

IFET COLLEGE OF ENGINEERING
(AN AUTONOMOUS INSTITUTION)
VILLUPURAM

A MINI PROJECT REPORT



DRIVER AND VEHICLES DOCUMENTS
AUTHENTICATION SYSTEM

Submitted By

RAGAPRIYA. N [191021076]

POOJA SRI. R [191021071]

COMPUTER SCIENCE AND ENGINEERING

Under the guidance of

KIRUBA.K, M.E



IFET COLLEGE OF ENGINEERING
(AN AUTONOMOUS INSTITUTION)
VILLUPURAM – 605108.

BONAFIDE CERTIFICATE

Certified that this mini project report **“DRIVER AND VEHICLES DOCUMENTS AUTHENTICATION SYSTEM”** is a bonafide work of **“POOJA SRI.R”** who carried out the project work under my supervision

SIGNATURE OF THE GUIDE

HEAD OF THE DEPARTMENT

The mini project report submitted for the viva voice held on _____

INTERNAL EXAMINER

EXTERNAL EXAMINER

DRIVER AND VEHICLES DOCUMENTS AUTHENTICATION SYSTEM

ABSTRACT

- The rapid growth on automobiles increases vehicle registration certificate, license, insurance and PUC certificate verification by traffic police.
- Sometimes the vehicle owner forgets to carry the vehicle related documents at the time of enquiry.
- This results in trouble to the owner and also to the traffic police.
- This paper proposed an approach to solve the issues by storing the vehicle related documents at the database by RTO department.
- This paper proposed an approach to solve the issues by storing the documents at the website of the RTO office. The documents stored at the RTO office are: Insurance, PUC certificate, Aadhaar card, voter id, RC book, smart card, pan card, passport, driving license
- This website can be accessed by the police department to ensure or cross verify the vehicle related documents of the person such as driving License, insurance certificate, offence, GPS tracking, RC book, preliminary test, stolen status.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	04
1.	INTRODUCTION	06
2.	OBJECTIVES	06
3.	EXISTING SYSTEM	07
	3.1 Disadvantages Of An Existing System	
4.	PROPOSED SYSTEM	07
	4.1 Advantages Of Proposed System	
5.	STEPS INVOLVED ARE	
	5.1 Rto / Emission Test Center / Insurance Companies.	09
	5.2 General User	
	5.3 Police	
6.	SYSTEM ARCHITECTURE	10
7.	MODULES	
	7.1 Rto Module Architecture	11
	7.2 Traffic Police Module Architecture	
	7.3 Civil Police Module Architecture	
8.	SOFTWARE REQUIREMENT	14
9.	METHODOLOGY	14
10.	OUTPUT	15
11.	CONCLUSION	18
	APPENDIX	19

CHAPTER 1

INTRODUCTION

In the current scenario, the key issue faced by commuters while travelling is to deal with the inefficiency and discrepancy in the traffic system and the people involved in managing the same. On being caught by the police, the commuter must submit his driving license, RC Book and other vehicle documents for verification. In case the commuter forgets or misplaces the documents, he is unnecessarily fined. In case a vehicle is booked for any traffic violations, details are uploaded onto a website. The owner is not updated with this information. If he fails to check this website regularly, he may not be aware of any pending traffic violations on his vehicles. In this age of science and technology, where the internet is in the palm of everyone's hands, a better system for managing the above process can be set up which would greatly reduce the burden on the daily commuters as well as the traffic police.

CHAPTER 2

OBJECTIVES

- Minimize the use of physical documentation.
- Ensure Authenticity of the e-documents and thereby eliminating usage of fake documents.
- Make it easy for residents to receive services.
- Anytime, anywhere access and share the documents.
- Enable e-Signing of documents and make them available electronically and online.

CHAPTER 3

EXISTING SYSTEM

In many countries they are using different website to access documents. Some of them have slow search, requires more information for the particulars, some aren't giving the required document etc.,

3.1. DRAWBACKS OF EXISTING SYSTEM:

- Existing system requires many information from the particulars.
- For obtaining many particulars, we need to enter the information for each and every time. It consumes more time.
- We can't share any of our particulars to others.
- If the traffic police want to check whether the person qualified in preliminary test, he does not have any source for that.
- If the person needs to know about the status of his/her website it is not possible

CHAPTER 4

PROPOSED SYSTEM

The proposed system is an enhanced web website for RTO Management System and also includes use of RFID System for vehicle authentication. Here we added more authorities to the current system. System consist of RTO as administrator, pollution agencies, insurance agencies, police department, and vehicle owners. In this proposed system, RTO is power user. He has the power to verify the data entered by the user, processing of data and provide appropriate solutions. Administration is controlled by each RTO officer based on their credential limits.

