**INTRODUCTION**



**ANS SMS SYSTEM**

An ANS **SMS**(Short Message Services) **API**(Application program Interface) System Is Digital SMS Services That have a Framework capacity for sending a message form source to destination Number. In this Project This services is used for sending a SMS to parents, if Any problems in collage form student, or any information related to Student education. This Is a WEB and ANDROID based project Where Used a various programming Languages. And Here the **FRONT END** programming is PHP, JAVA, JAVA script, CSS, XML and **BACK END** is MYSQL.

**WHY THIS PROJECT IS USED SERVER BASED CONNECTION ?.**

The main cause of server based connection is for Executing API Services, API is interface for the Application that handles Operations like

* SEND MESSAGE
* LOG THE MESSAGE
* Generate A Script key
* Authenticate User
* Check the Status of message That have delivered Destination or not

**SEND MESSAGE**

In SMS API the Source SMS will not Sent From Numbers, In-stored Of that It uses The Random CODE like NX-Title of the company.

**LOG THE MESSAGE**

The word LOG Means Storing the data. The data can be stored either in Text file with any file extinctions or SQL Database. But saving the data in File That can be accessed by any one by reading a file.

**GENERATE A SCRIPT KEY**

This is the main part of SMS API, without this the project will not work. The script Keys are the keys where that connects the API to Owner Server(Who provided the complete SMS Services).

**AUTHENTICATE USER**

Check the passwords that have signed any ware in other system. The main goal this phase is to identify the API services that have are used on Some other system.

**STATUS**

Check the message is delivered to destination number

If Code == 1

Then The message is delivered

If Code== 0

Message not delivered to destination Number

**OBJECTIVE**

Objectives of SMS API System

1. To support, those students who have been deprived of the services offered by the college, for which he / she is entitled.
2. To Send the SMS about Student is irregular class.
3. To report the Message, that has been sent form HOD or Teachers for identification.
4. To make student regular to class.
5. To inform any events that have takes place in collage.

**Problem definition**

The main problem definition is to send message to parent or student by typing a message Body at each time, it may take a lot of time.

Now a days Many Versions of smartphones released to platform. The Students parents are don't know how to use the Smartphones, or the Parents don't have a communication on Modern applications that are released now days or They are not in touch with smartphones. if the parents are using Button Phones, The button phones are not able to communicate with web based applications. In-stored of that The SMS System provides a Support to communicate with reaching the SMS message form collage.

**Methodology**

A SMS API System that runs with Android Using PHP, Mysql and SMS API,XML, HTML. A php that loaded from the host machine that checks the mysql connection to connect with server to dump the data to table database. SMS API that connects with Main server to current server to send the message for current server with destination address.

**SQL DATABASE CONNECTION CONFIGURATION**

****

**Android have a capability of connecting with sql server by two types**

**1. WebView Based**

**2. Standard connection**

***Webview Based Connection***

This projects is based on webview based connection. Because this project have a connection between SMS api package. The SMS API package is supported for Web Based applications to send and save the messages to database

1. SQL connection is configured by Config.php File. Where it checks the database is available or not at current time.

2. Sends the response by server to console write the status of database

3. If server is available then connect with server and operate insert, delete ,featch operations.

**Standard SQL Connection**

Here the connection is established directly on json or java source file, By specifying the host , Server Name , Password And user Name

**Life Cycle of SMS API System**

Life cycle of SMS API System cell works with the following steps

1. Open Android application load Index page from Server

2. Run Config file with index file

3. Check the database connection is is ok then go to success else failure

4. show The menu with 3 operations

* Send message
* Get report
* Set/reset data

5. check the sent message is not equal to NULL

6. if NULL display The message To Set The data

7. else go to send message Activity

8. Get the number form Number field

9. connect with SMS API services

10. Check Credit balance

11. if credit is more then 0 then send message and return success

12. If no Credit return failure and display alert message and exit.

**Report Generation In JAVA SCRIPT API**

1. Load main ACTIVITY with secure login

2. connect with external JS api

3. GET text id by passing variable

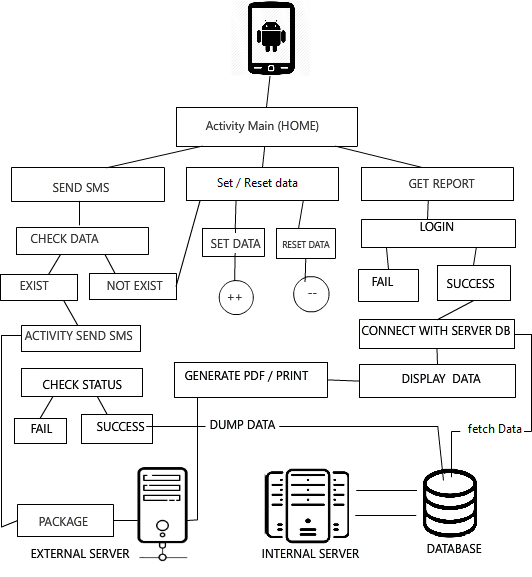
4. load JS API with operations

5. Ready to download

7. Complete Download

8. logout with closing session

**Working Of SMS API System**

****

**Hardware and software specifications**

**Hardware Requirements For development**

1. System Hard desk: 70 MB

2. System type: 32 Bit or 64 bit

3. Processor: Intel(R) 1 GHZ or AMD 1.33 GHZ

4. Monitor: 32 bit Color

5. Ram: 1GB

6. Network Speed: min 100 Kb / max 80 kb/sec

**Software Requirements For developmentt**

1. Operating system: Windows, MAC, Android, Linux.

2. Runtime: Browser Upgraded Chrome, IE 11, firefox,etc

3. Xampp Server

**Hardware Requirements For Execution**

1. System internal Storage : 70 MB

2. System type: 32 Bit or 64 bit

3. Ram : 1GB

4. Network Speed: min 100 Kb / max 80 kb/sec

**Software Requirements Execution**

1. Operating system: Android

2. OS Version : 4.0 to later

3. Sql server

**Tools/Environment Used**

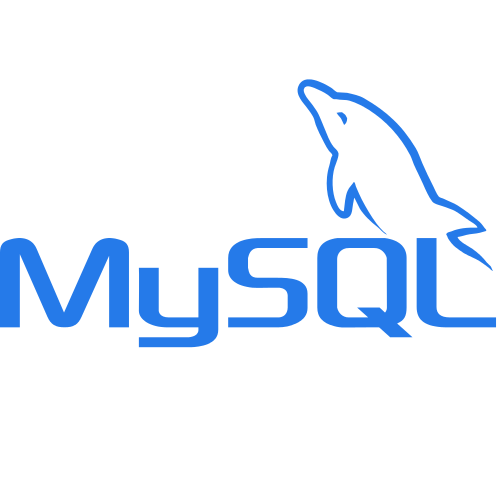
**Tools used**

1. ***Android Studio***

******

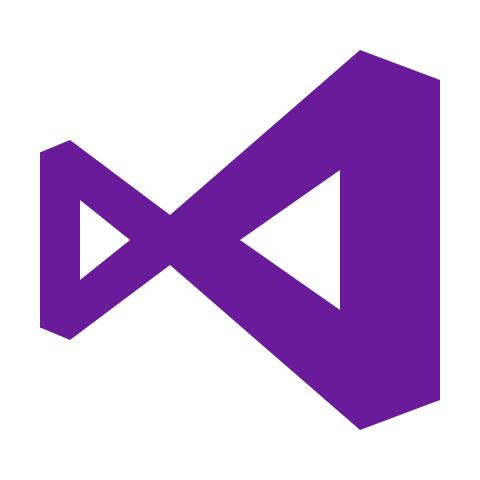
Android Studio is the official integrated development environment for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems.

1. ***Microsoft SQL Server***

******

Microsoft SQL Server is a relational database management system developed by Microsoft. As a database server, it is a software product with the primary function of storing and retrieving data as requested by other software applications—which may run either on the same computer or on another computer across a network.

1. ***Visual studio code***

******

Visual Studio Code is a source-code editor developed by Microsoft for Windows, Linux and macOS. It includes support for debugging, embedded Git control and GitHub, syntax highlighting, intelligent code completion, snippets, and code refactoring.

**Programming’s used**

***JAVA***

******

Java is a [general-purpose](https://en.wikipedia.org/wiki/General-purpose_language) [programming language](https://en.wikipedia.org/wiki/Programming_language) that is [class-based](https://en.wikipedia.org/wiki/Class-based_programming), [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming), and designed to have as few implementation [dependencies](https://en.wikipedia.org/wiki/Dependency_(computer_science)) as possible. It is intended to let [application developers](https://en.wikipedia.org/wiki/Application_developer) write once, run anywhere (WORA),meaning that [compiled](https://en.wikipedia.org/wiki/Compiler) Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to [bytecode](https://en.wikipedia.org/wiki/Java_bytecode) that can run on any [Java virtual machine](https://en.wikipedia.org/wiki/Java_virtual_machine) (JVM) regardless of the underlying [computer architecture](https://en.wikipedia.org/wiki/Computer_architecture). The [syntax](https://en.wikipedia.org/wiki/Syntax_(programming_languages)) of [Java](https://en.wikipedia.org/wiki/Java_(software_platform)) is similar to [C](https://en.wikipedia.org/wiki/C_(programming_language)) and [C++](https://en.wikipedia.org/wiki/C%2B%2B), but it has fewer [low-level](https://en.wikipedia.org/wiki/Low-level_programming_language) facilities than either of them. As of 2019, Java was one of the most [popular programming languages in use](https://en.wikipedia.org/wiki/Measuring_programming_language_popularity) according to [GitHub](https://en.wikipedia.org/wiki/GitHub),[[17]](https://en.wikipedia.org/wiki/Java_(programming_language)#cite_note-:0-17)[[18]](https://en.wikipedia.org/wiki/Java_(programming_language)#cite_note-18) particularly for [client-server](https://en.wikipedia.org/wiki/Client%E2%80%93server) [web applications](https://en.wikipedia.org/wiki/Web_applications), with a reported 9 million developers

1. ***PHP***

******

Hypertext Preprocessor is a general-purpose programming language originally designed for web development. It was originally created by Rasmus Lerdorf in 1994; the PHP reference implementation is now produced by The PHP Group.

