SHRI VILE PARLE KELVANI MANDAL‘s

Shri Bhagubhai Mafatlal Polytechnic

Vile Parle (West), Mumbal- 400 056.

Project Report On:

SMARTCAB

Submitted By:

Vidhi Panchal (1498020)

Nishi Pancholi (1498022)

Mayank Parvatia (1498024)

Guided By:

Mr. Pankaj Rathod

Head of Department:

Mrs. Neeta Kadukar

Technician:

Mrs. Reshma Parab

Department Of Information Technology

2016 - 2017

PREFACE

The curriculum of INFORMATION TECHNOLOGY at SHRI BHAGUBHAI MAFATLAL POLYTHENIC extends to us, students an opportunity to change our ideas into realities. All the knowledge and information gathered is expected to be put to use, and a fruitful output is to be generated. This Final year project, has given us the chance to develop an Android application.

This document shall guide you through our efforts, study and implementation along with aids to develop an application. This experience shall prove to be helpful in our future walks life. We regard it as a great start to our practical outlook and approach.

ACKNOWLEDGEMENT

The success of this project would not have been possible without constant advice, encouragement and support from a vast number of people. Ideas, they say, often remain ideas. Very few of them, if find support, are turned into actual working models. And rest of them are just forgotten, and remain what they started off as; ideas.

This idea wouldn’t have been in force if it wasn’t for our guide, to whom we owe gratitude. We, the members of team develop the **SMARTCAB** are delighted to take this opportunity, to acknowledge whole heartedly, the guidance and support extended to us, by our guide.

We would like to thank our guide **‘Mr. PANKAJ RATHOD’ ,** he never turned down any of our requests when we needed help and guidance. We would also like to thank him for the amount of support and freedom he bestowed upon us without which the completion of this project would not have been possible.

A big thanks to **‘Mrs. RESHMA PARAB’** for looking after our needs in the lab, being patient with our continual software demands, and her encouraging support.

We would like to thank to the entire staff of **'INFORMATION TECHNOLOGY AND COMPUTER DEPARTMENT'** for providing us a helping hand when we most needed it.

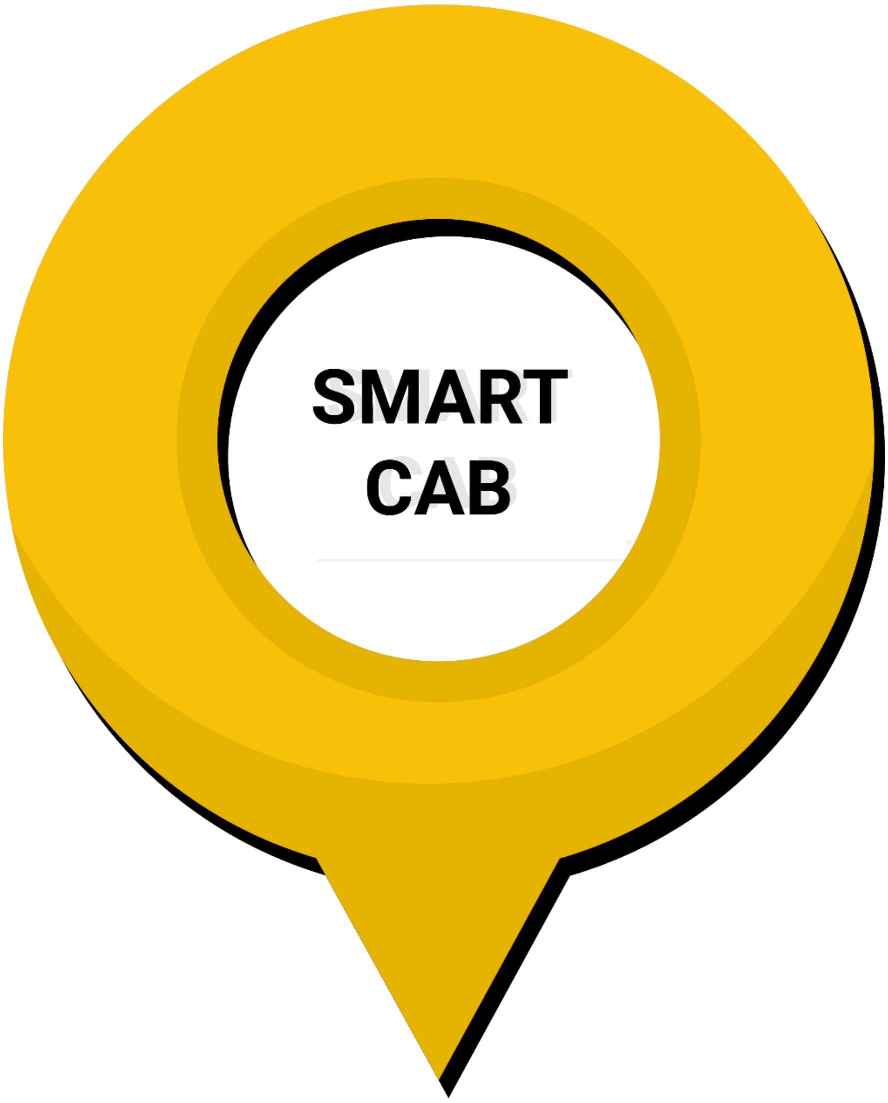
We also thank our H.O.D **'Mrs. NEETA KADUKAR'** for leading us the advice and helping us in our needs for making our project more successful.

Thanking you-

Vidhi Panchal (1498020)

Nishi Pancholi (1498022)

Mayank Parvatia (1498024)

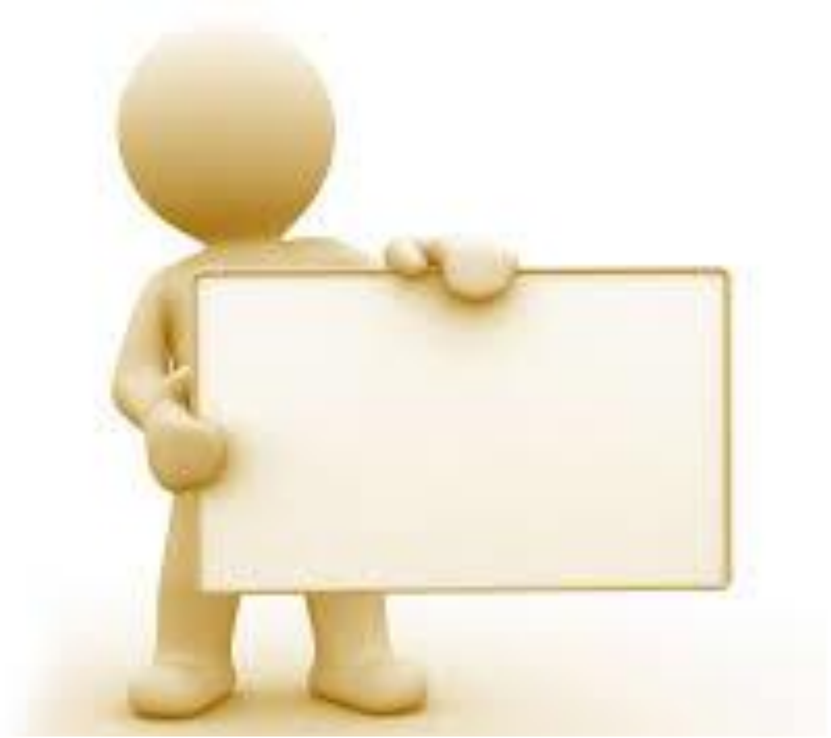

SMART
CAB

Index

ABSTRACT

Smart Cab is a cab search and price comparison application. Here, we provide a facility to the users for booking the car of his/her own choice. Users can compare fare prices, and then select a perfect cab. In this application, you don’t need to redirect to another application for booking the cab. Here, you make a booking directly with the application. One cannot access or alter your booking in any way but you can cancel your booking. Once you are done with the booking, you will get an email to the verified account. This application helps you to compare between the prices and get the best band for your buck.

INTRODUCTION



SMART CAB is an application which compares between the rates of the cars one wants to ride. This application will help you to get the best car at the best price. We can check the distance and estimated time to choose the perfect cab. This application will make it simple for you to get the required car as per your choice. For e.g. Normal Taxi, Sedan, SUV, Luxury Car. Here the user can also view the history of the rides he has booked through the application. The admin can know the amount of the cars booked and the total revenue generated by the companies and thus they can get their commission. This application is very useful to compare the taxis and book them in an easier way without any complications in the booking.

FEATURES

- Sign Up
- Login
- Different Cab Companies
- Comparing prices between different cabs.
- Easy booking
- Displays the least fare price cab.
- Provides multi-language support.
- User-friendly environment.
- Rides can be filtered by type of car (E.g. Sedan, SUV, etc.).
- Email to the verified account.
- History of the rides
- Easy cancellation

PROBLEM STATEMENT

Not long ago, booking a cab was complicated and time consuming. It was not easy to compare prices- always leaving you unsure if you got the best bang for your buck. Then came companies like OLA, UBER and TabCab which provides easy cab booking facilities.

SMART CAB allows users to efficiently compare fare prices of different companies. This improvement in the cab booking experience not only reduces the booking fear, but also makes sure that the cab companies deliver a great experience. For the first time, users could get a strong understanding of how much they have to pay before the ride. As people are more likely to use smart phones, this android application has a very user-friendly environment. This application also allows users to view their previous bookings.

SOFTWARE REQUIREMENTS

SOFTWARE USED:

ANDROID STUDIO

Android Studio is the official integrated development environment (IDE) for the Android platform.

It was announced on May 16, 2013 at the Google I/O conference. Android Studio is freely available under the Apache License 2.0.

Android Studio was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0.

Based on JetBrains' IntelliJ IDEA software, Android Studio is designed specifically for Android development.[8] It is available for download on Windows, macOS and Linux, and replaced Eclipse Android Development Tools (ADT) as Google's primary IDE for native Android application development.

SUBLIME TEXT

Sublime Text is a proprietary cross-platform source code editor with a Python application programming interface (API). It natively supports many programming languages and mark-up, and its functionality can be extended by users with plugins, typically community-built and maintained under free-software licenses.

- "Goto Anything," quick navigation to files, symbols, or lines
- "Command palette" uses adaptive matching for quick keyboard invocation of arbitrary commands
- Simultaneous editing: simultaneously make the same interactive changes to multiple selected areas
- Python-based plugin API
- Project-specific preferences
- Extensive customizability via JSON settings files, including project-specific and platform-specific settings
- Cross platform (Windows, OS X, Linux)
- Compatible with many language grammars from TextMate.

XAMPP

XAMPP is a free and open source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages XAMPP stands for Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) and Perl (P). It is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing and deployment purposes. Everything needed to set up a web server – server application (Apache), database (MariaDB), and scripting language (PHP) – is included in an extractable file. XAMPP is also cross-platform, which means it works equally well on Linux, Mac and Windows. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server extremely easy as well.

FILEZILLA

FileZilla is a free software, cross-platform FTP application, consisting of FileZilla Client and FileZilla Server. Client binaries are available for Windows, Linux, and macOS, server binaries are available for Windows only. The client supports FTP, SFTP and FTPS (FTP over SSL/TLS).

FileZilla's source code is hosted on SourceForge and the project was featured as Project of the Month in November 2003. However, there have been criticisms that SourceForge bundles malicious software with the application; and that FileZilla stores users' FTP passwords insecurely.

FileZilla Server is a sister product to FileZilla Client. It is an FTP server supported by the same project and features support for FTP and FTP over SSL/TLS.[24] FileZilla Server is currently available only on the Windows platform.

FileZilla Server is a free, open source FTP server. Its source code is hosted on SourceForge.net.

MYSQL WORKBENCH

MySQL Workbench is a visual database design tool that integrates SQL development, administration, database design, creation and maintenance into a single integrated development environment for the MySQL database system. It is the successor to DBDesigner 4 from fabFORCE.net, and replaces the previous package of software, MySQL GUI Tools Bundle.

- Database Connection & Instance Management
- Wizard driven action items
- Fully scriptable with Python and Lua.
- Support for custom plugins
- MSAA (Windows Accessibility API) compliant
- Supports MySQL Enterprise features (Audit Log, Firewall, and Enterprise Backup)
- Schema object browsing, inspection, and search
- SQL syntax highlighter and statement parser
- SQL code completion and context sensitive help.
- SQL snippets collections
- SSH connection tunneling
- Unicode support

HARDWARE REQUIREMENTS

ANDROID DEVICE

Android is a mobile operating system developed by Google, based on the Linux kernel and designed primarily for touchscreen mobile devices such as smartphones and tablets. Android's user interface is mainly based on direct manipulation, using touch gestures that loosely correspond to real-world actions, such as swiping, tapping and pinching, to manipulate on-screen objects, along with a virtual keyboard for text input. In addition to touchscreen devices, Google has further developed Android TV for televisions, Android Auto for cars, and Android Wear for wrist watches, each with a specialized user interface. Variants of Android are also used on notebooks, game consoles, digital cameras, and other electronics.

SERVER SPACE

A Server space is very essential in order to host some webpages across the web. A web hosting service is a type of Internet hosting service that allows individuals and organizations to make their website accessible via the World Wide Web. Web hosts are companies that provide space on a server owned or leased for use by clients, as well as providing Internet connectivity, typically in a data center. Web hosts can also provide data center space and connectivity to the Internet for other servers located in their data center, called colocation, also known as Housing in Latin America or France.

LANGUAGES USED

JAVA

Java is a general-purpose computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of computer architecture. As of 2016, Java is one of the most popular programming languages in use, particularly for client-server web applications, with a reported 9 million developers. Java was originally developed by James Gosling at Sun Microsystems (which has since been acquired by Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++, but it has fewer low-level facilities than either of them.

PHP

PHP is a server-side scripting language designed primarily for web development but also used as a general-purpose programming language. Originally created by Rasmus Lerdorf in 1994, the PHP reference implementation is now produced by The PHP Development Team. PHP originally stood for Personal Home Page, but it now stands for the recursive acronym PHP: Hypertext Preprocessor.

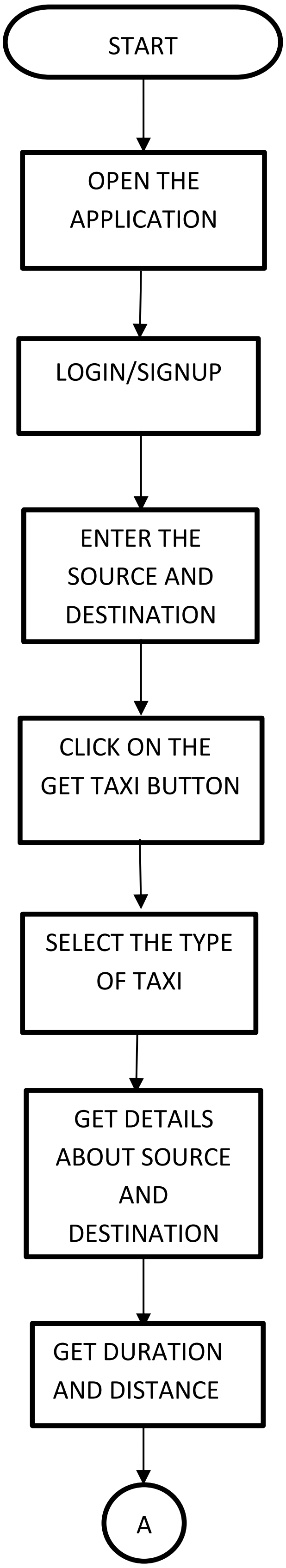
PHP code may be embedded into HTML or HTML5 code, or it can be used in combination with various web template systems, web content management systems and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable. The web server combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a command-line interface (CLI) and can be used to implement standalone graphical applications.

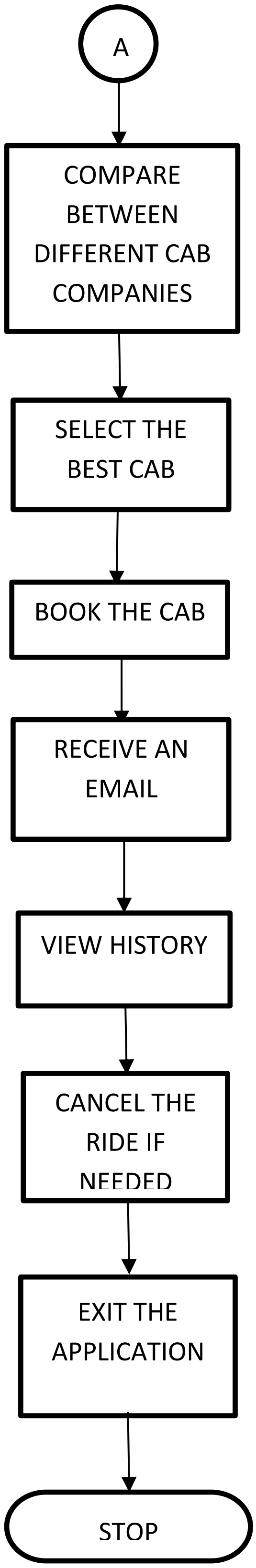
AJAX

Ajax is a set of Web development techniques using many Web technologies on the client side to create asynchronous Web applications. With Ajax, Web applications can send data to and retrieve from a server asynchronously (in the background) without interfering with the display and behavior of the existing page. By decoupling the data interchange layer from the presentation layer, Ajax allows for Web pages, and by extension Web applications, to change content dynamically without the need to reload the entire page.[4] In practice, modern implementations commonly substitute JSON for XML due to the advantages of being native to JavaScript.

Ajax is not a single technology, but rather a group of technologies. HTML and CSS can be used in combination to mark up and style information. The DOM is accessed with JavaScript to dynamically display – and allow the user to interact with – the information presented. JavaScript and the XMLHttpRequest object provide a method for exchanging data asynchronously between browser and server to avoid full page reloads.

FLOWCHART


START
OPEN THE APPLICATION
LOGIN/SIGNUP
ENTER THE SOURCE AND DESTINATION
CLICK ON THE GET TAXI BUTTON
SELECT THE TYPE OF TAXI
GET DETAILS ABOUT SOURCE AND DESTINATION
GET DURATION AND DISTANCE
A

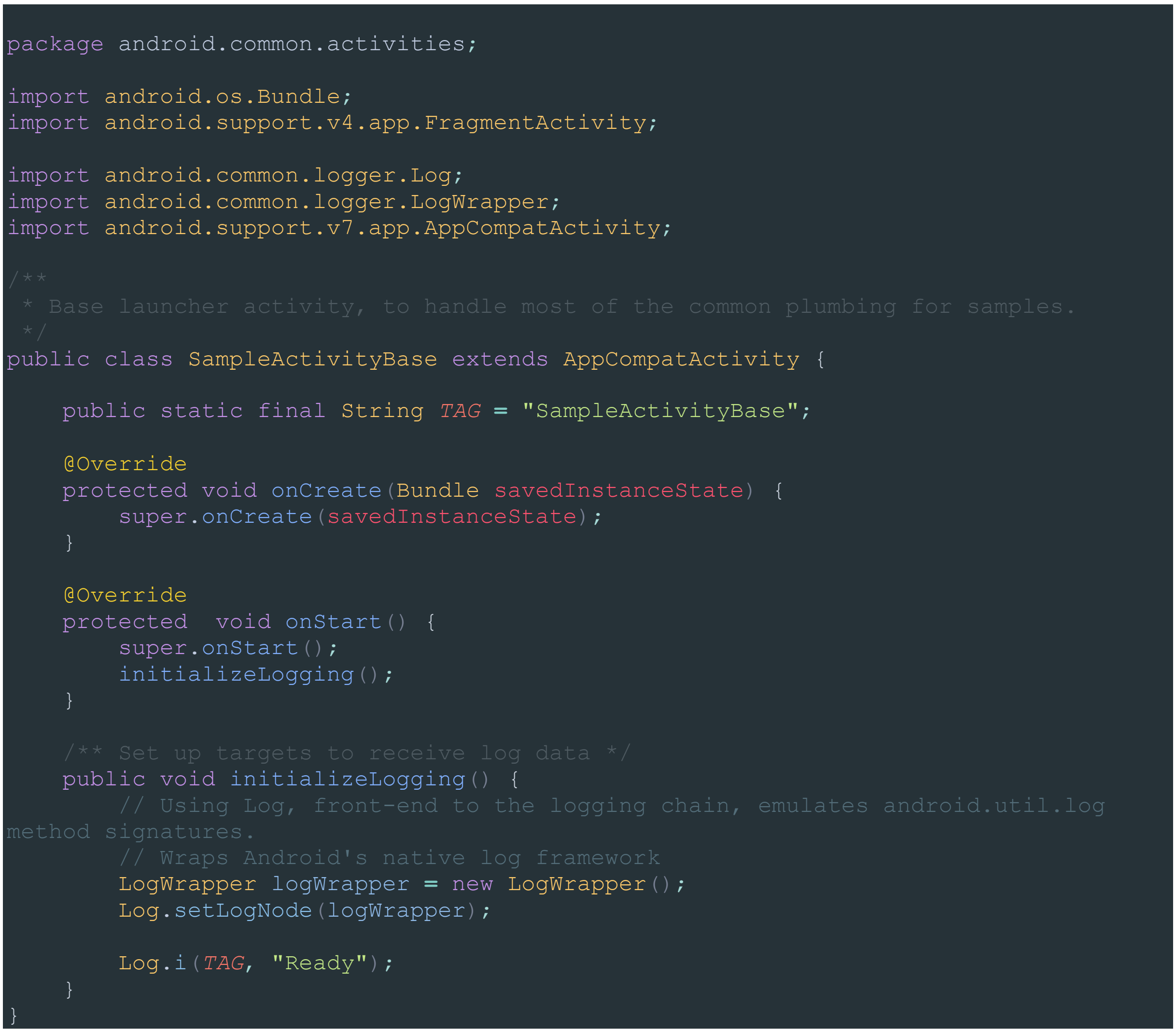

A
COMPARE BETWEEN DIFFERENT CAB COMPANIES
SELECT THE BEST CAB
BOOK THE CAB
RECEIVE AN EMAIL
VIEW HISTORY
CANCEL THE RIDE IF NEEDED
EXIT THE APPLICATION
STOP

SOURCE CODE

AndroidManifest.xml

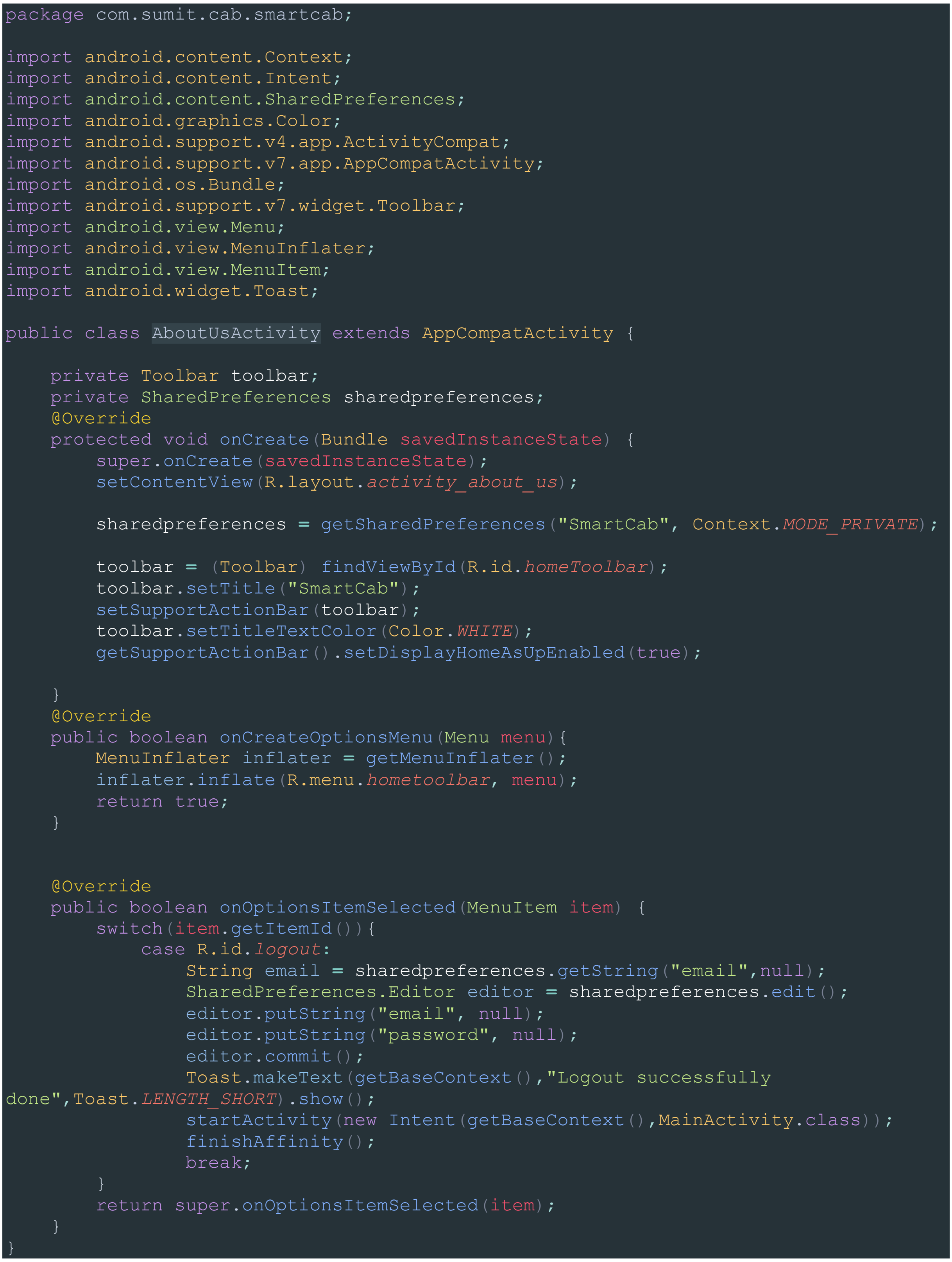
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.sumit.cab.smartcab">

