BASED APPLICATION (PUBLIC HELP SERVICE)

PROJECT REPORT

Submitted in fulfilment for the J Component of

SWE2005 – SOFTWARE TESTING

CAL COURSE

in

M. Tech (SE)

by

NITHISH KUMAR S (20MIS0024)

MURAHARI CHERMINI (20MIS0044)

VARSHA JHAVER (21MIS0164)

Under the guidance of

Dr. UMA K

SITE

School of Information Technology and Engineering

Fall Semester 2022-2023

TABLE OF CONTENTS

ABSTRACT

‘Public Help Service’ is an initiative to build a better environment for the citizens to live in. It is a website where people can share and complain the problems they face in their locality in a day to day basis. The Public need not visit any government offices to register their complaint. They can just access the website and simply register their complaint. After a complaint has been filed by a public citizen, the authorities concerned visit the locality to resolve the issue. Also, this system provides emergency contact details to the public which they can make use in case of any medical or fire emergency situation with just a single tap in the application with the help of live location tracking. This Project is implemented using Flutter and React (frontend and backend) which are one of the best tools for developing application and web interface. We use ProvenDB (Block chain database) in order to ensure the security of the user data and prevent any personal data breach or data leak issues.

- Decentralization
- Query
- Immutability
- Native Support of Multi assets
- Byzantine Fault Tolerant (BFT)
- Low Latency
- Customizable
- Open Source

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACRONYM	EXPANSION
API	Application Program Interface
RAM	Random Access Memory
CPU	Control Processing Unit
SHA	Secure Hash Algorithm

CHAPTER 1 – INTRODUCTION

1.1 MOTIVATION:

Block chain Database service that combines traditional database capabilities with Block chain characteristics such as immutability. Using it, you can write application which anchor data to a Public or Private Block chain, providing cryptographic proofs of the integrity and history of the data. It is for developers and organizations looking for a query able database with block chain characteristics such as decentralization, immutability and the ability to treat anything stored in the database as an asset. Whether it's atoms, bits or bytes of value, any real-world block chain application needs performance. The immutability of data within a block chain constitutes a revolution in computer science. For the first time, we have a way of storing data which cannot be altered and whose creation date and integrity can be cryptographically proven. Data in a traditional database can be altered at any time by a database administrator or a privileged developer. In contrast, data in a public block chain is immutable. For the first time, we can have absolute mathematical proof of the veracity and provenance of a data item. Block chain cryptographic proofs are increasingly being recognized as legal proofs. We can foresee a day in which block chain records are recognized as being of as high a standard of proof as DNA or fingerprint records. In this research, our motivation is to understand the application and test the application in order to evaluate and verify that the application does what it is supposed to do.

1.2 ISSUES:

Public Block chains– such as ethereum and bitcoin – are capable of only a handful of transactions per second. The cost of storing any non – trivial amount of data on these block chains is absolutely prohibitive. While there are alternative Block chains which have higher throughput and lower latencies, these Block chains do not provide the cryptographic strength of the major public Block chains such as those underlying Ethereum or Bitcoin. Therefore, testing modules with respect to the above mentioned issues will provide better application and user experience.

CHAPTER 2 – PROBLEM DEFINITION

2.1 CRITICAL ISSUES:

Block chain as a technology is just taking off. Notably, most of the solutions that are coming up are novel, and they are mostly technical to the majority of people in the industry. As block chain database is emerging technology, there are rare documentation for implementation and it not fully developed yet for every use case. Data in block chain database is not destroyed when update or delete operations are issued. Old versions of data remain available indefinitely. It might be problem for some industries to maintain the data that large scale.

2.2 PROBLEM DEFINITION:

Block chain database ‘Public Help Service’ is an initiative to build a better environment for a citizen to live in. It is a website/mobile application where people can share, complain about the problems they face in their locality. People need not visit any government offices to register their complaint. People can just open the website/mobile application and simply register their complaint. When the people complain, the respective officials visit the locality to resolve the issue. Also, helps with emergency contact where you can make medical, fire emergency with single tap 12 in application with the help of live location tracking. With data integrity block chain structure makes it virtually impossible for someone to change the data without breaking the chain.

CHAPTER 3 – PROJECT DESCRIPTION

3.1 LIST OF MODULES:

- User Side
    - Login / Register
    - View Dashboard
    - Register Complaint
    - View News
    - View / Edit Profile
    - View About
    - Logout

- Administrator Side
    - Login / Register
    - View / Edit Dashboard
    - View Complaint
    - Add / Edit News
    - View / Edit Profile
    - View / Edit About
    - Logout

3.2 REQUIREMENTS SPECIFICATION:

3.2.1 SOFTWARE REQUIREMENTS:

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the teams and tracking the team’s progress throughout the development activity

- Android 5 and above
- In PC works with Chromium based web browser (Chrome and Edge)

3.2.2 HARDWARE REQUIREMENTS:

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It should what the system do and not how it should be implemented.

- SOC or Dual core CPU with 4 Gigabyte of RAM

3.3 MODULE DESCRIPTION:

- **User Side**
  - **Login / Register:**

    Login is a process that allows user to type a user name and password to log-in and gain access to the app after proper authorization. Registration is the process of signing up in order to collect the data of the user and create a profile along with the password.

  - **View Dashboard:**

    Viewing Dashboard is a process of going through the services provided in the app and selecting the required service to register a complaint.

  - **Register Complaint:**

    Register Complaint is the process of registering a complaint from the user side so that the authorities concerned can take necessary actions regarding the complaint.

  - **View News:**

    View News is the process of viewing the report of recent events in order to update the user about the current scenario.

  - **View / Edit Profile:**

    View Profile is the process of viewing the information of the user. Edit Profile is the process of editing the profile of the user in order to change the information of the user.

- **View About:**

  View About is the process of viewing about the app in order to know about the app.

- **Logout:**

  Logout is the process to terminate the app.

- **Authorities Side**
  - **Login / Register:**

    Login is a process that allows user to type a user name and password to log-in and gain access to the app after proper authorization. Registration is the process of signing up in order to collect the data of the user and create a profile along with the password.

  - **View / Edit Dashboard:**

    Viewing Dashboard is a process of going through the services provided in the app and selecting the required service to register a complaint. Editing Dashboard is a process of editing the services provided in the app.

  - **View Complaint:**

    View Complaint is the process of viewing a complaint from the user side so that the authorities concerned can take necessary actions regarding the complaint.

  - **Add / Edit News:**

    Add News is the process of adding the report of recent events in order to update the user about the current scenario. Edit News is the process of editing the report of recent events in order to update the user about the current scenario

- **View / Edit Profile:**

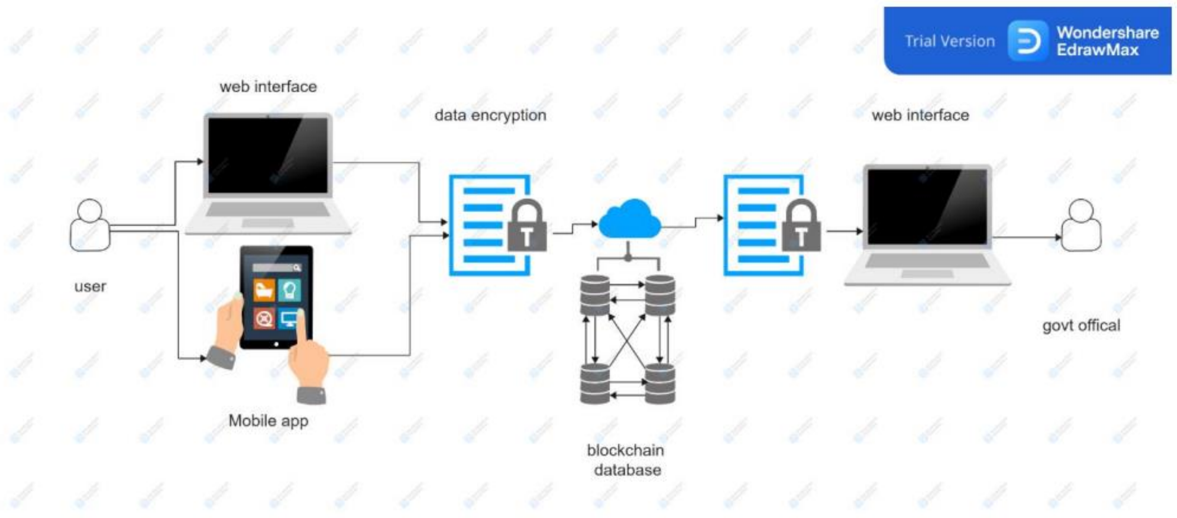
  View Profile is the process of viewing the information of the user. Edit Profile is the process of editing the profile of the user in order to change the information of the user.

- **View / Edit About:**

  View About is the process of viewing about the app in order to know about the app. Edit About is the process of editing the about of the app.