**Example**:

1. <?php
2. $string = "Hello Word";
3. echo "$string";
4. ?>

***2.HTML***

******

Hypertext Markup Language is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets and scripting languages such as JavaScript.

**Example**:

1. <html>
2. <head>
3. <title>Hello Word</title>
4. </head>
5. <body>
6. <p>Hello Word</h1>
7. </body>
8. </html>

***3.CSS***

******

Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

Example:

1. <html>
2. <head>
3. <title>Hello Word</title>
4. <style>
5. p{
6. color:gray;
7. font-size:11px;
8. background:white;
9. }
10. </style>
11. </head>
12. <body>
13. <p>Hello Word</h1>
14. </body>
15. </html>

***4.MYSQL***

******

MySQL is an open-source relational database management system. Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language.

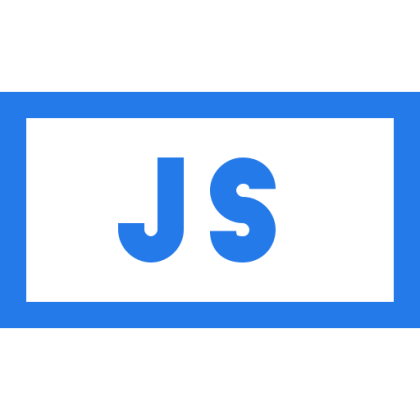
**Example**:

1. CREATE TABLE `project` (
2. `id` int(11) NOT NULL,
3. `name` text NOT NULL,
4. `email` text NOT NULL
5. )

//Dumping data for table `project`

1. INSERT INTO `project` (`id`, `name`, `email`, ) VALUES
2. (1, 'SQL', 'rajtharunir@gmail.com');

***5. Java script***

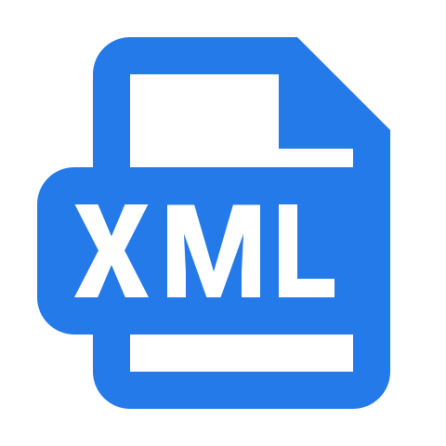
******

JavaScript, often abbreviated as JS, is a high-level, interpreted scripting language that conforms to the ECMAScript specification. JavaScript has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions.

**Example:**

1. var ir = document.getElementById('pay');
2. ir.innerText = "Hello word";

***6.XML***

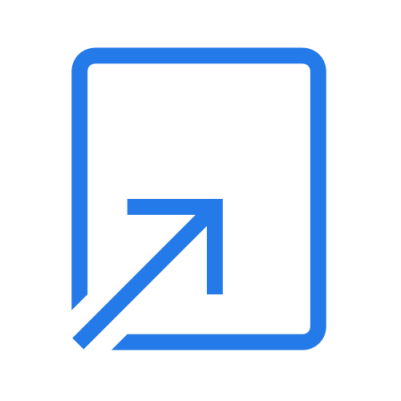
******

Extensible Markup Language is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. The W3C's XML 1.0 Specification and several other related specifications—all of them free open standards—define XML.

**Example**:

1. <?xml version="1.0" encoding="UTF-8"?>
2. <url>
3. <loc>https://rajsoft.ga/</loc>
4. <lastmod>2020-02-05T15:01:57+00:00</lastmod>
5. <priority>1.00</priority>
6. </url>

***7. JQUERY***

******

jQuery is a JavaScript library designed to simplify HTML DOM tree traversal and manipulation, as well as event handling, CSS animation, and Ajax. It is free, open-source software using the permissive MIT License. As of May 2019, jQuery is used by 73% of the 10 million most popular websites.

**Input Design and Output Design**

**INPUT DESIGN**

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

* What data should be given as input?
* How the data should be arranged or coded?
* The dialog to guide the operating personnel in providing input.
* Methods for preparing input validations and steps to follow when error occur.

**OBJECTIVES**

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2.It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system’s relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2. Select methods for presenting information.

3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

* Convey information about past activities, current status or projections of the
* Future.
* Signal important events, opportunities, problems, or warnings.
* Trigger an action.
* Confirm an action.

**TESTING**

The following are the Testing Methodologies:

* **Unit Testing.**
* **Integration Testing.**
* **User Acceptance Testing.**
* **Output Testing.**
* **Validation Testing.**

**Unit Testing**

Unit testing focuses verification effort on the smallest unit of Software design that is the module. Unit testing exercises specific paths in a module’s control structure to ensure complete coverage and maximum error detection. This test focuses on each module individually, ensuring that it functions properly as a unit. Hence, the naming is Unit Testing.

During this testing, each module is tested individually and the module interfaces are verified for the consistency with design specification. All important processing path are tested for the expected results. All error handling paths are also tested.

The main functionality of the module is to maintain the auditing details and generating the MAC along with security code for the file received from owner. It is tested to see if the files of multiple owners can be stored. Test is conducted to verify if the unique key is being generated for all the files.

**Integration Testing**

Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of high order tests are conducted. The main objective in this testing process is to take unit tested modules and builds a program structure that has been dictated by design.

The individual modules built and tested are integrated to obtain the fully functional system. In this step the system is tested to verify if all the modules interact with each other. The network connections are examined by using the ping system call.

Test is carried out to make sure the uploaded files by the owner are received by the TPA and cloud server. Multiple owners files have to be distinguished from each other by storing themalong with the owner name and port number. Automatic uploading of the file by the TPA to cloud server should be done after the time out and it should be done whenever the file is being deleted. Many to many network connections are established, multiple owners, multiple users, centralized TPA and cloud server functioning is tested.

**User Acceptance Testing**

User Acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes wherever required. The system developed provides a friendly user interface that can easily be understood even by a person who is new to the system.

**Output Testing**

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in the specified format. Asking the users about the format required by them tests the outputs generated or displayed by the system under consideration. Hence the output format is considered in 2 ways – one is on screen and another in printed format.

**Validation Checking**

Validation checks are performed on the following fields.

**Text Field:**

The text field can contain only the number of characters lesser than or equal to its size. The text fields are alphanumeric in some tables and alphabetic in other tables. Incorrect entry always flashes and error message.

**Preparation of Test Data**

Taking various kinds of test data does the above testing. Preparation of test data plays a vital role in the system testing. After preparing the test data the system under study is tested using that test data. While testing the system by using test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

**Using Live Test Data**

Live test data are those that are actually extracted from organization files. After a system is partially constructed, programmers or analysts often ask users to key in a set of data from their normal activities. Then, the systems person uses this data as a way to partially test the system. In other instances, programmers or analysts extract a set of live data from the files and have them entered themselves.

It is difficult to obtain live data in sufficient amounts to conduct extensive testing. And, although it is realistic data that will show how the system will perform for the typical processing requirement, assuming that the live data entered are in fact typical, such data generally will not test all combinations or formats that can enter the system. This bias toward typical values then does not provide a true systems test and in fact ignores the cases most likely to cause system failure.

**Using Artificial Test Data:**

Artificial test data are created solely for test purposes, since they can be generated to test all combinations of formats and values. In other words, the artificial data, which can quickly be prepared by a data generating utility program in the information systems department, make possible the testing of all login and control paths through the program.

The most effective test programs use artificial test data generated by persons other than those who wrote the programs. Often, an independent team of testers formulates a testing plan, using the systems specifications.

The package “Virtual Private Network” has satisfied all the requirements specified as per software requirement specification and was accepted.

**USER TRAINING**

Whenever a new system is developed, user training is required to educate them about the working of the system so that it can be put to efficient use by those for whom the system has been primarily designed. For this purpose the normal working of the project was demonstrated to the prospective users. Its working is easily understandable and since the expected users are people who have good knowledge of computers, the use of this system is very easy.

**MAINTAINANCE**

This covers a wide range of activities including correcting code and design errors. To reduce the need for maintenance in the long run, we have more accurately defined the user’s requirements during the process of system development. Depending on the requirements, this system has been developed to satisfy the needs to the largest possible extent. With development in technology, it may be possible to add many more features based on the requirements in future. The coding and designing is simple and easy to understand which will make maintenance easier.

**TESTING STRATEGY**

A strategy for system testing integrates system test cases and design techniques into a well-planned series of steps that results in the successful construction of software. The testing strategy must co-operate test planning, test case design, test execution, and the resultant data collection and evaluation .A strategy for software testing must accommodate low-level tests that are necessary to verify that a small source code segment has been correctly implemented as well as high level tests that validate major system functions against user requirements. Software testing is a critical element of software quality assurance and represents the ultimate review of specification design and coding. Testing represents an interesting anomaly for the software. Thus, a series of testing are performed for the proposed system before the system is ready for user acceptance testing.

**SYSTEM TESTING**

Software once validated must be combined with other system elements (e.g. Hardware, people, database). System testing verifies that all the elements are proper and that overall system function performance is achieved. It also tests to find discrepancies between the system and its original objective, current specifications and system documentation.

**Requirement Gathering Software Environment**

## Java & java Script Technology

Java technology is both a programming language and a platform.

### The Java Programming Language

### The Java programming language is a high-level language that can be characterized by all of the following buzzwords:

* + - Simple
    - Architecture neutral
    - Object oriented
    - Portable
    - Distributed
    - High performance
    - Interpreted
    - Multithreaded
    - Robust
    - Dynamic
    - Secure

With most programming languages, you either compile or interpret a program so that you can run it on your computer. The Java programming language is unusual in that a program is both compiled and interpreted. With the compiler, first you translate a program into an intermediate language called Java byte codes —the platform-independent codes interpreted by the interpreter on the Java platform. The interpreter parses and runs each Java byte code instruction on the computer. Compilation happens just once; interpretation occurs each time the program is executed. The following figure illustrates how this works.