    <uses-permission android:name="android.permission.CALL_PHONE" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.INTERNET" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <meta-data
            android:name="com.google.android.geo.API_KEY"
            android:value="AIzaSyBsVvSPrLvlqkrGR88uo4828-m9_PRpV0Y" />

        <activity android:name=".MapActivity" />
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".LoginActivity" />
        <activity android:name=".SignUpActivity" />
        <activity
            android:name=".PriceListActivity"
            android:parentActivityName=".TaxiListDemoActivity">
            <meta-data
                android:name="android.support.PARENT_ACTIVITY"
                android:value=".TaxiListDemoActivity" />
        </activity>
        <activity
            android:name=".TaxiListDemoActivity"
            android:label="@string/taxitype"
            android:parentActivityName=".MapActivity">
            <meta-data
                android:name="android.support.PARENT_ACTIVITY"
                android:value=".MainActivity" />
        </activity>
        <activity
            android:name=".AboutUsActivity"
            android:parentActivityName=".MapActivity">
            <meta-data
                android:name="android.support.PARENT_ACTIVITY"
                android:value=".MapActivity" />
        </activity>
        <activity
            android:name=".ContActivity"
            android:parentActivityName=".MapActivity">
            <meta-data
                android:name="android.support.PARENT_ACTIVITY"
                android:value=".MapActivity" />
        </activity>
        <activity
            android:name=".History.HistoryActivity"
            android:parentActivityName=".MapActivity">
            <meta-data
                android:name="android.support.PARENT_ACTIVITY"
                android:value=".MapActivity" />
        </activity>
        >
        <activity android:name=".History.ListAdapter" />
        <activity
            android:name=".History.DisplayActivty"
            android:parentActivityName=".History.HistoryActivity">
            <meta-data
                android:name="android.support.PARENT_ACTIVITY"
                android:value=".MapActivity" />
        </activity>
        >
        <activity android:name=".SuccessAcivity" />
        <activity android:name=".BookedActivity"
            android:label="Booking Done"></activity>
    </application>

</manifest>
```

SamplebaseActivity.java



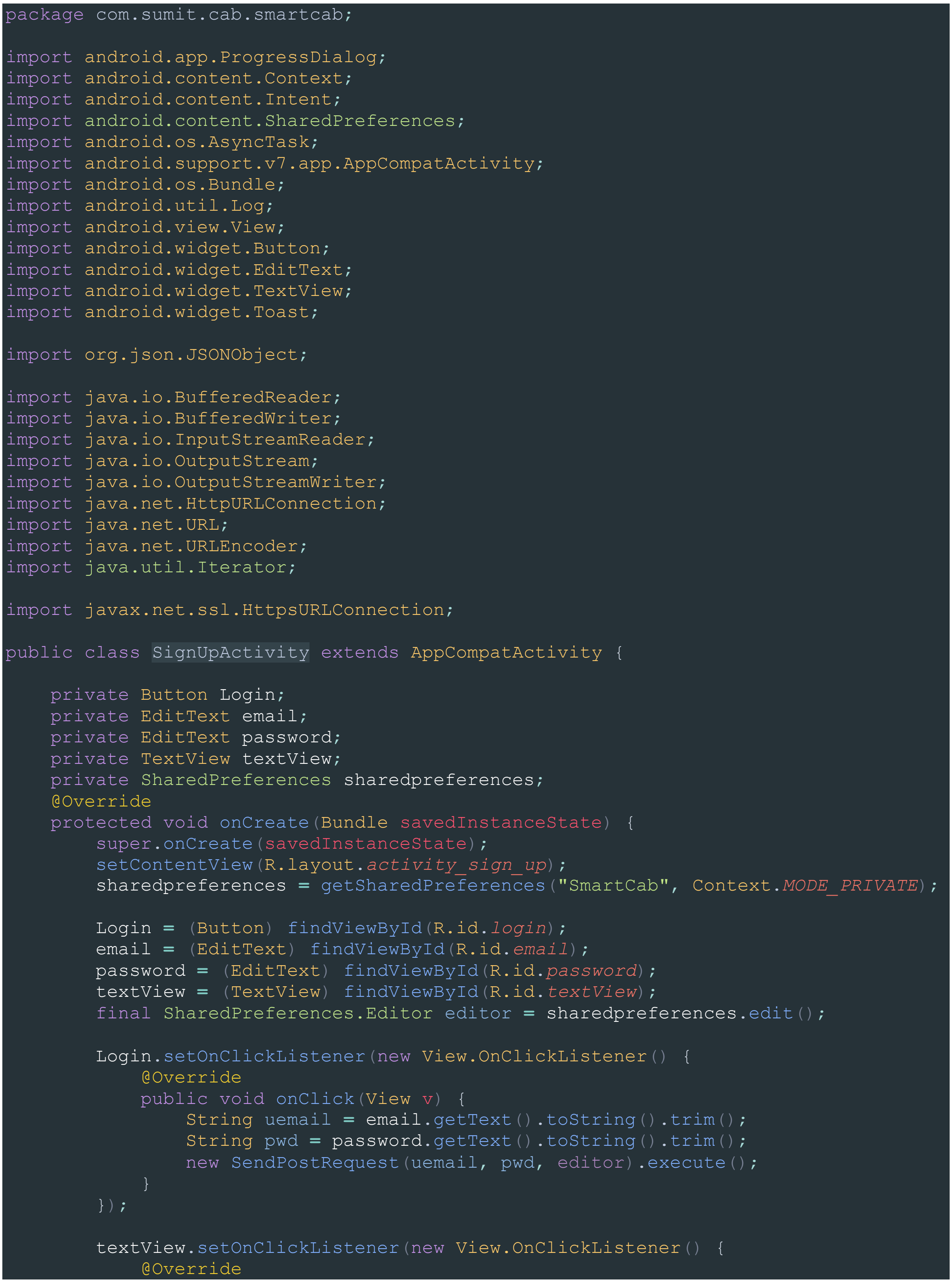
```java
package android.common.activities;

import android.os.Bundle;
import android.support.v4.app.FragmentActivity;

import android.common.logger.Log;
import android.common.logger.LogWrapper;
import android.support.v7.app.AppCompatActivity;

/**
 * Base launcher activity, to handle most of the common plumbing for samples.
 */
public class SampleActivityBase extends AppCompatActivity {

    public static final String TAG = "SampleActivityBase";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }

    @Override
    protected  void onStart() {
        super.onStart();
        initializeLogging();
    }

    /** Set up targets to receive log data */
    public void initializeLogging() {
        // Using Log, front-end to the logging chain, emulates android.util.log
method signatures.
        // Wraps Android's native log framework
        LogWrapper logWrapper = new LogWrapper();
        Log.setLogNode(logWrapper);

        Log.i(TAG, "Ready");
    }
}
```

AboutUsActivity.java

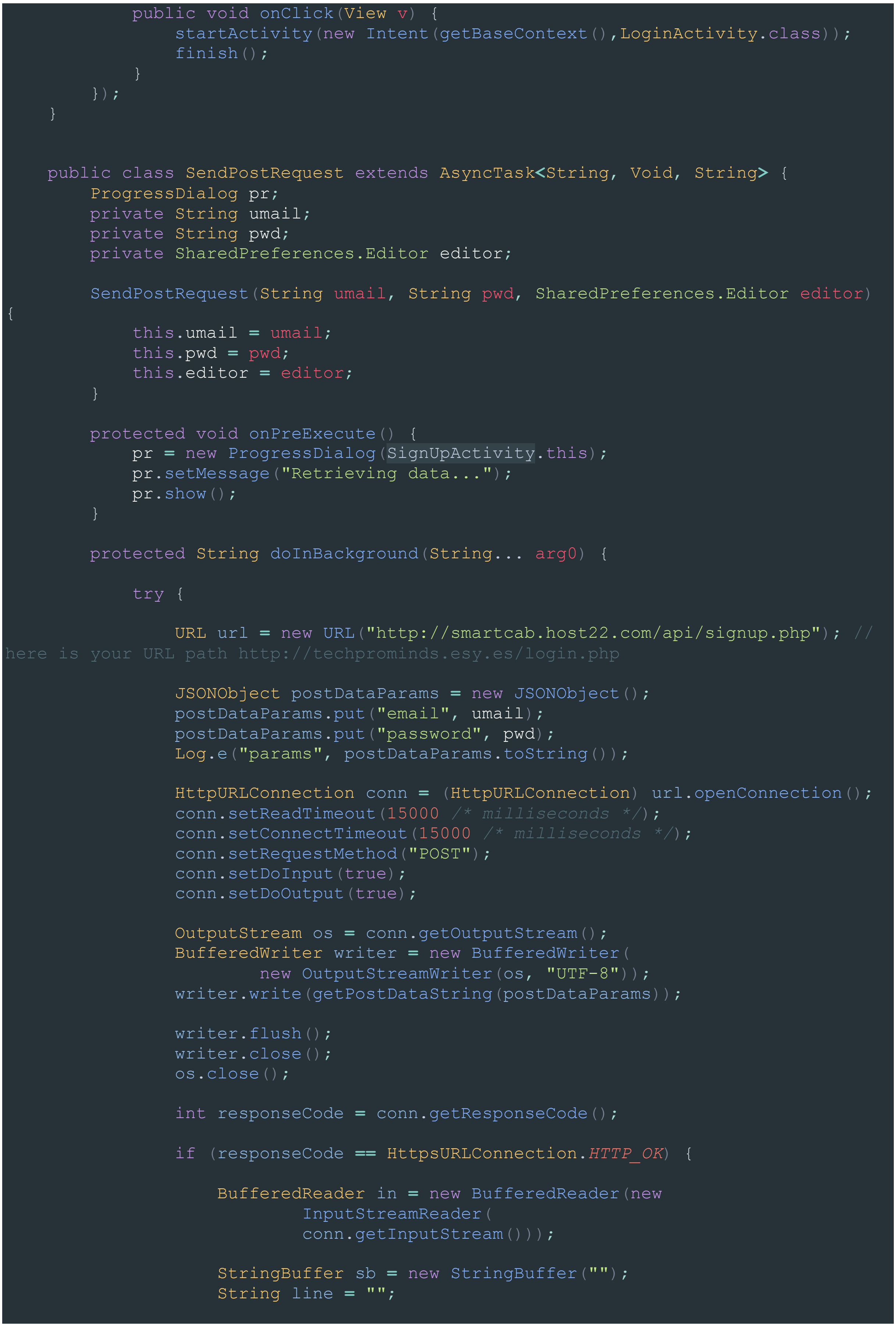


```java
package com.sumit.cab.smartcab;

import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.Color;
import android.support.v4.app.ActivityCompat;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.support.v7.widget.Toolbar;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.widget.Toast;

public class AboutUsActivity extends AppCompatActivity {

    private Toolbar toolbar;
    private SharedPreferences sharedpreferences;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_about_us);

        sharedpreferences = getSharedPreferences("SmartCab", Context.MODE_PRIVATE);

        toolbar = (Toolbar) findViewById(R.id.homeToolbar);
        toolbar.setTitle("SmartCab");
        setSupportActionBar(toolbar);
        toolbar.setTitleTextColor(Color.WHITE);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);

    }
    @Override
    public boolean onCreateOptionsMenu(Menu menu){
        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.hometoolbar, menu);
        return true;
    }


    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch(item.getItemId()){
            case R.id.logout:
                String email = sharedpreferences.getString("email",null);
                SharedPreferences.Editor editor = sharedpreferences.edit();
                editor.putString("email", null);
                editor.putString("password", null);
                editor.commit();
                Toast.makeText(getBaseContext(),"Logout successfully
done",Toast.LENGTH_SHORT).show();
                startActivity(new Intent(getBaseContext(),MainActivity.class));
                finishAffinity();
                break;
        }
        return super.onOptionsItemSelected(item);
    }
}
```

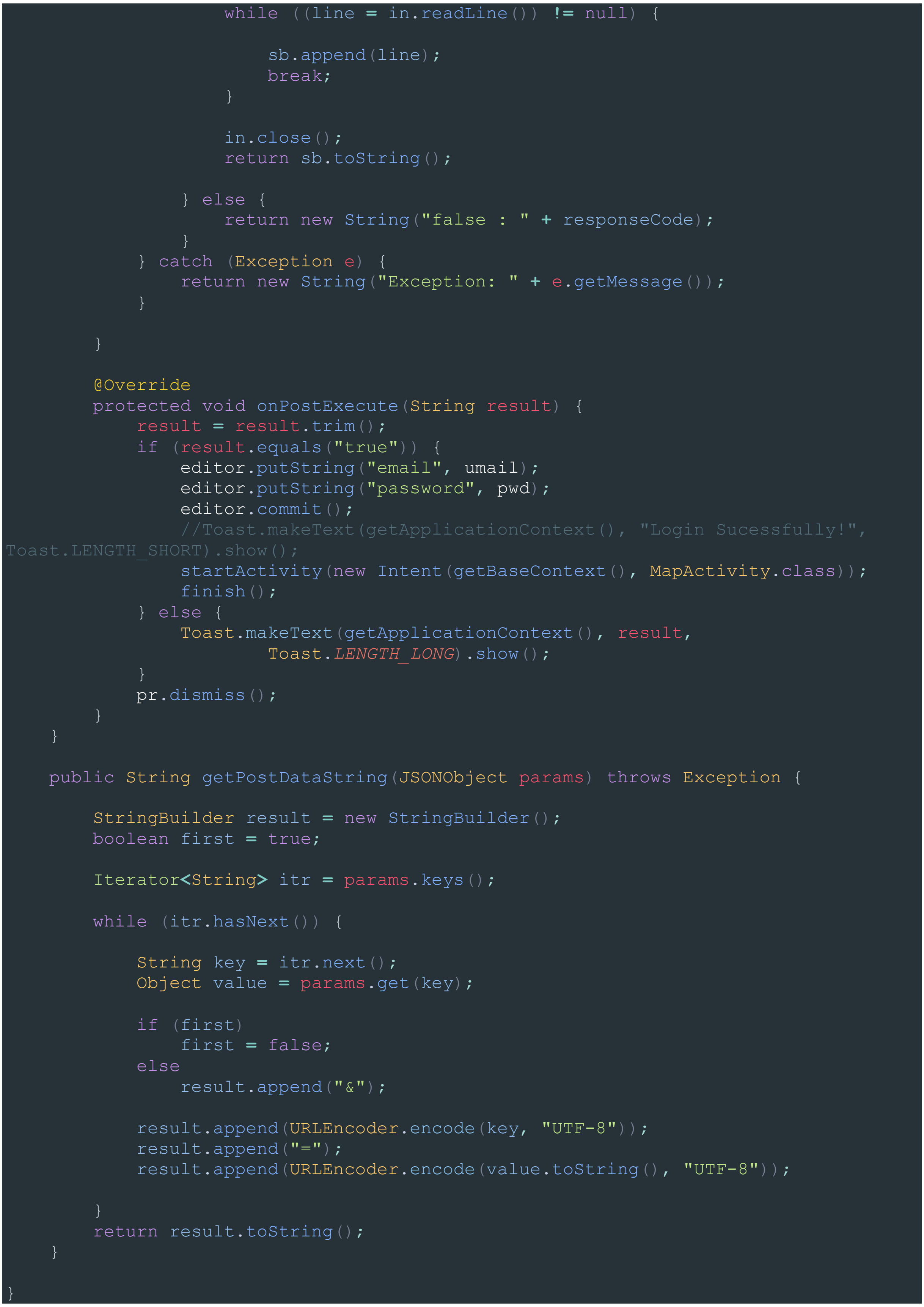
SignUpActivity.java



```java
package com.sumit.cab.smartcab;

import android.app.ProgressDialog;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.AsyncTask;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import org.json.JSONObject;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.URL;
import java.net.URLEncoder;
import java.util.Iterator;

import javax.net.ssl.HttpsURLConnection;

public class SignUpActivity extends AppCompatActivity {

    private Button Login;
    private EditText email;
    private EditText password;
    private TextView textView;
    private SharedPreferences sharedpreferences;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_sign_up);
        sharedpreferences = getSharedPreferences("SmartCab", Context.MODE_PRIVATE);

        Login = (Button) findViewById(R.id.login);
        email = (EditText) findViewById(R.id.email);
        password = (EditText) findViewById(R.id.password);
        textView = (TextView) findViewById(R.id.textView);
        final SharedPreferences.Editor editor = sharedpreferences.edit();

        Login.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String uemail = email.getText().toString().trim();
                String pwd = password.getText().toString().trim();
                new SendPostRequest(uemail, pwd, editor).execute();
            }
        });

        textView.setOnClickListener(new View.OnClickListener() {
            @Override
```



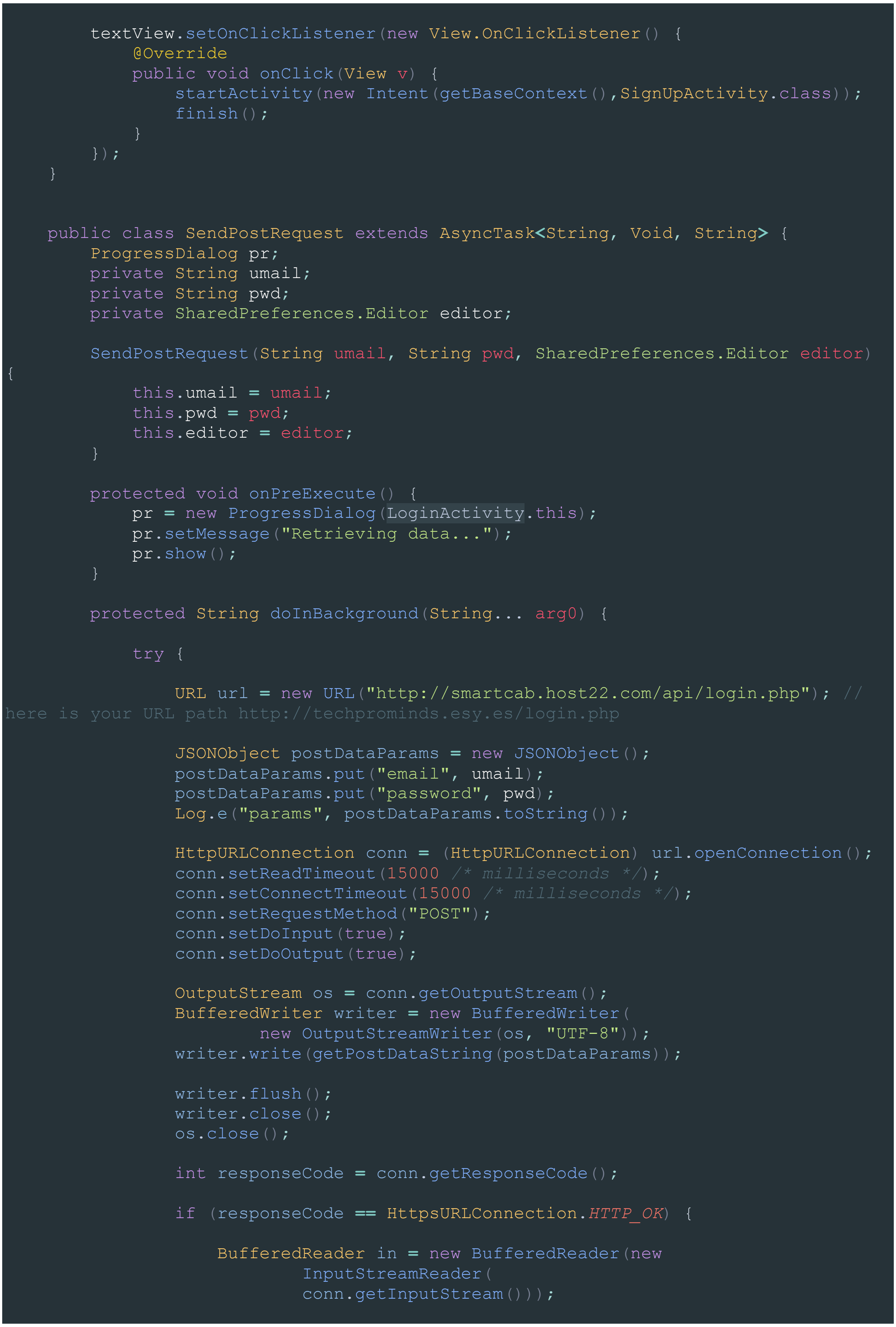
```
            public void onClick(View v) {
                startActivity(new Intent(getBaseContext(),LoginActivity.class));
                finish();
            }
        });
    }


    public class SendPostRequest extends AsyncTask<String, Void, String> {
        ProgressDialog pr;
        private String umail;
        private String pwd;
        private SharedPreferences.Editor editor;

        SendPostRequest(String umail, String pwd, SharedPreferences.Editor editor)
{
            this.umail = umail;
            this.pwd = pwd;
            this.editor = editor;
        }

        protected void onPreExecute() {
            pr = new ProgressDialog(SignUpActivity.this);
            pr.setMessage("Retrieving data...");
            pr.show();
        }

        protected String doInBackground(String... arg0) {

            try {

                URL url = new URL("http://smartcab.host22.com/api/signup.php"); //
here is your URL path http://techprominds.esy.es/login.php

                JSONObject postDataParams = new JSONObject();
                postDataParams.put("email", umail);
                postDataParams.put("password", pwd);
                Log.e("params", postDataParams.toString());

                HttpURLConnection conn = (HttpURLConnection) url.openConnection();
                conn.setReadTimeout(15000 /* milliseconds */);
                conn.setConnectTimeout(15000 /* milliseconds */);
                conn.setRequestMethod("POST");
                conn.setDoInput(true);
                conn.setDoOutput(true);

                OutputStream os = conn.getOutputStream();
                BufferedWriter writer = new BufferedWriter(
                        new OutputStreamWriter(os, "UTF-8"));
                writer.write(getPostDataString(postDataParams));

                writer.flush();
                writer.close();
                os.close();

                int responseCode = conn.getResponseCode();

                if (responseCode == HttpsURLConnection.HTTP_OK) {

                    BufferedReader in = new BufferedReader(new
                            InputStreamReader(
                            conn.getInputStream()));

                    StringBuffer sb = new StringBuffer("");
                    String line = "";
```



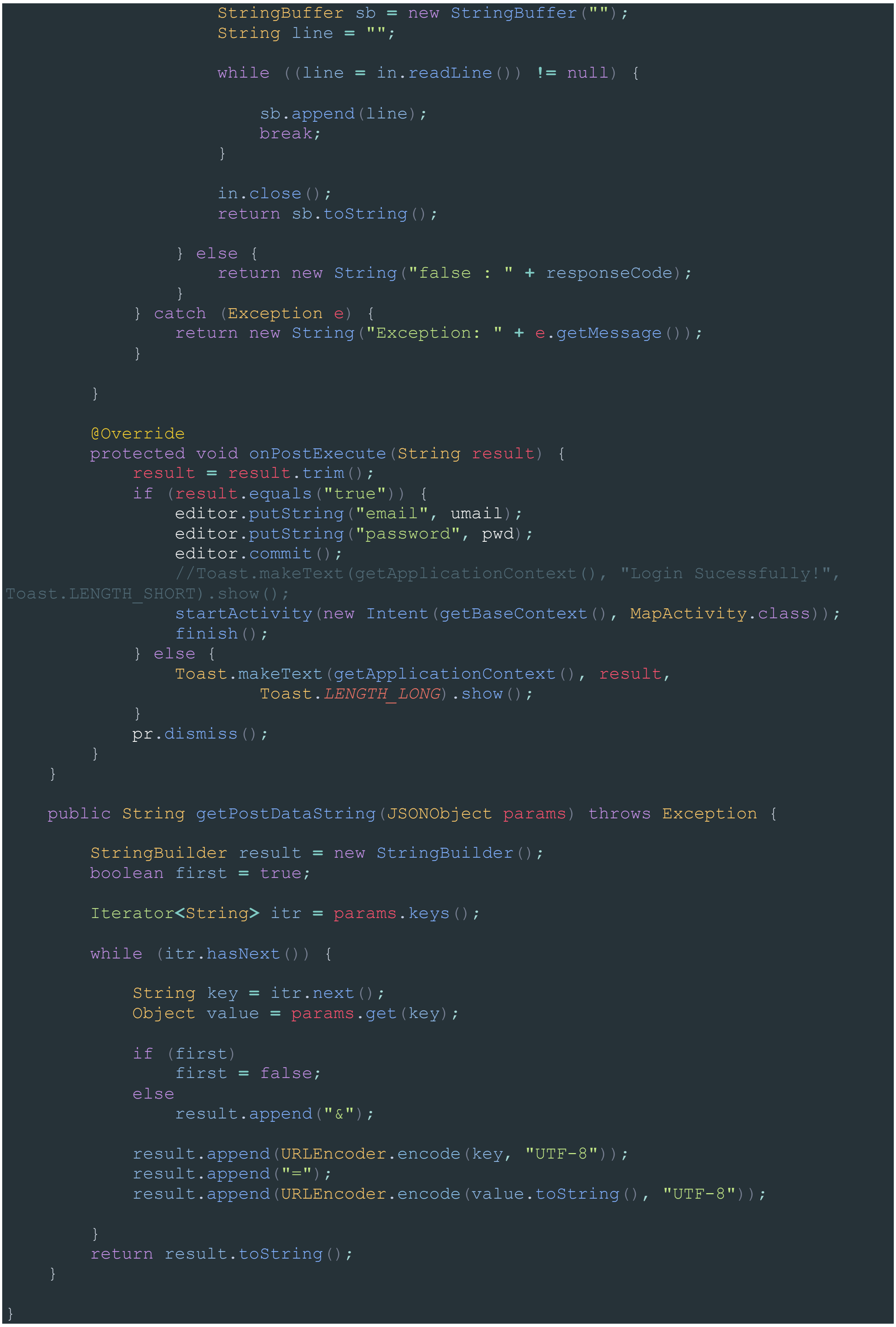
```
                    while ((line = in.readLine()) != null) {