- **Logout:**

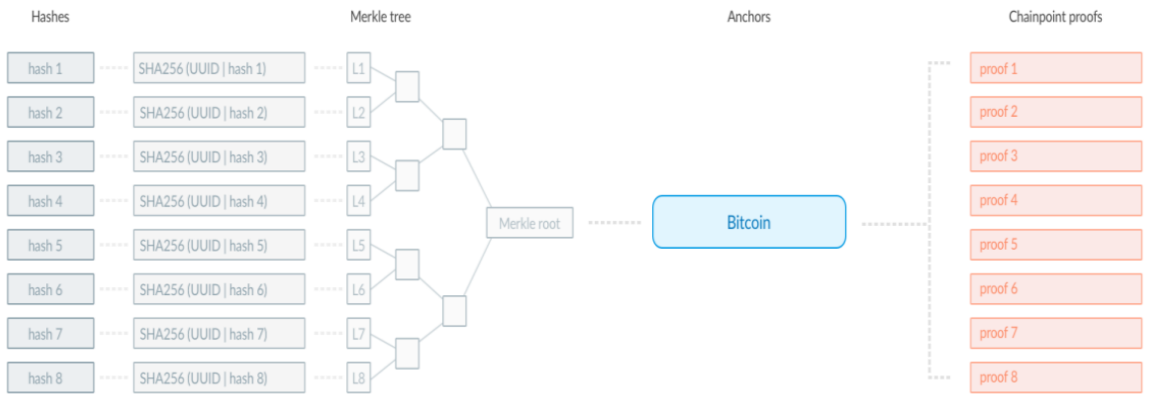
  Logout is the process to terminate the app.



CHAPTER 4 – IMPLEMENTATION

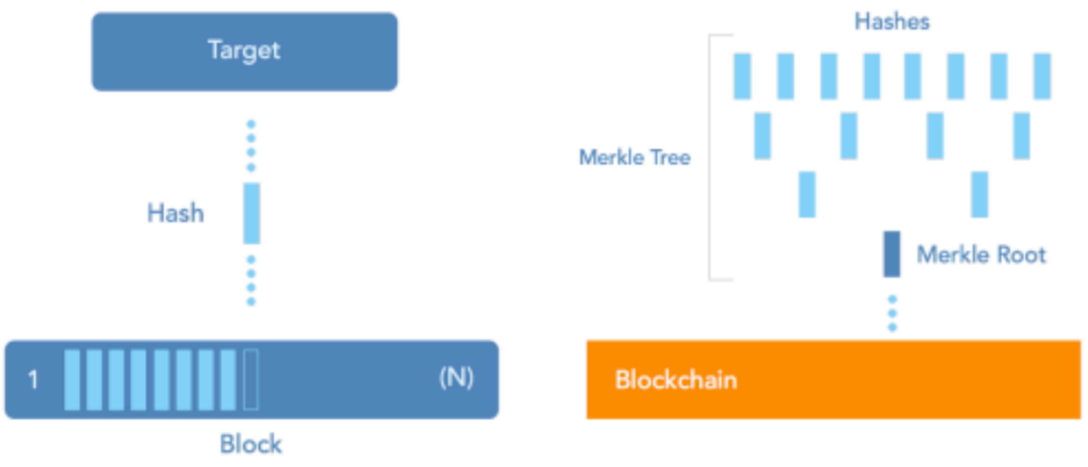
4.1 ALGORITHMS:

CHAINPOINT:



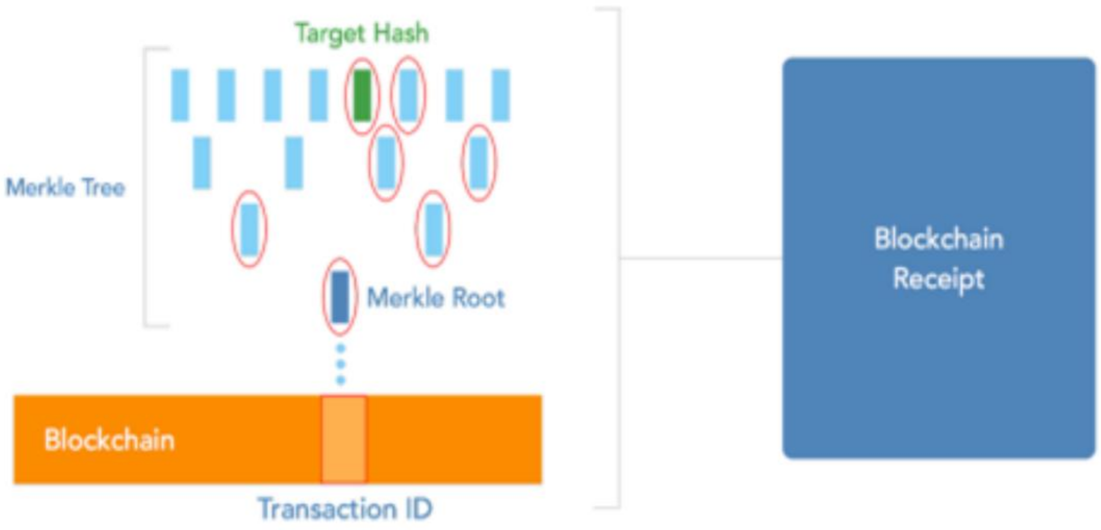
ANCHORING DATA:

To anchor data in the block chain, we start by using a standard hashing function such as SHA-256 to generate a unique hash of the target data. Multiple hashes are assembled into a block, which is simply a list of hashes. Periodically, these blocks are used to generate a Merkle Tree, and the Merkle Root is published in the block chain via a transaction. By collating multiple hashes into a Merkle Tree and publishing the Merkle Root, we can anchor large volumes of data in the block chain using a single transaction.



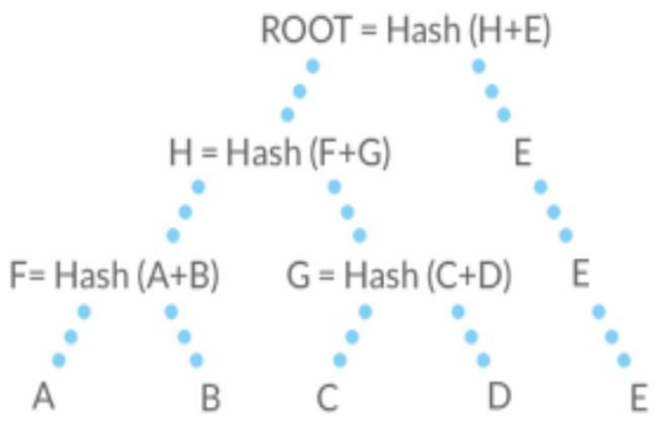
CREATING BLOCKCHAIN RECEIPTS:

A block chain receipt provides proof that some data existed at a specific time. It contains all the information needed to prove an individual hash was part of the Merkle Tree whose root was published in a transaction in the Block chain. By tracing a path from the Merkle root to the target hash, we can generate a Merkle Proof that proves any one of the elements is in the Merkle tree, without having to know the entire tree. These elements can be used to create a block chain receipt that contains, at minimum, the Target Hash, Merkle Proof, Merkle Root, and Transaction ID.



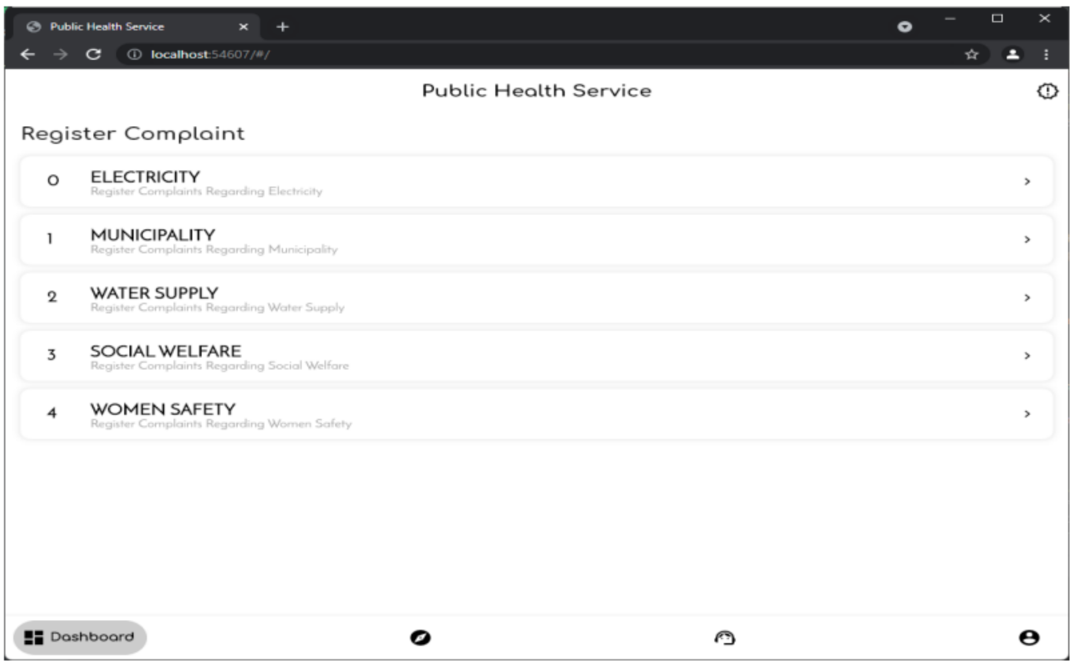
MERKLE TREE CONSTRUCTION:

When constructing merkle trees from which proofs will be generated for chain point receipts, lonely leaf (odd) end nodes on any given level should be promoted up to the next level, as depicted in the following diagram:



Chain point requires that hashes used in building merkle trees and proofs be handled internally in binary form only. When concatenating hashes and hashing those results, do not use the hex strings. The hex string representation of hashes is only used for displaying the hash values within the receipt.