**Source Of the project**

**package com.istudent;**

**import android.app.Activity;**

**import android.content.Intent;**

**import android.os.Bundle;**

**import android.os.Handler;**

**public class SplashActivity extends Activity {**

**// Splash screen timer**

**private static int SPLASH\_TIME\_OUT = 10000;**

**@Override**

**protected void onCreate(Bundle savedInstanceState) {**

**super.onCreate(savedInstanceState);**

**setContentView(R.layout.activity\_splash);**

**// Functionality for splash screen**

**new Handler().postDelayed(new Runnable() {**

**/\***

**\* Showing splash screen with a timer. This will be useful when you**

**\* want to show case your app logo / company**

**\*/**

**@Override**

**public void run() {**

**// This method will be executed once the timer is over**

**// Start your app main activity**

**Intent i = new Intent(SplashActivity.this, LoginActivity.class);**

**startActivity(i);**

**// close this activity**

**finish();**

**}**

**}, SPLASH\_TIME\_OUT);**

**}**

**}**

**AdminActivity**

**package com.istudent;**

**import java.util.ArrayList;**

**import android.annotation.SuppressLint;**

**import android.content.Intent;**

**import android.content.res.Configuration;**

**import android.content.res.TypedArray;**

**import android.os.Bundle;**

**import android.support.v4.app.ActionBarDrawerToggle;**

**import android.support.v4.app.Fragment;**

**import android.support.v4.app.FragmentManager;**

**import android.support.v4.widget.DrawerLayout;**

**import android.support.v7.app.ActionBarActivity;**

**import android.util.Log;**

**import android.view.Menu;**

**import android.view.MenuItem;**

**import android.view.View;**

**import android.widget.AdapterView;**

**import android.widget.ListView;**

**import android.widget.TextView;**

**import android.widget.Toast;**

**import com.istudent.adapter.NavDrawerListAdapter;**

**import com.istudent.fragment.admin.AdminHomeFragment;**

**import com.istudent.fragment.admin.EventFragment;**

**import com.istudent.fragment.admin.RegisterStaffFragment;**

**import com.istudent.fragment.admin.RegisterStudentFragment;**

**import com.istudent.fragment.admin.TimeTableFragment;**

**import com.istudent.fragment.student.EventCalendarFragment;**

**import com.istudent.fragment.student.ViewTimeTable;**

**import com.istudent.model.NavDrawerItem;**

**public class AdminActivity extends ActionBarActivity {**

**private DrawerLayout mDrawerLayout;**

**private ListView mDrawerList;**

**private ActionBarDrawerToggle mDrawerToggle;**

**private CharSequence mDrawerTitle;**

**// used to store app title**

**private CharSequence mTitle;**

**// slide menu items**

**private String[] navMenuTitles;**

**private TypedArray navMenuIcons;**

**private ArrayList<NavDrawerItem> navDrawerItems;**

**private NavDrawerListAdapter adapter;**

**String Id = null;**

**String firstNameText = null;**

**String lastNameText = null;**

**String emailText = null;**

**String mobileText = null;**

**String roleText = null;**

**TextView userText;**

**Bundle bundle = null;**

**@Override**

**protected void onCreate(Bundle savedInstanceState) {**

**super.onCreate(savedInstanceState);**

**setContentView(R.layout.activity\_options);**

**Bundle extras = getIntent().getExtras();**

**if (extras != null) {**

**Id = extras.getString("ID");**

**firstNameText = extras.getString("FIRST\_NAME");**

**lastNameText = extras.getString("LAST\_NAME");**

**emailText = extras.getString("USERNAME");**

**mobileText = extras.getString("MOBILE");**

**roleText = extras.getString("ROLE");**

**}**

**bundle = new Bundle();**

**bundle.putString("ID", Id);**

**bundle.putString("FIRST\_NAME", firstNameText);**

**bundle.putString("LAST\_NAME", lastNameText);**

**bundle.putString("USERNAME", emailText);**

**bundle.putString("MOBILE", mobileText);**

**bundle.putString("ROLE", roleText);**

**// userText = (TextView) findViewById(R.id.textViewUser);**

**// userText.setText("Welcome " + firstNameText);**

**mTitle = mDrawerTitle = getTitle();**

**// load slide menu items**

**navMenuTitles = getResources().getStringArray(R.array.nav\_drawer\_items);**

**// nav drawer icons from resources**

**navMenuIcons = getResources()**

**.obtainTypedArray(R.array.nav\_drawer\_icons);**

**mDrawerLayout = (DrawerLayout) findViewById(R.id.drawer\_layout);**

**mDrawerList = (ListView) findViewById(R.id.list\_slidermenu);**

**navDrawerItems = new ArrayList<NavDrawerItem>();**

**// adding nav drawer items to array**

**// Home**

**navDrawerItems.add(new NavDrawerItem(navMenuTitles[0], navMenuIcons**

**.getResourceId(0, -1)));**

**// Find People**

**navDrawerItems.add(new NavDrawerItem(navMenuTitles[1], navMenuIcons**

**.getResourceId(1, -1)));**

**// Photos**

**navDrawerItems.add(new NavDrawerItem(navMenuTitles[2], navMenuIcons**

**.getResourceId(2, -1)));**

**// Communities, Will add a counter here**

**navDrawerItems.add(new NavDrawerItem(navMenuTitles[3], navMenuIcons**

**.getResourceId(3, -1)));**

**// Logout**

**navDrawerItems.add(new NavDrawerItem(navMenuTitles[4], navMenuIcons**

**.getResourceId(4, -1)));**

**navDrawerItems.add(new NavDrawerItem(navMenuTitles[5], navMenuIcons**

**.getResourceId(5, -1)));**

**navDrawerItems.add(new NavDrawerItem(navMenuTitles[6], navMenuIcons**

**.getResourceId(6, -1)));**

**// Recycle the typed array**

**navMenuIcons.recycle();**

**mDrawerList.setOnItemClickListener(new SlideMenuClickListener());**

**// setting the nav drawer list adapter**

**adapter = new NavDrawerListAdapter(getApplicationContext(),**

**navDrawerItems);**

**mDrawerList.setAdapter(adapter);**

**// enabling action bar app icon and behaving it as toggle button**

**getSupportActionBar().setDisplayHomeAsUpEnabled(true);**

**getSupportActionBar().setHomeButtonEnabled(true);**

**mDrawerToggle = new ActionBarDrawerToggle(this, mDrawerLayout,**

**R.drawable.ic\_drawer, R.string.app\_name, 0) {**

**public void onDrawerClosed(View view) {**

**getSupportActionBar().setTitle(mTitle);**

**// calling onPrepareOptionsMenu() to show action bar icons**

**invalidateOptionsMenu();**

**}**

**public void onDrawerOpened(View drawerView) {**

**getSupportActionBar().setTitle(mDrawerTitle);**

**// calling onPrepareOptionsMenu() to hide action bar icons**

**invalidateOptionsMenu();**

**mDrawerLayout.bringToFront();**

**}**

**};**

**mDrawerLayout.setDrawerListener(mDrawerToggle);**

**if (savedInstanceState == null) {**

**// on first time display view for first nav item**

**displayView(0);**

**}**

**}**

**/\*\***

**\* Slide menu item click listener**

**\* \*/**

**private class SlideMenuClickListener implements**

**ListView.OnItemClickListener {**

**@Override**

**public void onItemClick(AdapterView<?> parent, View view, int position,**

**long id) {**

**// display view for selected nav drawer item**

**displayView(position);**

**}**

**}**

**@Override**

**public boolean onOptionsItemSelected(MenuItem item) {**

**// int id = item.getItemId();**

**if (item.getItemId() == R.id.action\_settings) {**

**Toast.makeText(getApplicationContext(), "Logging Out!",**

**Toast.LENGTH\_LONG).show();**

**Intent i = new Intent(AdminActivity.this, LoginActivity.class);**

**i.addFlags(Intent.FLAG\_ACTIVITY\_REORDER\_TO\_FRONT);**

**// i.addFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK);**

**startActivity(i);**

**// return true;**

**AdminActivity.this.finish();**

**}**

**if (mDrawerToggle.onOptionsItemSelected(item)) {**

**return true;**

**}**

**int itemId = item.getItemId();**

**if (itemId == R.id.action\_settings) {**

**return true;**

**} else {**

**return super.onOptionsItemSelected(item);**

**}**

**}**

**@Override**

**public boolean onCreateOptionsMenu(Menu menu) {**

**// Inflate the menu; this adds items to the action bar if it is present.