                        sb.append(line);
                        break;
                    }

                    in.close();
                    return sb.toString();

                } else {
                    return new String("false : " + responseCode);
                }
            } catch (Exception e) {
                return new String("Exception: " + e.getMessage());
            }

        }

        @Override
        protected void onPostExecute(String result) {
            result = result.trim();
            if (result.equals("true")) {
                editor.putString("email", umail);
                editor.putString("password", pwd);
                editor.commit();
                //Toast.makeText(getApplicationContext(), "Login Sucessfully!",
Toast.LENGTH_SHORT).show();
                startActivity(new Intent(getBaseContext(), MapActivity.class));
                finish();
            } else {
                Toast.makeText(getApplicationContext(), result,
                        Toast.LENGTH_LONG).show();
            }
            pr.dismiss();
        }
    }

    public String getPostDataString(JSONObject params) throws Exception {

        StringBuilder result = new StringBuilder();
        boolean first = true;

        Iterator<String> itr = params.keys();

        while (itr.hasNext()) {

            String key = itr.next();
            Object value = params.get(key);

            if (first)
                first = false;
            else
                result.append("&");

            result.append(URLEncoder.encode(key, "UTF-8"));
            result.append("=");
            result.append(URLEncoder.encode(value.toString(), "UTF-8"));

        }
        return result.toString();
    }

}
```

LoginActivity.java

```java
package com.sumit.cab.smartcab;

import android.app.ProgressDialog;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.AsyncTask;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import org.json.JSONObject;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.URL;
import java.net.URLEncoder;
import java.util.Iterator;

import javax.net.ssl.HttpsURLConnection;

public class LoginActivity extends AppCompatActivity {

    private Button Login;
    private EditText email;
    private EditText password;

    private SharedPreferences sharedpreferences;
    private TextView textView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);

        sharedpreferences = getSharedPreferences("SmartCab", Context.MODE_PRIVATE);

        textView = (TextView) findViewById(R.id.textView);
        Login = (Button) findViewById(R.id.login);
        email = (EditText) findViewById(R.id.email);
        password = (EditText) findViewById(R.id.password);
        final SharedPreferences.Editor editor = sharedpreferences.edit();

        Login.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String uemail = email.getText().toString().trim();
                String pwd = password.getText().toString().trim();
                new SendPostRequest(uemail, pwd, editor).execute();
            }
        });
```



```
        textView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new Intent(getBaseContext(),SignUpActivity.class));
                finish();
            }
        });
    }


    public class SendPostRequest extends AsyncTask<String, Void, String> {
        ProgressDialog pr;
        private String umail;
        private String pwd;
        private SharedPreferences.Editor editor;

        SendPostRequest(String umail, String pwd, SharedPreferences.Editor editor)
{
            this.umail = umail;
            this.pwd = pwd;
            this.editor = editor;
        }

        protected void onPreExecute() {
            pr = new ProgressDialog(LoginActivity.this);
            pr.setMessage("Retrieving data...");
            pr.show();
        }

        protected String doInBackground(String... arg0) {

            try {

                URL url = new URL("http://smartcab.host22.com/api/login.php"); //
here is your URL path http://techprominds.esy.es/login.php

                JSONObject postDataParams = new JSONObject();
                postDataParams.put("email", umail);
                postDataParams.put("password", pwd);
                Log.e("params", postDataParams.toString());

                HttpURLConnection conn = (HttpURLConnection) url.openConnection();
                conn.setReadTimeout(15000 /* milliseconds */);
                conn.setConnectTimeout(15000 /* milliseconds */);
                conn.setRequestMethod("POST");
                conn.setDoInput(true);
                conn.setDoOutput(true);

                OutputStream os = conn.getOutputStream();
                BufferedWriter writer = new BufferedWriter(
                        new OutputStreamWriter(os, "UTF-8"));
                writer.write(getPostDataString(postDataParams));

                writer.flush();
                writer.close();
                os.close();

                int responseCode = conn.getResponseCode();

                if (responseCode == HttpsURLConnection.HTTP_OK) {

                    BufferedReader in = new BufferedReader(new
                            InputStreamReader(
                            conn.getInputStream()));
```



```
                    StringBuffer sb = new StringBuffer("");
                    String line = "";

                    while ((line = in.readLine()) != null) {

                        sb.append(line);
                        break;
                    }

                    in.close();
                    return sb.toString();

                } else {
                    return new String("false : " + responseCode);
                }
            } catch (Exception e) {
                return new String("Exception: " + e.getMessage());
            }

        }

        @Override
        protected void onPostExecute(String result) {
            result = result.trim();
            if (result.equals("true")) {
                editor.putString("email", umail);
                editor.putString("password", pwd);
                editor.commit();
                //Toast.makeText(getApplicationContext(), "Login Sucessfully!",
Toast.LENGTH_SHORT).show();
                startActivity(new Intent(getBaseContext(), MapActivity.class));
                finish();
            } else {
                Toast.makeText(getApplicationContext(), result,
                        Toast.LENGTH_LONG).show();
            }
            pr.dismiss();
        }
    }

    public String getPostDataString(JSONObject params) throws Exception {

        StringBuilder result = new StringBuilder();
        boolean first = true;

        Iterator<String> itr = params.keys();

        while (itr.hasNext()) {

            String key = itr.next();
            Object value = params.get(key);

            if (first)
                first = false;
            else
                result.append("&");

            result.append(URLEncoder.encode(key, "UTF-8"));
            result.append("=");
            result.append(URLEncoder.encode(value.toString(), "UTF-8"));

        }
        return result.toString();
    }

}
```

MainActivity.java

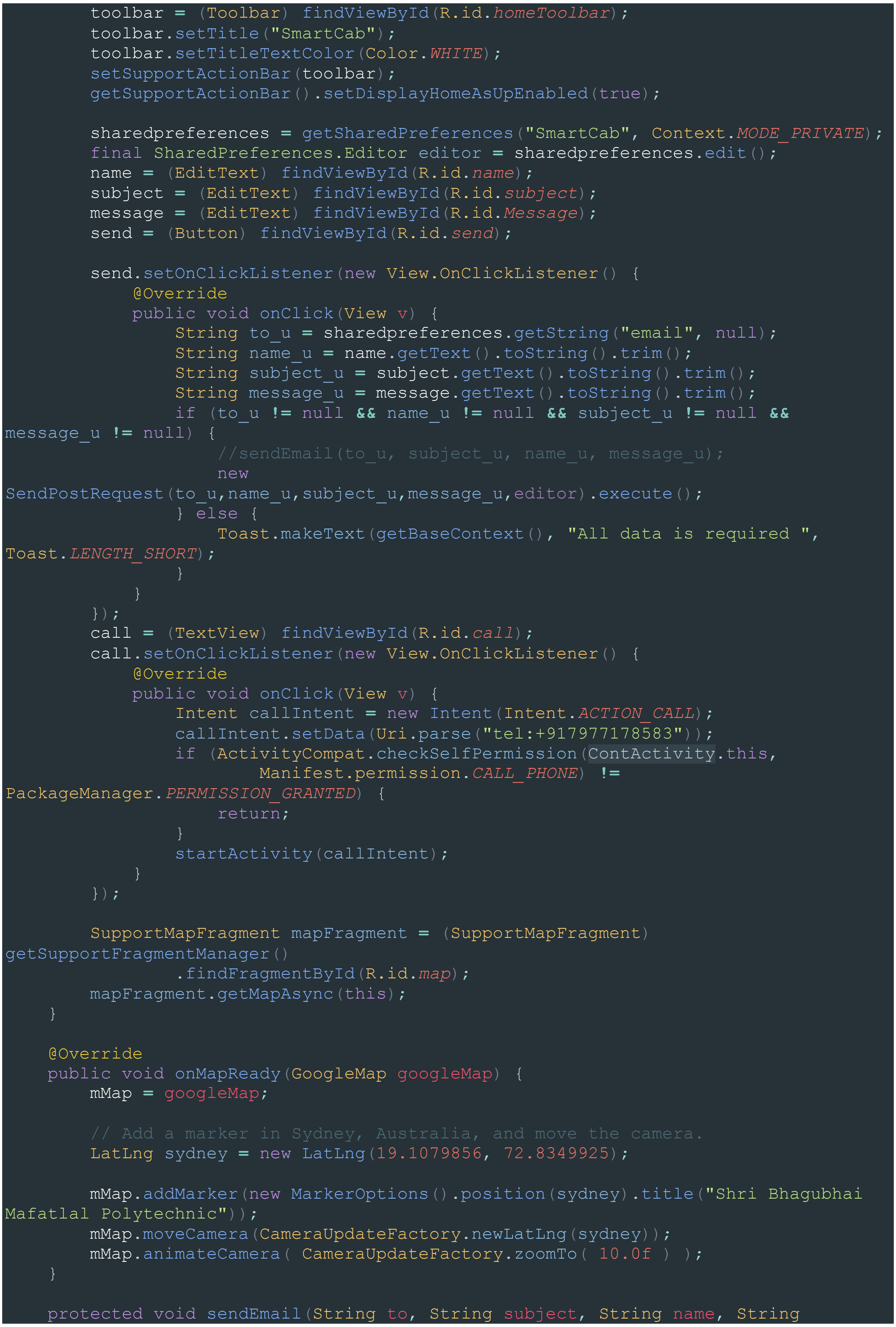
```
package com.sumit.cab.smartcab;

import android.Manifest;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.pm.PackageManager;
import android.support.v4.app.ActivityCompat;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {
    static final int MY_REQUEST=10;
    private Button getStated;
    private SharedPreferences sharedpreferences;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        if(ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED
                    ||
ActivityCompat.checkSelfPermission(this,Manifest.permission.CALL_PHONE) !=
PackageManager.PERMISSION_GRANTED
                ||
ActivityCompat.checkSelfPermission(this,Manifest.permission.INTERNET) !=
PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.ACCESS_FINE_LOCATION,
                    Manifest.permission.INTERNET,
                    Manifest.permission.CALL_PHONE}, MY_REQUEST);
        }

        final SharedPreferences sharedPreferences =
getSharedPreferences("SmartCab", Context.MODE_PRIVATE);

        getStated = (Button) findViewById(R.id.getStated);

        getStated.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String email = sharedPreferences.getString("email",null);
                if(email == null) {
                    startActivity(new Intent(getBaseContext(),
LoginActivity.class));
                    finish();
                }
                else{
                    startActivity(new Intent(getBaseContext(), MapActivity.class));
                    finish();
                }
            }
        });
    }
}
```

AboutUsActivity.java

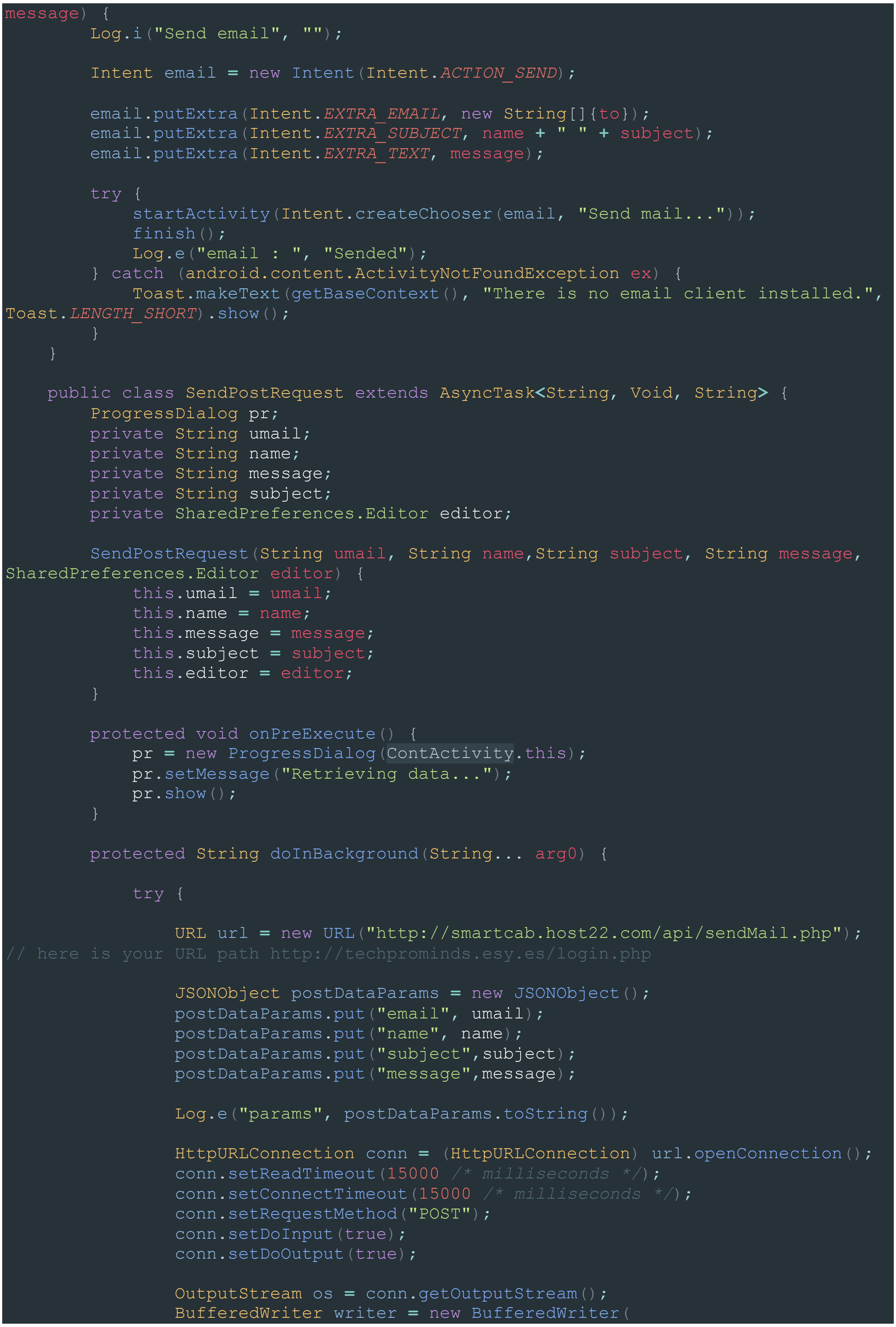
```java
package com.sumit.cab.smartcab;

import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.Color;
import android.support.v4.app.ActivityCompat;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.support.v7.widget.Toolbar;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.widget.Toast;

public class AboutUsActivity extends AppCompatActivity {

    private Toolbar toolbar;
    private SharedPreferences sharedpreferences;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_about_us);

        sharedpreferences = getSharedPreferences("SmartCab", Context.MODE_PRIVATE);

        toolbar = (Toolbar) findViewById(R.id.homeToolbar);
        toolbar.setTitle("SmartCab");
        setSupportActionBar(toolbar);
        toolbar.setTitleTextColor(Color.WHITE);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);

    }
    @Override
    public boolean onCreateOptionsMenu(Menu menu){
        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.hometoolbar, menu);
        return true;
    }


    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch(item.getItemId()){
            case R.id.logout:
                String email = sharedpreferences.getString("email",null);
                SharedPreferences.Editor editor = sharedpreferences.edit();
                editor.putString("email", null);
                editor.putString("password", null);
                editor.commit();
                Toast.makeText(getBaseContext(),"Logout successfully
done",Toast.LENGTH_SHORT).show();
                startActivity(new Intent(getBaseContext(),MainActivity.class));
                finishAffinity();
                break;
        }
        return super.onOptionsItemSelected(item);
    }
}
```

ContUs.java

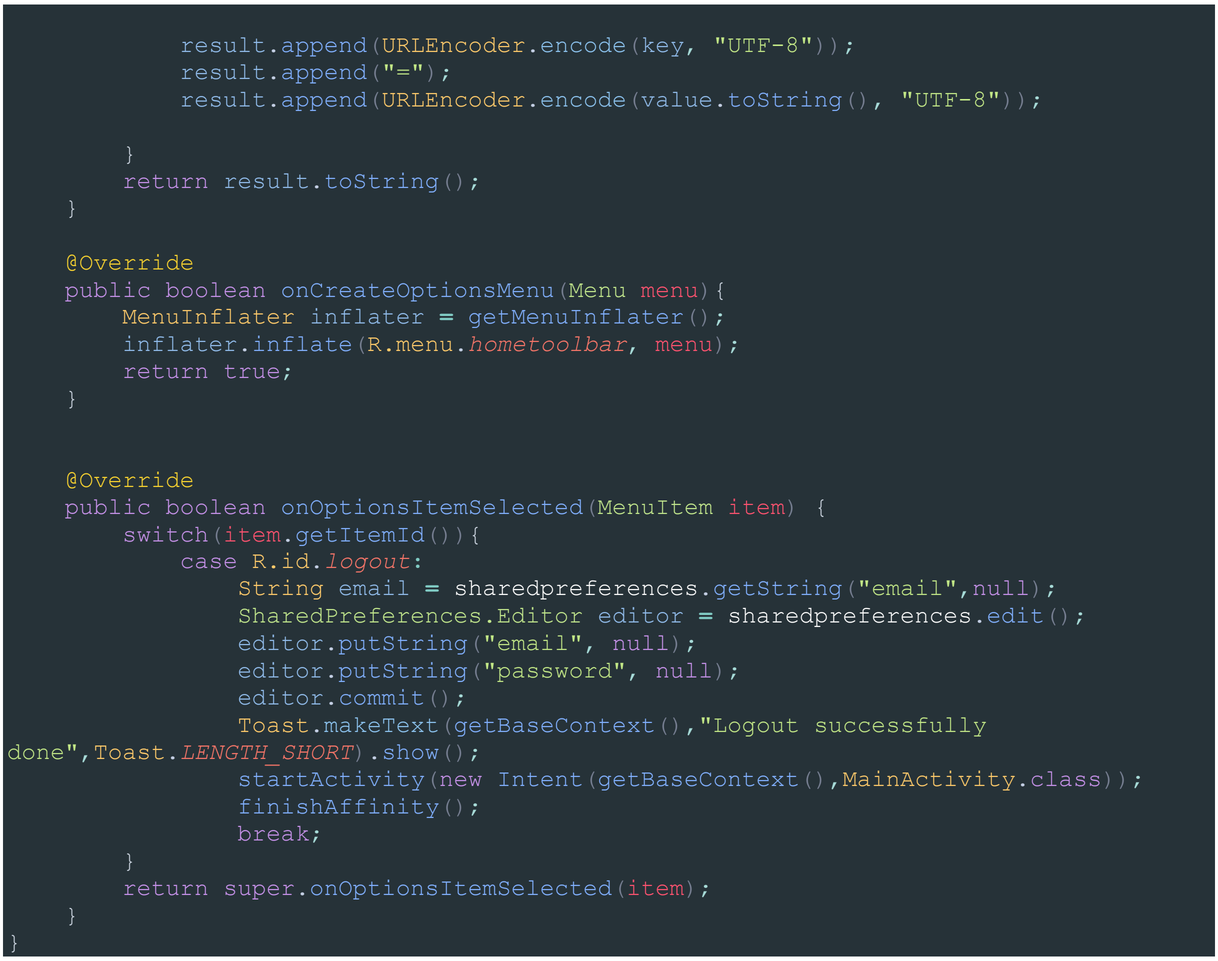
```java
package com.sumit.cab.smartcab;

import android.Manifest;
import android.app.ProgressDialog;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.pm.PackageManager;
import android.graphics.Color;
import android.net.Uri;
import android.os.AsyncTask;
import android.support.v4.app.ActivityCompat;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.support.v7.widget.Toolbar;
import android.util.Log;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import android.os.Bundle;
import android.support.v4.app.FragmentActivity;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;

import org.json.JSONObject;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.URL;
import java.net.URLEncoder;
import java.util.Iterator;

import javax.net.ssl.HttpsURLConnection;

public class ContActivity extends AppCompatActivity implements OnMapReadyCallback{
    private Toolbar toolbar;
    private EditText name, subject, message;
    private Button send;
    private SharedPreferences sharedpreferences;
    private TextView call;
    private GoogleMap mMap;

    @Override
    protected void onCreate(final Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_cont);
```



```
        toolbar = (Toolbar) findViewById(R.id.homeToolbar);
        toolbar.setTitle("SmartCab");
        toolbar.setTitleTextColor(Color.WHITE);
        setSupportActionBar(toolbar);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);

        sharedpreferences = getSharedPreferences("SmartCab", Context.MODE_PRIVATE);
        final SharedPreferences.Editor editor = sharedpreferences.edit();
        name = (EditText) findViewById(R.id.name);
        subject = (EditText) findViewById(R.id.subject);
        message = (EditText) findViewById(R.id.Message);
        send = (Button) findViewById(R.id.send);

        send.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String to_u = sharedpreferences.getString("email", null);
                String name_u = name.getText().toString().trim();
                String subject_u = subject.getText().toString().trim();
                String message_u = message.getText().toString().trim();
                if (to_u != null && name_u != null && subject_u != null &&
message_u != null) {
                    //sendEmail(to_u, subject_u, name_u, message_u);
                    new
SendPostRequest(to_u,name_u,subject_u,message_u,editor).execute();
                } else {
                    Toast.makeText(getBaseContext(), "All data is required ",
Toast.LENGTH_SHORT);
                }
            }
        });
        call = (TextView) findViewById(R.id.call);
        call.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent callIntent = new Intent(Intent.ACTION_CALL);
                callIntent.setData(Uri.parse("tel:+917977178583"));
                if (ActivityCompat.checkSelfPermission(ContActivity.this,
                        Manifest.permission.CALL_PHONE) !=
PackageManager.PERMISSION_GRANTED) {
                    return;
                }
                startActivity(callIntent);
            }
        });

        SupportMapFragment mapFragment = (SupportMapFragment)
getSupportFragmentManager()
                .findFragmentById(R.id.map);
        mapFragment.getMapAsync(this);
    }

    @Override
    public void onMapReady(GoogleMap googleMap) {
        mMap = googleMap;

        // Add a marker in Sydney, Australia, and move the camera.
        LatLng sydney = new LatLng(19.1079856, 72.8349925);

        mMap.addMarker(new MarkerOptions().position(sydney).title("Shri Bhagubhai
Mafatlal Polytechnic"));
        mMap.moveCamera(CameraUpdateFactory.newLatLng(sydney));
        mMap.animateCamera( CameraUpdateFactory.zoomTo( 10.0f ) );
    }

    protected void sendEmail(String to, String subject, String name, String
```



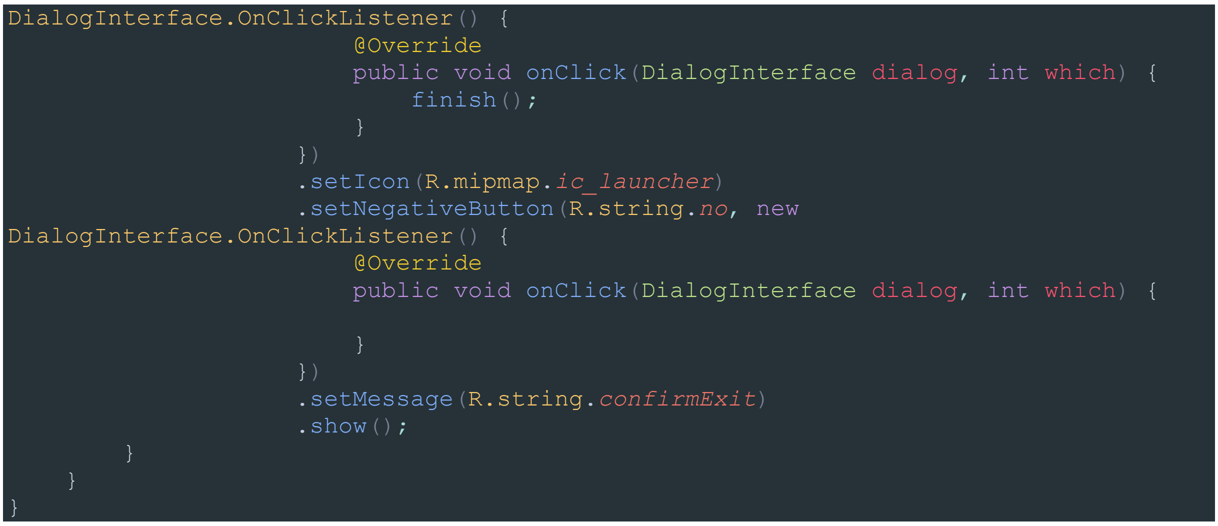
```
message) {
        Log.i("Send email", "");

        Intent email = new Intent(Intent.ACTION_SEND);

        email.putExtra(Intent.EXTRA_EMAIL, new String[]{to});
        email.putExtra(Intent.EXTRA_SUBJECT, name + " " + subject);
        email.putExtra(Intent.EXTRA_TEXT, message);

        try {
            startActivity(Intent.createChooser(email, "Send mail..."));
            finish();
            Log.e("email : ", "Sended");
        } catch (android.content.ActivityNotFoundException ex) {
            Toast.makeText(getBaseContext(), "There is no email client installed.",
Toast.LENGTH_SHORT).show();
        }
    }

    public class SendPostRequest extends AsyncTask<String, Void, String> {
        ProgressDialog pr;
        private String umail;
        private String name;
        private String message;
        private String subject;
        private SharedPreferences.Editor editor;

        SendPostRequest(String umail, String name,String subject, String message,
SharedPreferences.Editor editor) {
            this.umail = umail;
            this.name = name;
            this.message = message;
            this.subject = subject;
            this.editor = editor;
        }

        protected void onPreExecute() {
            pr = new ProgressDialog(ContActivity.this);
            pr.setMessage("Retrieving data...");
            pr.show();
        }

        protected String doInBackground(String... arg0) {

            try {

                URL url = new URL("http://smartcab.host22.com/api/sendMail.php");
// here is your URL path http://techprominds.esy.es/login.php

                JSONObject postDataParams = new JSONObject();
                postDataParams.put("email", umail);
                postDataParams.put("name", name);
                postDataParams.put("subject",subject);
                postDataParams.put("message",message);