4.1. ADVANTAGES OF PROPOSED SYSTEM:

- New proposed system requires only candidate name, candidate's father /husband name, date of birth, Aadhar number.
- In this we can obtain all the particulars, by entering the information only once.
- We can share the particulars.
- The traffic police can easily view the person preliminary test video using this new proposed system and he can also get the information like how many times the person attend the test.
- The person can easily get the status of his/her website.
- In the above section, we have briefly examined security issue are OTP password which is not capable to hold a large amount of data it is the weakest part of the existing system but in the proposed system we using password-based security as well as QR code technique.
- A large amount of data store in the inventory means QR code image has more capacity to hold data.
- In case of any part of QR code image is corrupt or some error has occurred that time to provide security.

CHAPTER 5

STEPS INVOLVED ARE

5.1.RTO / EMISSION TEST CENTERS / INSURANCE

COMPANIES:

1. RTO registers the users with a driving license and provides a unique user id.
2. They upload the necessary documents onto the drivers' and the vehicles' profiles.

5.2. GENERAL USER:

1. Signs in through the android website using the user id provided to him as authentication.
2. Can view all the latest documents such as driving license, owned vehicle details, RC Book, latest emission test certificate, insurance copy, etc. on his profile page.
3. Can report stolen vehicle to notify the nearest police.
4. Can check any unpaid offences on his vehicle and can check the status of his website.

5.3. POLICE:

1. Signs in through the android website using the user id provided to him as authentication.
2. Can enter a vehicle number to view the owner's or vehicle's documents and previous unpaid offences.
3. Can report any offences committed by the driver using the website.
4. Receives reports about vehicles stolen under his jurisdiction/working location.

CHAPTER 6

SYSTEM ARCHITECTURE

- The larger system is decomposed into sub systems. That provides some related set of services.
- In which we aim to provide better services. Through android phones and that requires internet for the data transfer.
- The architecture we can see that a new applier will provide his document's hard copy forms to the administrator of RTO. This information is stored in database at server through online registration. And server-side end is in JSP. On client side an android website will be provided to police. After police logging into the system can retrieve vehicle and license related information from the RTO database.
- If authentication fails the information is provided to the police to retry else information about the user is displayed.
- The architecture mainly consists of four modules.