4.2 SCREENSHOTS OF THE PROJECT:



Public Health Service

localhost:54607/#/

Public Health Service

Register Complaint

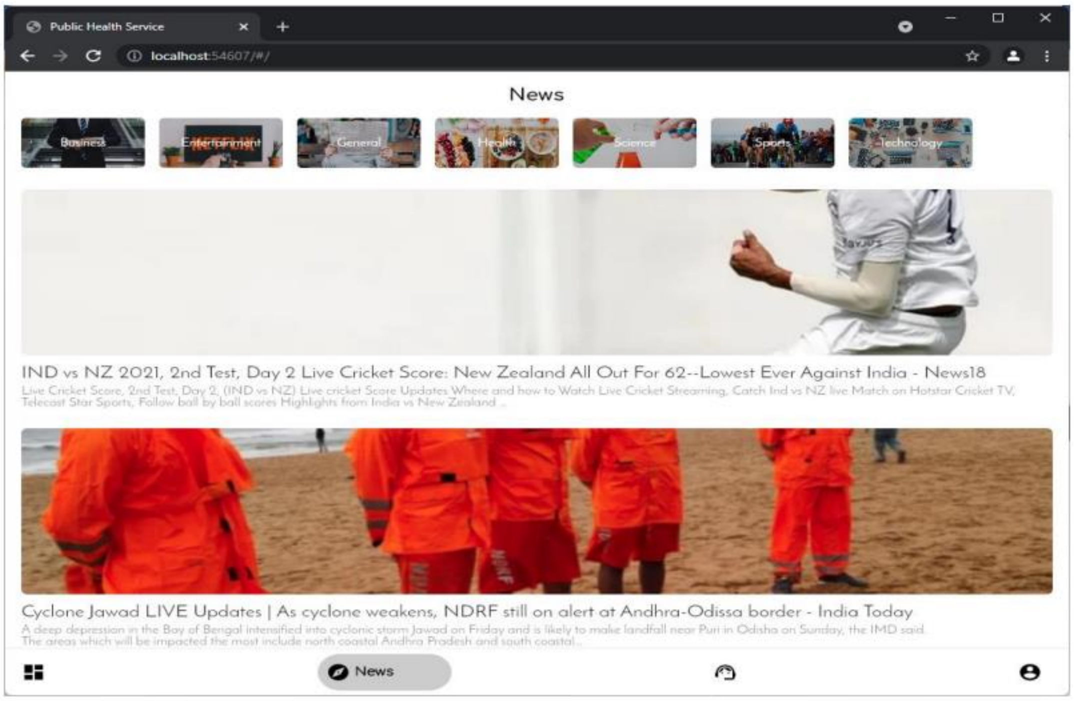
0 ELECTRICITY
Register Complaints Regarding Electricity

1 MUNICIPALITY
Register Complaints Regarding Municipality

2 WATER SUPPLY
Register Complaints Regarding Water Supply

3 SOCIAL WELFARE
Register Complaints Regarding Social Welfare

4 WOMEN SAFETY
Register Complaints Regarding Women Safety

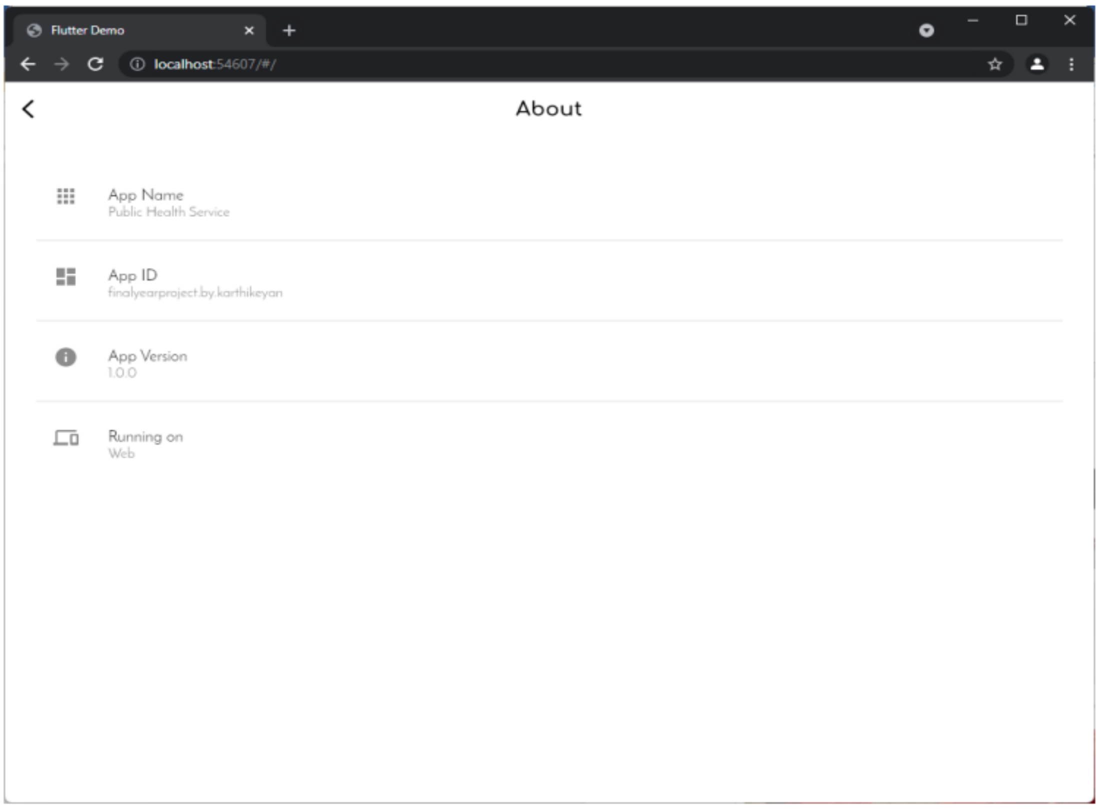


IND vs NZ 2021, 2nd Test, Day 2 Live Cricket Score: New Zealand All Out For 62--Lowest Ever Against India - News18

Live Cricket Score, 2nd Test, Day 2, (IND vs NZ) Live cricket Score Updates Where and how to Watch Live Cricket Streaming, Catch Ind vs NZ live Match on Hotstar Cricket TV, Telecast Star Sports, Follow ball by ball scores Highlights from India vs New Zealand ...

Cyclone Jawad LIVE Updates | As cyclone weakens, NDRF still on alert at Andhra-Odissa border - India Today

A deep depression in the Bay of Bengal intensified into cyclonic storm Jawad on Friday and is likely to make landfall near Puri in Odisha on Sunday, the IMD said. The areas which will be impacted the most include north coastal Andhra Pradesh and south coastal...