                Log.e("params", postDataParams.toString());

                HttpURLConnection conn = (HttpURLConnection) url.openConnection();
                conn.setReadTimeout(15000 /* milliseconds */);
                conn.setConnectTimeout(15000 /* milliseconds */);
                conn.setRequestMethod("POST");
                conn.setDoInput(true);
                conn.setDoOutput(true);

                OutputStream os = conn.getOutputStream();
                BufferedWriter writer = new BufferedWriter(
                        new OutputStreamWriter(os, "UTF-8"));
                writer.write(getPostDataString(postDataParams));

                writer.flush();
                writer.close();
                os.close();

                int responseCode = conn.getResponseCode();

                if (responseCode == HttpsURLConnection.HTTP_OK) {

                    BufferedReader in = new BufferedReader(new
                            InputStreamReader(
                            conn.getInputStream()));

                    StringBuffer sb = new StringBuffer("");
                    String line = "";

                    while ((line = in.readLine()) != null) {

                        sb.append(line);
                        break;
                    }

                    in.close();
                    return sb.toString();

                } else {
                    return new String("false : " + responseCode);
                }
            } catch (Exception e) {
                return new String("Exception: " + e.getMessage());
            }

        }

        @Override
        protected void onPostExecute(String result) {
            result = result.trim();
            if (result.equals("true")) {
                Toast.makeText(getBaseContext(),"Thank
you!",Toast.LENGTH_SHORT).show();
            } else {
                Toast.makeText(getApplicationContext(), result,
                        Toast.LENGTH_LONG).show();
            }
            pr.dismiss();
        }
    }

    public String getPostDataString(JSONObject params) throws Exception {

        StringBuilder result = new StringBuilder();
        boolean first = true;

        Iterator<String> itr = params.keys();

        while (itr.hasNext()) {

            String key = itr.next();
            Object value = params.get(key);

            if (first)
                first = false;
            else
                result.append("&");
```



```
            result.append(URLEncoder.encode(key, "UTF-8"));
            result.append("=");
            result.append(URLEncoder.encode(value.toString(), "UTF-8"));

        }
        return result.toString();
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu){
        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.hometoolbar, menu);
        return true;
    }


    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch(item.getItemId()){
            case R.id.logout:
                String email = sharedpreferences.getString("email",null);
                SharedPreferences.Editor editor = sharedpreferences.edit();
                editor.putString("email", null);
                editor.putString("password", null);
                editor.commit();
                Toast.makeText(getBaseContext(),"Logout successfully
done",Toast.LENGTH_SHORT).show();
                startActivity(new Intent(getBaseContext(),MainActivity.class));
                finishAffinity();
                break;
        }
        return super.onOptionsItemSelected(item);
    }
}
```

MapActivity.java

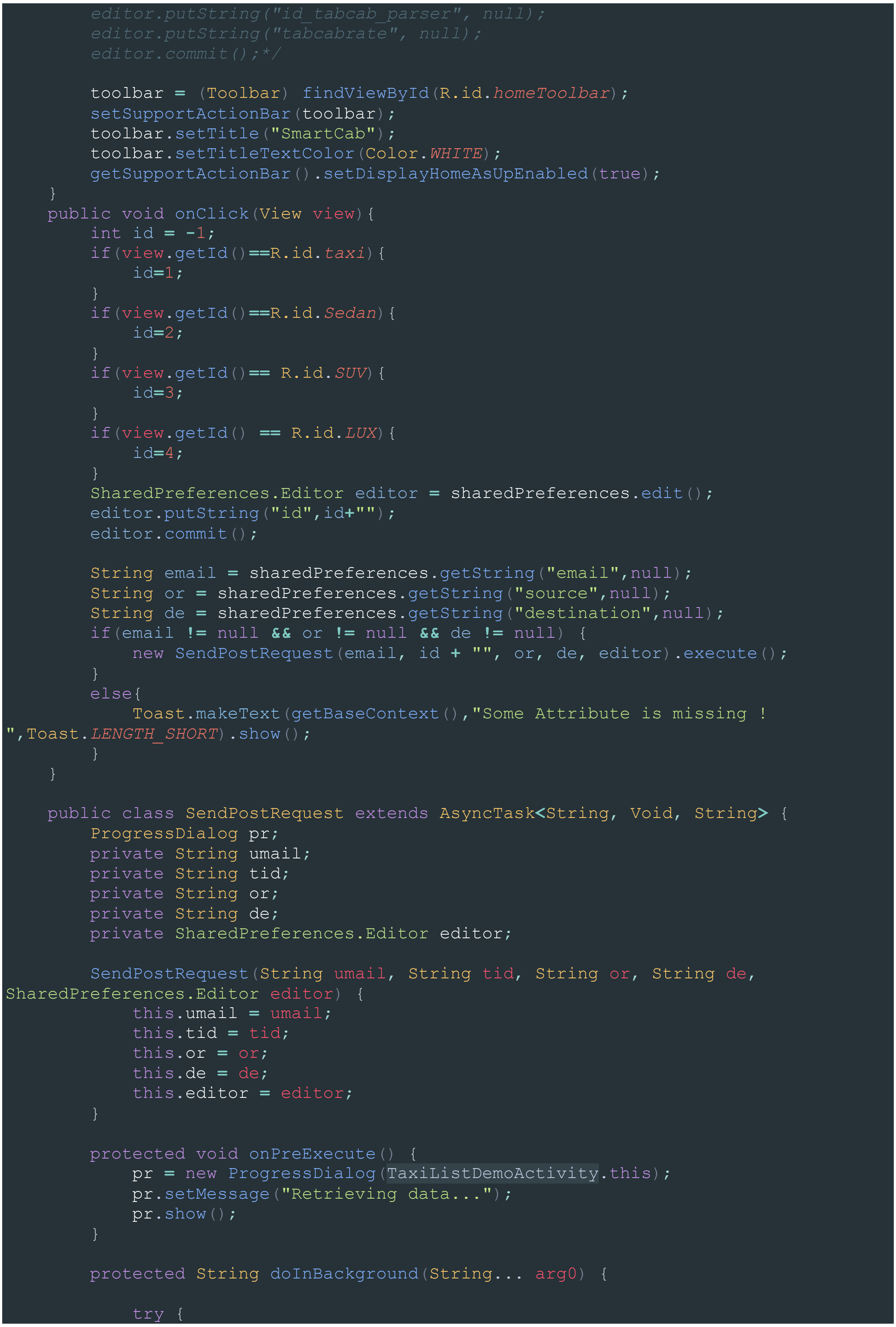
```
/*
 * Copyright (C) 2015 Google Inc. All Rights Reserved.
 *
 *  Licensed under the Apache License, Version 2.0 (the "License");
 *  you may not use this file except in compliance with the License.
 *  You may obtain a copy of the License at
 *
 *      http://www.apache.org/licenses/LICENSE-2.0
 *
 *  Unless required by applicable law or agreed to in writing, software
 *  distributed under the License is distributed on an "AS IS" BASIS,
 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 *  See the License for the specific language governing permissions and
 *  limitations under the License.
 */

package com.sumit.cab.smartcab;

import com.google.android.gms.common.SignInButton;
import com.google.android.gms.common.api.Status;
import com.google.android.gms.location.places.Place;
import com.google.android.gms.location.places.ui.PlaceAutocompleteFragment;
import com.google.android.gms.location.places.ui.PlaceSelectionListener;
import com.google.android.gms.maps.model.LatLngBounds;
import com.sumit.cab.smartcab.History.HistoryActivity;

import android.app.AlertDialog;
import android.common.activities.SampleActivityBase;
import android.common.logger.Log;

import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.res.Resources;
import android.graphics.Color;
import android.net.Uri;
import android.os.Bundle;

import android.support.v7.widget.CardView;
import android.support.v7.widget.Toolbar;
import android.text.Html;
import android.text.Spanned;
import android.text.TextUtils;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.ImageButton;
import android.widget.TextView;
import android.widget.Toast;


public class MapActivity extends SampleActivityBase implements
PlaceSelectionListener {

    private TextView mPlaceDetailsText;

    private TextView mPlaceAttribution;
    private TextView mPlaceDetailsText_dest;

    private Button CompTaxi;
    Toolbar toolbar;

    private ImageButton about_us, cont_us;

    private SharedPreferences sharedpreferences;

    private ImageButton history;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_map);
        Log.e("Back Button : ", "button");
        sharedpreferences = getSharedPreferences("SmartCab", Context.MODE_PRIVATE);
        SharedPreferences.Editor editor = sharedpreferences.edit();
        editor.putString("source", null);
        editor.putString("destination", null);
        editor.commit();

        String email = sharedpreferences.getString("email", null);
        if (email == null) {
            Log.e("Null data : ", "email");
            startActivity(new Intent(getBaseContext(), LoginActivity.class));
        }

        history = (ImageButton) findViewById(R.id.historyCard);
        CompTaxi = (Button) findViewById(R.id.Search);
        toolbar = (Toolbar) findViewById(R.id.homeToolbar);
        toolbar.setTitle("SmartCab");
        setSupportActionBar(toolbar);
        toolbar.setTitleTextColor(Color.WHITE);

        about_us = (ImageButton) findViewById(R.id.about_us);
        cont_us = (ImageButton) findViewById(R.id.con_us);

        about_us.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new Intent(getBaseContext(), AboutUsActivity.class));
            }
        });

        cont_us.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new Intent(getBaseContext(), ContActivity.class));
            }
        });
        // Retrieve the PlaceAutocompleteFragment.
        PlaceAutocompleteFragment autocompleteFragment =
(PlaceAutocompleteFragment)
                getFragmentManager().findFragmentById(R.id.autocomplete_fragment);

        PlaceAutocompleteFragment autocompleteFragment2 =
(PlaceAutocompleteFragment)
                getFragmentManager().findFragmentById(R.id.autocomplete_fragment2);

        // Register a listener to receive callbacks when a place has been selected
or an error has
        // occurred.
        autocompleteFragment.setOnPlaceSelectedListener(this);
        autocompleteFragment2.setOnPlaceSelectedListener(this);

        // Retrieve the TextViews that will display details about the selected 
place.
        mPlaceDetailsText = (TextView) findViewById(R.id.place_details);
        //mPlaceAttribution = (TextView) findViewById(R.id.place_attribution);

        mPlaceDetailsText_dest = (TextView) findViewById(R.id.place_details_dest);

        CompTaxi.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                String source = sharedpreferences.getString("source", null);
                String destination = sharedpreferences.getString("destination", 
null);
                if (source != null && destination != null) {
                    Log.e("source ", source);
                    Log.e("destination", destination);
                    startActivity(new Intent(getBaseContext(), 
TaxiListDemoActivity.class));
                    //finish();
                } else {
                    Toast.makeText(getBaseContext(), "Please Select your source and 
destionation", Toast.LENGTH_SHORT).show();
                }
            }
        });

        history.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new Intent(getBaseContext(), HistoryActivity.class));
            }
        });
    }

    /**
     * Callback invoked when a place has been selected from the 
PlaceAutocompleteFragment.
     */
    @Override
    public void onPlaceSelected(Place place) {
        Log.i(TAG, "Place Selected: " + place.getName());

        SharedPreferences.Editor editor = sharedpreferences.edit();
        String source = sharedpreferences.getString("source", null);
        if (source == null) {
            editor.putString("source", 
place.getLatLng().toString().substring(place.getLatLng().toString().indexOf('(') + 
1, place.getLatLng().toString().indexOf(')')));
            Log.e("Latlang : ", 
place.getLatLng().toString().substring(place.getLatLng().toString().indexOf('(') + 
1, place.getLatLng().toString().indexOf(')')));
        } else {
            editor.putString("destination", 
place.getLatLng().toString().substring(place.getLatLng().toString().indexOf('(') + 
1, place.getLatLng().toString().indexOf(')')));
            Log.e("Latlang : ", 
place.getLatLng().toString().substring(place.getLatLng().toString().indexOf('(') + 
1, place.getLatLng().toString().indexOf(')')));
        }
        editor.commit();
        // Format the returned place's details and display them in the TextView.
        /*
        mPlaceDetailsText.setText(place.getLatLng()+"");
        mPlaceDetailsText_dest.setText(place.getLatLng()+"");*/
    }

    /**
     * Callback invoked when PlaceAutocompleteFragment encounters an error.
     */
    @Override
    public void onError(Status status) {
        Log.e(TAG, "onError: Status = " + status.toString());

        Toast.makeText(this, "Place selection failed: " +
status.getStatusMessage(),
                Toast.LENGTH_SHORT).show();
    }

    /**
     * Helper method to format information about a place nicely.
     */
    /*
    private static Spanned formatPlaceDetails(Resources res, CharSequence name,
String id,
                                              CharSequence address, CharSequence
phoneNumber, Uri websiteUri) {
        Log.e(TAG, res.getString(R.string.place_details, name, id, address,
phoneNumber,
                websiteUri));
        return Html.fromHtml(res.getString(R.string.place_details, name, id,
address, phoneNumber,
                websiteUri));

    }*/
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.hometoolbar, menu);
        return true;
    }


    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
            case R.id.logout:
                String email = sharedpreferences.getString("email", null);
                SharedPreferences.Editor editor = sharedpreferences.edit();
                editor.putString("email", null);
                editor.putString("password", null);
                editor.commit();
                Toast.makeText(getBaseContext(), "Logout successfully done",
Toast.LENGTH_SHORT).show();
                startActivity(new Intent(getBaseContext(), MainActivity.class));
                finishAffinity();
                break;
        }
        return super.onOptionsItemSelected(item);
    }

    @Override
    public void onBackPressed() {
        if (getFragmentManager().getBackStackEntryCount() > 0) {
            getFragmentManager().popBackStackImmediate();
        } else {
            new AlertDialog.Builder(this)
                    .setTitle(R.string.exit)
                    .setPositiveButton(R.string.yes, new
```



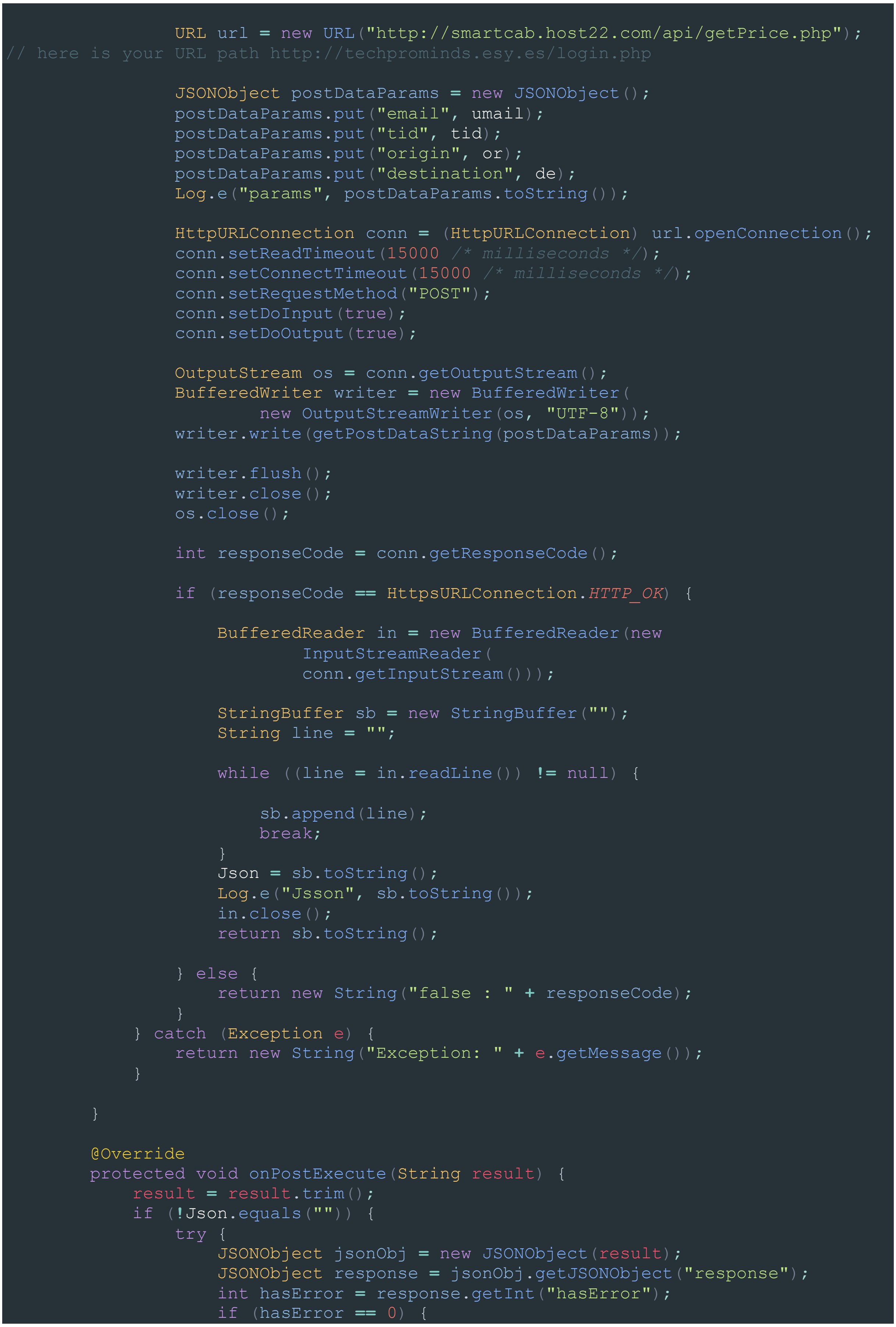
```
DialogInterface.OnClickListener() {
                        @Override
                        public void onClick(DialogInterface dialog, int which) {
                            finish();
                        }
                    })
                    .setIcon(R.mipmap.ic_launcher)
                    .setNegativeButton(R.string.no, new
DialogInterface.OnClickListener() {
                        @Override
                        public void onClick(DialogInterface dialog, int which) {

                        }
                    })
                    .setMessage(R.string.confirmExit)
                    .show();
        }
    }
}
```

TaxiListDemo.java

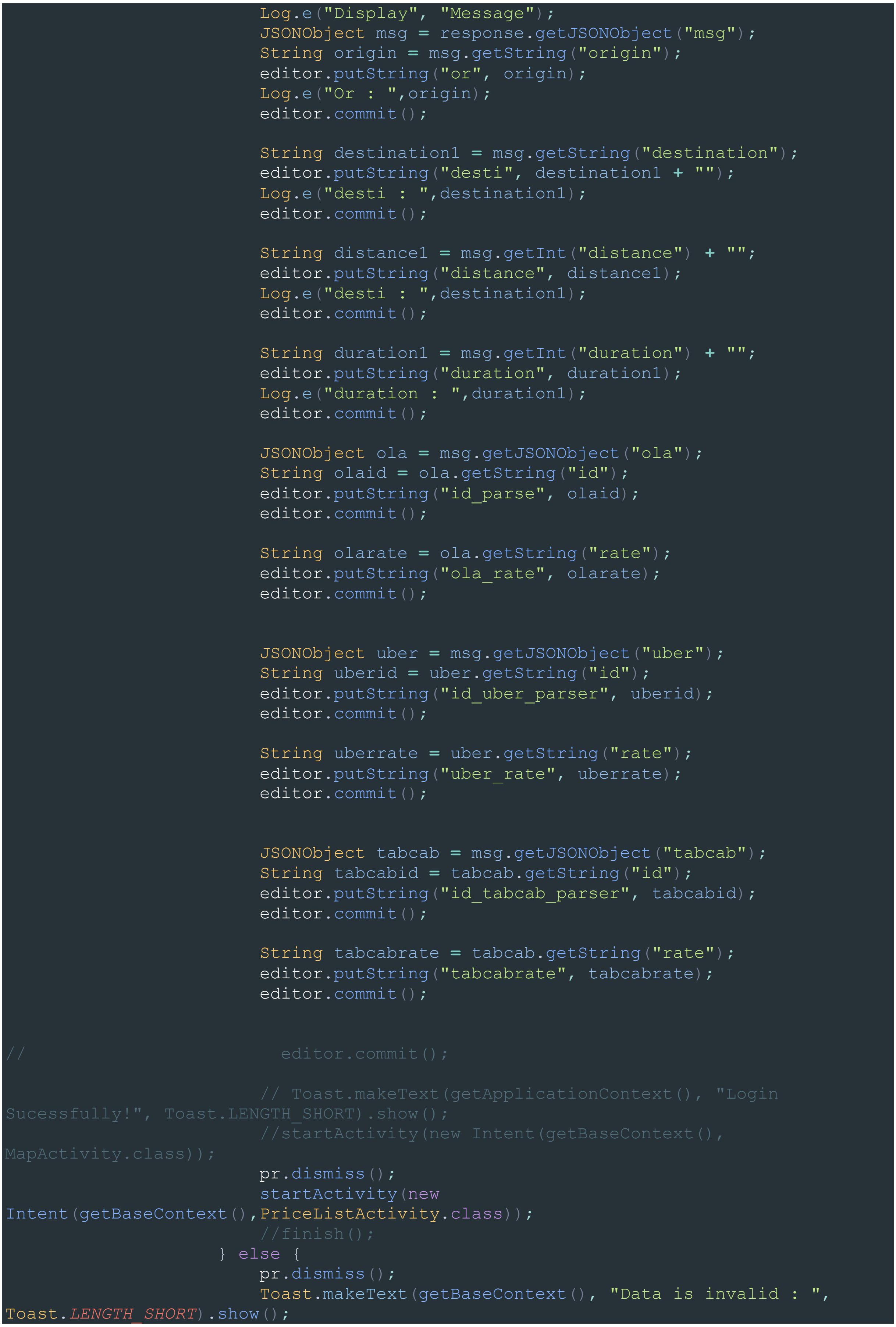
```
package com.sumit.cab.smartcab;

import android.app.ProgressDialog;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.Color;
import android.os.AsyncTask;
import android.support.v4.app.ActivityCompat;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.support.v7.widget.Toolbar;
import android.util.Log;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.TextView;
import android.widget.Toast;

import org.json.JSONException;
import org.json.JSONObject;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.URL;
import java.net.URLEncoder;
import java.util.Iterator;

import javax.net.ssl.HttpsURLConnection;

public class TaxiListDemoActivity extends AppCompatActivity {
    private String Json = "";
    private Toolbar toolbar;

    private SharedPreferences sharedPreferences;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_taxi_list_demo);

        sharedPreferences = getSharedPreferences("SmartCab", Context.MODE_PRIVATE);

        SharedPreferences.Editor editor = sharedPreferences.edit();
        /*
            Null SharedPrefernces
         */
        /*
        editor.putString("or", null);
        editor.putString("destination", null);
        editor.putString("distance", null);
        editor.putString("duration", null);
        editor.putString("id_parse", null);
        editor.putString("ola_rate", null);
        editor.putString("id_uber_parser", null);
        editor.putString("uber_rate", null);
```



```
        editor.putString("id_tabcab_parser", null);
        editor.putString("tabcabrate", null);
        editor.commit();*/

        toolbar = (Toolbar) findViewById(R.id.homeToolbar);
        setSupportActionBar(toolbar);
        toolbar.setTitle("SmartCab");
        toolbar.setTitleTextColor(Color.WHITE);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
    }
    public void onClick(View view){
        int id = -1;
        if(view.getId()==R.id.taxi){
            id=1;
        }
        if(view.getId()==R.id.Sedan){
            id=2;
        }
        if(view.getId()== R.id.SUV){
            id=3;
        }
        if(view.getId() == R.id.LUX){
            id=4;
        }
        SharedPreferences.Editor editor = sharedPreferences.edit();
        editor.putString("id",id+"");
        editor.commit();

        String email = sharedPreferences.getString("email",null);
        String or = sharedPreferences.getString("source",null);
        String de = sharedPreferences.getString("destination",null);
        if(email != null && or != null && de != null) {
            new SendPostRequest(email, id + "", or, de, editor).execute();
        }
        else{
            Toast.makeText(getBaseContext(),"Some Attribute is missing !
",Toast.LENGTH_SHORT).show();
        }
    }

    public class SendPostRequest extends AsyncTask<String, Void, String> {
        ProgressDialog pr;
        private String umail;
        private String tid;
        private String or;
        private String de;
        private SharedPreferences.Editor editor;