**

**getMenuInflater().inflate(R.menu.options, menu);**

**return true;**

**}**

**/\*\*\***

**\* Called when invalidateOptionsMenu() is triggered**

**\*/**

**@Override**

**public boolean onPrepareOptionsMenu(Menu menu) {**

**// if nav drawer is opened, hide the action items**

**boolean drawerOpen = mDrawerLayout.isDrawerOpen(mDrawerList);**

**menu.findItem(R.id.action\_settings).setVisible(!drawerOpen);**

**return super.onPrepareOptionsMenu(menu);**

**}**

**/\*\***

**\* Diplaying fragment view for selected nav drawer list item**

**\* \*/**

**private void displayView(int position) {**

**// update the main content by replacing fragments**

**// System.out.println("inside method");**

**Fragment fragment = null;**

**switch (position) {**

**case 0:**

**fragment = new AdminHomeFragment();**

**fragment.setArguments(bundle);**

**break;**

**case 1:**

**fragment = new RegisterStudentFragment();**

**fragment.setArguments(bundle);**

**break;**

**case 2:**

**fragment = new RegisterStaffFragment();**

**fragment.setArguments(bundle);**

**break;**

**case 3:**

**fragment = new EventFragment();**

**fragment.setArguments(bundle);**

**break;**

**case 4:**

**fragment = new TimeTableFragment();**

**fragment.setArguments(bundle);**

**break;**

**case 5:**

**fragment = new EventCalendarFragment();**

**fragment.setArguments(bundle);**

**break;**

**case 6:**

**fragment = new ViewTimeTable();**

**fragment.setArguments(bundle);**

**break;**

**// case 6:**

**// fragment = new CmritForum();**

**// break;**

**default:**

**break;**

**}**

**if (fragment != null) {**

**FragmentManager fragmentManager = getSupportFragmentManager();**

**fragmentManager.beginTransaction()**

**.replace(R.id.frame\_container, fragment).commit();**

**// update selected item and title, then close the drawer**

**mDrawerList.setItemChecked(position, true);**

**mDrawerList.setSelection(position);**

**setTitle(navMenuTitles[position]);**

**mDrawerLayout.closeDrawer(mDrawerList);**

**} else {**

**// error in creating fragment**

**Log.e("MainActivity", "Error in creating fragment");**

**}**

**}**

**@Override**

**public void setTitle(CharSequence title) {**

**mTitle = title;**

**getSupportActionBar().setTitle(mTitle);**

**}**

**/\*\***

**\* When using the ActionBarDrawerToggle, you must call it during**

**\* onPostCreate() and onConfigurationChanged()...**

**\*/**

**@Override**

**protected void onPostCreate(Bundle savedInstanceState) {**

**super.onPostCreate(savedInstanceState);**

**// Sync the toggle state after onRestoreInstanceState has occurred.**

**mDrawerToggle.syncState();**

**}**

**@Override**

**public void onConfigurationChanged(Configuration newConfig) {**

**super.onConfigurationChanged(newConfig);**

**// Pass any configuration change to the drawer toggls**

**mDrawerToggle.onConfigurationChanged(newConfig);**

**}**

**}**

**LoginActivity**

**package com.istudent;**

**import java.util.List;**

**import android.app.Activity;**

**import android.content.Intent;**

**import android.os.Bundle;**

**import android.util.Log;**

**import android.view.View;**

**import android.view.View.OnClickListener;**

**import android.widget.Button;**

**import android.widget.EditText;**

**import android.widget.RadioGroup;**

**import android.widget.Toast;**

**import com.istudent.database.DatabaseHandler;**

**import com.istudent.model.Personal\_Details;**

**public class LoginActivity extends Activity {**

**EditText et\_username = null;**

**EditText et\_password = null;**

**RadioGroup radiogroup = null;**

**private DatabaseHandler mySQLiteAdapter;**

**String str\_firstName = null;**

**String str\_lastName = null;**

**String str\_username = null;**

**String str\_mobile = null;**

**String str\_role = null;**

**String str\_id = null;**

**String str\_password = null;**

**@Override**

**protected void onCreate(Bundle savedInstanceState) {**

**super.onCreate(savedInstanceState);**

**setContentView(R.layout.activity\_login);**

**// To connect to database**

**mySQLiteAdapter = new DatabaseHandler(getApplicationContext());**

**Button btn\_reset = (Button) findViewById(R.id.btn\_reset);**

**Button btn\_login = (Button) findViewById(R.id.login);**

**et\_username = (EditText) findViewById(R.id.username);**

**et\_password = (EditText) findViewById(R.id.password);**

**radiogroup = (RadioGroup) findViewById(R.id.grp\_role);**

**// Reset button functionality**

**btn\_reset.setOnClickListener(new OnClickListener() {**

**@Override**

**public void onClick(View arg0) {**

**Log.i("Login Activity", "Reset button clicked");**

**et\_username.setText("");**

**et\_password.setText("");**

**}**

**});**

**// Login button functionality**

**btn\_login.setOnClickListener(new OnClickListener() {**

**@Override**

**public void onClick(View arg0) {**

**int checkedId = radiogroup.getCheckedRadioButtonId();**

**if (checkedId == R.id.rb\_student) {**

**str\_role = "Student";**

**} else if (checkedId == R.id.rb\_staff) {**

**str\_role = "Staff";**

**} else if (checkedId == R.id.rb\_admin) {**

**str\_role = "Admin";**

**}**

**Log.i("Login Activity", "Login button clicked");**

**String str\_username = et\_username.getText().toString();**

**String str\_password = et\_password.getText().toString();**

**if (str\_username.isEmpty() && str\_password.isEmpty()) {**

**Log.i("Login Activity", "enter username and password");**

**Toast.makeText(getApplicationContext(),**

**"enter username and password", Toast.LENGTH\_SHORT)**

**.show();**

**} else if (str\_username.isEmpty()) {**

**Log.i("Login Activity", "enter username");**

**Toast.makeText(getApplicationContext(), "enter username ",**

**Toast.LENGTH\_SHORT).show();**

**} else if (str\_password.isEmpty()) {**

**Log.i("Login Activity", "enter password");**

**Toast.makeText(getApplicationContext(), "enter password",**

**Toast.LENGTH\_SHORT).show();**

**} else if (str\_username.equalsIgnoreCase("Abhiram")**

**&& str\_password.equals("mind")) {**

**if (str\_role.equals("Student") || str\_role.equals("Staff")) {**

**Toast.makeText(getApplicationContext(),**

**"Select the valid role", Toast.LENGTH\_SHORT)**

**.show();**

**}**

**else {**

**Intent i = new Intent(LoginActivity.this,**

**AdminActivity.class);**

**i.putExtra("ID", "Admin");**

**i.putExtra("FIRST\_NAME", "Abhiram");**

**i.putExtra("LAST\_NAME", "DM");**

**i.putExtra("USERNAME", "Abhiram");**

**i.putExtra("ROLE", "Admin");**

**i.putExtra("MOBILE", "9900165295");**

**startActivity(i);**

**}**

**<?php**

**include('config.php');**

**date\_default\_timezone\_set('Asia/Kolkata');**

**$time = date( 'h:i:s A', time () );**

**$cdate = date( 'y-m-d');**

**?>**

**<html lang="en">**

**<head>**

**<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">**

**<meta charset="UTF-8">**

**<meta name="viewport" content="width=device-width, initial-scale=1">**

**<style>**

**@font-face {**

**font-family:rajsoft;**

**src: url(rajsoftn.ttf);**

**}**

**body**

**{**

**margin: 0px;**

**font-family:rajsoft;**

**background-image: linear-gradient(to bottom , #f33649, #fa2545);**

**background-repeat: no-repeat;**

**color: white;**

**}**

**.text**

**{**

**background:white;**

**padding: 11px 40px;**

**border:solid white 1px;**

**text-decoration: none;**

**overflow: hidden;**

**color:black;**

**border-radius:22px;**

**outline: none;**

**}**

**select**

**{**

**background-color: transparent;**

**padding:11px;**

**border:solid white 1px;**

**text-decoration: none;**

**overflow: hidden;**

**color:white;**

**border-radius:22px;**

**outline: none;**

**width: 69%;**

**}**

**.send**

**{**

**background-image: linear-gradient(to left,white,white );**

**padding: 11px 20px;**

**border: none;**

**text-decoration: none;**

**overflow: hidden;**

**color: black;**

**border-radius:20px;**

**border:solid white 1px;**

**}**

**.rep**

**{**

**background-color: transparent;**

**padding: 11px ;**

**border: none;**

**text-decoration: none;**

**overflow: hidden;**

**color:white;**

**outline:none;**

**}**

**.set**

**{**

**background-color: transparent;**

**padding: 11px ;**

**border: none;**

**text-decoration: none;**

**overflow: hidden;**

**color:white;**

**outline:none;**

**}**

**.url**

**{**

**padding: 7px;**

**width: 250px;**

**}**

**.form1**

**{**

**padding: 11px;**

**width: 400px;**

**border: solid whitesmoke 2px;**

**height: 150px;**

**}**

**form**

**{**

**margin-top: 27%;**

**}**

**</style>**

**</head>**

**<body>**

**<form method="POST" action="#">**

**<center><div class="img">**

**<center>**

**<img src="icon.png"><br><BR><h1>Ready To Send</h1>**

**A Digital Automated SMS Messages<br>Version : 1.0<br><br>**

**<div class="url"><input name="userMobile" class="text" type="tel" pattern="[6789][0-9]{9}" placeholder="Mobile Number" id="number" required maxlength="10" oninvalid="Please Enter Proper Mobile Number" ><br></div><div>**

**<select name="userMessage" class="form-control clid" id="classid" required="required">**

**<?php**

**$sql = "SELECT \* FROM setting ORDER BY id DESC";**

**$result = $con->query($sql);**

**if ($result->num\_rows > 0) {**

**// output data of each row**

**while($row = $result->fetch\_assoc()) {**

**echo "<option value='{$row['message']}'>{$row['name']}</option>";**

**}**

**} else {**

**echo "<h1 style='color:coral;'>0</h1> Completed Grievances Found";**

**}**

**$con->close();**

**?>**

**</select><br></div>**

**<div class="url"><div style="float: right;">**

**<button type="button" onclick="window.location='home.php';" class="set"><i class="fa fa-home"></i>&nbsp;Home</button>**

**<button type="button" class="rep"onclick="clearAllFields()"><i class="fa fa-close"></i>&nbsp;Clear</button>**

**<button type="submit" class="send" name="SubmitButton" id="btnSend"><i class="fa fa-send-o"></i>&nbsp;Send</button></div>**