App Name
Public Health Service

App ID
finalyearproject.by.karthikeyan

App Version
1.0.0

Running on
Web

4.3 CODE:

DASHBOARDSCREEN.DART

```
import 'package:finalyearproject/UI/AboutScreen/about.dart';
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
class DashboardScreen extends StatefulWidget
{
  @override
  _DashboardScreenState createState() => _DashboardScreenState();
}
class _DashboardScreenState extends State<DashboardScreen>
{
  List chapters = [
    'Electricity',
    'Municipality',
    'Water Supply',
    'Social Welfare',
    'Women Safety',
  ];
  List topics = [
    'Register Complaints Regarding Electricity',
    'Register Complaints Regarding Municipality',
    'Register Complaints Regarding Water Supply',
    'Register Complaints Regarding Social Welfare',
    'Register Complaints Regarding Women Safety',
  ];
  @override
  Widget build(BuildContext context)
  {
    return Scaffold(
      backgroundColor: Colors.white,
      appBar: AppBar(
        actions: <Widget>[
        IconButton(
          onPressed: ()
          {
            Navigator.of(context).push(MaterialPageRoute(builder: (context)
=> AboutScreen()));
          },
          icon: Icon(Icons.new_releases_outlined, color: Colors.black,)
        ),
        ],
        automaticallyImplyLeading: false,
        centerTitle: true,
        title: Text('Public Health Service', style:
GoogleFonts.comfortaa(color: Colors.black, fontWeight: FontWeight.bold)),
        backgroundColor: Colors.white,
        iconTheme: IconThemeData(color: Colors.black),
        elevation: 0,
      ),
      body: Padding(
        padding: const EdgeInsets.all(15.0),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: <Widget>[
          Text(
            'Register Complaint',
            style: GoogleFonts.comfortaa(
              fontSize: 22,
              fontWeight: FontWeight.bold,
            ),
          ),
          Padding(
            padding: EdgeInsets.only(top: 15),
          ),
          Expanded(
            child: ListView.builder(
              itemCount: chapters.length,
              shrinkWrap: true,
              itemBuilder: (BuildContext context, int index) {
                return Container(
                  margin: EdgeInsets.only(bottom: 10),
                  decoration: BoxDecoration(
                    color: Colors.white,
                    borderRadius: BorderRadius.circular(10),
                    boxShadow: [
                    BoxShadow(
                      color: Colors.grey[300],
                      offset: Offset(0, 0),
                      blurRadius: 5,
                    ),
                    ],
                  ),
                  child: ListTile(
                    leading: Container(
                      padding: EdgeInsets.fromLTRB(10, 10, 0, 10),
                      child: Text(
                        '$index',
                        style: TextStyle(
                          color: Colors.black,
                          fontSize: 18,
                          fontWeight: FontWeight.bold,
                        ),
                      ),
                    ),
                    title: Text(
                      chapters[index].toString().toUpperCase(),
                      style: TextStyle(
                        color: Colors.black,
                        fontSize: 18,
                        fontWeight: FontWeight.bold,
                      ),
                    ),
                    subtitle: Text(
                      topics[index],
                      style: TextStyle(
                        color: Colors.grey,
                        fontSize: 14,
                      ),
                    ),
                    trailing: Icon(
                      Icons.chevron_right,
                      size: 18,
                      color: Colors.black,
                    ),
                  ),
                );
              },
            ),
          ),
          ],
        ),
      ),
    );
  }
}
```

NEWSSCREEN.DART

```
import 'package:cached_network_image/cached_network_image.dart';
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'categorie_news.dart';
import 'helper/data.dart';
import 'helper/news.dart';
import 'helper/widgets.dart';
import 'models/categorie_model.dart';
class NewsScreen extends StatefulWidget {
  @override
  _NewsScreenState createState() => _NewsScreenState();
}
class _NewsScreenState extends State<NewsScreen> {
  bool _loading;
  var newslist;
  List<CategorieModel> categories = List<CategorieModel>();
  void getNews() async {
    News news = News();
    await news.getNews();
    newslist = news.news;
    setState(() {
      _loading = false;
    });
  }
  @override
  void initState() {
    _loading = true;
    // TODO: implement initState
    super.initState();
    categories = getCategories();
    getNews();
  }
  @override
  Widget build(BuildContext context) {
    return Scaffold(
        backgroundColor: Colors.white,
        appBar: AppBar(
        automaticallyImplyLeading: false,
        centerTitle: true,
        title: Text('News', style: GoogleFonts.comfortaa(color:
Colors.black, fontWeight:
        FontWeight.bold)),
    backgroundColor: Colors.white,
    iconTheme: IconThemeData(color: Colors.black),
    elevation: 0,
    ),
      body: SafeArea(
          child: _loading
              ? Center(
            child: CircularProgressIndicator(),
          )
              : SingleChildScrollView(
          child: Container(
          child: Column(
          children: <Widget>[
      /// Categories
      Container(
      padding: EdgeInsets.symmetric(horizontal: 16),
      height: 70,
      child: ListView.builder(
          scrollDirection: Axis.horizontal,
          itemCount: categories.length,
          itemBuilder: (context, index) {
            return CategoryCard(
              imageAssetUrl: categories[index].imageAssetUrl,
              categoryName: categories[index].categorieName,
            );
          }),
    ),
            /// News Article
            Container(
              margin: EdgeInsets.only(top: 16),
              child: ListView.builder(
                  itemCount: newslist.length,
                  shrinkWrap: true,
                  physics: ClampingScrollPhysics(),
                  itemBuilder: (context, index) {
                    return NewsTile(
                      imgUrl: newslist[index].urlToImage ?? "",
                      title: newslist[index].title ?? "",
                      desc: newslist[index].description ?? "",
                      content: newslist[index].content ?? "",
                      posturl: newslist[index].articleUrl ?? "",
                    );
                  }),
            ),
          ],
          ),
          ),
          ),
      ),
    );
  }
}
class CategoryCard extends StatelessWidget {
  final String imageAssetUrl, categoryName;
  CategoryCard({this.imageAssetUrl, this.categoryName});
  @override
  Widget build(BuildContext context) {
    return GestureDetector(
        onTap: (){
          Navigator.push(context, MaterialPageRoute(
              builder: (context) => CategoryNews(
                newsCategory: categoryName.toLowerCase(),
              )
          ));
        },
        child: Container(
        margin: EdgeInsets.only(right: 14),
    child: Stack(
    children: <Widget>[
    ClipRRect(
    borderRadius: BorderRadius.circular(5),
    child: CachedNetworkImage(
      imageUrl: imageAssetUrl,
      height: 60,
      width: 120,
      fit: BoxFit.cover,
    ),
    ),
    Container(
    alignment: Alignment.center,
    height: 60,
    width: 120,
    decoration: BoxDecoration(
    borderRadius: BorderRadius.circular(5),
    color: Colors.black26
    ),
    child: Text(
    categoryName,
    textAlign: TextAlign.center,
    style: TextStyle(
    color: Colors.white,
    fontSize: 14,
    fontWeight: FontWeight.w500),
    ),
    )
    ],
    ),
        ),
    );
  }
}
```