        SendPostRequest(String umail, String tid, String or, String de,
SharedPreferences.Editor editor) {
            this.umail = umail;
            this.tid = tid;
            this.or = or;
            this.de = de;
            this.editor = editor;
        }

        protected void onPreExecute() {
            pr = new ProgressDialog(TaxiListDemoActivity.this);
            pr.setMessage("Retrieving data...");
            pr.show();
        }

        protected String doInBackground(String... arg0) {

            try {
```



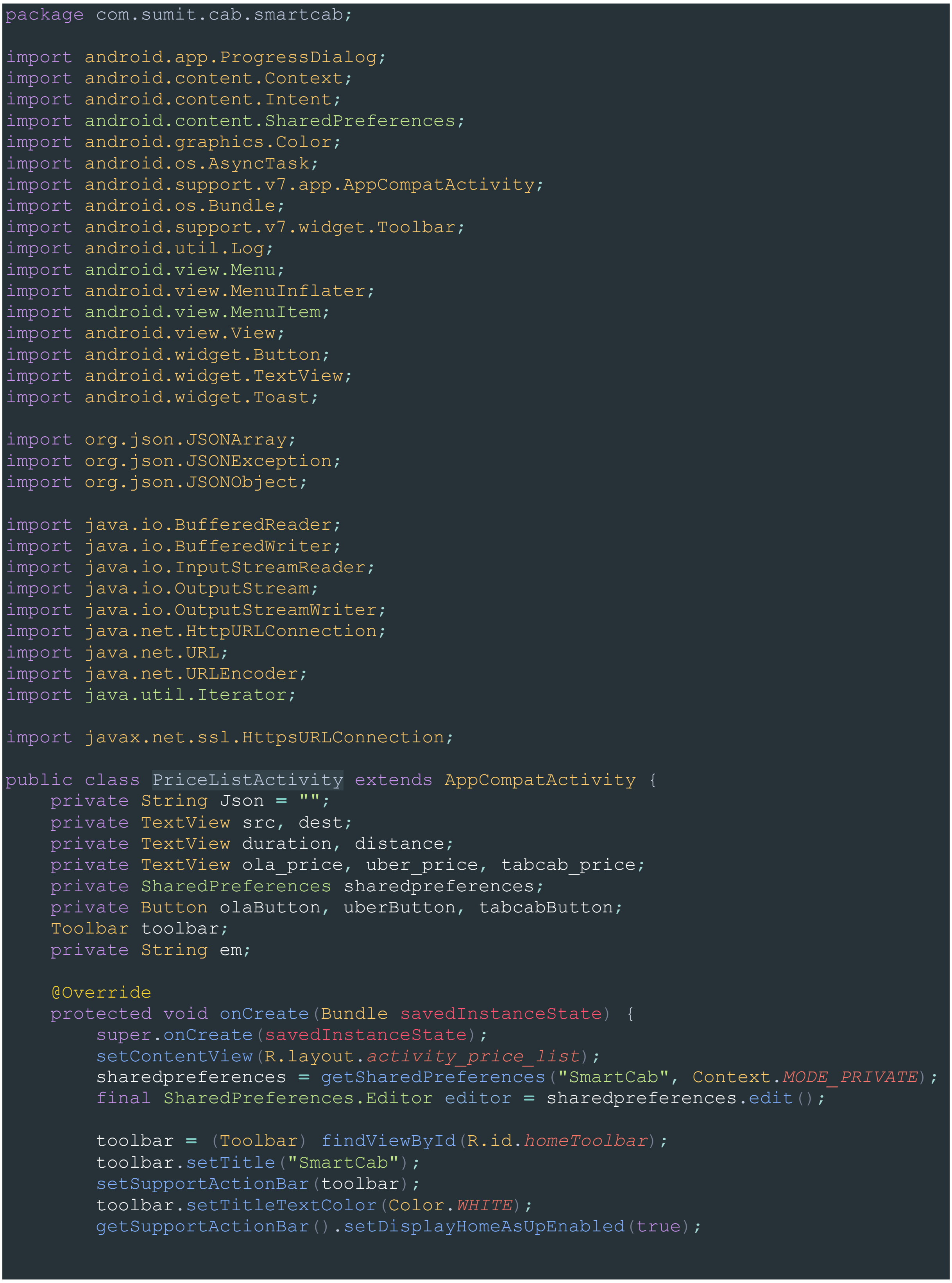
```
                URL url = new URL("http://smartcab.host22.com/api/getPrice.php");
// here is your URL path http://techprominds.esy.es/login.php

                JSONObject postDataParams = new JSONObject();
                postDataParams.put("email", umail);
                postDataParams.put("tid", tid);
                postDataParams.put("origin", or);
                postDataParams.put("destination", de);
                Log.e("params", postDataParams.toString());

                HttpURLConnection conn = (HttpURLConnection) url.openConnection();
                conn.setReadTimeout(15000 /* milliseconds */);
                conn.setConnectTimeout(15000 /* milliseconds */);
                conn.setRequestMethod("POST");
                conn.setDoInput(true);
                conn.setDoOutput(true);

                OutputStream os = conn.getOutputStream();
                BufferedWriter writer = new BufferedWriter(
                        new OutputStreamWriter(os, "UTF-8"));
                writer.write(getPostDataString(postDataParams));

                writer.flush();
                writer.close();
                os.close();

                int responseCode = conn.getResponseCode();

                if (responseCode == HttpsURLConnection.HTTP_OK) {

                    BufferedReader in = new BufferedReader(new
                            InputStreamReader(
                            conn.getInputStream()));

                    StringBuffer sb = new StringBuffer("");
                    String line = "";

                    while ((line = in.readLine()) != null) {

                        sb.append(line);
                        break;
                    }
                    Json = sb.toString();
                    Log.e("Jsson", sb.toString());
                    in.close();
                    return sb.toString();

                } else {
                    return new String("false : " + responseCode);
                }
            } catch (Exception e) {
                return new String("Exception: " + e.getMessage());
            }

        }

        @Override
        protected void onPostExecute(String result) {
            result = result.trim();
            if (!Json.equals("")) {
                try {
                    JSONObject jsonObj = new JSONObject(result);
                    JSONObject response = jsonObj.getJSONObject("response");
                    int hasError = response.getInt("hasError");
                    if (hasError == 0) {
```



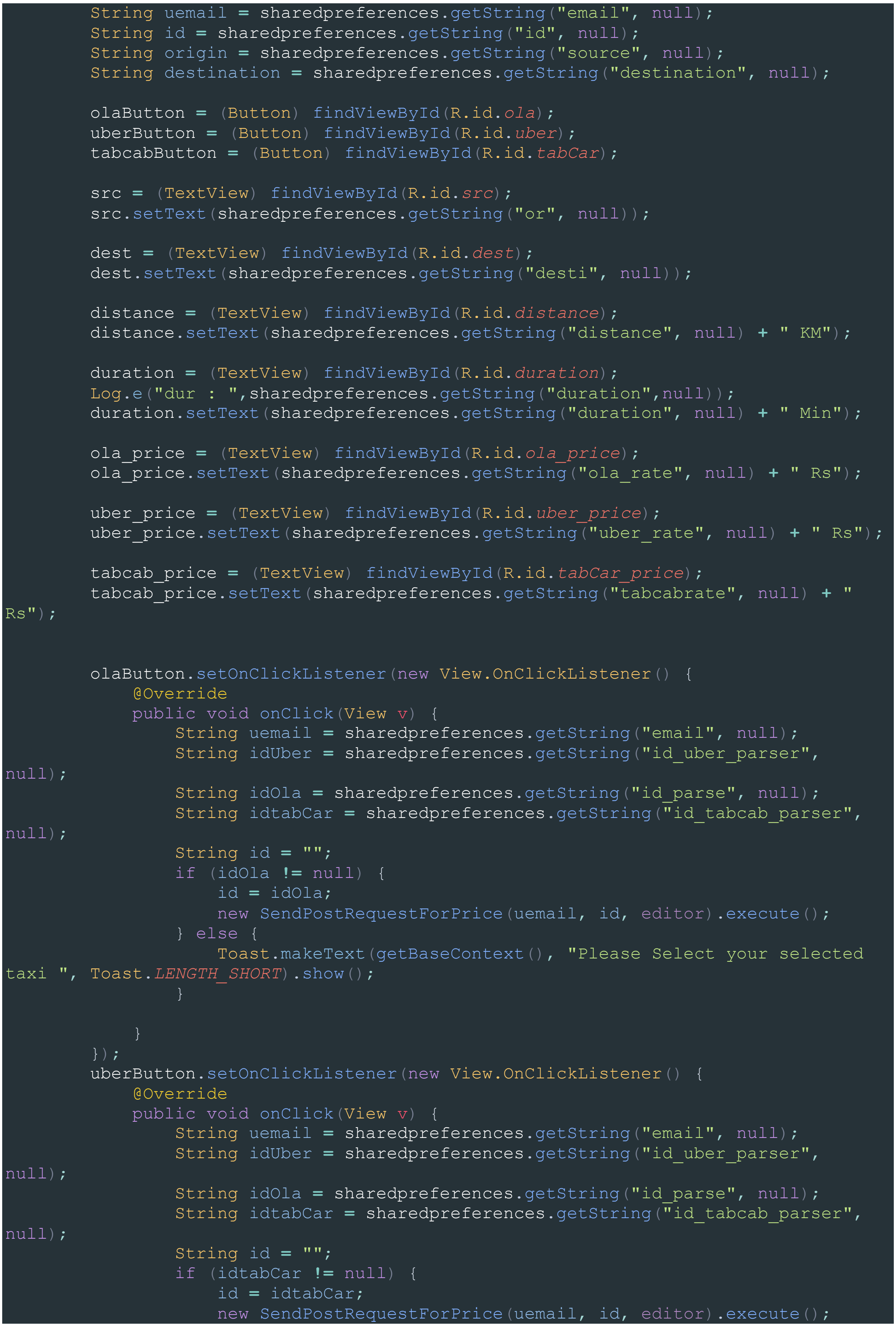
```
                        Log.e("Display", "Message");
                        JSONObject msg = response.getJSONObject("msg");
                        String origin = msg.getString("origin");
                        editor.putString("or", origin);
                        Log.e("Or : ",origin);
                        editor.commit();

                        String destination1 = msg.getString("destination");
                        editor.putString("desti", destination1 + "");
                        Log.e("desti : ",destination1);
                        editor.commit();

                        String distance1 = msg.getInt("distance") + "";
                        editor.putString("distance", distance1);
                        Log.e("desti : ",destination1);
                        editor.commit();

                        String duration1 = msg.getInt("duration") + "";
                        editor.putString("duration", duration1);
                        Log.e("duration : ",duration1);
                        editor.commit();

                        JSONObject ola = msg.getJSONObject("ola");
                        String olaid = ola.getString("id");
                        editor.putString("id_parse", olaid);
                        editor.commit();

                        String olarate = ola.getString("rate");
                        editor.putString("ola_rate", olarate);
                        editor.commit();


                        JSONObject uber = msg.getJSONObject("uber");
                        String uberid = uber.getString("id");
                        editor.putString("id_uber_parser", uberid);
                        editor.commit();

                        String uberrate = uber.getString("rate");
                        editor.putString("uber_rate", uberrate);
                        editor.commit();


                        JSONObject tabcab = msg.getJSONObject("tabcab");
                        String tabcabid = tabcab.getString("id");
                        editor.putString("id_tabcab_parser", tabcabid);
                        editor.commit();

                        String tabcabrate = tabcab.getString("rate");
                        editor.putString("tabcabrate", tabcabrate);
                        editor.commit();


//                        editor.commit();

                        // Toast.makeText(getApplicationContext(), "Login
Sucessfully!", Toast.LENGTH_SHORT).show();
                        //startActivity(new Intent(getBaseContext(),
MapActivity.class));
                        pr.dismiss();
                        startActivity(new
Intent(getBaseContext(),PriceListActivity.class));
                        //finish();
                    } else {
                        pr.dismiss();
                        Toast.makeText(getBaseContext(), "Data is invalid : ",
Toast.LENGTH_SHORT).show();
```

```
                    }
                } catch (JSONException e) {
                    e.printStackTrace();
                }
            }
        }
    }

    public String getPostDataString(JSONObject params) throws Exception {

        StringBuilder result = new StringBuilder();
        boolean first = true;

        Iterator<String> itr = params.keys();

        while (itr.hasNext()) {

            String key = itr.next();
            Object value = params.get(key);

            if (first)
                first = false;
            else
                result.append("&");

            result.append(URLEncoder.encode(key, "UTF-8"));
            result.append("=");
            result.append(URLEncoder.encode(value.toString(), "UTF-8"));

        }
        return result.toString();
    }
    @Override
    public boolean onCreateOptionsMenu(Menu menu){
        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.hometoolbar, menu);
        return true;
    }


    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch(item.getItemId()){
            case R.id.logout:
                String email = sharedPreferences.getString("email",null);
                SharedPreferences.Editor editor = sharedPreferences.edit();
                editor.putString("email", null);
                editor.putString("password", null);
                editor.commit();
                Toast.makeText(getBaseContext(),"Logout successfully
done",Toast.LENGTH_SHORT).show();
                startActivity(new Intent(getBaseContext(),MainActivity.class));
                finishAffinity();
                break;
        }
        return super.onOptionsItemSelected(item);
    }
}
```

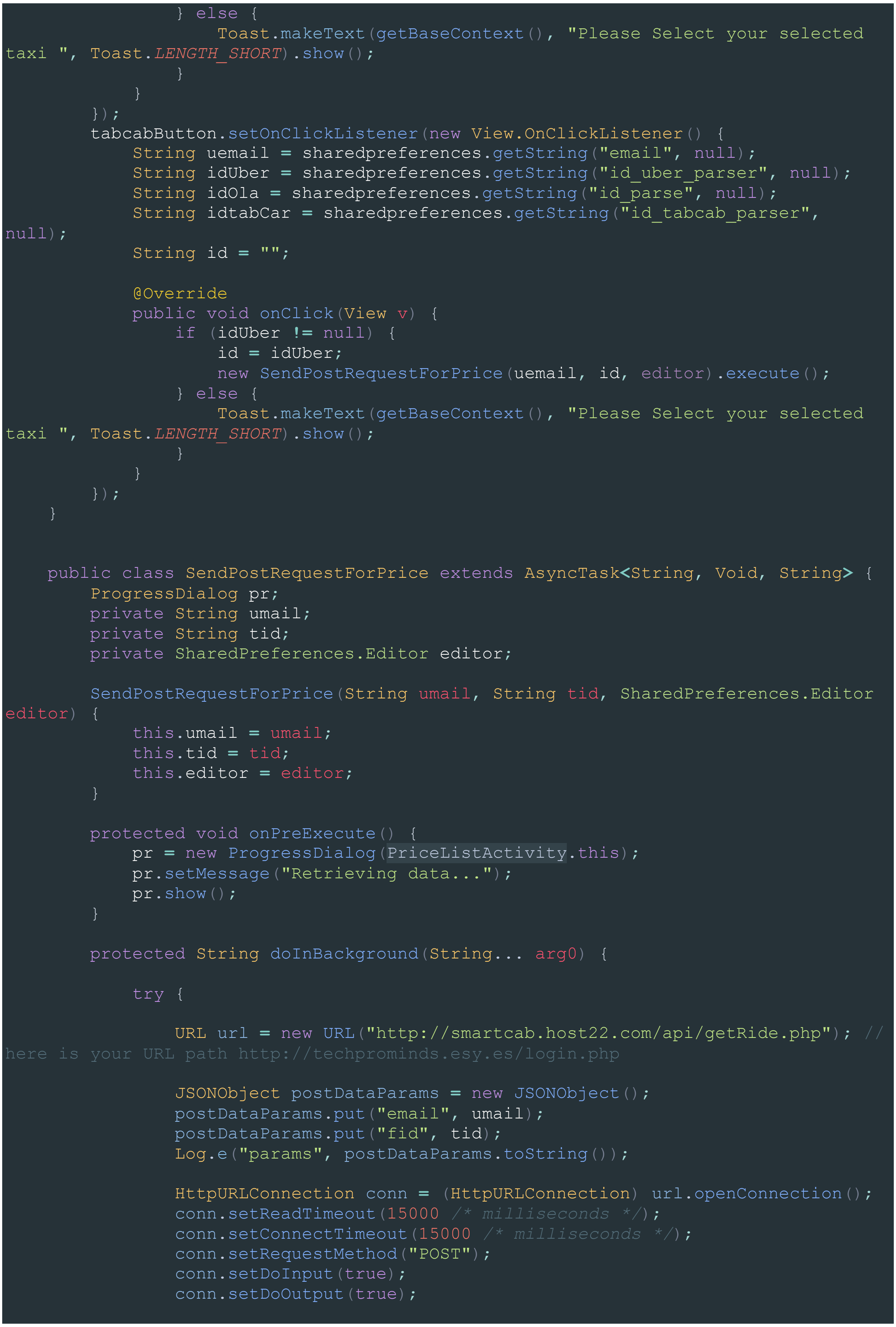
PriceListActivity.java



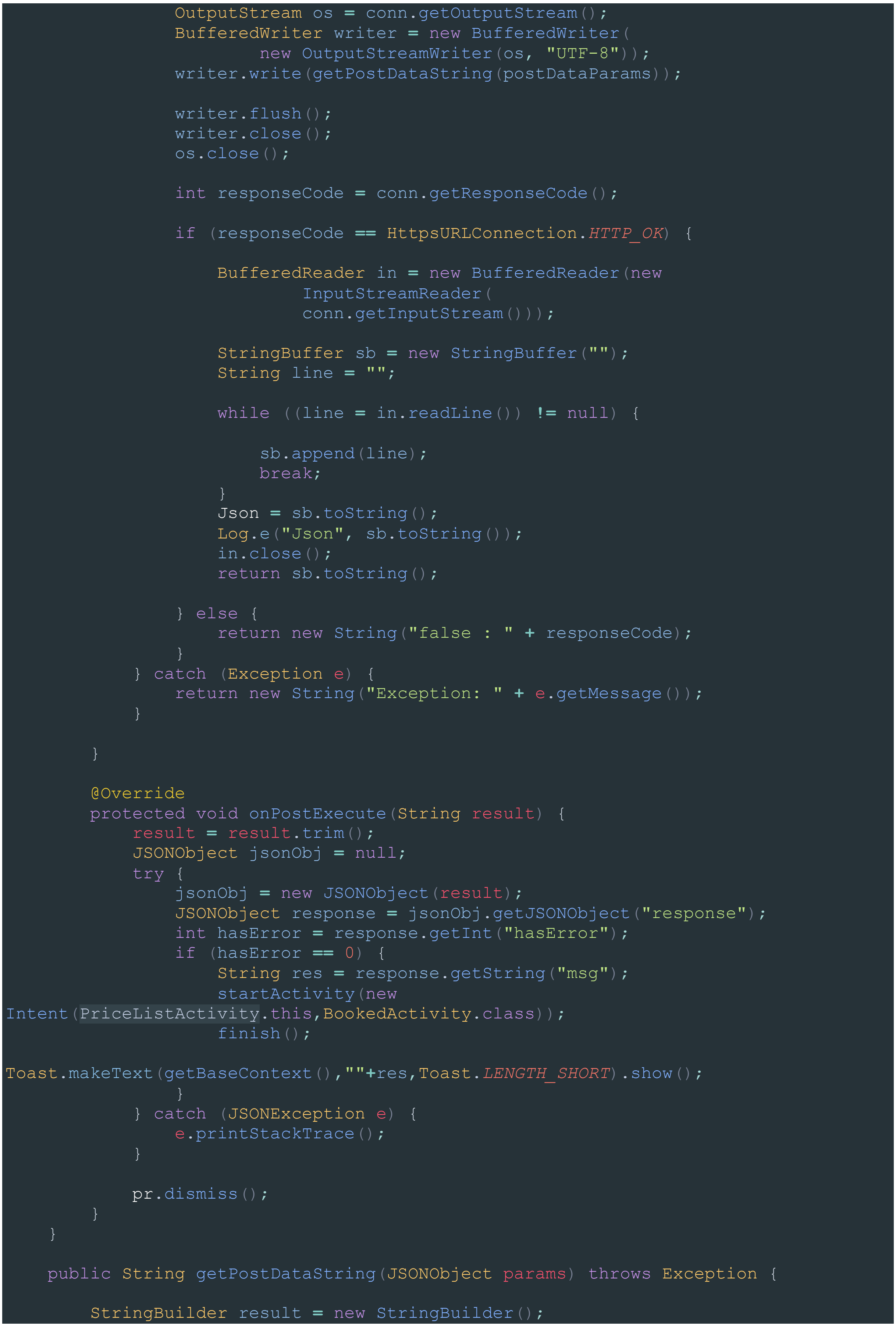
```java
package com.sumit.cab.smartcab;

import android.app.ProgressDialog;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.Color;
import android.os.AsyncTask;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.support.v7.widget.Toolbar;
import android.util.Log;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.URL;
import java.net.URLEncoder;
import java.util.Iterator;

import javax.net.ssl.HttpsURLConnection;

public class PriceListActivity extends AppCompatActivity {
    private String Json = "";
    private TextView src, dest;
    private TextView duration, distance;
    private TextView ola_price, uber_price, tabcab_price;
    private SharedPreferences sharedpreferences;
    private Button olaButton, uberButton, tabcabButton;
    Toolbar toolbar;
    private String em;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_price_list);
        sharedpreferences = getSharedPreferences("SmartCab", Context.MODE_PRIVATE);
        final SharedPreferences.Editor editor = sharedpreferences.edit();

        toolbar = (Toolbar) findViewById(R.id.homeToolbar);
        toolbar.setTitle("SmartCab");
        setSupportActionBar(toolbar);
        toolbar.setTitleTextColor(Color.WHITE);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
```



```
        String uemail = sharedpreferences.getString("email", null);
        String id = sharedpreferences.getString("id", null);
        String origin = sharedpreferences.getString("source", null);
        String destination = sharedpreferences.getString("destination", null);

        olaButton = (Button) findViewById(R.id.ola);
        uberButton = (Button) findViewById(R.id.uber);
        tabcabButton = (Button) findViewById(R.id.tabCar);

        src = (TextView) findViewById(R.id.src);
        src.setText(sharedpreferences.getString("or", null));

        dest = (TextView) findViewById(R.id.dest);
        dest.setText(sharedpreferences.getString("desti", null));

        distance = (TextView) findViewById(R.id.distance);
        distance.setText(sharedpreferences.getString("distance", null) + " KM");

        duration = (TextView) findViewById(R.id.duration);
        Log.e("dur : ",sharedpreferences.getString("duration",null));
        duration.setText(sharedpreferences.getString("duration", null) + " Min");

        ola_price = (TextView) findViewById(R.id.ola_price);
        ola_price.setText(sharedpreferences.getString("ola_rate", null) + " Rs");

        uber_price = (TextView) findViewById(R.id.uber_price);
        uber_price.setText(sharedpreferences.getString("uber_rate", null) + " Rs");

        tabcab_price = (TextView) findViewById(R.id.tabCar_price);
        tabcab_price.setText(sharedpreferences.getString("tabcabrate", null) + "
Rs");


        olaButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String uemail = sharedpreferences.getString("email", null);
                String idUber = sharedpreferences.getString("id_uber_parser",
null);
                String idOla = sharedpreferences.getString("id_parse", null);
                String idtabCar = sharedpreferences.getString("id_tabcab_parser",
null);
                String id = "";
                if (idOla != null) {
                    id = idOla;
                    new SendPostRequestForPrice(uemail, id, editor).execute();
                } else {
                    Toast.makeText(getBaseContext(), "Please Select your selected
taxi ", Toast.LENGTH_SHORT).show();
                }

            }
        });
        uberButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String uemail = sharedpreferences.getString("email", null);
                String idUber = sharedpreferences.getString("id_uber_parser",
null);
                String idOla = sharedpreferences.getString("id_parse", null);
                String idtabCar = sharedpreferences.getString("id_tabcab_parser",
null);
                String id = "";
                if (idtabCar != null) {
                    id = idtabCar;
                    new SendPostRequestForPrice(uemail, id, editor).execute();
```



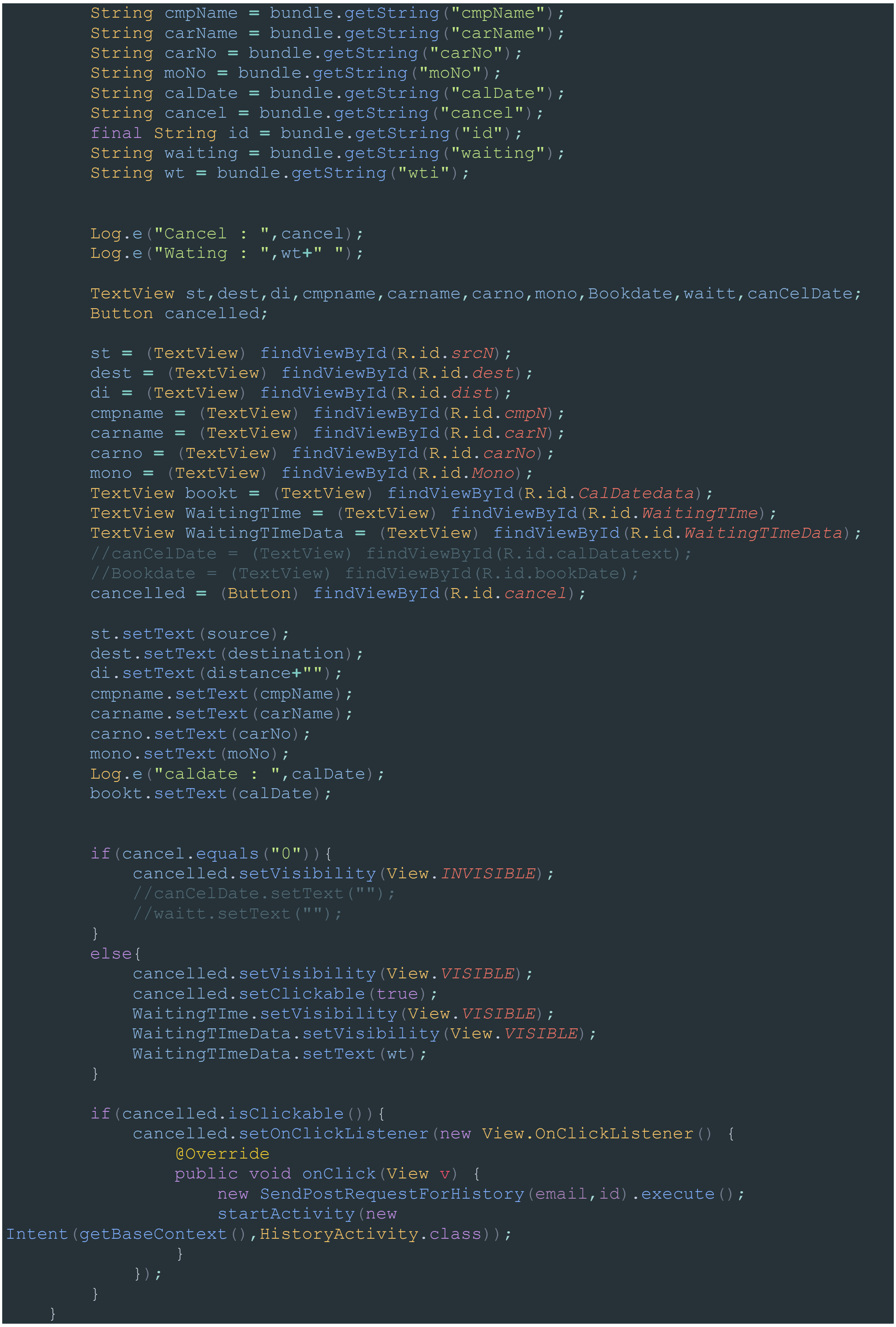
```
                } else {
                    Toast.makeText(getBaseContext(), "Please Select your selected
taxi ", Toast.LENGTH_SHORT).show();
                }
            }
        });
        tabcabButton.setOnClickListener(new View.OnClickListener() {
            String uemail = sharedpreferences.getString("email", null);
            String idUber = sharedpreferences.getString("id_uber_parser", null);
            String idOla = sharedpreferences.getString("id_parse", null);
            String idtabCar = sharedpreferences.getString("id_tabcab_parser",
null);
            String id = "";