**</div></center></div></center></form>**

**</body>**

**<script type="text/javascript">**

**function clearAllFields(){**

**number.value="";**

**}**

**</script>**

**</html>**

**<?php**

**if(isset($\_POST['SubmitButton']))**

**{**

**$textMessage=$\_POST["userMessage"];**

**$mobileNumber=$\_POST["userMobile"];**

**$apiKey = urlencode('WsKqUfka5Wk-HdUjQkM13i0X86ei7dPxrgZXPsL17C');**

**// Message details**

**$numbers = array($mobileNumber);**

**$sender = urlencode('TXTLCL');**

**$message = rawurlencode($textMessage);**

**$numbers = implode(',', $numbers);**

**// Prepare data for POST request**

**$data = array('apikey' => $apiKey, 'numbers' => $numbers, "sender" => $sender, "message" => $message);**

**// Send the POST request with cURL**

**$ch = curl\_init('https://api.textlocal.in/send/');**

**curl\_setopt($ch, CURLOPT\_POST, true);**

**curl\_setopt($ch, CURLOPT\_POSTFIELDS, $data);**

**curl\_setopt($ch, CURLOPT\_RETURNTRANSFER, true);**

**$response = curl\_exec($ch);**

**curl\_close($ch);**

**$haystack = $response;**

**//$haystack = "success";**

**$needle = "success";**

**if( strpos( $haystack, $needle ) !== false) {**

**include('config.php');**

**$query = mysqli\_query($con,"insert into report (phone, message, date,** time)values('$mobileNumber','$textMessage','$cdate','$time')");

**if($query == true)**

**{**

**echo "<script>alert('Message sent Successfully!.');</script>";**

**}**

**else**

**{**

**echo "<script>alert('Could Not Save message\\nMessage sent Successfully!.');</script>";**

**}**

**}**

**else**

**{**

**echo "<script>alert('Could Not Send Message!.\\nPlease check your Credit Balance.');</script>";**

**}**

**}**

**?>**

**<?php**

**include('config.php');**

**?>**

**<html>**

**<head>**

**<link rel="stylesheet" href="w3.css">**

**<script type="text/javascript" src="jquery-1.7.1.min.js"></script>**

**<script type="text/javascript" src="jspdf.debug.js"></script>**

**<meta charset="UTF-8">**

**<meta name="viewport" content="width=device-width, initial-scale=1">**

**<style>**

**body**

**{**

**margin: 0px;**

**font-family: lato;**

**}**

**.head**

**{**

**position: fixed;**

**background-color: #2d2e30;**

**color: white;**

**width: 100%;**

**padding: 11px;**

**}**

**.link**

**{**

**float: right;**

**padding: 9px;**

**}**

**a**

**{**

**color: silver;**

**overflow: hidden;**

**text-decoration: none;**

**}**

**.send**

**{**

**background-image: linear-gradient(to left,#1953ac,#2067d6 );**

**padding: 11px;**

**border: none;**

**text-decoration: none;**

**overflow: hidden;**

**color: white;**

**border-radius:2px;**

**}**

**.form**

**{**

**position:static;**

**padding: 11px;**

**}**

**@media print {**

**.res {**

**visibility: hidden;**

**}**

**}**

**</style>**

**<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">**

**</head>**

**<body>**

**<div class="head"><font style="font-size:17px;"><i class="fa fa-cog fa-fw"></i></font><font style="font-size:24px;">&nbsp;History</font>**

**<div class="link"><a href="home.php">Home</a> | <a href="#" style="color:white">History&nbsp;&nbsp;&nbsp;&nbsp;</a></div>**

**</div>**

**<br><br><br>**

**<form method="POST" action="#">**

**<center> <div class="form">**

**<h2 style="color:#1953ac">Message History**

**</h2>**

**<font color="gray">Full details About messages which you have been sent!.</font><br><button style="padding: 6px 20px;background-color:#0080ff;color: white;cursor:pointer;outline: none;border: 0px;" OnClick="demoFromHTML()">Print All Message History</button><a href="#" onclick="window.open('https://www.google.com/', '\_system');" >Google</a>**

**<br>**

**<div class="res" id="fromHTMLtestdiv">**

**<?php**

**$sql = "SELECT \* FROM `report` ORDER BY id DESC";**

**$result = $con->query($sql);**

**if ($result->num\_rows > 0) {**

**// output data of each row**

**while($row = $result->fetch\_assoc()) {**

**echo "<br><div id={$row['id']}><div style='font-size:17;background-image:whitesmoke;padding:11px; background-image: linear-gradient(to top , whitesmoke , white);'>Message Id :{$row['id']}</div><table border='0' width='100%' style='border:solid silver 1px;'><tr><th style='border:solid whitesmoke 1px;background-color:whitesmoke;width:150px;'>".**

**"<tr><th style='border:solid silver 1px;text-align:left;padding:11px;background-color:whitesmoke'>Phone Number<br></th></tr>".**

**"<tr><td style=padding:11px>{$row['phone']} <br> </td></tr> ".**

**"<tr><th style='border:solid silver 1px;text-align:left;padding:11px;background-color:whitesmoke'>Message<br></th></tr>".**

**"<tr><td style=padding:11px>{$row['message']} <br> </td></tr> ".**

**"<tr><th style='border:solid silver 1px;text-align:left;padding:11px;background-color:whitesmoke'>Date<br></th></tr>".**

**"<tr><td style=padding:11px>{$row['date']} <br> </td></tr> ".**

**"<tr><th style='border:solid silver 1px;text-align:left;padding:11px;background-color:whitesmoke'>Time**

**<br></th></tr>".**

**"<tr><td style=padding:11px>{$row['time']} <br> </td></tr><br></table></div><div> <div style='font-size:17;background:whitesmoke;padding:11px;border:solid silver 1px;height:55px;'><input type='button' style='float:right;padding:4px;background-color:#0080ff;cursor:pointer;color:white;outline: none;border: 0px;'**

**onclick=CallPrin(this) name={$row['id']} value='Print Current'/>&nbsp;&nbsp;<input type='button' style='float:left;padding:4px;background-color:#fd5a5e;cursor:pointer;color:white;outline: none;border: 0px;'**

**onclick=window.location='re.php?cid={$row['id']}' value='Delete'/></div>";**

**}**

**} else {**

**echo "<h1 style='color:#1953ac;'>0</h1>Message History found";**

**}**

**$con->close();**

**?>**

**</div></div></center>**

**</form>**

**<script>**

**function CallPrin(str) {**

**var c = "#" + str.name;**

**var pdf = new jsPDF('p', 'pt', 'letter')**

**// source can be HTML-formatted string, or a reference**

**// to an actual DOM element from which the text will be scraped.**

**, source = $(c)[0]**

**// we support special element handlers. Register them with jQuery-style**

**// ID selector for either ID or node name. ("#iAmID", "div", "span" etc.)**

**// There is no support for any other type of selectors**

**// (class, of compound) at this time.**

**, specialElementHandlers = {**

**// element with id of "bypass" - jQuery style selector**

**'#bypassme': function(element, renderer){**

**// true = "handled elsewhere, bypass text extraction"**

**return true**

**}**

**}**

**margins = {**

**top: 40,**

**bottom: 40,**

**left: 45,**

**width: 522**

**};**

**// all coords and widths are in jsPDF instance's declared units**

**// 'inches' in this case**

**pdf.fromHTML(**

**source // HTML string or DOM elem ref.**

**, margins.left // x coord**

**, margins.top // y coord**

**, {**

**'width': margins.width // max width of content on PDF**

**, 'elementHandlers': specialElementHandlers**

**},**

**function (dispose) {**

**// dispose: object with X, Y of the last line add to the PDF**

**// this allow the insertion of new lines after html**

**var now = Math.floor(Math.random() \* 105675471);**

**var name = "Message\_history-" + now;**

**pdf.save(name);**

**},**

**margins**

**)**

**}**

**function demoFromHTML() {**

**var pdf = new jsPDF('p', 'pt', 'letter')**

**// source can be HTML-formatted string, or a reference**

**// to an actual DOM element from which the text will be scraped.**

**, source = $('#fromHTMLtestdiv')[0]**

**// we support special element handlers. Register them with jQuery-style**

**// ID selector for either ID or node name. ("#iAmID", "div", "span" etc.)**

**// There is no support for any other type of selectors**

**// (class, of compound) at this time.**

**, specialElementHandlers = {**

**// element with id of "bypass" - jQuery style selector**

**'#bypassme': function(element, renderer){**

**// true = "handled elsewhere, bypass text extraction"**

**return true**

**}**

**}**

**margins = {**

**top: 40,**

**bottom: 40,**

**left: 45,**

**width: 522**

**};**

**// all coords and widths are in jsPDF instance's declared units**

**// 'inches' in this case**

**pdf.fromHTML(**

**source // HTML string or DOM elem ref.**

**, margins.left // x coord**

**, margins.top // y coord**

**, {**

**'width': margins.width // max width of content on PDF**

**, 'elementHandlers': specialElementHandlers**

**},**

**function (dispose) {**

**// dispose: object with X, Y of the last line add to the PDF**

**// this allow the insertion of new lines after html**

**var now = Math.floor(Math.random() \* 105675471);**

**var name = "Message\_history-" + now;**

**pdf.save(name);**

**},**

**margins**

**)**

**}**

**</script>**

**</body>**

**</html>**

**<?php**

**include('config.php');**

**if(isset($\_POST['SubmitButton'])){**

**$name = $\_POST['name'];**

**$message = $\_POST['mesaage'];**

**$query = mysqli\_query($con,"insert into setting (name, message)values('$name','$message')");**

**if($query == true)**

**{**

**echo "<script>alert('Message Added Successfully\\nThis Message will appeared in Send SMS form You can Select by Message Subject.');</script>";**

**}**

**else**

**{**

**echo "<script>alert('Could Not add message\\nSmoothing went wrong please Contact admin to fix this problems!.');</script>";**