PROFILE.DART

```
import 'package:flutter/material.dart';
import 'package:flutter/cupertino.dart';
class ProfileScreen extends StatefulWidget {
  @override
  ProfileScreenState createState() => ProfileScreenState();
}
class ProfileScreenState extends State<ProfileScreen>
    with SingleTickerProviderStateMixin {
  bool _status = true;
  final FocusNode myFocusNode = FocusNode();
  @override
  void initState() {
    // TODO: implement initState
    super.initState();
  }
  @override
  Widget build(BuildContext context) {
    return new Scaffold(
        appBar: AppBar(
        automaticallyImplyLeading: false,
        backgroundColor: Colors.white,
        elevation: 0.0,
        title: Text(
        'Profile',
        style: TextStyle(
        fontWeight: FontWeight.bold,
        color: Colors.blue
    ),
    ),
    ),
    body: new Container(
    color: Colors.white,
    child: new ListView(
    children: <Widget>[
    Column(
    children: <Widget>[
    new Container(
    height: 250.0,
    color: Colors.white,
    child: new Column(
    children: <Widget>[
    Padding(
    padding: EdgeInsets.only(top: 20.0),
    child: new Stack(fit: StackFit.loose, children: <Widget>[
    new Row(
    crossAxisAlignment: CrossAxisAlignment.center,
    mainAxisAlignment: MainAxisAlignment.center,
    children: <Widget>[
    new Container(
    width: 140.0,
    height: 140.0,
    decoration: new BoxDecoration(
    shape: BoxShape.circle,
    image: new DecorationImage(
    image: new ExactAssetImage(
    'assets/images/as.png'),
    fit: BoxFit.cover,
    ),
    )),
    ],
    ),
    Padding(
    padding: EdgeInsets.only(top: 90.0, right: 100.0),
    child: new Row(
    mainAxisAlignment: MainAxisAlignment.center,
    children: <Widget>[
    new CircleAvatar(
    backgroundColor: Colors.red,
    radius: 25.0,
    child: new Icon(
    Icons.camera_alt,
    color: Colors.white,
    ),
    )
    ],
    )),
    ]),
    )
    ],
    ),
    ),
    new Container(
    color: Color(0xffFFFFFF),
    child: Padding(
    padding: EdgeInsets.only(bottom: 25.0),
    child: new Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    mainAxisAlignment: MainAxisAlignment.start,
    children: <Widget>[
    Padding(
    padding: EdgeInsets.only(
    left: 25.0, right: 25.0, top: 25.0),
    child: new Row(
    mainAxisAlignment: MainAxisAlignment.spaceBetween,
    mainAxisSize: MainAxisSize.max,
    children: <Widget>[
    new Column(
    mainAxisAlignment: MainAxisAlignment.start,
    mainAxisSize: MainAxisSize.min,
    children: <Widget>[
    new Text(
    'Parsonal Information',
    style: TextStyle(
    fontSize: 18.0,
    fontWeight: FontWeight.bold),
    ),
    ],
    ),
    new Column(
    mainAxisAlignment: MainAxisAlignment.end,
    mainAxisSize: MainAxisSize.min,
    children: <Widget>[
    _status ? _getEditIcon() : new Container(),
    ],
    )
    ],
    )),
    Padding(
    padding: EdgeInsets.only(
    left: 25.0, right: 25.0, top: 25.0),
    child: new Row(
    mainAxisSize: MainAxisSize.max,
    children: <Widget>[
    new Column(
    mainAxisAlignment: MainAxisAlignment.start,
    mainAxisSize: MainAxisSize.min,
    children: <Widget>[
    new Text(
    'Name',
    style: TextStyle(
    fontSize: 16.0,
    fontWeight: FontWeight.bold),
    ),
    ],
    ),
    ],
    )),
    Padding(
    padding: EdgeInsets.only(
    left: 25.0, right: 25.0, top: 2.0),
    child: new Row(
    mainAxisSize: MainAxisSize.max,
    children: <Widget>[
    new Flexible(
    child: new TextField(
    decoration: const InputDecoration(
    hintText: "Enter Your Name",
    ),
    enabled: !_status,
    autofocus: !_status,
    ),
    ),
    ],
    )),
    Padding(
    padding: EdgeInsets.only(
    left: 25.0, right: 25.0, top: 25.0),
    child: new Row(
    mainAxisSize: MainAxisSize.max,
    children: <Widget>[
    new Column(
    mainAxisAlignment: MainAxisAlignment.start,
    mainAxisSize: MainAxisSize.min,
    children: <Widget>[
    new Text(
    'Email ID',
    style: TextStyle(
    fontSize: 16.0,
    fontWeight: FontWeight.bold),
    ),
    ],
    ),
    ],
    )),
    Padding(
    padding: EdgeInsets.only(
    left: 25.0, right: 25.0, top: 2.0),
    child: new Row(
    mainAxisSize: MainAxisSize.max,
    children: <Widget>[
    new Flexible(
    child: new TextField(
    decoration: const InputDecoration(
    hintText: "Enter Email ID"),
    enabled: !_status,
    ),
    ),
    ],
    )),
    Padding(
    padding: EdgeInsets.only(
    left: 25.0, right: 25.0, top: 25.0),
    child: new Row(
    mainAxisSize: MainAxisSize.max,
    children: <Widget>[
    new Column(
    mainAxisAlignment: MainAxisAlignment.start,
    mainAxisSize: MainAxisSize.min,
    children: <Widget>[
    new Text(
    'Mobile',
    style: TextStyle(
    fontSize: 16.0,
    fontWeight: FontWeight.bold),
    ),
    ],
    ),
    ],
    )),
    Padding(
    padding: EdgeInsets.only(
    left: 25.0, right: 25.0, top: 2.0),
    child: new Row(
    mainAxisSize: MainAxisSize.max,
    children: <Widget>[
    new Flexible(
    child: new TextField(
    decoration: const InputDecoration(
    hintText: "Enter Mobile Number"),
    enabled: !_status,
    ),
    ),
    ],
    )),
    Padding(
    padding: EdgeInsets.only(
    left: 25.0, right: 25.0, top: 25.0),
    child: new Row(
    mainAxisSize: MainAxisSize.max,
    mainAxisAlignment: MainAxisAlignment.start,
    children: <Widget>[
    Expanded(
    child: Container(
    child: new Text(
    'Pin Code',
    style: TextStyle(
    fontSize: 16.0,
    fontWeight: FontWeight.bold),
    ),
    ),
    flex: 2,
    ),
    Expanded(
    child: Container(
    child: new Text(
    'State',
    style: TextStyle(
    fontSize: 16.0,
    fontWeight: FontWeight.bold),
    ),
    ),
    flex: 2,
    ),
    ],
    )),
    Padding(
    padding: EdgeInsets.only(
    left: 25.0, right: 25.0, top: 2.0),
    child: new Row(
    mainAxisSize: MainAxisSize.max,
    mainAxisAlignment: MainAxisAlignment.start,
    children: <Widget>[
    Flexible(
    child: Padding(
    padding: EdgeInsets.only(right: 10.0),
    child: new TextField(
    decoration: const InputDecoration(
    hintText: "Enter Pin Code"),
    enabled: !_status,
    ),
    ),
    flex: 2,
    ),
    Flexible(
    child: new TextField(
    decoration: const InputDecoration(
    hintText: "Enter State"),
```

ABOUTSCREEN.DART

```
import 'dart:async';
import 'package:flutter/material.dart';
// IMPORTING CROSS PLATFORM
import 'package:cross_platform/cross_platform.dart';
// IMPORTING GOOGLE FONTS
import 'package:google_fonts/google_fonts.dart';
class AboutScreen extends StatefulWidget
{
  @override
  _AboutScreenState createState() => _AboutScreenState();
}
class _AboutScreenState extends State<AboutScreen>
{
  Widget _buildContent()
  {
    if (Platform.isWeb)
    {
      return Text('Web');
    }
    else if (Platform.isAndroid)
    {
      return Text('Android');
    }
    else if (Platform.isIOS)
    {
      return Text('iOS');
    }
    else if (Platform.isMacOS)
    {
      return Text('MacOS');
    }
    else if (Platform.isLinux)
    {
      return Text('Linux');
    }
    else if (Platform.isWindows)
    {
      return Text('Windows');
    }
    else if (Platform.isFuchsia)
    {
      return Text('Fuchsia');
    }
    else
    {
      return Text('Undefined');
    }
  }
  @override
```

```
Widget build(BuildContext context)
{
  return Scaffold(
    backgroundColor: Colors.white,
    appBar: AppBar(
      backgroundColor: Colors.white,
      centerTitle: true,
      elevation: 0.0,
      leading: IconButton(
        icon: Icon(Icons.arrow_back_ios, color: Colors.black,),
        onPressed: ()
        {
          Navigator.pop(context);
        },
      ),
      title: Text(
        "About",
        style: GoogleFonts.comfortaa(color: Colors.black, fontWeight:
        FontWeight.bold),
      ),
    ),
    body: Container(
      child: Column(
        children: [
        Container(
          padding: EdgeInsets.fromLTRB(30, 25, 30, 30),
          alignment: Alignment.centerLeft,
          child: new ListBody(
            children: <Widget>[
            new Container(height: 10.0,),
            new ListTile(
              leading: const Icon(Icons.apps),
              title: const Text('App Name'),
              subtitle: new Text("Public Health Service"),
            ),
            const Divider(height: 20.0),
            new ListTile(
              leading: const Icon(Icons.dashboard),
              title: const Text('App ID'),
              subtitle: new Text("finalyearproject.by.karthikeyan"),
            ),
            const Divider(height: 20.0),
            new ListTile(
              leading: const Icon(Icons.info),
              title: const Text('App Version'),
              subtitle: new Text("1.0.0"),
            ),
            const Divider(height: 20.0),
            new ListTile(
              leading: const Icon(Icons.devices),
              title: const Text('Running on'),
              subtitle: _buildContent(),
            ),
            ],
          ),
        ),
        ],
      ),
    ),
  );
```

CHAPTER 5 – TESTING

SOFTWARE TESTING:

Software testing is the process of evaluating and verifying that a software product or application does what it is supposed to do. The benefits of testing include preventing bugs, reducing development costs and improving performance. Even a simple application can be subject to a large number and variety of tests. A test management plan helps to prioritize which types of testing provide the most value – given available time and resources. Testing effectiveness is optimized by running the fewest number of tests to find the largest number of defects. Doing test activities earlier in the cycle helps keep the testing effort at the forefront rather than as an afterthought to development. Earlier software tests also mean that defects are less expensive to resolve.

UNIT TESTING:

Unit testing is the testing of each module and the integration of the overall system is done. Unit testing becomes verification efforts on the smallest unit of software design in the module. This is also known as 'module testing'. The modules of the system are tested separately. This testing is carried out during the programming itself. In this testing step, each model is found to be working satisfactorily as regard to the expected output from the module. There are some validation checks for the fields. For example, the validation check is done for verifying the data given by the user where both format and validity of the data entered is included. It is very easy to find error and debug the system.