            @Override
            public void onClick(View v) {
                if (idUber != null) {
                    id = idUber;
                    new SendPostRequestForPrice(uemail, id, editor).execute();
                } else {
                    Toast.makeText(getBaseContext(), "Please Select your selected
taxi ", Toast.LENGTH_SHORT).show();
                }
            }
        });
    }


    public class SendPostRequestForPrice extends AsyncTask<String, Void, String> {
        ProgressDialog pr;
        private String umail;
        private String tid;
        private SharedPreferences.Editor editor;

        SendPostRequestForPrice(String umail, String tid, SharedPreferences.Editor
editor) {
            this.umail = umail;
            this.tid = tid;
            this.editor = editor;
        }

        protected void onPreExecute() {
            pr = new ProgressDialog(PriceListActivity.this);
            pr.setMessage("Retrieving data...");
            pr.show();
        }

        protected String doInBackground(String... arg0) {

            try {

                URL url = new URL("http://smartcab.host22.com/api/getRide.php"); //
here is your URL path http://techprominds.esy.es/login.php

                JSONObject postDataParams = new JSONObject();
                postDataParams.put("email", umail);
                postDataParams.put("fid", tid);
                Log.e("params", postDataParams.toString());

                HttpURLConnection conn = (HttpURLConnection) url.openConnection();
                conn.setReadTimeout(15000 /* milliseconds */);
                conn.setConnectTimeout(15000 /* milliseconds */);
                conn.setRequestMethod("POST");
                conn.setDoInput(true);
                conn.setDoOutput(true);
```



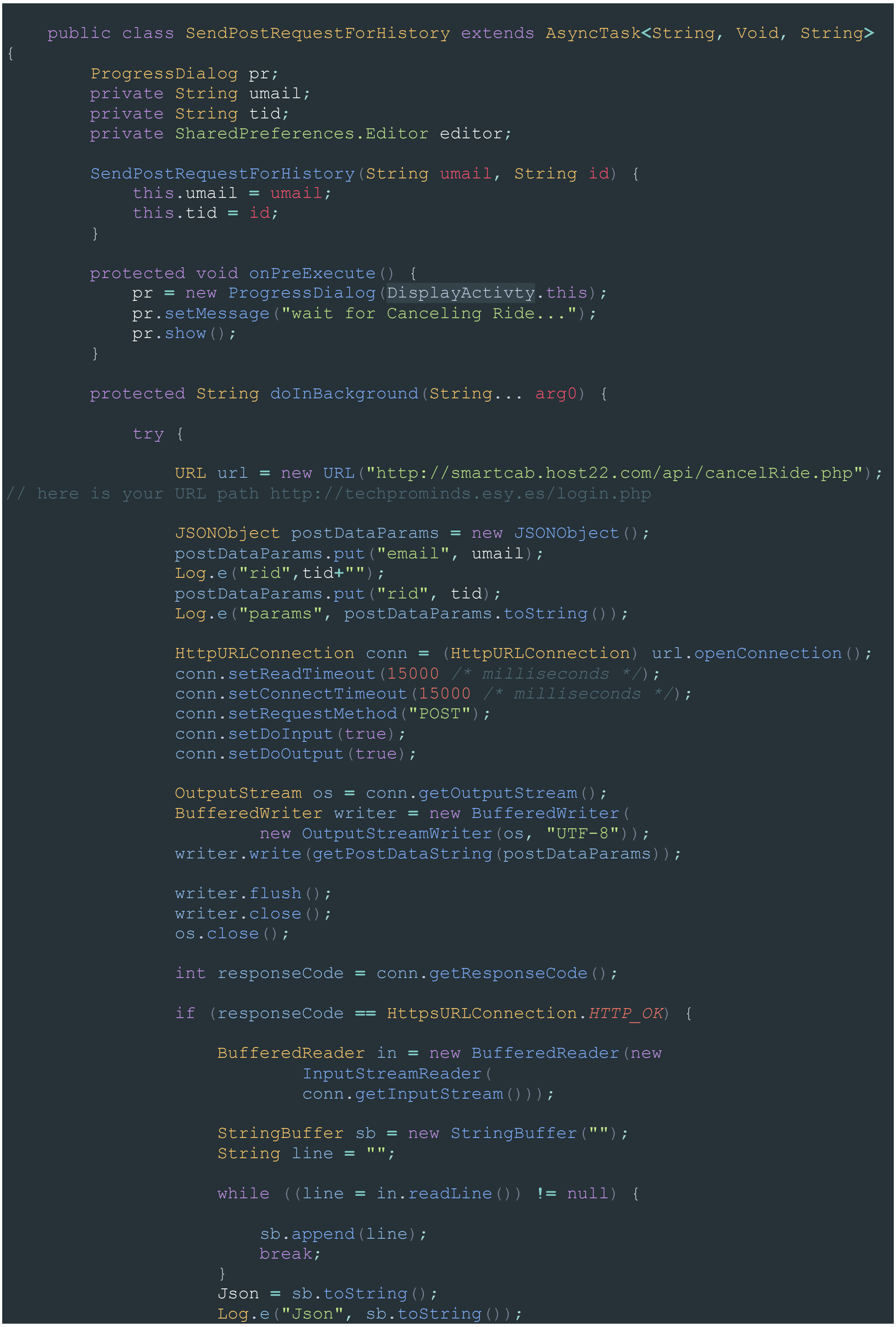
```
                OutputStream os = conn.getOutputStream();
                BufferedWriter writer = new BufferedWriter(
                        new OutputStreamWriter(os, "UTF-8"));
                writer.write(getPostDataString(postDataParams));

                writer.flush();
                writer.close();
                os.close();

                int responseCode = conn.getResponseCode();

                if (responseCode == HttpsURLConnection.HTTP_OK) {

                    BufferedReader in = new BufferedReader(new
                            InputStreamReader(
                            conn.getInputStream()));

                    StringBuffer sb = new StringBuffer("");
                    String line = "";

                    while ((line = in.readLine()) != null) {

                        sb.append(line);
                        break;
                    }
                    Json = sb.toString();
                    Log.e("Json", sb.toString());
                    in.close();
                    return sb.toString();

                } else {
                    return new String("false : " + responseCode);
                }
            } catch (Exception e) {
                return new String("Exception: " + e.getMessage());
            }

        }

        @Override
        protected void onPostExecute(String result) {
            result = result.trim();
            JSONObject jsonObj = null;
            try {
                jsonObj = new JSONObject(result);
                JSONObject response = jsonObj.getJSONObject("response");
                int hasError = response.getInt("hasError");
                if (hasError == 0) {
                    String res = response.getString("msg");
                    startActivity(new
Intent(PriceListActivity.this,BookedActivity.class));
                    finish();

Toast.makeText(getBaseContext(),""+res,Toast.LENGTH_SHORT).show();
                }
            } catch (JSONException e) {
                e.printStackTrace();
            }

            pr.dismiss();
        }
    }

    public String getPostDataString(JSONObject params) throws Exception {

        StringBuilder result = new StringBuilder();
        boolean first = true;

        Iterator<String> itr = params.keys();

        while (itr.hasNext()) {

            String key = itr.next();
            Object value = params.get(key);

            if (first)
                first = false;
            else
                result.append("&");

            result.append(URLEncoder.encode(key, "UTF-8"));
            result.append("=");
            result.append(URLEncoder.encode(value.toString(), "UTF-8"));

        }
        return result.toString();
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu){
        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.hometoolbar, menu);
        return true;
    }


    @Override
    public boolean onOptionsItemSelected(MenuItem item)  {
        switch(item.getItemId()){
            case R.id.logout:
                String email = sharedpreferences.getString("email",null);
                SharedPreferences.Editor editor = sharedpreferences.edit();
                editor.putString("email", null);
                editor.putString("password", null);
                editor.commit();
                Toast.makeText(getBaseContext(),"Logout successfully
done",Toast.LENGTH_SHORT).show();
                startActivity(new Intent(getBaseContext(),MainActivity.class));
                finishAffinity();
                break;
        }
        return super.onOptionsItemSelected(item);
    }
}
```

DisplayActivity.java

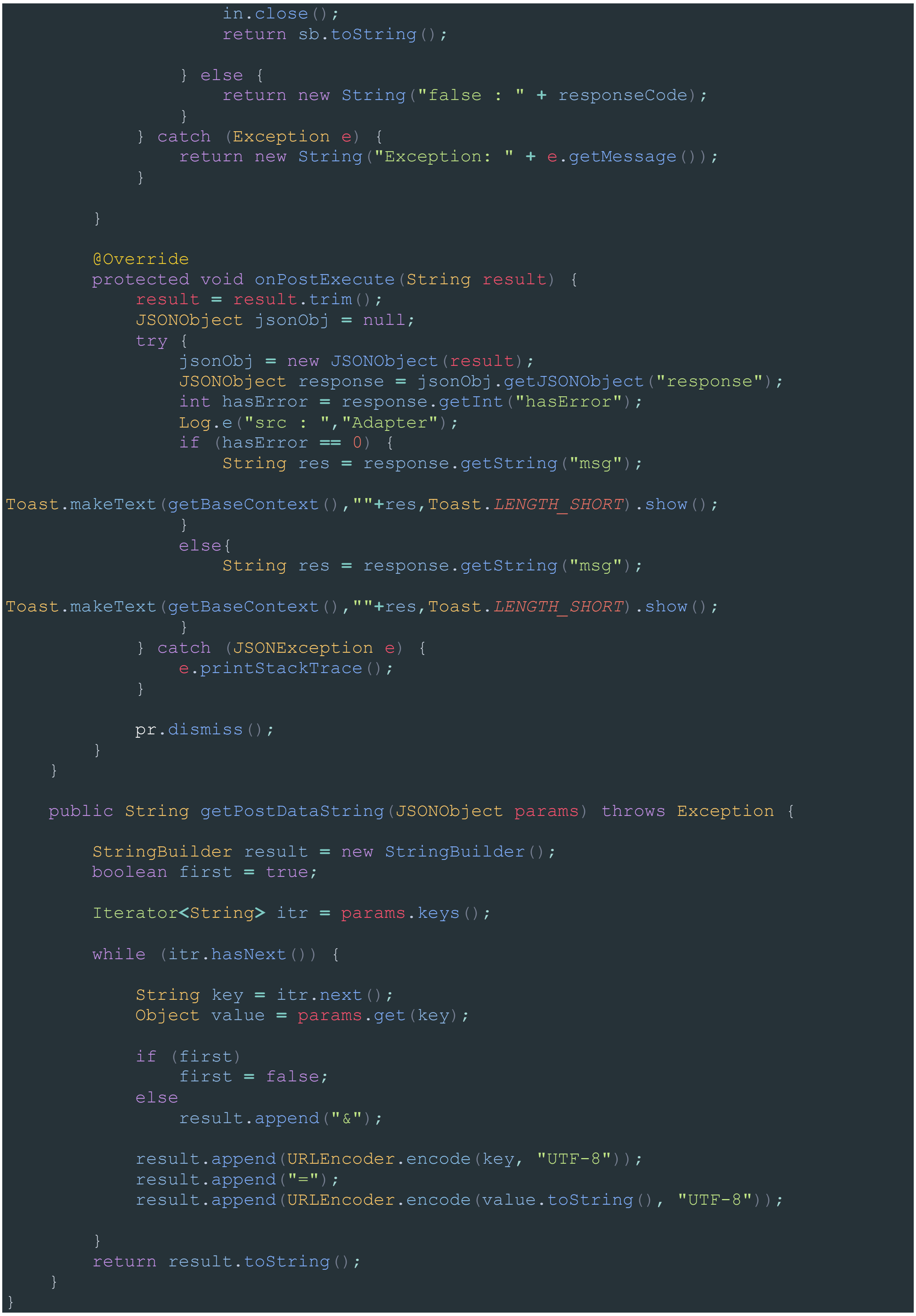
```java
package com.sumit.cab.smartcab.History;

import android.app.ProgressDialog;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.Color;
import android.os.AsyncTask;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.support.v7.widget.ButtonBarLayout;
import android.support.v7.widget.Toolbar;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

import com.sumit.cab.smartcab.R;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.URL;
import java.net.URLEncoder;
import java.util.ArrayList;
import java.util.Iterator;

import javax.net.ssl.HttpsURLConnection;

public class DisplayActivty extends AppCompatActivity {

    private String Json;
    SharedPreferences sharedPreferences;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_display_activty);

        Toolbar toolbar = (Toolbar) findViewById(R.id.homeToolbar);
        setSupportActionBar(toolbar);
        toolbar.setTitle("History");
        toolbar.setTitleTextColor(Color.WHITE);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);

        Bundle bundle = getIntent().getExtras();

        sharedPreferences = getSharedPreferences("SmartCab", MODE_PRIVATE);
        final String email = sharedPreferences.getString("email", null);

        String source = bundle.getString("source");
        String destination = bundle.getString("destination");
        String distance = bundle.getString("Distance");
```



```
        String cmpName = bundle.getString("cmpName");
        String carName = bundle.getString("carName");
        String carNo = bundle.getString("carNo");
        String moNo = bundle.getString("moNo");
        String calDate = bundle.getString("calDate");
        String cancel = bundle.getString("cancel");
        final String id = bundle.getString("id");
        String waiting = bundle.getString("waiting");
        String wt = bundle.getString("wti");


        Log.e("Cancel : ",cancel);
        Log.e("Wating : ",wt+" ");

        TextView st,dest,di,cmpname,carname,carno,mono,Bookdate,waitt,canCelDate;
        Button cancelled;

        st = (TextView) findViewById(R.id.srcN);
        dest = (TextView) findViewById(R.id.dest);
        di = (TextView) findViewById(R.id.dist);
        cmpname = (TextView) findViewById(R.id.cmpN);
        carname = (TextView) findViewById(R.id.carN);
        carno = (TextView) findViewById(R.id.carNo);
        mono = (TextView) findViewById(R.id.Mono);
        TextView bookt = (TextView) findViewById(R.id.CalDatedata);
        TextView WaitingTIme = (TextView) findViewById(R.id.WaitingTIme);
        TextView WaitingTImeData = (TextView) findViewById(R.id.WaitingTImeData);
        //canCelDate = (TextView) findViewById(R.id.calDatatext);
        //Bookdate = (TextView) findViewById(R.id.bookDate);
        cancelled = (Button) findViewById(R.id.cancel);

        st.setText(source);
        dest.setText(destination);
        di.setText(distance+"");
        cmpname.setText(cmpName);
        carname.setText(carName);
        carno.setText(carNo);
        mono.setText(moNo);
        Log.e("caldate : ",calDate);
        bookt.setText(calDate);


        if(cancel.equals("0")){
            cancelled.setVisibility(View.INVISIBLE);
            //canCelDate.setText("");
            //waitt.setText("");
        }
        else{
            cancelled.setVisibility(View.VISIBLE);
            cancelled.setClickable(true);
            WaitingTIme.setVisibility(View.VISIBLE);
            WaitingTImeData.setVisibility(View.VISIBLE);
            WaitingTImeData.setText(wt);
        }

        if(cancelled.isClickable()){
            cancelled.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    new SendPostRequestForHistory(email,id).execute();
                    startActivity(new
Intent(getBaseContext(),HistoryActivity.class));
                }
            });
        }
    }
```



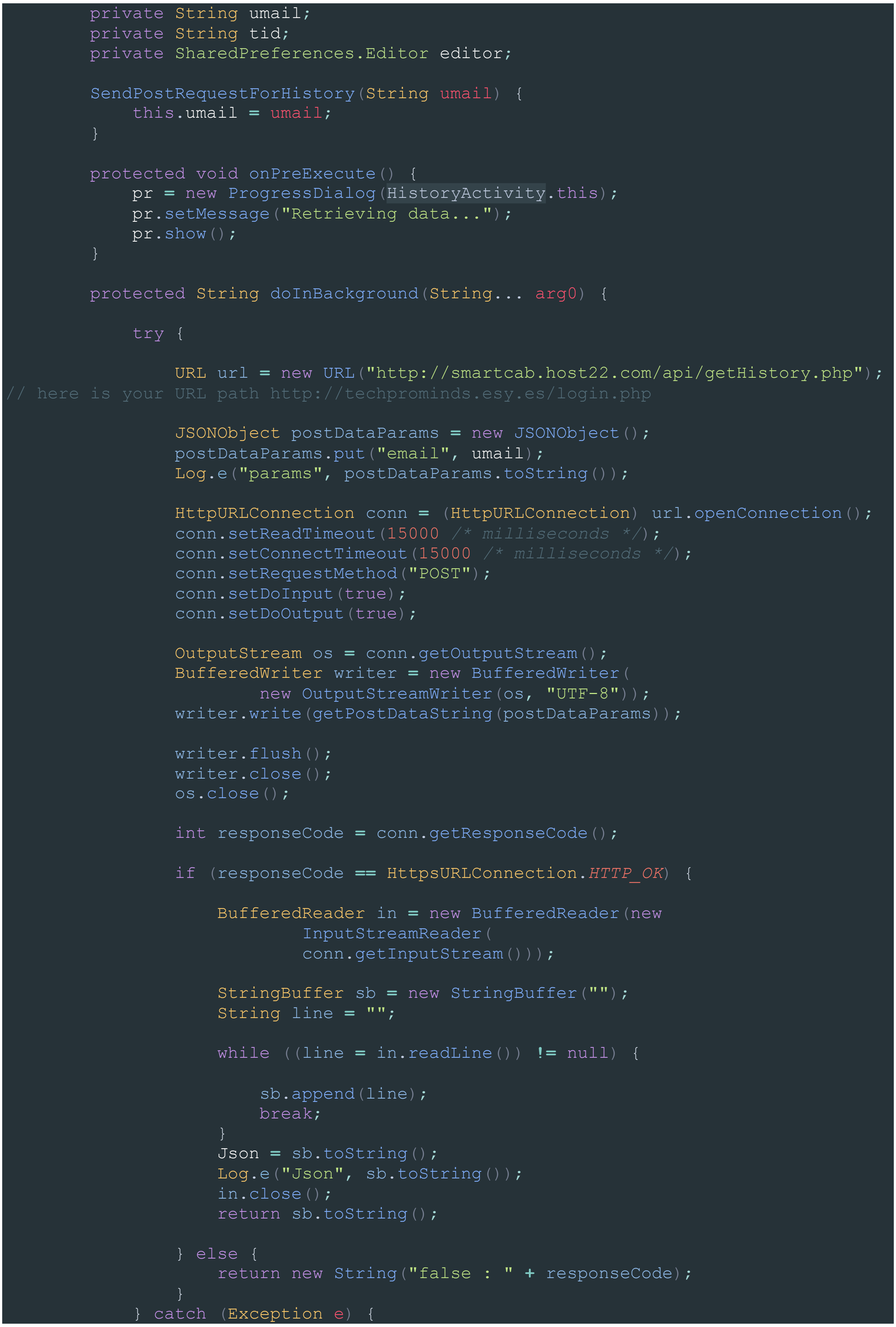
```java
    public class SendPostRequestForHistory extends AsyncTask<String, Void, String>
{
        ProgressDialog pr;
        private String umail;
        private String tid;
        private SharedPreferences.Editor editor;

        SendPostRequestForHistory(String umail, String id) {
            this.umail = umail;
            this.tid = id;
        }

        protected void onPreExecute() {
            pr = new ProgressDialog(DisplayActivty.this);
            pr.setMessage("wait for Canceling Ride...");
            pr.show();
        }

        protected String doInBackground(String... arg0) {

            try {

                URL url = new URL("http://smartcab.host22.com/api/cancelRide.php");
// here is your URL path http://techpromindss.esy.es/login.php

                JSONObject postDataParams = new JSONObject();
                postDataParams.put("email", umail);
                Log.e("rid",tid+"");
                postDataParams.put("rid", tid);
                Log.e("params", postDataParams.toString());

                HttpURLConnection conn = (HttpURLConnection) url.openConnection();
                conn.setReadTimeout(15000 /* milliseconds */);
                conn.setConnectTimeout(15000 /* milliseconds */);
                conn.setRequestMethod("POST");
                conn.setDoInput(true);
                conn.setDoOutput(true);

                OutputStream os = conn.getOutputStream();
                BufferedWriter writer = new BufferedWriter(
                        new OutputStreamWriter(os, "UTF-8"));
                writer.write(getPostDataString(postDataParams));

                writer.flush();
                writer.close();
                os.close();

                int responseCode = conn.getResponseCode();

                if (responseCode == HttpsURLConnection.HTTP_OK) {

                    BufferedReader in = new BufferedReader(new
                            InputStreamReader(
                            conn.getInputStream()));

                    StringBuffer sb = new StringBuffer("");
                    String line = "";

                    while ((line = in.readLine()) != null) {

                        sb.append(line);
                        break;
                    }
                    Json = sb.toString();
                    Log.e("Json", sb.toString());
```



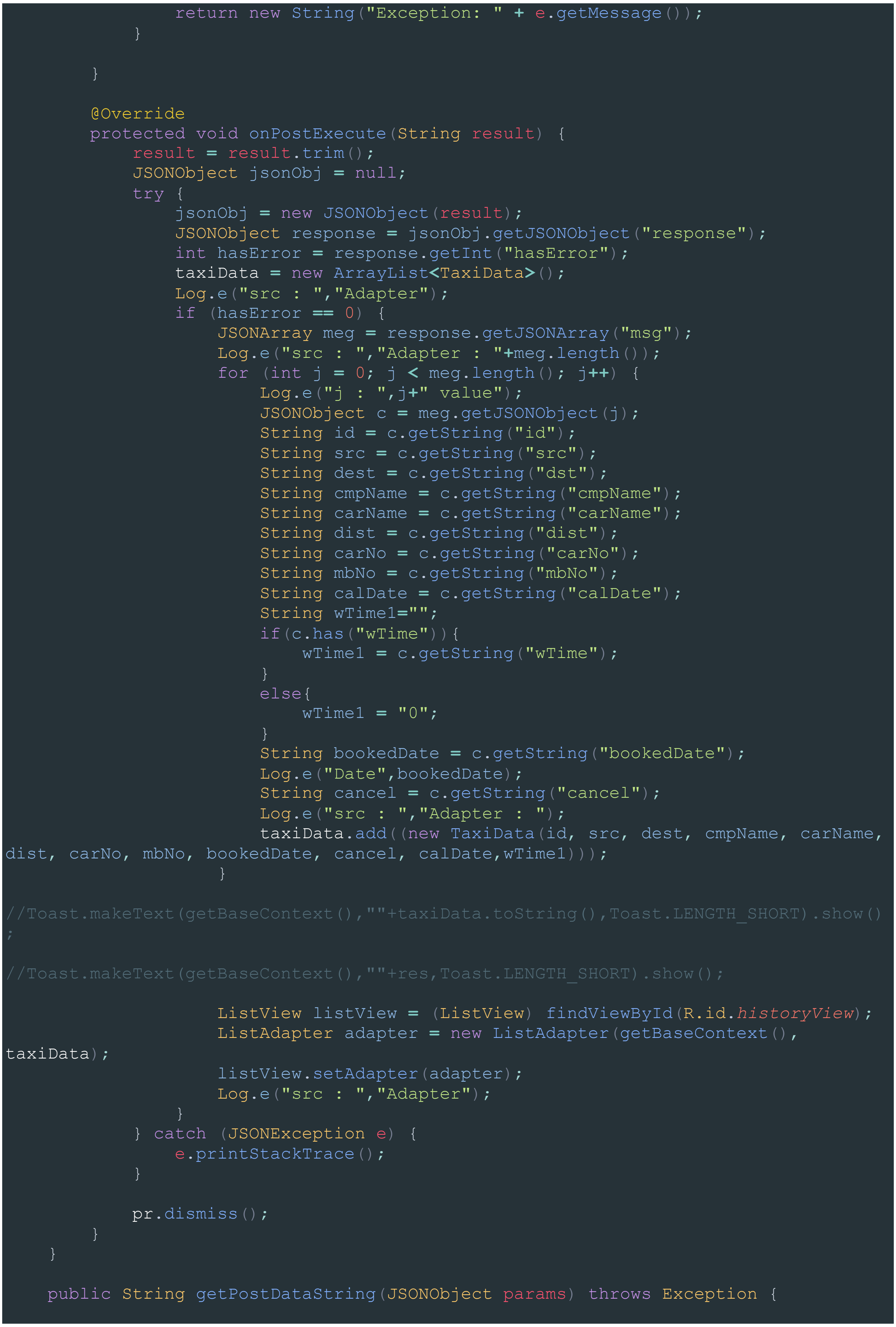
```java
                    in.close();
                    return sb.toString();

                } else {
                    return new String("false : " + responseCode);
                }
            } catch (Exception e) {
                return new String("Exception: " + e.getMessage());
            }

        }

        @Override
        protected void onPostExecute(String result) {
            result = result.trim();
            JSONObject jsonObj = null;
            try {
                jsonObj = new JSONObject(result);
                JSONObject response = jsonObj.getJSONObject("response");
                int hasError = response.getInt("hasError");
                Log.e("src : ","Adapter");
                if (hasError == 0) {
                    String res = response.getString("msg");

Toast.makeText(getBaseContext(),""+res,Toast.LENGTH_SHORT).show();
                }
                else{
                    String res = response.getString("msg");