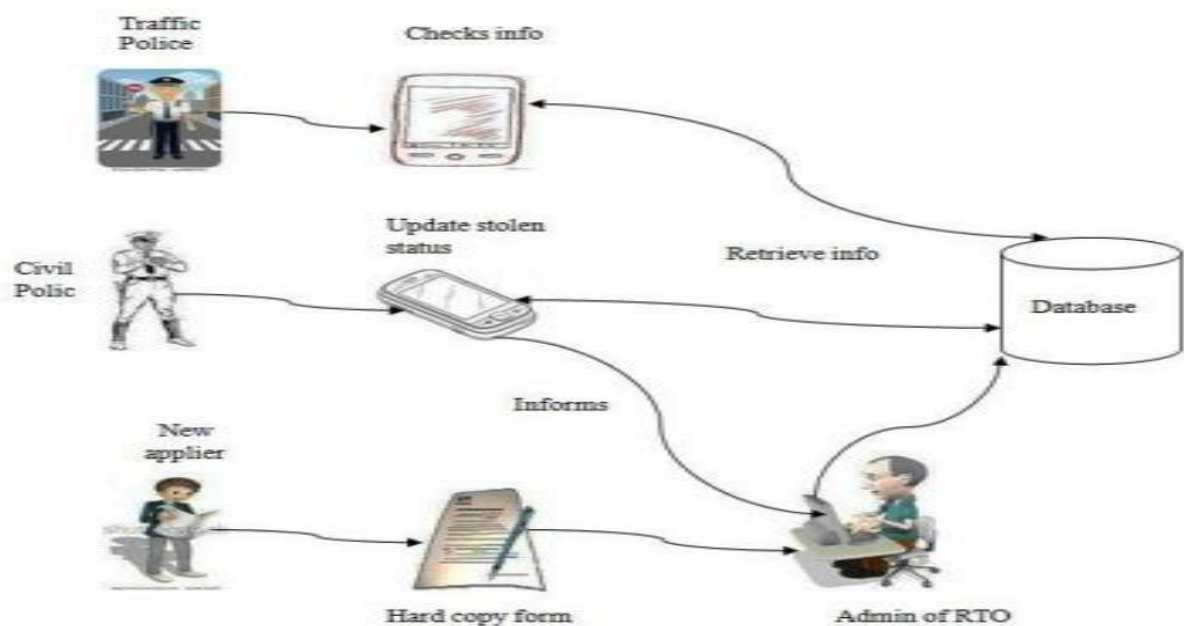


Fig. 1 System Architecture

CHAPTER 7

MODULES

7.1.USER MODULE:

User will fill the online form followings are:

A. LICENSE REGISTRATION FORM:

The form display information of particular license holder with photo and other details such as first name, middle name, last name, email, date of birth, gender, taluka, contact number, ration card number, address, Aadhar card number, and voter ID.

B. VEHICLE REGISTRATION FORM:

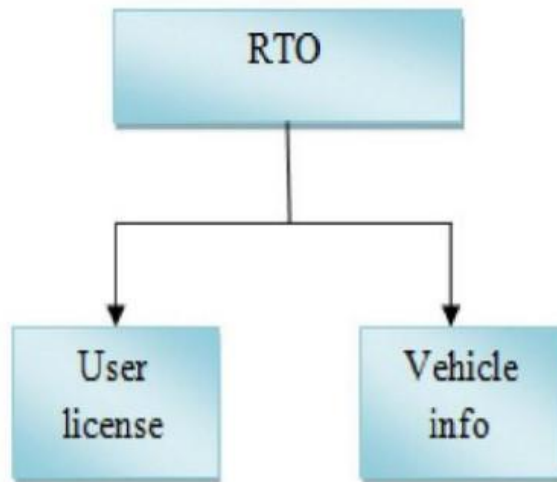
In vehicle Registration form the RTO administrator gets various details of vehicle and to enter vehicle information first select the vehicle type such as two-wheeler, three-wheeler, four-wheeler, and other types. After entering Registration no such as AP 22 CA 7613 then click check vehicle button it gives pop up message found or not found and displays all records in the data grid below figure shows the vehicle registration form of two-wheeler vehicle.

C. INSURANCE FORM:

The owner will enter his information as well as vehicle information. User information like first name, last name, middle name, mobile number, email, Vehicle Information like vehicle make, vehicle sub-type, vehicle model, engine number, CC, manufacturing year, city, state, chassis number, fuel used, registration date, your bike's IDV, Research Article Volume 7 Issue No.6 International Journal of Engineering Science and Computing, June 2017 12577 <http://ijesc.org/> house number, and select premium plan. We search the insurance formation by using insurance number.

7.2. RTO MODULE ARCHITECTURE:

This module is specifically designed for the RTO administrator and it consists of information's related to the user license and vehicle. This information will be stored in the database.



7.3. TRAFFIC POLICE MODULE ARCHITECTURE:

This module mainly focuses on providing the information only to the traffic police officers. It consists of vehicle information and license information's. It also generates the fine.

A. CHECK LICENSE:

The Check License module takes a License number. as an input and returns the respective License information like Name, Photo, address, DOB, issued date, License status and Validity of the license. In case if the record doesn't exist, it shows a respective message that record not found.

B. CHECK VEHICLE:

The Check Vehicle module takes a Vehicle Registration number as an input and returns the Respective Owner details like Name and Address. It also includes the Vehicle issued date and valid date of vehicle.

C. GENERATE FINE:

The Generate Fine module mainly focuses on generating fines for the offences committed. It provides for selecting a variety of offences from a set of given offences for which the fine amount is auto-generated. The user (R.T.O officer) needs to enter other information related to Officer like Officer Id and Other important Vehicle information. This overall process requires internet for data transfer between the client and the server and the data is stored on server.