CODE:

```
from tabulate import tabulate
import numpy as np

username_database = ['20MIS0001', '20MIS0002', '20MIS0003', '20MIS0004'
, '20MIS0005', '20MIS0006', '20MIS0007', '20MIS0008', '20MIS0009', '20M
IS0010']
password_database = ['qwertyqwe', 'asdfghjkl', 'zxcvbnmjk', 'mkijnhujk'
, 'gfychdnjs', 'oiuergfhj', 'jdufbeehj', 'sjdfuehew', 'sjdhaicji', 'skd
isnbah']
testcase = ["TC01", "TC02", "TC03", "TC04", "TC05", "TC06", "TC07", "TC
08", "TC09", "TC10", "TC11", "TC12", "TC13", "TC14", "TC15", "TC16", "T
C17", "TC18", "TC19", "TC20", "TC21", "TC22"]
username_testcase = ['20MIS0001', '20MIS0002', '20MIS0003', '20MIS0004'
, '20MIS0005', '20MIS0006', '20MIS0007', '20MIS0008', '20MIS0009', '20M
IS0010', '20MIS0011', '20MIS0012', '20MIS0013', '20MIS0014', '20MIS0015
', '20MIS0016', '20MIS0017', '20MIS0018', '20MIS0019', '20MIS0020', '20
MIS0001', '']
password_testcase = ['qwertyqwe', 'asdfghjkl', 'zxcvbnmjk', 'mkijnhujk'
, 'gfychdnjs', '123456789', '789654123', '456852364', '753951458', '596
321448', 'gfychdnjs', 'oiuergfhj', 'jdufbeehj', 'sjdfuehew', 'sjdhaicji
', '123456789', '789654123', '456852364', '753951458', '596321448', '',
 '']
expected_output = ['Login Successful', 'Login Successful', 'Login Succe
ssful', 'Login Successful', 'Login Successful', 'Invalid Password', 'In
valid Password', 'Invalid Password', 'Invalid Password', 'Invalid Passw
ord', 'Invalid Username', 'Invalid Username', 'Invalid Username', 'Inva
lid Username', 'Invalid Username', 'Invalid Username', 'Invalid Usernam
e', 'Invalid Username', 'Invalid Username', 'Invalid Username', 'Enter 
Password', 'Enter Username']

#print("\t\t\t\t\t\t\tWelcome to Public Help Service\n")
#print("\t\t\t\t\t\t\t\tLogin Page")
print("\t\t\t\t\t\t\t\t\tSoftware Testing\n")
print("\t\t\t\t\t\t\t\t\tSWE2005\n")
print("\t\t\t\t\t\t\t\t\tJ - Component\n")
print("\t\t\t\t\t\t\t\tThis is the Login Module\n")
print("\t\t\t\t\t\t\t\t\tUnit Testing\n")
actual_output = []
count = 0
for i in range(len(testcase)):
  username_flag = False
  username = username_testcase[i]
  #print("\nEnter Username:", username)
  try:
    if(username not in username_database or len(username) != 9 or type(
int(username[:2])) != int or type(username[2:5]) != str or type(int(use
rname[5:])) != int):
      username_flag = True
  except:
    username_flag = True

  if(username == ''):
    #print("Enter Username")
    actual_output.append('Enter Username')
  elif(username_flag):
    #print("Invalid Username")
    actual_output.append('Invalid Username')
  else:
    password = password_testcase[i]
    #print("Enter Password:", password)
    index = username_database.index(username)
    if(password == ''):
      #print("Enter Password")
```

```
          actual_output.append('Enter Password')
        elif(password != password_database[index]):
          #print("Invalid Password")
          actual_output.append('Invalid Password')
        else:
          #print("Login Successful")
          actual_output.append('Login Successful')

data = np.array([testcase, username_testcase, password_testcase, expect
ed_output, actual_output])
data = data.T
print(tabulate(data, headers = ["Testcase ID", "Username Testcase", "Pa
ssword Testcase", "Expected Output", "Actual Output"]))

for i in range(len(testcase)):
  if(data[i][3] == data[i][4]):
    count += 1

print("\nNumber of Total Testcases:", len(testcase))
print("Number of Testcases Passed:", count)
print("Number of Testcases Failed:", len(testcase) - count)
```

MUTATION TESTING:

Mutation Testing is a type of software testing in which certain statements of the source code are changed/mutated to check if the test cases are able to find errors in source code. The goal of Mutation Testing is ensuring the quality of test cases in terms of robustness that it should fail the mutated source code. The changes made in the mutant program should be kept extremely small that it does not affect the overall objective of the program. Mutation Testing is also called Fault-based testing strategy as it involves creating a fault in the program and it is a type of White Box Testing which is mainly used for Unit Testing.

CODE:

```
from tabulate import tabulate
import numpy as np

username_database = ['20MIS0001', '20MIS0002', '20MIS0003', '20MIS0004'
, '20MIS0005', '20MIS0006', '20MIS0007', '20MIS0008', '20MIS0009', '20M
IS0010']
password_database = ['qwertyqwe', 'asdfghjkl', 'zxcvbnmjk', 'mkijnhujk'
, 'gfychdnjs', 'oiuergfhj', 'jdufbeehj', 'sjdfuehew', 'sjdhaicji', 'skd
isnbah']
```

```
testcase = ["TC01", "TC02", "TC03", "TC04", "TC05", "TC06", "TC07", "TC
08", "TC09", "TC10", "TC11", "TC12", "TC13", "TC14", "TC15", "TC16", "T
C17", "TC18", "TC19", "TC20", "TC21", "TC22"]
username_testcase = ['20MIS0001', '20MIS0002', '20MIS0003', '20MIS0004'
, '20MIS0005', '20MIS0006', '20MIS0007', '20MIS0008', '20MIS0009', '20M
IS0010', '20MIS0011', '20MIS0012', '20MIS0013', '20MIS0014', '20MIS0015
', '20MIS0016', '20MIS0017', '20MIS0018', '20MIS0019', '20MIS0020', '20
MIS0001', '']
password_testcase = ['qwertyqwe', 'asdfghjkl', 'zxcvbnmjk', 'mkijnhujk'
, 'gfychdnjs', '123456789', '789654123', '456852364', '753951458', '596
321448', 'gfychdnjs', 'oiuergfhj', 'jdufbeehj', 'sjdfuehew', 'sjdhaicji
', '123456789', '789654123', '456852364', '753951458', '596321448', '',
 '']
expected_output = ['Login Successful', 'Login Successful', 'Login Succe
ssful', 'Login Successful', 'Login Successful', 'Invalid Password', 'In
valid Password', 'Invalid Password', 'Invalid Password', 'Invalid Passw
ord', 'Invalid Username', 'Invalid Username', 'Invalid Username', 'Inva
lid Username', 'Invalid Username', 'Invalid Username', 'Invalid Usernam
e', 'Invalid Username', 'Invalid Username', 'Invalid Username', 'Enter
Password', 'Enter Username']

#print("\t\t\t\t\t\t\tWelcome to Public Help Service\n")
#print("\t\t\t\t\t\t\t\tLogin Page")
print("\t\t\t\t\t\t\t\tSoftware Testing\n")
print("\t\t\t\t\t\t\t\tSWE2005\n")
print("\t\t\t\t\t\t\t\tJ - Component\n")
print("\t\t\t\t\t\t\tThis is the Login Module\n")
print("\t\t\t\t\t\t\t\tMutation Testing\n")
actual_output = []
count = 0
for i in range(len(testcase)):
  username_flag = False
  username = username_testcase[i]
  #print("\nEnter Username:", username)
  try:
    if(username not in username_database or len(username) != 9 or type(
int(username[:2])) != int or type(username[2:5]) != str or type(int(use
rname[5:])) != int):
      username_flag = True
  except:
    username_flag = True
  if(username_flag):
    #print("Invalid Username")
    actual_output.append('Invalid Username')
  elif(not username_flag):
    password = password_testcase[i]
    #print("Enter Password:", password)
    index = username_database.index(username)
    if(password != password_database[index]):
      #print("Invalid Password")
      #actual_output.append('Invalid Password')
      pass
    #print("Login Successful")
    actual_output.append('Login Successful')

data = np.array([testcase, username_testcase, password_testcase, expect
ed_output, actual_output])
data = data.T
print(tabulate(data, headers = ["Testcase ID", "Username Testcase", "Pa
ssword Testcase", "Expected Output", "Actual Output"]))

for i in range(len(testcase)):
  if(data[i][3] == data[i][4]):
    count += 1

print("\nNumber of Total Testcases:", len(testcase))
print("Number of Testcases Passed:", count)
print("Number of Testcases Failed:", len(testcase) - count)
```

TEST PLAN

Project Number:	PC0024
Project Name:	Block chain Database Based Application (Public Help Service)
Description:	‘Public Help Service’ is an initiative to build a better environment for a citizen to live in. It is a website/mobile application where people can share, complain about the problems they face in their locality.
Project Manager:	Nithish Kumar
Date Updated:	18 Nov 2022

Project Test Plan

Overview	
Test plan objectives	To ensure that the Block Chain Database Based Application (Public Help Service) will: - Meet or exceed user requirements and technical specifications. - Not adversely impact other systems or the existing technology environment.
Testing Assumptions	- Valid and Invalid cases for particular module. - Login Module: - Invalid Username, Valid Password - Invalid Password, Valid Username - Invalid Username, Invalid Password - Null Username, Valid/Invalid Password - Valid Username, Null password
Risks & Contingencies	The following risks apply to the testing process and may impact either the proposed date of readiness for the deployment of Block Chain Database Based Application (Public Help Service), or the comprehensive level of testing that can be performed in each of the Functional Units: - Data in block chain database is not destroyed when update or delete operations are issued. Old versions of data remain available

	indefinitely. It might be problem for some industries to maintain the data that large scale. Password and Username to Login.