Toast.makeText(getBaseContext(),""+res,Toast.LENGTH_SHORT).show();
                }
            } catch (JSONException e) {
                e.printStackTrace();
            }

            pr.dismiss();
        }
    }

    public String getPostDataString(JSONObject params) throws Exception {

        StringBuilder result = new StringBuilder();
        boolean first = true;

        Iterator<String> itr = params.keys();

        while (itr.hasNext()) {

            String key = itr.next();
            Object value = params.get(key);

            if (first)
                first = false;
            else
                result.append("&");

            result.append(URLEncoder.encode(key, "UTF-8"));
            result.append("=");
            result.append(URLEncoder.encode(value.toString(), "UTF-8"));

        }
        return result.toString();
    }
}
```

HistoryActivity.java

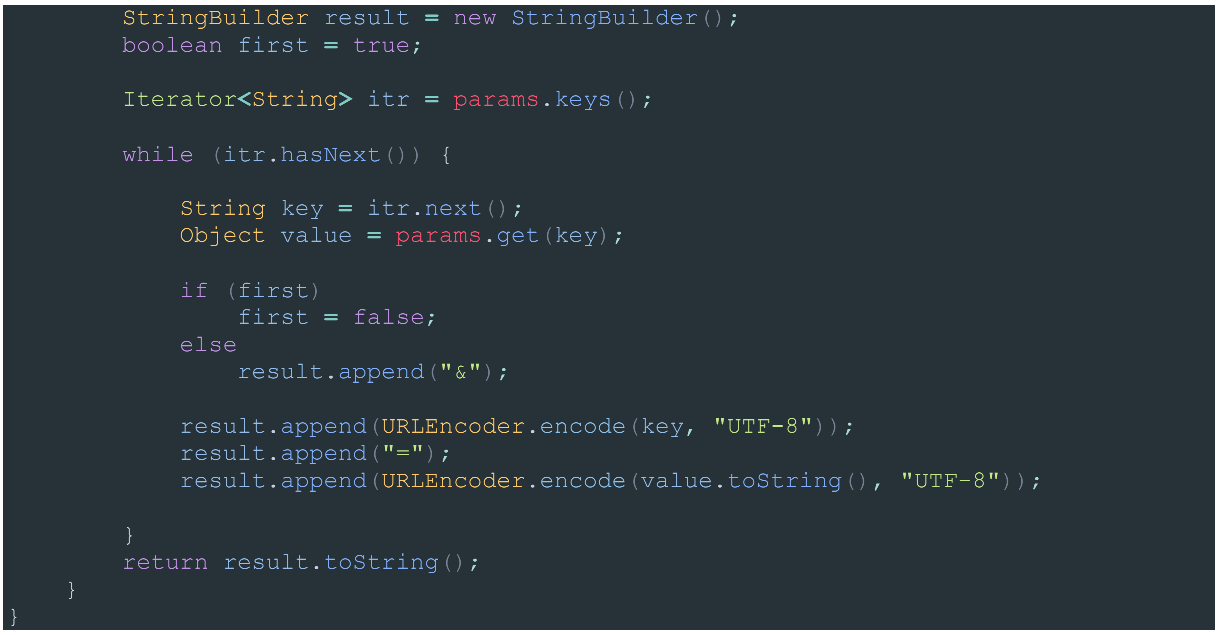
```
package com.sumit.cab.smartcab.History;

import android.app.ProgressDialog;
import android.content.SharedPreferences;
import android.graphics.Color;
import android.os.AsyncTask;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.support.v7.widget.Toolbar;
import android.util.Log;
import android.widget.ListView;
import android.widget.Toast;

import com.sumit.cab.smartcab.PriceListActivity;
import com.sumit.cab.smartcab.R;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.io.StringReader;
import java.net.HttpURLConnection;
import java.net.URL;
import java.net.URLEncoder;
import java.util.ArrayList;
import java.util.Iterator;

import javax.net.ssl.HttpsURLConnection;

public class HistoryActivity extends AppCompatActivity {

    private String Json = "";
    ArrayList<TaxiData> taxiData;
    SharedPreferences sharedPreferences;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_history);

        Toolbar toolbar = (Toolbar) findViewById(R.id.homeToolbar);
        toolbar.setTitle("History");
        setSupportActionBar(toolbar);
        toolbar.setTitleTextColor(Color.WHITE);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);

        sharedPreferences = getSharedPreferences("SmartCab", MODE_PRIVATE);
        String email = sharedPreferences.getString("email", null);
        new SendPostRequestForHistory(email).execute();

    }

    public class SendPostRequestForHistory extends AsyncTask<String, Void, String>
{
        ProgressDialog pr;
```



```
        private String umail;
        private String tid;
        private SharedPreferences.Editor editor;

        SendPostRequestForHistory(String umail) {
            this.umail = umail;
        }

        protected void onPreExecute() {
            pr = new ProgressDialog(HistoryActivity.this);
            pr.setMessage("Retrieving data...");
            pr.show();
        }

        protected String doInBackground(String... arg0) {

            try {

                URL url = new URL("http://smartcab.host22.com/api/getHistory.php");
// here is your URL path http://techprominds.esy.es/login.php

                JSONObject postDataParams = new JSONObject();
                postDataParams.put("email", umail);
                Log.e("params", postDataParams.toString());

                HttpURLConnection conn = (HttpURLConnection) url.openConnection();
                conn.setReadTimeout(15000 /* milliseconds */);
                conn.setConnectTimeout(15000 /* milliseconds */);
                conn.setRequestMethod("POST");
                conn.setDoInput(true);
                conn.setDoOutput(true);

                OutputStream os = conn.getOutputStream();
                BufferedWriter writer = new BufferedWriter(
                        new OutputStreamWriter(os, "UTF-8"));
                writer.write(getPostDataString(postDataParams));

                writer.flush();
                writer.close();
                os.close();

                int responseCode = conn.getResponseCode();

                if (responseCode == HttpsURLConnection.HTTP_OK) {

                    BufferedReader in = new BufferedReader(new
                            InputStreamReader(
                            conn.getInputStream()));

                    StringBuffer sb = new StringBuffer("");
                    String line = "";

                    while ((line = in.readLine()) != null) {

                        sb.append(line);
                        break;
                    }
                    Json = sb.toString();
                    Log.e("Json", sb.toString());
                    in.close();
                    return sb.toString();

                } else {
                    return new String("false : " + responseCode);
                }
            } catch (Exception e) {
```

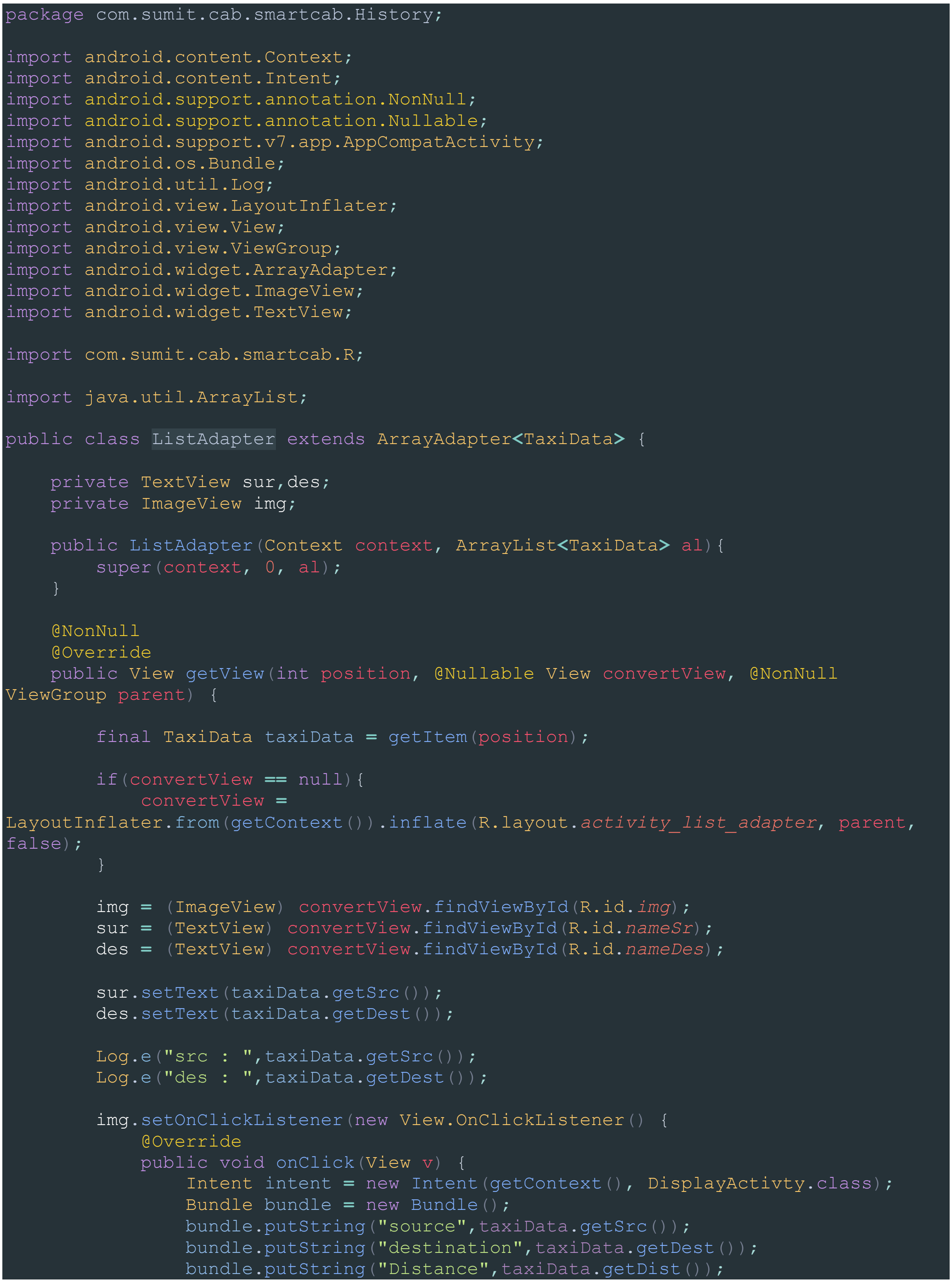


```
                return new String("Exception: " + e.getMessage());
            }

        }

        @Override
        protected void onPostExecute(String result) {
            result = result.trim();
            JSONObject jsonObj = null;
            try {
                jsonObj = new JSONObject(result);
                JSONObject response = jsonObj.getJSONObject("response");
                int hasError = response.getInt("hasError");
                taxiData = new ArrayList<TaxiData>();
                Log.e("src : ","Adapter");
                if (hasError == 0) {
                    JSONArray meg = response.getJSONArray("msg");
                    Log.e("src : ","Adapter : "+meg.length());
                    for (int j = 0; j < meg.length(); j++) {
                        Log.e("j : ",j+" value");
                        JSONObject c = meg.getJSONObject(j);
                        String id = c.getString("id");
                        String src = c.getString("src");
                        String dest = c.getString("dst");
                        String cmpName = c.getString("cmpName");
                        String carName = c.getString("carName");
                        String dist = c.getString("dist");
                        String carNo = c.getString("carNo");
                        String mbNo = c.getString("mbNo");
                        String calDate = c.getString("calDate");
                        String wTime1="";
                        if(c.has("wTime")){
                            wTime1 = c.getString("wTime");
                        }
                        else{
                            wTime1 = "0";
                        }
                        String bookedDate = c.getString("bookedDate");
                        Log.e("Date",bookedDate);
                        String cancel = c.getString("cancel");
                        Log.e("src : ","Adapter : ");
                        taxiData.add((new TaxiData(id, src, dest, cmpName, carName,
dist, carNo, mbNo, bookedDate, cancel, calDate,wTime1)));
                    }

//Toast.makeText(getBaseContext(),""+taxiData.toString(),Toast.LENGTH_SHORT).show()
;

//Toast.makeText(getBaseContext(),""+res,Toast.LENGTH_SHORT).show();

                    ListView listView = (ListView) findViewById(R.id.historyView);
                    ListAdapter adapter = new ListAdapter(getBaseContext(),
taxiData);

                    listView.setAdapter(adapter);
                    Log.e("src : ","Adapter");
                }
            } catch (JSONException e) {
                e.printStackTrace();
            }

            pr.dismiss();
        }
    }

    public String getPostDataString(JSONObject params) throws Exception {
```



```
        StringBuilder result = new StringBuilder();
        boolean first = true;

        Iterator<String> itr = params.keys();

        while (itr.hasNext()) {

            String key = itr.next();
            Object value = params.get(key);

            if (first)
                first = false;
            else
                result.append("&");

            result.append(URLEncoder.encode(key, "UTF-8"));
            result.append("=");
            result.append(URLEncoder.encode(value.toString(), "UTF-8"));

        }
        return result.toString();
    }
}
```

ListAdapter.java

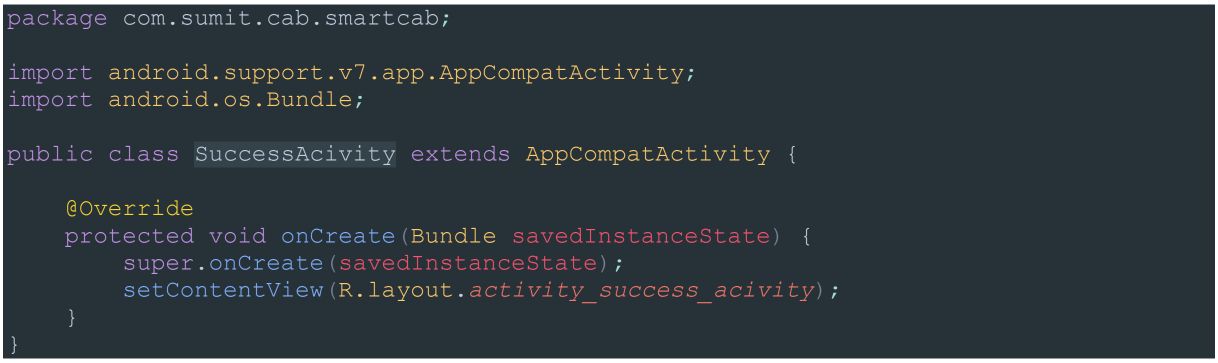


```
package com.sumit.cab.smartcab.History;

import android.content.Context;
import android.content.Intent;
import android.support.annotation.NonNull;
import android.support.annotation.Nullable;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.ImageView;
import android.widget.TextView;

import com.sumit.cab.smartcab.R;

import java.util.ArrayList;

public class ListAdapter extends ArrayAdapter<TaxiData> {

    private TextView sur,des;
    private ImageView img;

    public ListAdapter(Context context, ArrayList<TaxiData> al){
        super(context, 0, al);
    }

    @NonNull
    @Override
    public View getView(int position, @Nullable View convertView, @NonNull
ViewGroup parent) {

        final TaxiData taxiData = getItem(position);

        if(convertView == null){
            convertView =
LayoutInflater.from(getContext()).inflate(R.layout.activity_list_adapter, parent,
false);
        }

        img = (ImageView) convertView.findViewById(R.id.img);
        sur = (TextView) convertView.findViewById(R.id.nameSr);
        des = (TextView) convertView.findViewById(R.id.nameDes);

        sur.setText(taxiData.getSrc());
        des.setText(taxiData.getDest());

        Log.e("src : ",taxiData.getSrc());
        Log.e("des : ",taxiData.getDest());

        img.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(getContext(), DisplayActivty.class);
                Bundle bundle = new Bundle();
                bundle.putString("source",taxiData.getSrc());
                bundle.putString("destination",taxiData.getDest());
                bundle.putString("Distance",taxiData.getDist());
                bundle.putString("cmpName",taxiData.getCmpName());
                bundle.putString("carName",taxiData.getCarName());
                bundle.putString("carNo",taxiData.getCarNo());
                bundle.putString("moNo",taxiData.getMoNo());
                bundle.putString("calDate",taxiData.getCalDate());
                bundle.putString("cancel",taxiData.getCancel());
                bundle.putString("id",taxiData.getId());
                bundle.putString("waiting",taxiData.getWTime());
                bundle.putString("wti",taxiData.getwTime1());
                intent.putExtras(bundle);
                intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
                getContext().startActivity(intent);
            }
        });

        return convertView;
    }
}
```

TaxiData.java

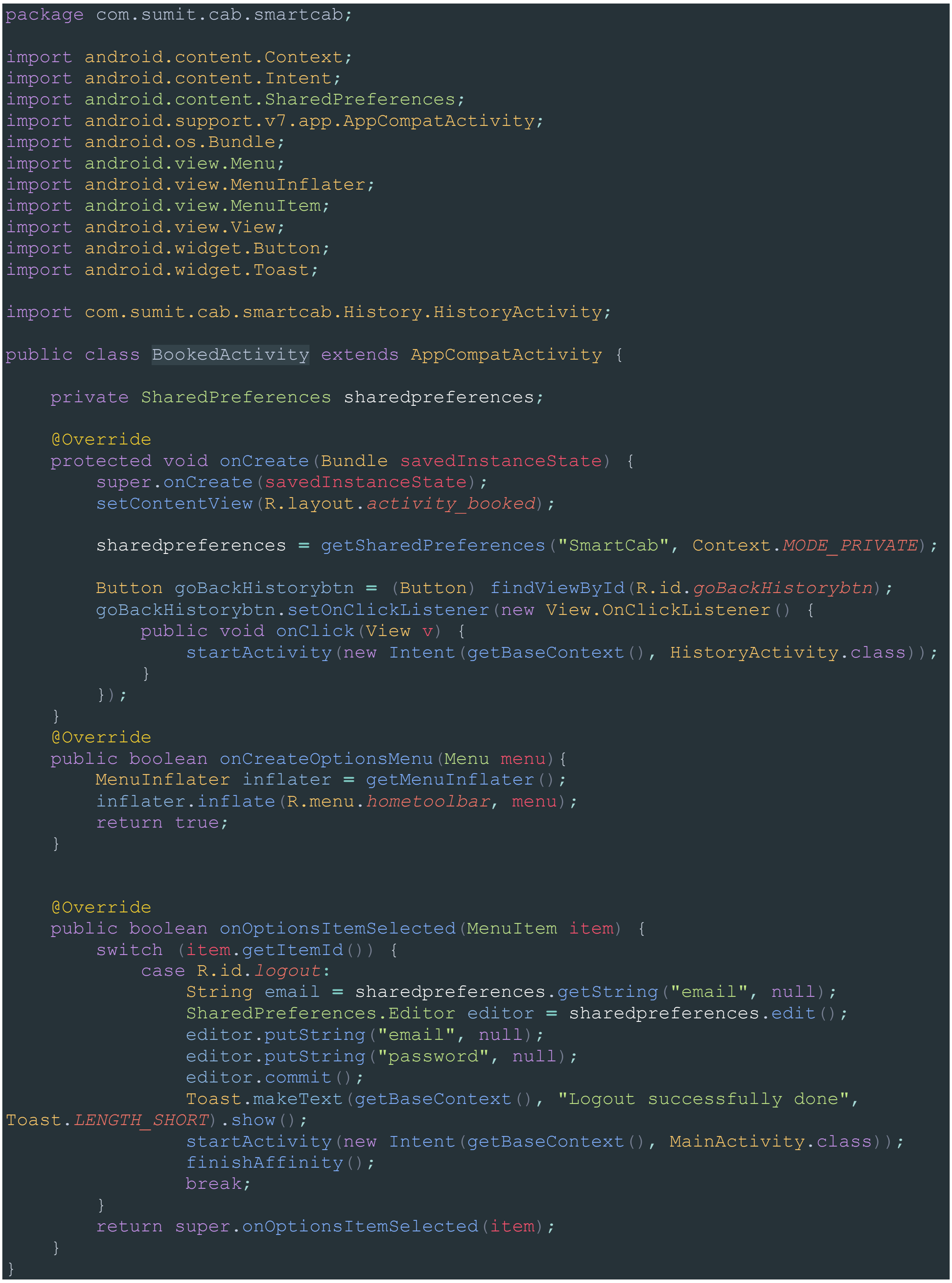
```
package com.sumit.cab.smartcab.History;

/**
 * Created by DELL on 06-04-2017.
 */

public class TaxiData {
    String id;
    String src;
    String dest;
    String cmpName;
    String carName;
    String dist;
    String carNo;
    String moNo;
    String calDate;
    String cancel;
    String WTime;
    String wTime1;

    public TaxiData(String id, String src, String dest, String cmpName, String
carName, String dist, String carNo, String moNo, String calDate, String cancel,
String WTime, String wtime1) {
        this.id = id;
        this.src = src;
        this.dest = dest;
        this.cmpName = cmpName;
        this.carName = carName;
        this.dist = dist;
        this.carNo = carNo;
        this.moNo = moNo;
        this.calDate = calDate;
        this.cancel = cancel;
        this.WTime = WTime;
        this.wTime1 = wtime1;
    }

    public String getId() {
        return id;
    }

    public String getCalDate() {
        return calDate;
    }

    public String getCmpName() {
        return cmpName;
    }

    public String getCancel() {
        return cancel;
    }

    public String getCarName() {
        return carName;
    }

    public String getDest() {
        return dest;
    }
    public String getCarNo() {
        return carNo;
    }

    public String getDist() {
        return dist;
    }

    public String getMoNo() {
        return moNo;
    }

    public String getSrc() {
        return src;
    }

    public String getWTime() {
        return WTime;
    }

    public String getwTime1() {
        return wTime1;
    }
}
```

SuccessActivity.java



```java
package com.sumit.cab.smartcab;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

public class SuccessAcivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_success_acivity);
    }
}
```

BookedActivity.java



```java
package com.sumit.cab.smartcab;

import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

import com.sumit.cab.smartcab.History.HistoryActivity;

public class BookedActivity extends AppCompatActivity {

    private SharedPreferences sharedpreferences;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_booked);

        sharedpreferences = getSharedPreferences("SmartCab", Context.MODE_PRIVATE);

        Button goBackHistorybtn = (Button) findViewById(R.id.goBackHistorybtn);
        goBackHistorybtn.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                startActivity(new Intent(getBaseContext(), HistoryActivity.class));
            }
        });
    }
    @Override
    public boolean onCreateOptionsMenu(Menu menu){
        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.hometoolbar, menu);
        return true;
    }


    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
            case R.id.logout:
                String email = sharedpreferences.getString("email", null);
                SharedPreferences.Editor editor = sharedpreferences.edit();
                editor.putString("email", null);
                editor.putString("password", null);
                editor.commit();
                Toast.makeText(getBaseContext(), "Logout successfully done",
Toast.LENGTH_SHORT).show();
                startActivity(new Intent(getBaseContext(), MainActivity.class));
                finishAffinity();
                break;
        }
        return super.onOptionsItemSelected(item);
    }
}
```

Build.gradle(Project)

```
// Top-level build file where you can add configuration options common to all sub-
projects/modules.

buildscript {
    repositories {
        jcenter()
    }
    dependencies {
        classpath 'com.android.tools.build:gradle:2.3.1'

        // NOTE: Do not place your application dependencies here; they belong
        // in the individual module build.gradle files
    }
}

allprojects {
    repositories {
        jcenter()
    }
}

task clean(type: Delete) {
    delete rootProject.buildDir
}
```

Build.gradle(Module.app)

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 25
    buildToolsVersion "25.0.2"
    defaultConfig {
        applicationId "com.sumit.cab.smartcab"
        minSdkVersion 16
        targetSdkVersion 25
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'),
'proguard-rules.pro'
        }
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {
        exclude group: 'com.android.support', module: 'support-annotations'
    })

    compile 'com.android.support:cardview-v7:23.1.1'
    compile 'com.google.android.gms:play-services-location:8.4.0'
    compile 'com.android.support:appcompat-v7:25.2.0'
    compile 'com.android.support:design:25.1.0'
    compile 'com.android.support.constraint:constraint-layout:1.0.1'
    testCompile 'junit:junit:4.12'
}
```