D. CHECK INSURANCES:

Check insurance model which gives the information of the person such as name, city, vehicle make, vehicle model, plan premium, issued date, valid date.

7.4. CIVIL POLICE MODULE ARCHITECTURE:

This module is mainly provided to the civil police officer. It provides a web page to update the stolen status of vehicle.

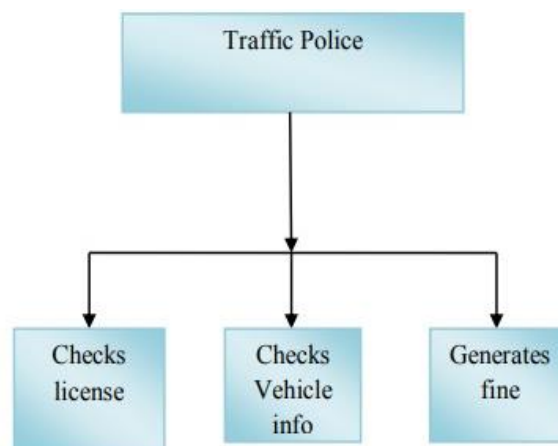


Fig. 3 Traffic police module architecture

CHAPTER 8

SOFTWARE REQUIREMENTS

- Android
- Web Server
- MySQL
- PHP
- HTTP

CHAPTER 9

METHODOLOGY

The proposed system has an android website designed using JAVA and XML with dedicated user interfaces for each of the actors on the scene. The android website connects to a webserver by making a HTTP request so that when the android website is executed, it connects the android device to the PHP script. The PHP script fetches the data from the database, which is then encoded and sent to the device. This encoded data is then parsed and displayed on the android device.

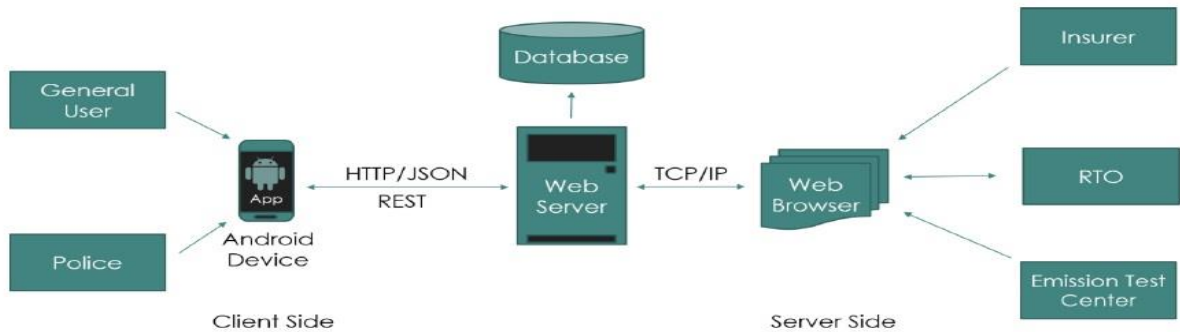
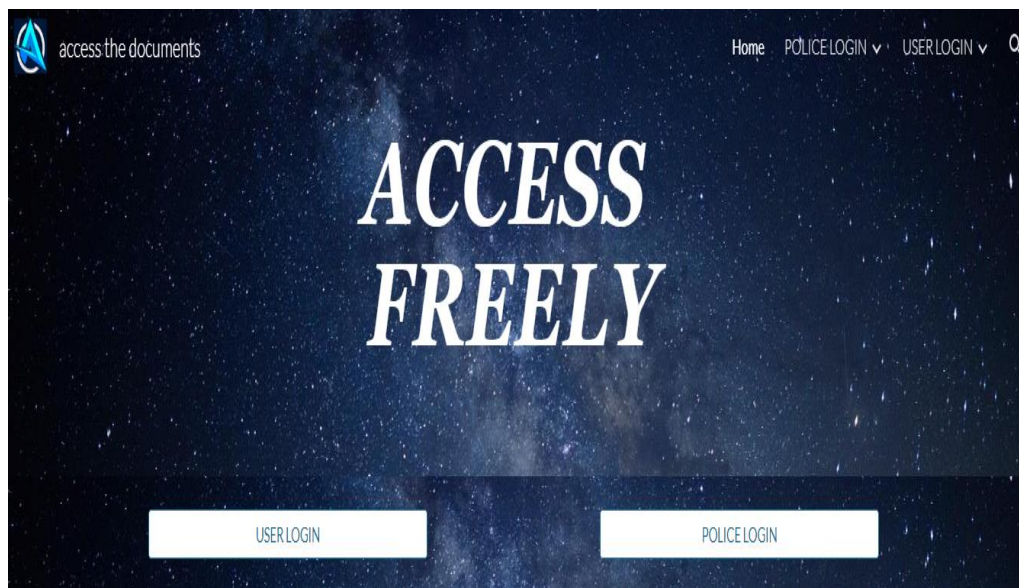