**}**

**}**

**?>**

**<html>**

**<head>**

**<meta charset="UTF-8">**

**<meta name="viewport" content="width=device-width, initial-scale=1">**

**<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">**

**<style>**

**@font-face {**

**font-family:rajsoft;**

**src: url(rajsoftn.ttf);**

**}**

**body**

**{**

**margin: 0px;**

**font-family:rajsoft;**

**}**

**.head**

**{**

**position: fixed;**

**background-image: linear-gradient(to bottom , #f33649, #fa2545);**

**color: white;**

**width: 100%;**

**padding: 10px;**

**}**

**.link**

**{**

**float: right;**

**padding: 9px;**

**}**

**a**

**{**

**color: silver;**

**overflow: hidden;**

**text-decoration: none;**

**}**

**.text**

**{**

**background-color: white;**

**border: solid #f2f2f2 1px;**

**padding: 11px;**

**width: 250px;**

**outline: none;**

**text-decoration: none;**

**overflow: hidden;**

**color: #2c2c2c;**

**border-radius:2px;**

**}**

**.text1**

**{**

**background-color: white;**

**border: solid #f2f2f2 1px;**

**padding: 11px;**

**width: 250px;**

**height: 130px;**

**outline: none;**

**text-decoration: none;**

**overflow: hidden;**

**color: #2c2c2c;**

**border-radius:2px;**

**}**

**.send**

**{**

**background-image: linear-gradient(to bottom , #f33649, #fa2545);**

**padding: 11px;**

**border: none;**

**text-decoration: none;**

**overflow: hidden;**

**color: white;**

**border-radius:2px;**

**}**

**.form**

**{**

**position: static;**

**}**

**.url**

**{**

**padding: 7px;**

**width: 250px;**

**text-align: left;**

**}**

**.url1**

**{**

**padding: 7px;**

**width: 250px;**

**text-align: right;**

**}**

**</style></head>**

**<body>**

**<div class="head"><font style="font-size:17px;"><i class="fa fa-cog fa-fw"></i></font><font style="font-size:24px;">&nbsp;Settings</font>**

**<div class="link"><a href="home.php"><i class="fa fa-home"></i></a>&nbsp; | &nbsp;<a href="#" style="color:white"><i class="fa fa-plus"></i></a> &nbsp;|&nbsp; <a href="remove.php"><i class="fa fa-trash"></i>&nbsp;&nbsp;&nbsp;&nbsp;</a></div>**

**</div><br><br><br><br>**

**<form method="POST" action="#">**

**<center><div class="form">**

**<h2 style="color:#f12e33">Add New Message</h2>**

**<div style="width:84%;"><font color="gray">Please enter the subject and message description, And click on add button.</font></div><br><br>**

**<div class="url">**

**<label>Subject</label><br><br>**

**<input required name="name" type="text" class="text" placeholder="Subject"></div>**

**<div class="url">**

**<label>Message Description**

**</label><br><br>**

**<textarea required name="mesaage" maxlength="160" class="text1" placeholder="Message Description**

**"></textarea></div>**

**<div class="url1">**

**<button type="submit" class="send" name="SubmitButton" id="btnSend"><i class="fa fa-save"></i>&nbsp;&nbsp;Add & Save</button></div>**

**</div></center>**

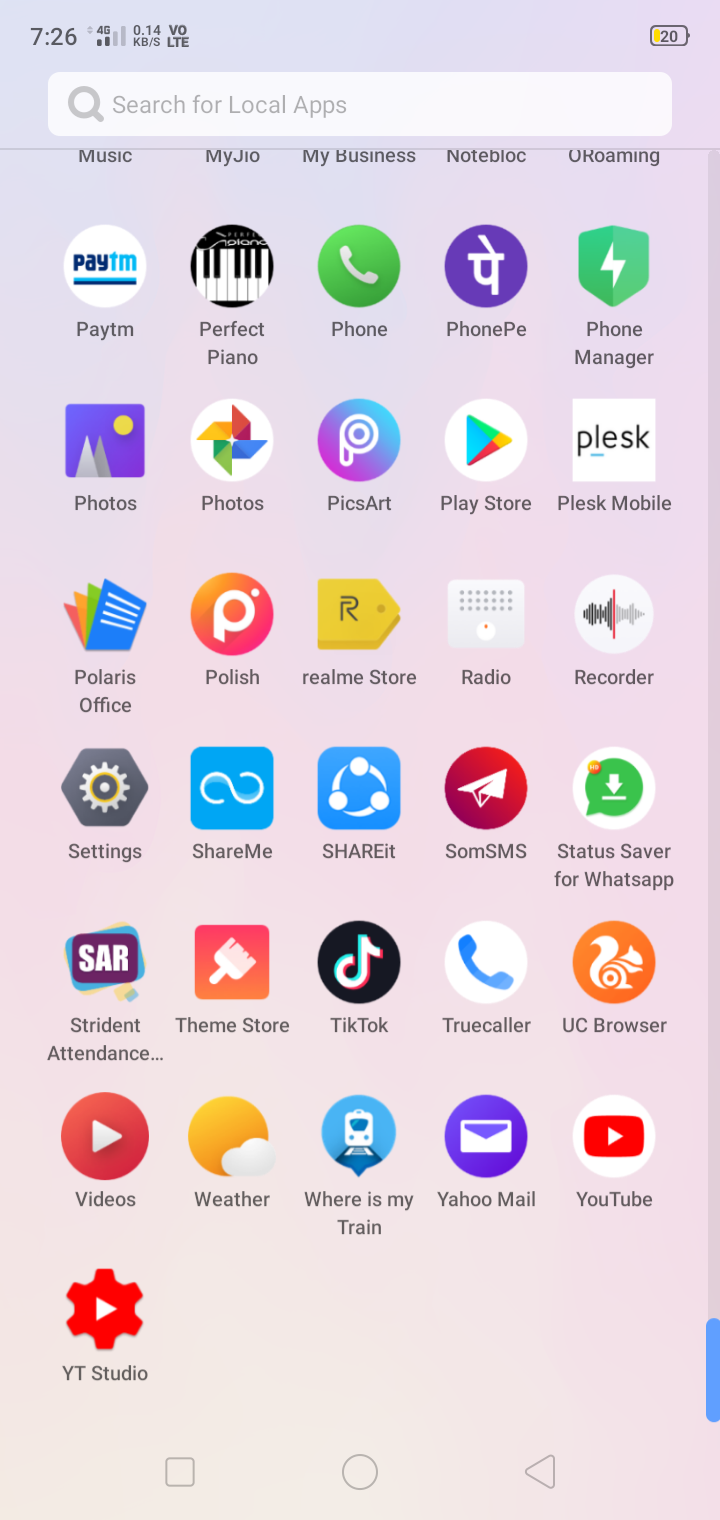
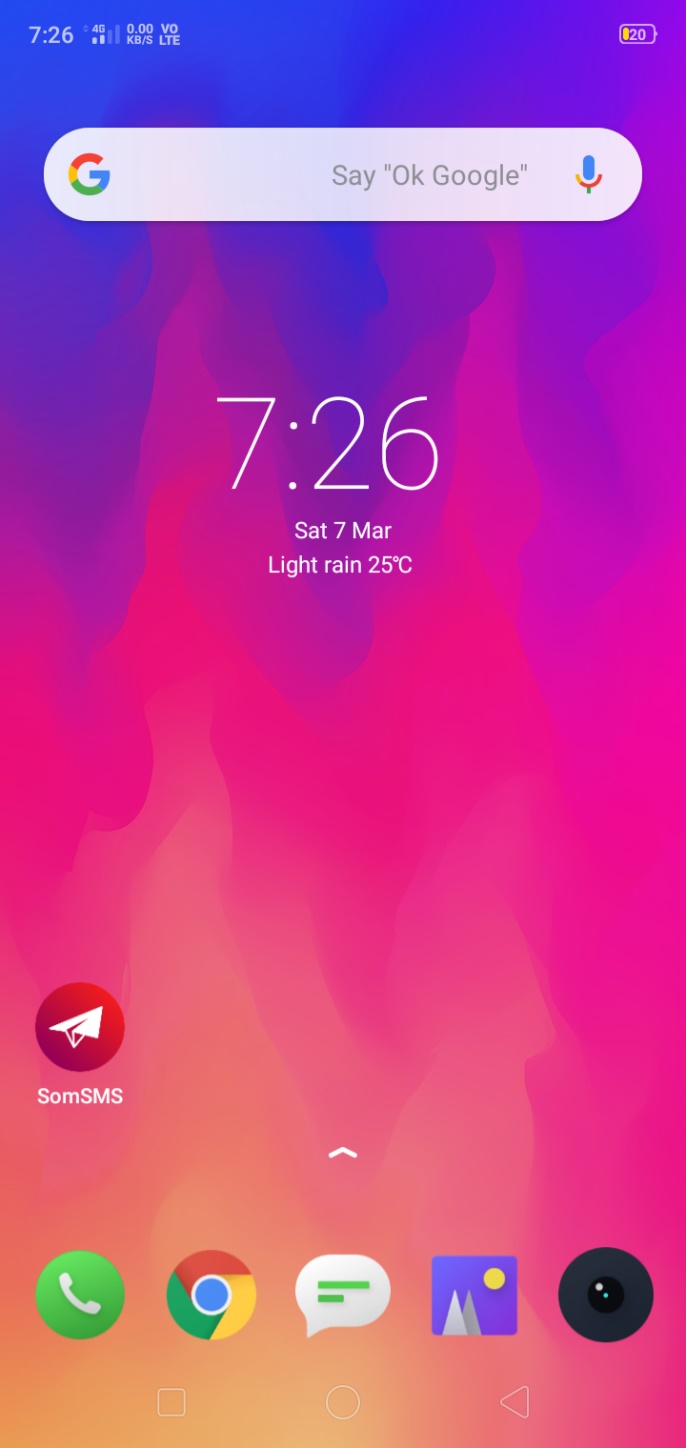
**</form>**