Test Scope	
Features to be Tested	All features, forms, reports and interfaces affected by the Block Chain Database Based Application (Public Help Service) will be tested. These include: - Login - Register Complaint - News - Logout - Database used for store the complaints of the users.
Features Not to be Tested	- System functionality - scope of this project.

Test Methodologies	
Testing Approach	- The following approach will be used to test the Block Chain Database Based Application (Public Help Service) Unit testing will be conducted to provide an initial stable testing environment as follows: - For testing the functional specification, the unit testing isperformed by the developer. - Once user sign-off is received, then the actual deployment isperformed. - Mutation Testing is for testing the Application quality.
Test Documents	The following test documents will be created and maintained throughout the project lifecycle: - Block chain Database Based Application (Public Help Service) Test Plan - Master test case lists for each of the following functional units: user side and authorities side. - Test case scripts for each test case recorded in the master test case list

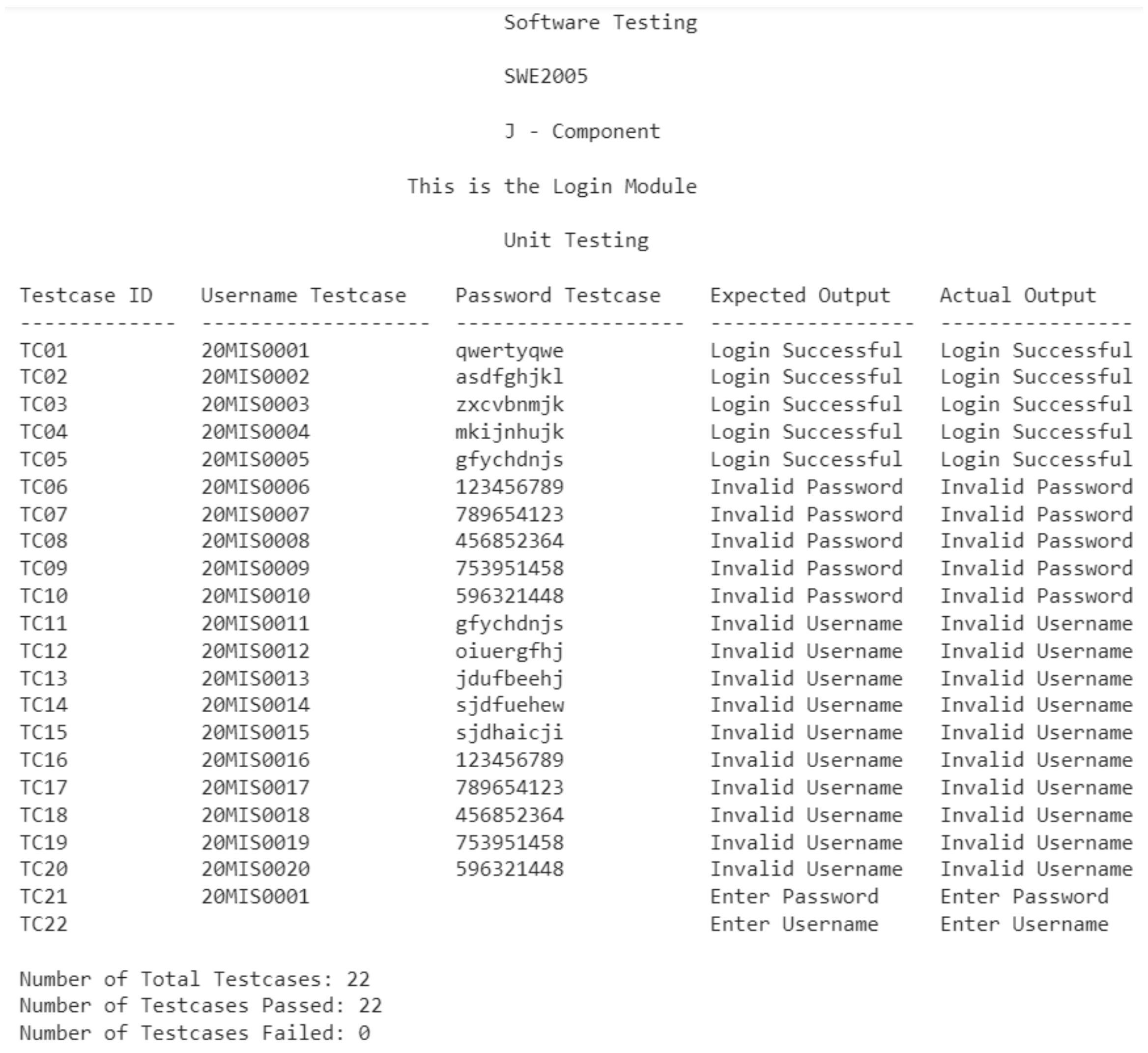
	- Log of all problems encountered during the testing phase of the project
Test Case Pass/Fail Criteria	Each Test Case will be evaluated against the acceptance criteria as outlined in the test case scripts to determine if the test passed or failed. In the case of a failure, the tester will assign a severity to the problem using the appropriate priority rating system established within Tracker for each application.
Suspension/Resumption Criteria	Test Cases that do not run to completion will be evaluated on a case by case basis to determine if the testing must start over or resume at the point where the failure occurred. In extremely long test cases, checkpoints will be established for resumption in the middle of a test case where appropriate. In general, a test may be resumed in the middle when the error is not critical.
Problem Logging/Resolution	Errors identified through testing will be logged. Resolve the problem according to the deemed severity level, and update the master test case list. Once the problem has been fixed, record the resolution into the database. The failed test case will then be retested using the same test case script that detected the error in order to verify that the problem has been rectified.

Resources	
Environmental Needs	In order to conduct comprehensive end-to-end system and user testing are the networked system where the functional and non-functional testing is performed.
Staffing Requirements	Developer

CHAPTER 6 – RESULTS AND DISCUSSIONS

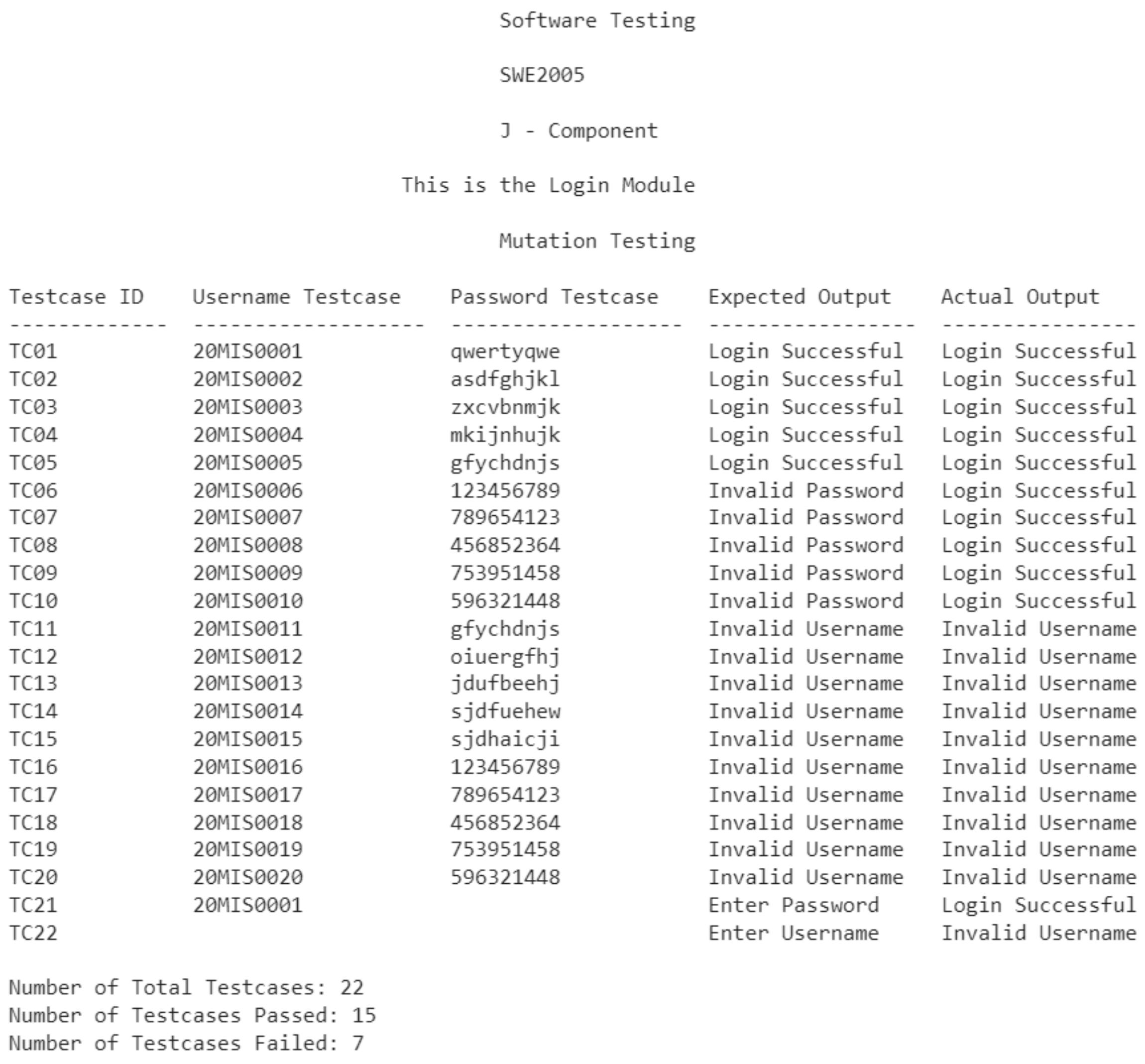
RESULTS:

UNIT TESTING:



```
                                Software Testing

                                SWE2005

                                J - Component

                        This is the Login Module

                                Unit Testing

Testcase ID    Username Testcase    Password Testcase    Expected Output    Actual Output
-------------  -------------------  -------------------  -----------------  ----------------
TC01           20MIS0001            qwertyqwe            Login Successful   Login Successful
TC02           20MIS0002            asdfghjkl            Login Successful   Login Successful
TC03           20MIS0003            zxcvbnmjk            Login Successful   Login Successful
TC04           20MIS0004            mkijnhujk            Login Successful   Login Successful
TC05           20MIS0005            gfychdnjs            Login Successful   Login Successful
TC06           20MIS0006            123456789            Invalid Password   Invalid Password
TC07           20MIS0007            789654123            Invalid Password   Invalid Password
TC08           20MIS0008            456852364            Invalid Password   Invalid Password
TC09           20MIS0009            753951458            Invalid Password   Invalid Password
TC10           20MIS0010            596321448            Invalid Password   Invalid Password
TC11           20MIS0011            gfychdnjs            Invalid Username   Invalid Username
TC12           20MIS0012            oiuergfhj            Invalid Username   Invalid Username
TC13           20MIS0013            jdufbeehj            Invalid Username   Invalid Username
TC14           20MIS0014            sjdfuehew            Invalid Username   Invalid Username
TC15           20MIS0015            sjdhaicji            Invalid Username   Invalid Username
TC16           20MIS0016            123456789            Invalid Username   Invalid Username
TC17           20MIS0017            789654123            Invalid Username   Invalid Username
TC18           20MIS0018            456852364            Invalid Username   Invalid Username
TC19           20MIS0019            753951458            Invalid Username   Invalid Username
TC20           20MIS0020            596321448            Invalid Username   Invalid Username
TC21           20MIS0001                                 Enter Password     Enter Password
TC22                                                     Enter Username     Enter Username

Number of Total Testcases: 22
Number of Testcases Passed: 22
Number of Testcases Failed: 0
```

MUTATION TESTING:



```
                                        Software Testing

                                        SWE2005

                                        J - Component

                                This is the Login Module

                                        Mutation Testing

Testcase ID    Username Testcase    Password Testcase    Expected Output    Actual Output
-------------  -------------------  -------------------  -----------------  ----------------
TC01           20MIS0001            qwertyqwe            Login Successful   Login Successful
TC02           20MIS0002            asdfghjkl            Login Successful   Login Successful
TC03           20MIS0003            zxcvbnmjk            Login Successful   Login Successful
TC04           20MIS0004            mkijnhujk            Login Successful   Login Successful
TC05           20MIS0005            gfychdnjs            Login Successful   Login Successful
TC06           20MIS0006            123456789            Invalid Password   Login Successful
TC07           20MIS0007            789654123            Invalid Password   Login Successful
TC08           20MIS0008            456852364            Invalid Password   Login Successful
TC09           20MIS0009            753951458            Invalid Password   Login Successful
TC10           20MIS0010            596321448            Invalid Password   Login Successful
TC11           20MIS0011            gfychdnjs            Invalid Username   Invalid Username
TC12           20MIS0012            oiuergfhj            Invalid Username   Invalid Username
TC13           20MIS0013            jdufbeehj            Invalid Username   Invalid Username
TC14           20MIS0014            sjdfuehew            Invalid Username   Invalid Username
TC15           20MIS0015            sjdhaicji            Invalid Username   Invalid Username
TC16           20MIS0016            123456789            Invalid Username   Invalid Username
TC17           20MIS0017            789654123            Invalid Username   Invalid Username
TC18           20MIS0018            456852364            Invalid Username   Invalid Username
TC19           20MIS0019            753951458            Invalid Username   Invalid Username
TC20           20MIS0020            596321448            Invalid Username   Invalid Username
TC21           20MIS0001                                 Enter Password     Login Successful
TC22                                                     Enter Username     Invalid Username

Number of Total Testcases: 22
Number of Testcases Passed: 15
Number of Testcases Failed: 7
```

CHAPTER 7 – CONCLUSION AND FUTURE WORK

7.1 CONCLUSION:

In this project, we have understood and tested the application with Unit and Mutation testing in order to ensure that the application is bug free and efficient. This proves that the application does what it is supposed to do. We have implemented the Unit and Mutation testing using Google Colab in Python language. We can understand from the results that the piece of code used for testing is well implemented in order to test for multiple test cases and ensure the quality of the application. In Unit testing, we have tested the login module of the application with more than 20 test cases and conclude that the module is bug – free. But, in order to ensure that the application is completely error – free, we perform Mutation testing to further verify the robustness of the software. Therefore, we can state from the actual output of the Unit testing that the application does what it is supposed to do and from the actual output of Mutation testing, we can further conclude that the application responds the way we expect it to. Hence, it is safe to conclude that the login module which has been tested thoroughly is bug – free.

7.2 FUTURE WORK:

In this work, we explored the way of working of block chain in database. There are so many developments going which utilizes the block chain database in certain application like in game purchase items of games now started using block chain and It will have used for data to trace back the data of the original data. It creates a permanent and immutable record of every transaction. This impenetrable digital ledger makes fraud, hacking, data theft, and information loss impossible. Which helps in data in integrity of user's data. It may be complicated for now. Like any new technology, the block chain is an idea that initially disrupts, and over time it could promote the development of a larger ecosystem that includes both the old way and the new innovation.

REFERENCES

[1] Guy Harrison Introducing ProvenDB, ProvenDB - a Blockchain enabled database service layered on MongoDB (Apr 29, 2019), medium.com.

[2] Mayank Raikwar, Danilo Gligoroski, Goran Velinov, Trends in Development of Databases and Blockchain, Norwegian University of Science and Technology (NTNU) Trondheim, Norway, University Ss. Cyril and Methodius Skopje, Macedonia.

[3] Muhammad ElHindi, Carsten Binnig, Arvind Arasu, Donald Kossmann, Ravi Ramamurthy, BlockchainDB - A Shared Database on Blockchains, Microsoft Research.

[4] Senthil Nathan, Chander Govindarajan, Adarsh Saraf, Manish Sethi, and Praveen Jayachandran, Blockchain Meets Database: Design and Implementation of a Blockchain Relational Database, IBM Research India, IBM Industry Platforms, USA.

[5] Mohammad Jabed Morshed Chowdhury, Alan Colman, Muhammad Ashad Kabir, Jun Han and Paul Sarda, Blockchain versus Database: A Critical Analysis, School of Software and Electrical Engineering, Swinburne University of Technology, Melbourne, Australia.

[6] Alisha Nalavade, Divya Rawat, Prof. Harshil Kanakia, Blockchain Technology: Most Secure Database, Sardar Patel Institute of Technology, Andheri (West), Mumbai.

[7] Hye-Young Paik, Xiwei Xu, H. M. N. Dilum Bandara, Sung Une Lee and Sin Kuang Lo, Analysis of Data Management in Blockchain-Based Systems: From Architecture to Governance, Data61, CSIRO, Sydney, NSW 2015, Australia, School of Computer Science and Engineering, University of New South Wales, Sydney, NSW 2052, Australia, School of Information Systems, Queensland University of Technology, Brisbane, QLD 4000, Australia.