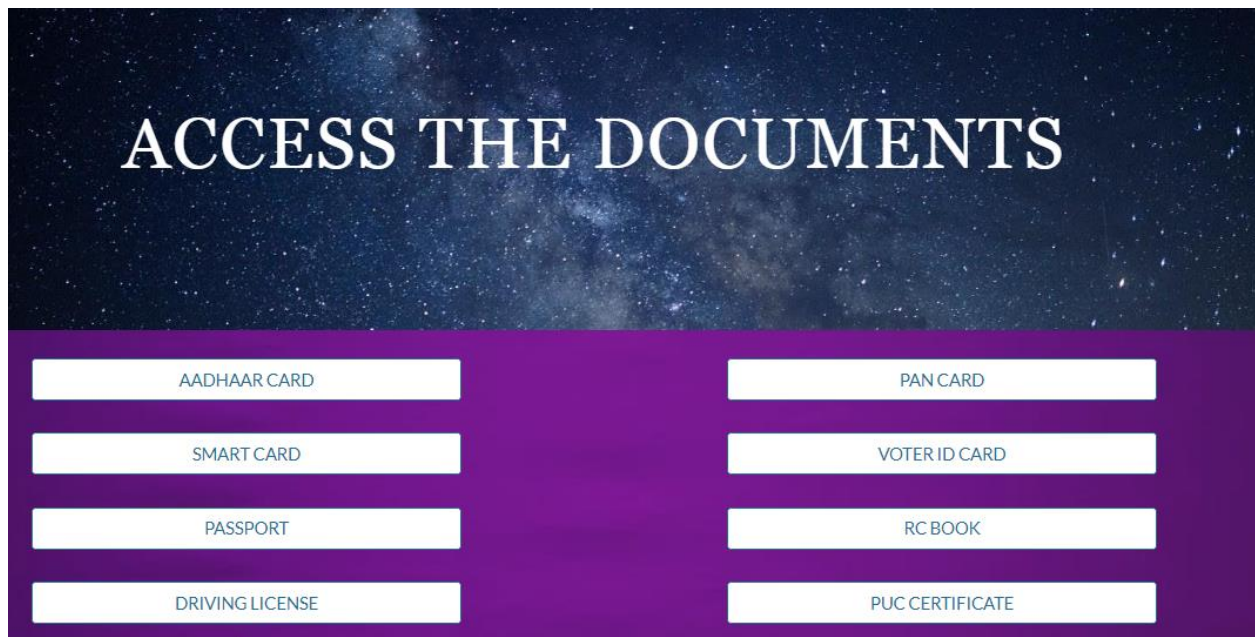
Figure 1: Methodology

CHAPTER 10

OUTPUT



This screenshot shows the 'USER LOGIN' form. The background is dark with a starry pattern. At the top, there is a hamburger menu icon, a logo, and a search icon. The title 'USER LOGIN' is displayed in a large, white, serif font. Below the title, there is a form with five input fields, each with a red asterisk indicating a required field. The fields are: 'USER LOGIN' (with a red asterisk and a red 'Required' message), 'USER NAME' (with a red asterisk), 'DATE OF BIRTH' (with a red asterisk and a 'Date' label), 'FATHER'S NAME' (with a red asterisk), and 'AADHAAR NUMBER' (with a red asterisk). Each field has a 'Your answer' placeholder text. At the bottom, there is a green 'Submit' button.



DRIVING LICENSE



POLICE LOGIN

ACCESS FREELY

Your response has been recorded.