**</body>**

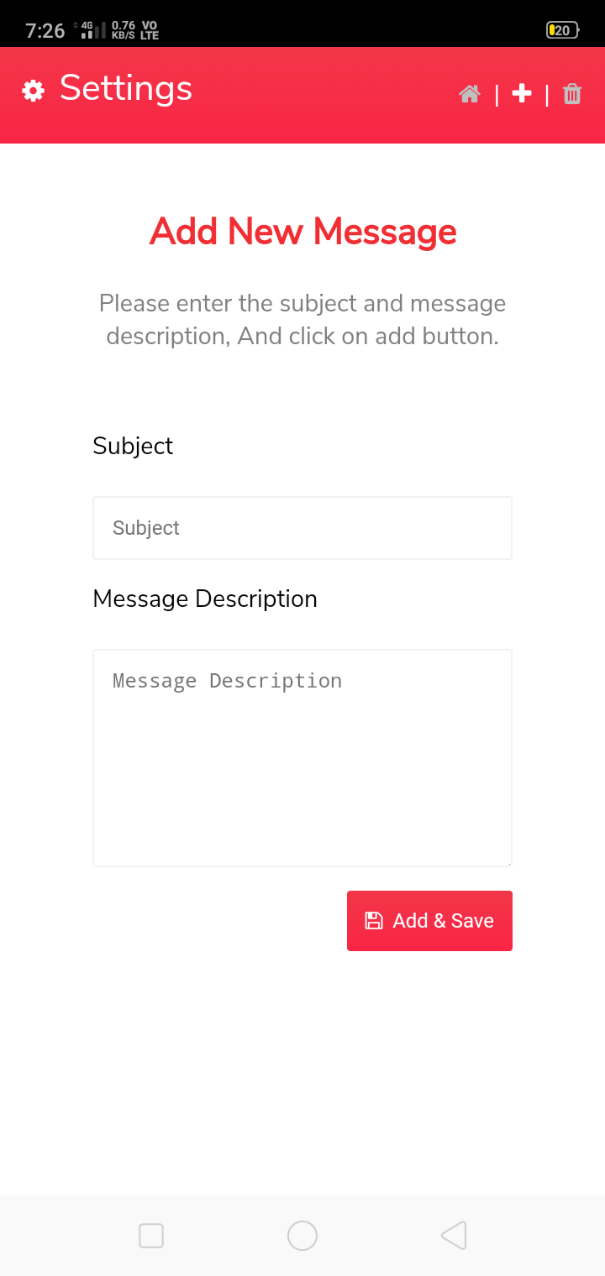
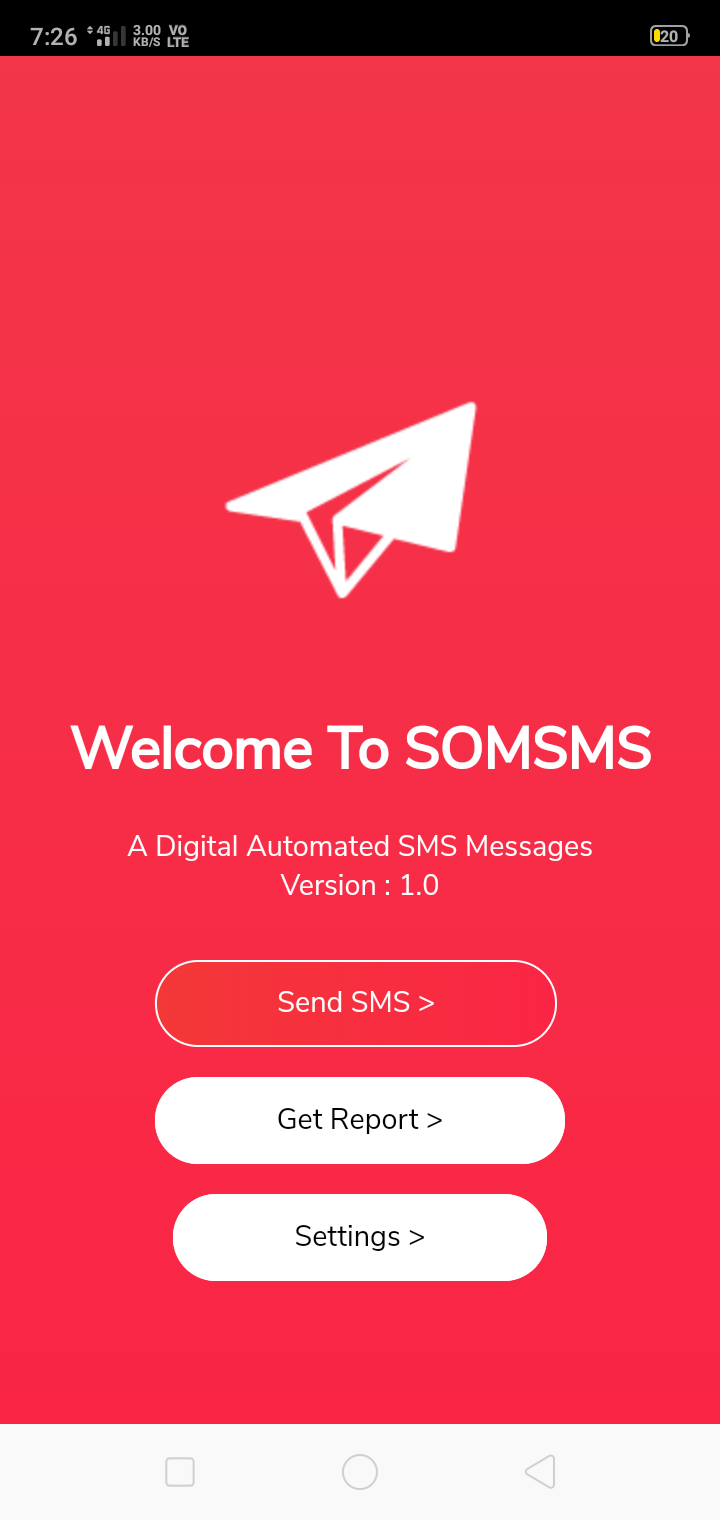
**</html>**

**Screen Shots**

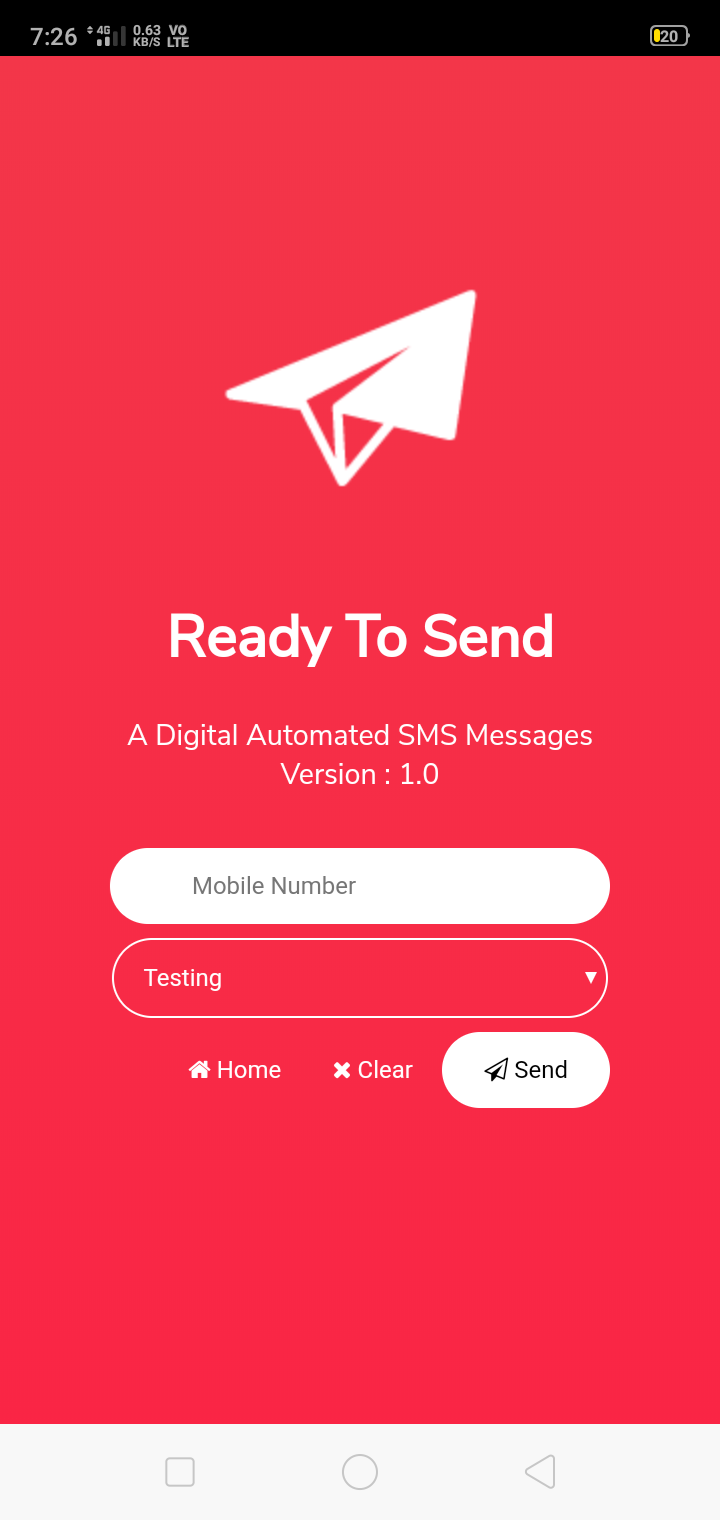
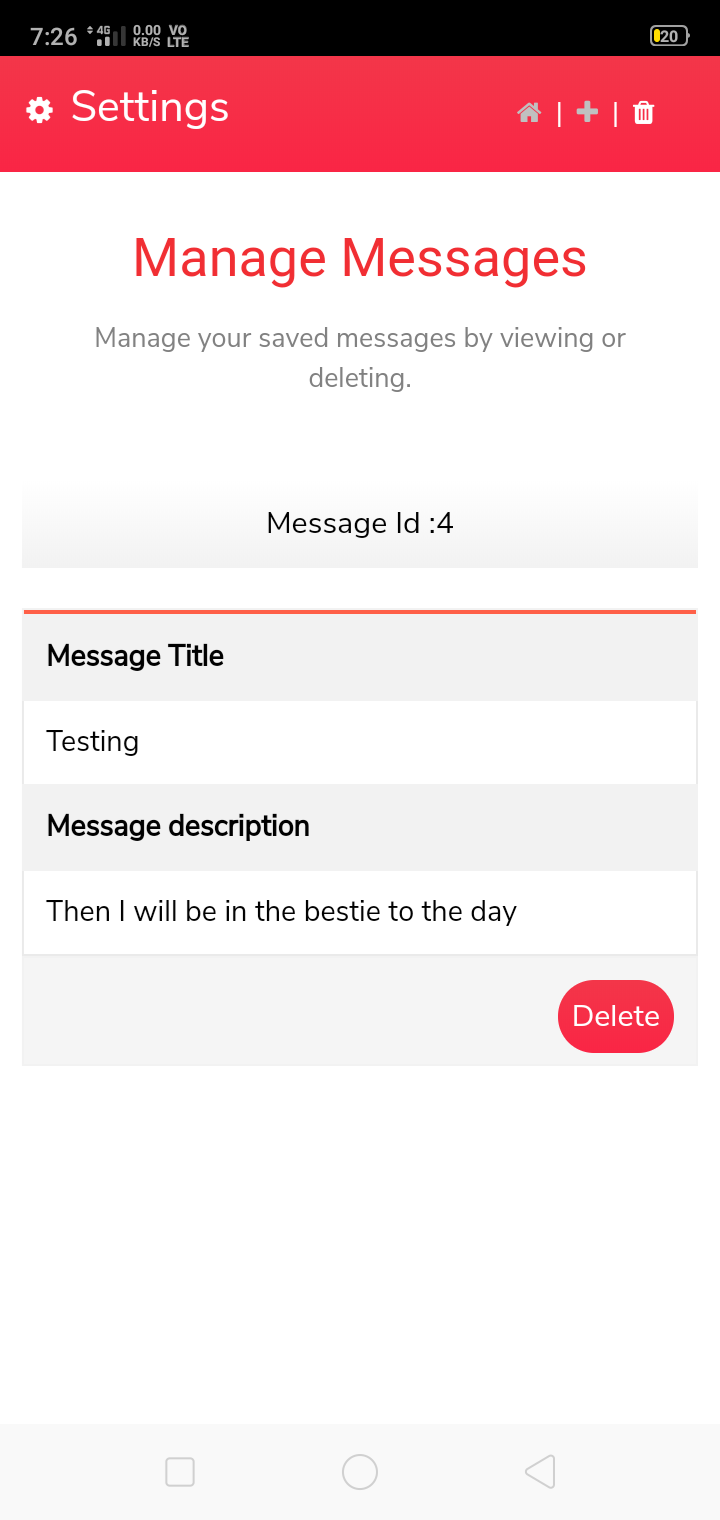
**Screen Shortcut**