[Submit another response](#)

VEHICLE NUMBER

ACCESS FREELY

POLICE VERIFICATION CODE

vehicle registration number

Your answer

Submit

CHAPTER 11

CONCLUSION

This system also enables a simplified system for commuters to handle various vehicle documents as they now no longer have to worry about managing the hard copies of each. With every important document present as a soft copy in their mobiles, the commuters can now just show these to the police for verification. Along with all of the above-mentioned benefits, the system now greatly helps vehicle owners lodge a stolen vehicle report from within the app. This can help notify the police in the nearby locality faster and help track the vehicle in a more efficient manner. This integrated system greatly improves efficiency and makes way for a better approach to manage the work of the traffic department. Thus, the major expected outcome of this project is to improve public welfare using the latest technologies.

APPENDIX

```
package com.example.anichopr.accessfreelycker;

import android.content.Context;
import android.graphics.BitmapFactory;
import android.os.StrictMode;


import org.jsoup.Connection;
import org.jsoup.Jsoup;
import org.jsoup.nodes.Document;
import org.jsoup.nodes.Element;
import org.jsoup.select.Elements;


import java.io.IOException;
import java.text.DateFormat;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.List;
import java.util.Locale;
import java.util.Map;


public class AccLibrary {
    public static List<access freely> otherDocs = null;
    public static List<Access freely> essentialDocs = null;
    static List<Access freely> allDocs = null;


    static String[] importantDocNames = {
        "Passport",
        "Pan",
```

```

"Voter",
"Aadhar",
"Driving",
"Birth"
};

```

```

public static int getImagefromName(String name) {
    if (name.contains("Passport")) {
        return R.mipmap.passport;
    } else if (name.contains("Pan")) {
        return R.mipmap.pancard;
    } else if (name.contains("Voter")) {
        return R.mipmap.voterid;
    } else if (name.contains("Aadhar")) {
        return R.mipmap.aadhaar;
    }
    else if (name.contains("Birth")) {
        return R.mipmap.aadhaar;
    }
    else if (name.contains("Driving")) {
        return R.mipmap.aadhaar;
    }
    else return R.mipmap.docwhite;
}

```

```

public static void refreshEssentialAndOthersAccessfreely(Context context) {
    ArrayList<Access freely> otherDocList = new ArrayList<Access freely>();
    ArrayList<Access freely> essentialDocList = new ArrayList<Access freely>();
    boolean[] essentialDocAvailableList = new boolean[importantDocNames.length];
}

```

```

// refresh all documents

```

```
refreshAccessfreelyFromDocument();
```

```
if (allDocs==null)
return;
for (inti=0;i<allDocs.size();i++) {
booleanfEssentialDoc=false;
for (int j=0; j<importantDocNames.length;j++) {
if (allDocs.get(i).documentName.contains(importantDocNames[j])) {
int id = getImagefromName(allDocs.get(i).documentName);
allDocs.get(i).bitmap=BitmapFactory.decodeResource(context.getResources(), id);
essentialDocAvailableList[j] =true;
```

```
essentialDocList.add(allDocs.get(i));
fEssentialDoc=true;
break;
    }
}
```

```
if(!fEssentialDoc) {
otherDocList.add(allDocs.get(i));
    }
}
```

```
for (inti=0;i<essentialDocAvailableList.length;i++) {
if(!essentialDocAvailableList[i]) {
int id =R.mipmap.plus;
```

```
Access freely =newAccess freely(importantDocNames[i], "");
accessfreely.bitmap=BitmapFactory.decodeResource(context.getResources(), id);
```



```
essentialDocList.add(access freely);
    }
}
```

```
otherDocs=otherDocList;
essentialDocs=essentialDocList;
}
```

```
private static void refreshAccessfreelyFromDocument() {
try {
StrictMode.ThreadPolicy policy
=new StrictMode.ThreadPolicy.Builder().permitAll().build();
StrictMode.setThreadPolicy(policy);
Connection.Response res = Jsoup.connect("https://access freely.gov.in/Signin.aspx")
.timeout(100000)
.execute();
Map<String, String> cookie_map=res.cookies();

Document doc =res.parse();
Element viewstateValue=doc.getElementById("__VIEWSTATE");
Element eventvalidationValue=doc.getElementById("__EVENTVALIDATION");
Element eventargumentValue=doc.getElementById("__EVENTARGUMENT");
Element eventtargetValue=doc.getElementById("__EVENTTARGET");
Element viewstateencryptedValue=doc.getElementById("__VIEWSTATEENCRYPTED");
String seed =doc.getElementById("SitePH_btnLogin").attr("onClick").split("")[1];
String ss =viewstateencryptedValue.attr("value");
```

```

Connection.Response signin=Jsoup.connect("https://access.freelycker.gov.in/Signin.aspx").
data("__VIEWSTATE", viewstateValue.attr("value")).
data("__EVENTVALIDATION", eventalidationValue.attr("value")).
data("__EVENTARGUMENT", "").
data("__EVENTTARGET", "").
data("__VIEWSTATEENCRYPTED", viewstateencryptedValue.attr("value")).
data("ToolkitScriptManager1_HiddenField", "").
data("ctl00$SitePH$HV1", "").
data("ctl00$SitePH$btnLogin", "Sign In").
data("ctl00$SitePH$hdnFBEmail", "").
data("ctl00$SitePH$hdnFBFName", "").
data("ctl00$SitePH$hdnFBID", "").
data("ctl00$SitePH$hdnFBLName", "").
data("ctl00$SitePH$hdnUserId", "").
data("ctl00$SitePH$hdngmail", "").
data("cctl00$SitePH$hid_LoginAttempt", "").
data("ctl00$SitePH$hid_btnText", "").
data("ctl00$SitePH$txtEIDNew", "").
data("ctl00$SitePH$txtPassword",
PasswordEncryption.GetEncryptedPassword("Indrachand@22", seed)).
data("ctl00$SitePH$txtUIDNew", "").
data("ctl00$SitePH$txtUserID", "Icchopra").
data("ctl00$SitePH$txt_UID", "Aadhaar Number").
        cookies(cookie_map).
method(Connection.Method.POST).
timeout(10000).
execute();