****

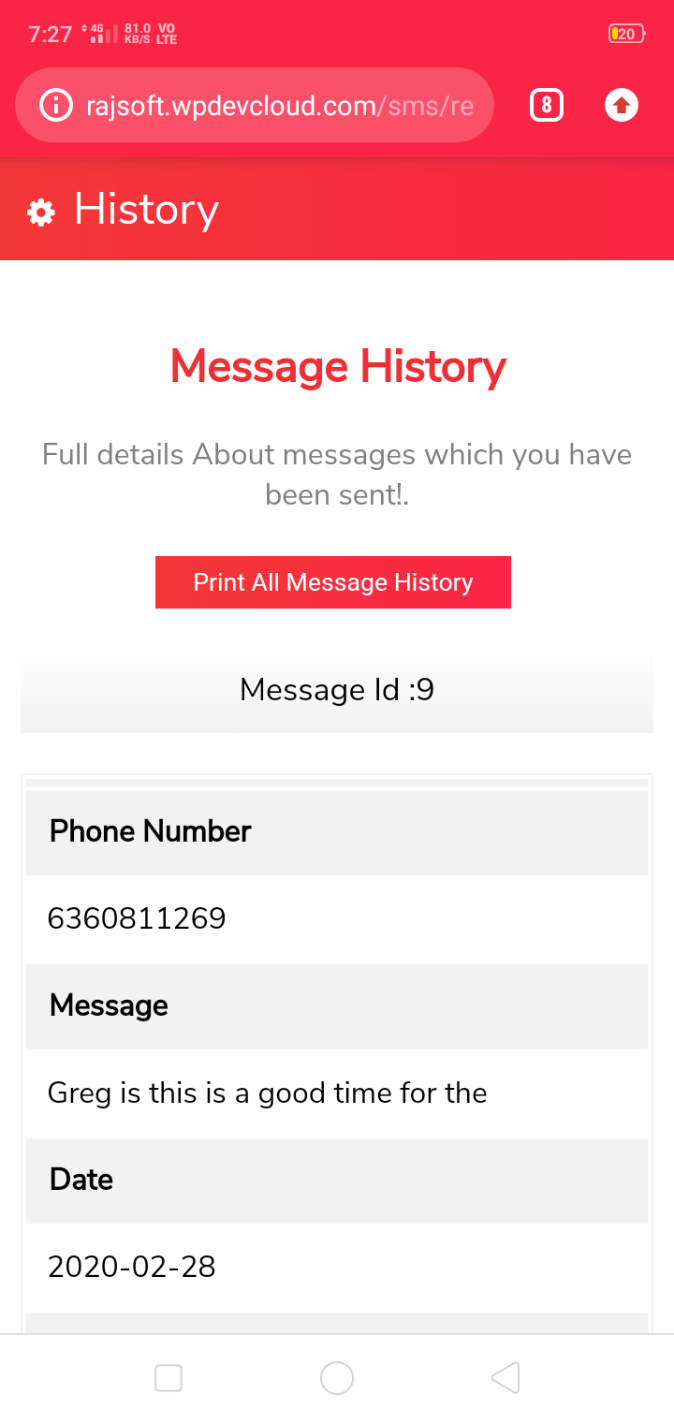
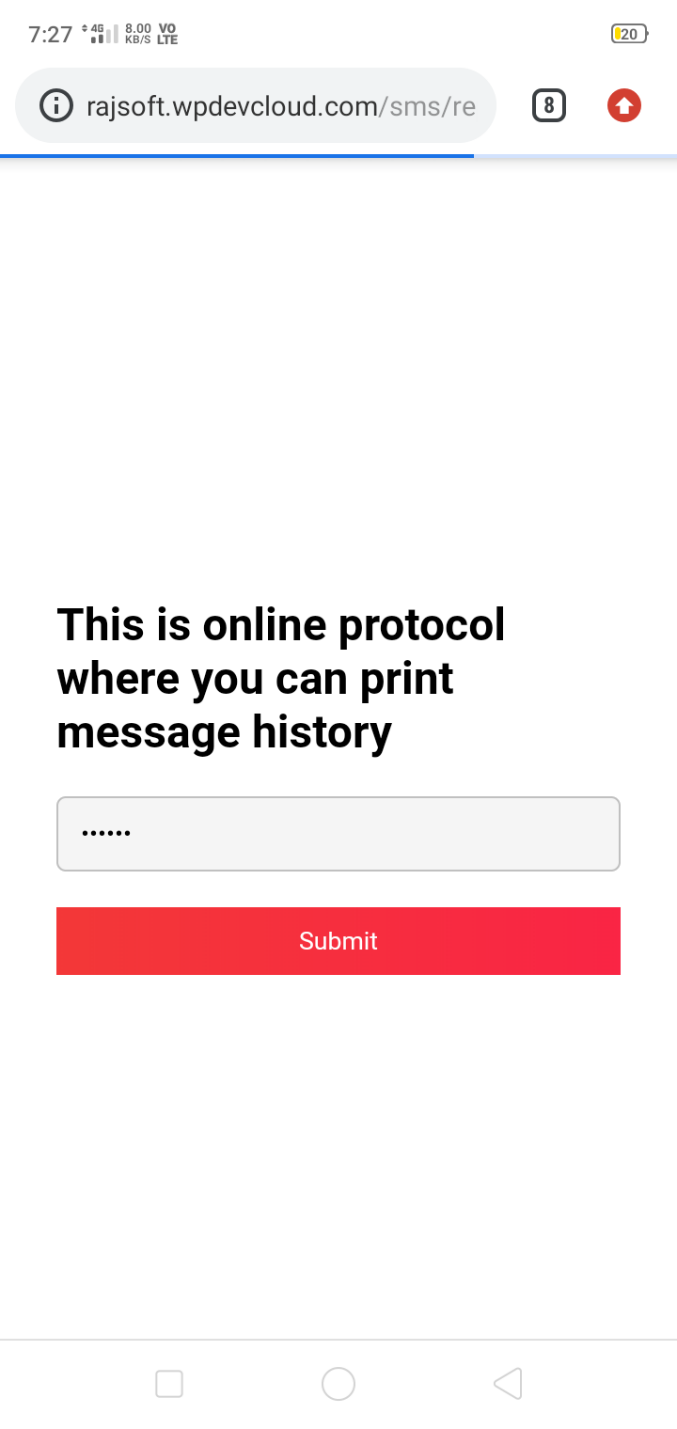
**App Home Screen & App Settings Activity**

****

**App Manage Activity & App Send SMS Activity**

****

**Get report Activity**

****

**Security**

Each Request and response activities are encrypted and decrypted in instruction between application and server.

**Features of project**

1. Less Storage Used
2. Supports Android version 4.0 to later
3. Simple and mobile compellable
4. High speed performance
5. SMS with there delivered status
6. Login Is secured
7. Admin can manage users
8. Get all record details as a PDF or Direct print.
9. Easy and secured updates by POST method

**Conclusion**

The project is designed for the purpose to reduce the internet quality assurance cell time and to reduce the burden of maintaining details of student while attendance shortage. Hence its very useful software for sending a SMS to parents.Automated report generation either print direct or PDF. Using this software it reduces Student absence and bunking of classes.

**References and Biography**

<https://www.w3schools.com/w3css/>

https://developer.android.com/guide

<https://www.tutorialspoint.com/android/index.htm>

<https://www.tutorialspoint.com/php/index.htm>

<https://github.com/>