```

```

//      File input = new File("C:\\Users\\israut\\Desktop\\Access freelycker.htm");
//      Document signin = Jsoup.parse(input, "UTF-8", "https://access.freelycker.gov.in/");

```

```

ArrayList<Access
freely>allDocsList=DocLibrary.getAccessfreelyFromDocument(signin.parse());
allDocs=allDocsList;
    } catch (IOException e) {
e.printStackTrace();
    }
}

```

```

publicstaticArrayList<Access freely>getAccessfreelysFromDocument(Documentdoc) {
ArrayList<Access freely>access freelyArray=newArrayList<Access freely>();
ElementstableAccessfreely=doc.getElementsByClass("tblFlexi");
Element table =tableAccessfreely.first();

```

```

//for self uploaded Docs
    table =tableAccessfreelys.last();
booleannotAvailable=table.html().contains("Not Available");

```

```

if(!notAvailable) {
ElementsdocRows=table.select("tr");
for (inti=1; i<docRows.size(); i++) {
Elementsdocrow=docRows.eq(i);
ElementsdocColumns=docrow.first().select("td");

```

```

StringdocumentName=docColumns.eq(1).first().text();
StringdocumentURL=docColumns.select("a[href]").first().attr("href");

```

```
Access freelyaccess freely=newAccess freely(documentName, documentURL);
```

```
access freely.serialNo=Integer.parseInt(docColumns.eq(0).first().text());
```

```
String date =docColumns.eq(2).first().text();
```

```
DateFormat df =newSimpleDateFormat("dd-mm-yyyy", Locale.ENGLISH);
```

```
try {
```

```
access freely.uploadDate=df.parse(date);
```

```
    } catch (ParseException e) {
```

```
// TODO Auto-generated catch block
```

```
System.out.println("during date");
```

```
e.printStackTrace();
```

```
    }
```

```
//doc status
```

```
access freely.status=docColumns.eq(3).first().text();
```

```
//share id
```

```
access freely.shareId=Integer.parseInt(docColumns.eq(5).toString().split("&quot;")[3]);
```

```
Access freely.noOfOtherDocs++;
```

```
access freelyArray.add(access freely);
```

```
    }
```

```
  }
```

```
returnaccess freelyArray;
```

```
}
```

```
public static boolean IsDocAvailable(String text) {  
    if (allDocs == null)  
        return false;  
  
    for (Access freelydoc : allDocs) {  
        if (doc.documentName.contains(text)) {  
            return true;  
        }  
    }  
    return false;  
}  
}
```