Admin Panel

signUp.php

```
<?php
   $resJson = ['hasError'=>0];
   if(!isset($_POST['email']) || !isset($_POST['password'])){
       echo 'Field is not set!';
       exit;
   }
   if(empty($_POST['email']) || empty($_POST['password'])){
       echo 'Fields value cannot be empty!';
       exit;
   }
   if(strlen(trim($_POST['email'])) <= 0){
       echo 'Email cannot be white space!';
       exit;
   }
   if(strlen(trim($_POST['password'])) <= 0){
       echo 'Password cannot be white space!';
       exit;
   }

   // Remove all illegal characters from email
   $email = filter_var($_POST['email'], FILTER_SANITIZE_EMAIL);
   //Validate e-mail
```

```
    if(!filter_var($email, FILTER_VALIDATE_EMAIL) == true){
        echo 'Email id is not valid!';
        exit;


    }


    include("../process/config.php");
    $email =
mysqli_real_escape_string($con,trim(filter_var($_POST['email'],
FILTER_SANITIZE_EMAIL)));
    $password =
mysqli_real_escape_string($con,trim($_POST['password']));


    $query = "SELECT `id` FROM `user` WHERE `email`='$email'";
    $run = mysqli_query($con,$query);
    if(mysqli_num_rows($run) == 1){
        mysqli_close($con);
        echo 'User is already exist!';
        exit;
    }
    $md5pass = md5($email.$password);


    $query = "INSERT INTO `user`(`email`,`password`) VALUES
('$email','$md5pass')";
    if(!mysqli_query($con,$query)){
        mysqli_close($con);

        echo 'User not signuped due to some reason!';

        exit;

    }

    mysqli_close($con);

    echo "true";

    exit;

?>
```

login.php

```
<?php
   $resJson = ['hasError'=>0];
   if(!isset($_POST['email']) || !isset($_POST['password'])){
      echo 'Field is not set!';
      exit;
   }
   if(empty($_POST['email']) || empty($_POST['password'])){
      echo 'Fields value cannot be empty!';
      exit;
   }
   if(strlen(trim($_POST['email'])) <= 0){
      echo 'Email cannot be white space!';
      exit;
   }
   if(strlen(trim($_POST['password'])) <= 0){
        echo 'Password cannot be white space!';
      exit;
   }

   // Remove all illegal characters from email
   $email = filter_var($_POST['email'], FILTER_SANITIZE_EMAIL);
   //Validate e-mail
   if(!filter_var($email, FILTER_VALIDATE_EMAIL) == true){

      echo 'Email id is not valid!';

      exit;


   }


   include("../process/config.php");

   $email =
mysqli_real_escape_string($con,trim(filter_var($_POST['email'],
FILTER_SANITIZE_EMAIL)));

   $password =
mysqli_real_escape_string($con,trim($_POST['password']));

   $md5pass = md5($email.$password);


   $query = "SELECT `id` FROM `user` WHERE `email`='$email' AND
`password`='$md5pass'";

   $run = mysqli_query($con,$query);

   if(mysqli_num_rows($run) != 1){

      mysqli_close($con);

      echo 'Incorrect username and password!';

      exit;

   } mysqli_close($con);

   echo 'true';

   exit;

?>
```

getRide.php

```
<?php
   $resJson = ['hasError'=>0];
   if(!isset($_POST['email']) || !isset($_POST['fid'])){
      $resJson['hasError']++;
      $resJson['error'] = 'Field is not set!';
      print_r(json_encode(array("response"=>($resJson))));
      exit;
   }
   if(empty($_POST['email']) || empty($_POST['fid'])){
      $resJson['hasError']++;
      $resJson['error'] = 'Fields value cannot be empty!';
      print_r(json_encode(array("response"=>($resJson))));
      exit;
   }
   if(strlen(trim($_POST['email'])) <= 0){
      $resJson['hasError']++;
      $resJson['error'] = 'Email cannot be white space!';
      print_r(json_encode(array("response"=>($resJson))));
      exit;
   }
   if(strlen(trim($_POST['fid'])) <= 0){
      $resJson['hasError']++;
```

```
        $resJson['error'] = 'Fare ID cannot be white space!';
        print_r(json_encode(array("response"=>($resJson))));
        exit;
    }

    // Remove all illegal characters from email
    $email = filter_var($_POST['email'], FILTER_SANITIZE_EMAIL);
    //Validate e-mail
    if(!filter_var($email, FILTER_VALIDATE_EMAIL) == true){
        $resJson['hasError']++;
        $resJson['error'] = 'Email id is not valid!';
        print_r(json_encode(array("response"=>($resJson))));
        exit;
    }

    include("../process/config.php");
    $fid = mysqli_real_escape_string($con,trim($_POST['fid']));
    $email =
mysqli_real_escape_string($con,trim(filter_var($_POST['email'],
FILTER_SANITIZE_EMAIL)));

    $query = "SELECT `id`, SUBSTRING_INDEX(`email`, '@', 1) AS
`uname` FROM `user` WHERE `email`='$email'";
    $run = mysqli_query($con,$query);
    if(mysqli_num_rows($run) != 1){
        mysqli_close($con);

        $resJson['hasError']++;

        $resJson['error'] = 'User does not exist!';

        print_r(json_encode(array("response"=>($resJson))));

        exit;

    }

    $row = mysqli_fetch_assoc($run);

    $uid = $row['id'];

    $uname = $row['uname'];


    $query = "SELECT vr.*,ROUND(f.`price`*vr.`dist`) AS `eprice`,
cp.`name` as cpname, c.`name` as cname FROM `viewrides` vr, fare f,
company cp, car c WHERE f.`id`=$fid AND f.`cid`=cp.`id` AND
f.`tcid`=c.`id` AND vr.`id`=(SELECT MAX(`id`) FROM `viewrides`
WHERE `uid`=$uid)";

    $run = mysqli_query($con,$query);

    if(mysqli_num_rows($run) != 1){

        mysqli_close($con);

        $resJson['hasError']++;

        $resJson['error'] = 'Ride not booked due to invalid data!';

        print_r(json_encode(array("response"=>($resJson))));

        exit;

    }

    $row = mysqli_fetch_assoc($run);
    $carNo = "MH 04 DE ".rand(100,9999);

    $mbNo = "+919987".rand(000000,999999);

    $min = rand(20,80);


    $query = "INSERT INTO `rides`(`src`, `dst`, `fid`, `eprice`, `date`, `uid`, 
`dist`, `time`, `carNo`, `mbNo`, `calDate`) VALUES 
('".$row['src']."','".$row['dst']."',$fid,".$row['eprice'].",now(),$uid,".$row
['dist'].",".$row['time'].", '$carNo', '$mbNo',now()+INTERVAL ".$min." 
MINUTE)";

    if(!mysqli_query($con,$query)){

        mysqli_close($con);

        $resJson['hasError']++;

        $resJson['error'] = 'Ride not booked due to some reason!';

        print_r(json_encode(array("response"=>($resJson))));

        exit;

    }


    $query = "DELETE FROM `viewrides` WHERE `id`=".$row['id'];

    mysqli_query($con,$query);


    $to   = $email;

    $subject ="Ride Booked";

    $headers  = 'MIME-Version: 1.0' . "\r\n";

    $headers .= 'Content-type: text/html; charset=iso-8859-1' . "\r\n";
```

```
   // Additional headers
   $headers .= 'From: Smart Cab <smartcab22@gmail.com>' . "\r\n";
    $headers .= "CC: smartcab22@gmail.com\r\n";
      //begin of HTML message
   $message = '
   <html>
   <head>
    <title>Ride Booked!</title>
    <link
href="http://fonts.googleapis.com/css?family=Montserrat:400,700"
rel="stylesheet" type="text/css">
    <style>
      button
      {
         border-width:2px;
         border-color:white;
         display:block;
         padding:5px;
         background : #062D53;
         cursor:pointer;
         float:left;
         margin-left:15px;
         color:white;
         text-decoration:none;
```

```
        }
        button:hover
        {
            background: rgba(0,0,0,0.9);
            color:white;
        }
      </style>
    </head>
    <body>
    <div style="width:100%; background-color:#eae5e5; color:black;">
    <div style="background-color:#eae5e5; float:left; width:100%;
padding:10px; float:left;">
      <div style="float:left; width:70%;">
        <div style="float:left"><img
src="https://webstermind.000webhostapp.com/images/favicon.png"
height=30px;></div>
        <div style="float:left; margin-left:10px; color:#339AE5; font-
size:30px;">Smart Cab</div>
      </div>
    </div>
    <div style=" padding:5px; padding-top:10px; width:100%; float:left;
background-color:white; color:black;">
    Dear '.$uname.',
      <p>Thank you for booking ride at Smart Cab. </p><br>
      <div><b>Source : </b>'.$row["src"].'</div>
      <div><b>Destination : </b>'.$row["dst"].'</div>

      <div><b>Company : </b>'.$row["cpname"].'</div>

      <div><b>Car Type : </b>'.$row["cname"].'</div>

      <div><b>Estimated Distance : </b>'.$row["dist"].' km</div>

      <div><b>Estimated Price : </b>&#8377; '.$row["eprice"].' </div>

      <div><b>Estimated time : </b>'.$row["time"]. 'min</div>

        <br>

       <p>If you are experiencing any technical difficulties, please send an
email to <a
href="mailto:smartcab22@gmail.com">smartcab22@gmail.com</a> and
we will get back to you as soon as possible.</p>

    <br><br><br>

    Regards,<br>

    Smart Cab Team<br><br>

    </div>


    <footer>

    <div style=" width:100%; float:left; padding:10px; padding-top:5px;
background-color:#eae5e5;">

      <div style="width:50%; float:left;">

         <span style="font-size:120%;">Contact Us</span>

          <address style="margin-top:10px;">

            <span >Irla, N. R. G Marg, </span><br>

            <span >Vileparle (W), Mumbai: </span><span style=" text-
decoration:none;">400056.</span><br/>
```

```
            <abbr title="Telephone" >Telephone: </abbr> <span style=" text-
decoration:none;">+917977178583</span><br/>

            <abbr title="Email" >Email: </abbr> <a
href="mailto:smartcab22@gmail.com" >smartcab22@gmail.com</a>

          </address>

      </div>

      <div style="float:left; width:50%;">

        <table>

            <tr>

            <td><p style="padding-right:20px; color:black;">Follow
us:</p></td>

            <td style="margin-left:15px;">

            <a href="https://twitter.com/"  style="text-decoration:none;"><img
src="https://webstermind.000webhostapp.com/images/twitter.png"
height=20px; style="margin-top:-2%; padding:10px; padding-
left:0px;"></a>

            </td>

            <td>

            <a href="https://www.facebook.com/"  style="text-
decoration:none;"><img
src="https://webstermind.000webhostapp.com/images/facebook.png"
height=20px; style="margin-top:-2%; padding:10px; padding-
left:0px;"></a>

            </td>

            <td>

            <a href="https://plus.google.com/"  style="text-
decoration:none;"><img
src="https://webstermind.000webhostapp.com/images/google-plus.png"
```

```
height=20px; style="margin-top:-2%; padding:10px; padding-left:0px;"></a>
            </td>
            <td>
            <a href="https://www.pinterest.com/" style="text-decoration:none;"><img src="https://webstermind.000webhostapp.com/images/pinterest.png" height=20px; style="margin-top:-2%; padding:10px; padding-left:0px;"></a>
            </td>
        </table>
      </div>
    </div>
    </footer>
    </div>
    </body>
    </html>
    ';


    if(!mail($to, $subject, $message, $headers)){
        mysqli_close($con);
        $resJson['hasError']++;
        $resJson['error'] = 'Ride booked but mail does not send due to some reason!';
        print_r(json_encode(array("response"=>($resJson))));
        exit;
    }


    mysqli_close($con);
    $resJson['msg'] = "Your Ride has been successfully booked !";
    print_r(json_encode(array("response"=>($resJson))));
    exit;
?>
```

getPrice.php

```
<?php
   $resJson = ['hasError'=>0];
   if(!isset($_POST['origin']) || !isset($_POST['destination']) ||
!isset($_POST['email']) || !isset($_POST['tid'])){
      $resJson['hasError']++;
      $resJson['error'] = 'Field is not set!';
      print_r(json_encode(array("response"=>($resJson))));
      exit;
   }
   if(empty($_POST['origin']) || empty($_POST['destination']) ||
empty($_POST['email']) || empty($_POST['tid'])){
      $resJson['hasError']++;
      $resJson['error'] = 'Fields value cannot be empty!';
      print_r(json_encode(array("response"=>($resJson))));
      exit;
   }
   if(strlen(trim($_POST['origin'])) <= 0){
      $resJson['hasError']++;
      $resJson['error'] = 'Origin cannot be white space!';
      print_r(json_encode(array("response"=>($resJson))));
      exit;
   }
```

```
if(strlen(trim($_POST['destination'])) <= 0){
    $resJson['hasError']++;
    $resJson['error'] = 'Destination cannot be white space!';
    print_r(json_encode(array("response"=>($resJson))));
    exit;
}
if(strlen(trim($_POST['email'])) <= 0){
    $resJson['hasError']++;
    $resJson['error'] = 'Email cannot be white space!';
    print_r(json_encode(array("response"=>($resJson))));
    exit;
}
if(strlen(trim($_POST['tid'])) <= 0){
    $resJson['hasError']++;
    $resJson['error'] = 'Type ID cannot be white space!';
    print_r(json_encode(array("response"=>($resJson))));
    exit;
}

// Remove all illegal characters from email
$email = filter_var($_POST['email'], FILTER_SANITIZE_EMAIL);
//Validate e-mail
if(!filter_var($email, FILTER_VALIDATE_EMAIL) == true){
    $resJson['hasError']++;
```

```
        $resJson['error'] = 'Email id is not valid!';

        print_r(json_encode(array("response"=>($resJson))));

        exit;


    }


    include("../process/config.php");

    $email =
mysqli_real_escape_string($con,trim(filter_var($_POST['email'],
FILTER_SANITIZE_EMAIL)));

    $origin = mysqli_real_escape_string($con,trim($_POST['origin']));

    $destination =
mysqli_real_escape_string($con,trim($_POST['destination']));

    $tid = mysqli_real_escape_string($con,trim($_POST['tid']));

    $key = "AIzaSyAW1n8vtFckqBym8CWvBfPVhsU8KpefZvo";

$parameters="units=imperial&origins=".$origin."&destinations=".$destin
ation."&key=".$key;


    $query = "SELECT `id` FROM `user` WHERE `email`='$email'";

    $run = mysqli_query($con,$query);

    if(mysqli_num_rows($run) != 1){

        mysqli_close($con);

        $resJson['hasError']++;

        $resJson['error'] = 'User does not exist!';
        print_r(json_encode(array("response"=>($resJson))));
        exit;
    }
    $row = mysqli_fetch_assoc($run);
    $id = $row['id'];

    $getData =
json_decode(file_get_contents('https://maps.googleapis.com/maps/api/distancematrix/json?'.$parameters, false));
    if($getData->status == "OK" && $getData->rows[0]->elements[0]->status == "OK"){

        $query = "SELECT `id`,`price` FROM `fare` WHERE `tcid`='$tid' ORDER BY `cid` ASC";
        $run = mysqli_query($con,$query);
        if(!mysqli_num_rows($run)){
            mysqli_close($con);
            $resJson['hasError']++;
            $resJson['error'] = 'Car Type dooes not exist!';
            print_r(json_encode(array("response"=>($resJson))));
            exit;
        }
        $data = array();
        while($row = mysqli_fetch_assoc($run))
            array_push($data,$row);
        $distance = round((($getData->rows[0]->elements[0]->distance->value)/1000),1);

        $origin = $getData->destination_addresses[0];

        $destination = $getData->origin_addresses[0];

        $duration = round((($getData->rows[0]->elements[0]->duration->value)/60),0);


        $query = "INSERT INTO `viewrides`(`uid`, `src`, `dst`, `dist`,`time`) VALUES ($id,'$origin','$destination',$distance,$duration)";

        if(!mysqli_query($con,$query)){

            mysqli_close($con);

            $resJson['hasError']++;

            $resJson['error'] = 'Rides not viewed due to some reason!';

            print_r(json_encode(array("response"=>($resJson))));

            exit;

        }


        $resJson["msg"]= array(

            "origin"       =>   $origin,

            "destination"   =>   $destination,

            "distance"      =>   $distance,

            "duration"      =>   $duration,

            "ola"=>     array(

                "id"    =>   $data[0]['id'],

                "rate"  =>   round($data[0]['price']*$distance)
            ),
            "uber"=>    array(
                "id"    =>   $data[1]['id'],
                "rate"  =>   round($data[1]['price']*$distance)
            ),
            "tabcab"=>  array(
                "id"    =>   $data[2]['id'],
                "rate"  =>   round($data[2]['price']*$distance)
            )
        );

        mysqli_close($con);
        print_r(json_encode(array("response"=>($resJson))));
        exit;

    }
    else{
        $resJson['hasError']++;
        $resJson['error'] = 'Requested data is invalid! :(';
        print_r(json_encode(array("response"=>($resJson))));
        exit;
    }
?>
```

sendMail.php

```
<?php
   $resJson = ['hasError'=>0];
   if(!isset($_POST['email']) || !isset($_POST['name'])  ||
!isset($_POST['subject'])  || !isset($_POST['message'])){
       echo  'Field is not set!';
       exit;
   }
   if(empty($_POST['email']) || empty($_POST['name']) ||
empty($_POST['subject']) || empty($_POST['message'])){
       echo 'Fields value cannot be empty!';
       exit;
   }
   if(strlen(trim($_POST['email'])) <= 0){
       echo 'Email cannot be white space!';
       exit;
   }
   if(strlen(trim($_POST['name'])) <= 0){
       echo 'Name cannot be white space!';
       exit;
   }
   if(strlen(trim($_POST['subject'])) <= 0){
       echo 'Subject cannot be white space!';
```

```
        exit;
    }
    if(strlen(trim($_POST['message'])) <= 0){
        echo 'Message cannot be white space!';
        exit;
    }

    // Remove all illegal characters from email
    $email = filter_var($_POST['email'], FILTER_SANITIZE_EMAIL);
    //Validate e-mail
    if(!filter_var($email, FILTER_VALIDATE_EMAIL) == true){
        echo 'Email id is not valid!';
        exit;

    }

    include("../process/config.php");
    $name = mysqli_real_escape_string($con,trim($_POST['name']));
    $sub = mysqli_real_escape_string($con,trim($_POST['subject']));
    $msg = mysqli_real_escape_string($con,trim($_POST['message']));
    $email =
mysqli_real_escape_string($con,trim(filter_var($_POST['email'],
FILTER_SANITIZE_EMAIL)));
```

```
  $subject = "Contact From: ".$name;
  $headers  = 'MIME-Version: 1.0' . "\r\n";


    $headers .= 'From: '. $name . '<' . $email . '>' . "\r\n";
  $headers .= 'Reply-To: ' . $email . "\r\n";
    $headers .= "CC: ".$email."\r\n";


  $headers .= 'MIME-Version: 1.0' . "\r\n";
  $headers .= 'Content-type: text/html; charset=iso-8859-1' . "\r\n";


      //begin of HTML message
  $message = '
  <html>
  <head>
    <title>Contact form</title>
    <link
href="http://fonts.googleapis.com/css?family=Montserrat:400,700"
rel="stylesheet" type="text/css">
    <style>
      button
      {
          border-width:2px;
          border-color:white;
          display:block;
```

```
            padding:5px;

            background : #062D53;

            cursor:pointer;

            float:left;

            margin-left:15px;

            color:white;

            text-decoration:none;

        }

        button:hover

        {

            background: rgba(0,0,0,0.9);

            color:white;

        }

      </style>

    </head>

    <body>

    <div style="width:100%; background-color:#eae5e5; color:black;">

    <div style="background-color:#eae5e5; float:left; width:100%;
padding:10px; float:left;">

      <div style="float:left; width:70%;">

        <div style="float:left"><img
src="https://webstermind.000webhostapp.com/images/favicon.png"
height=30px;></div>

        <div style="float:left; margin-left:10px; color:#339AE5; font-
size:30px;">Smart Cab</div>
```

```
      </div>
    </div>
    <div style=" padding:5px; padding-top:10px; width:100%; float:left;
background-color:white; color:black;">
    <p>Dear Smart Cab team</p>
      <p>Contact From : </p>
      <div><b>Name : </b>'.$name.'</div>
      <div><b>Email : </b>'.$email.'</div>
      <div><b>Subject : </b>'.$sub.'</div>
      <div><b>Message : </b>'.$msg.'</div>
        <br><br>

    <footer>
    <div style=" width:100%; float:left; padding:10px; padding-top:5px;
background-color:#eae5e5;">
      <div style="width:50%; float:left;">
         <span style="font-size:120%;">Contact Us</span>
          <address style="margin-top:10px;">
            <span >Irla, N. R. G Marg, </span><br>
            <span >Vileparle (W), Mumbai: </span><span style=" text-
decoration:none;">400056.</span><br/>
            <abbr title="Telephone" >Telephone: </abbr> <span style=" text-
decoration:none;">+917977178583</span><br/>
            <abbr title="Email" >Email: </abbr> <a
href="mailto:smartcab22@gmail.com" >smartcab22@gmail.com</a>
```

```
        </address>
      </div>
      <div style="float:left; width:50%;">
        <table>
          <tr>
          <td><p style="padding-right:20px; color:black;">Follow us:</p></td>
          <td style="margin-left:15px;">
          <a href="https://twitter.com/"  style="text-decoration:none;"><img src="https://webstermind.000webhostapp.com/images/twitter.png" height=20px; style="margin-top:-2%; padding:10px; padding-left:0px;"></a>
          </td>
          <td>
          <a href="https://www.facebook.com/"  style="text-decoration:none;"><img src="https://webstermind.000webhostapp.com/images/facebook.png" height=20px; style="margin-top:-2%; padding:10px; padding-left:0px;"></a>
          </td>
          <td>
          <a href="https://plus.google.com/"  style="text-decoration:none;"><img src="https://webstermind.000webhostapp.com/images/google-plus.png" height=20px; style="margin-top:-2%; padding:10px; padding-left:0px;"></a>
          </td>
          <td>
      <a href="https://www.pinterest.com/" style="text-
decoration:none;"><img
src="https://webstermind.000webhostapp.com/images/pinterest.png"
height=20px; style="margin-top:-2%; padding:10px; padding-
left:0px;"></a>
      </td>
    </table>
   </div>
  </div>
  </footer>
  </div>
  </body>
  </html>
  ';

  if(!mail('smartcab22@gmail.com', $subject, $message, $headers)){
    echo "email not send due to some reason!";
  }

  echo "true"
?>
```

getHistory.php

```
<?php
   $resJson = ['hasError'=>0];
   if(!isset($_POST['email'])){
      $resJson['hasError']++;
      $resJson['error'] = 'Field is not set!';
      print_r(json_encode(array("response"=>($resJson))));
      exit;
   }
   if(empty($_POST['email'])){
      $resJson['hasError']++;
      $resJson['error'] = 'Fields value cannot be empty!';
      print_r(json_encode(array("response"=>($resJson))));
      exit;
   }
   if(strlen(trim($_POST['email'])) <= 0){
      $resJson['hasError']++;
      $resJson['error'] = 'Email cannot be white space!';
      print_r(json_encode(array("response"=>($resJson))));
      exit;
   }

   // Remove all illegal characters from email
```

```
$email = filter_var($_POST['email'], FILTER_SANITIZE_EMAIL);

//Validate e-mail

if(!filter_var($email, FILTER_VALIDATE_EMAIL) == true){

    $resJson['hasError']++;

    $resJson['error'] = 'Email id is not valid!';

    print_r(json_encode(array("response"=>($resJson))));

    exit;


}


include("../process/config.php");

$email =
mysqli_real_escape_string($con,trim(filter_var($_POST['email'],
FILTER_SANITIZE_EMAIL)));


$query = "SELECT `id` FROM `user` WHERE `email`='$email'";

$run = mysqli_query($con,$query);

if(mysqli_num_rows($run) != 1){

    mysqli_close($con);

    $resJson['hasError']++;

    $resJson['error'] = 'User does not exist!';

    print_r(json_encode(array("response"=>($resJson))));

    exit;

}
```

```
   $row = mysqli_fetch_assoc($run);
   $uid = $row['id'];


   $query = "SELECT r.`id` AS `id`, r.`src` AS `src`, r.`dst` AS `dst`,
cp.`name` as `cmpName`, c.`name` as `carName`, r.`dist` AS `dist`,
r.`carNo` AS `carNo`, r.`mbNo` AS `mbNo`, r.`calDate` AS `calDate`
FROM `rides` r,`fare` f, `company` cp, `car` c WHERE f.`id`=r.`fid` AND
f.`cid`=cp.`id` AND f.`tcid`=c.`id` AND r.`uid`=$uid";
   $run = mysqli_query($con,$query);
   if(mysqli_num_rows($run) <= 0){
      mysqli_close($con);
      $resJson['hasError']++;
      $resJson['error'] = 'No Data to show!';
      print_r(json_encode(array("response"=>($resJson))));
      exit;
   }
   $resJson['msg']=array();
   $i=0;
   while($row = mysqli_fetch_assoc($run)):
        date_default_timezone_set('Asia/Kolkata');
        if($row['calDate']>=date("Y-m-d H:i:s"))
                $row['cancel']=1;
        else
                $row['cancel']=0;
        array_push($resJson['msg'], $row);
```

```
    endwhile;


    mysqli_close($con);
    print_r(json_encode(array("response"=>($resJson))));
    exit;
?>
```

cancelRide.php

```
<?php
   $resJson = ['hasError'=>0];
   if(!isset($_POST['email']) || !isset($_POST['fid'])){
      $resJson['hasError']++;
      $resJson['error'] = 'Field is not set!';
      print_r(json_encode(array("response"=>($resJson))));
      exit;
   }
   if(empty($_POST['email']) || empty($_POST['fid'])){
      $resJson['hasError']++;
      $resJson['error'] = 'Fields value cannot be empty!';
      print_r(json_encode(array("response"=>($resJson))));
      exit;
   }
   if(strlen(trim($_POST['email'])) <= 0){
      $resJson['hasError']++;
      $resJson['error'] = 'Email cannot be white space!';
      print_r(json_encode(array("response"=>($resJson))));
      exit;
   }
   if(strlen(trim($_POST['fid'])) <= 0){
      $resJson['hasError']++;
```

```
        $resJson['error'] = 'Fare ID cannot be white space!';
        print_r(json_encode(array("response"=>($resJson))));
        exit;
    }


    // Remove all illegal characters from email
    $email = filter_var($_POST['email'], FILTER_SANITIZE_EMAIL);
    //Validate e-mail
    if(!filter_var($email, FILTER_VALIDATE_EMAIL) == true){
        $resJson['hasError']++;
        $resJson['error'] = 'Email id is not valid!';
        print_r(json_encode(array("response"=>($resJson))));
        exit;
    }


    include("../process/config.php");
    $fid = mysqli_real_escape_string($con,trim($_POST['fid']));
    $email =
mysqli_real_escape_string($con,trim(filter_var($_POST['email'],
FILTER_SANITIZE_EMAIL)));


    $query = "SELECT `id` FROM `user` WHERE `email`='$email'";
    $run = mysqli_query($con,$query);
    if(mysqli_num_rows($run) != 1){
        mysqli_close($con);

        $resJson['hasError']++;

        $resJson['error'] = 'User does not exist!';

        print_r(json_encode(array("response"=>($resJson))));

        exit;

    }

    $row = mysqli_fetch_assoc($run);

    $uid = $row['id'];


    $query = "DELETE FROM `rides` WHERE `id`=$fid AND ` uid`=$uid
AND `calDate`>=NOW()";

    if(!mysqli_query($con,$query)){

        mysqli_close($con);

        $resJson['hasError']++;

        $resJson['error'] = 'Sorry! You are late.';

        print_r(json_encode(array("response"=>($resJson))));

        exit;

    }


    mysqli_close($con);

    $resJson['msg'] = "Your Ride has been successfully canceled!";

    print_r(json_encode(array("response"=>($resJson))));

    exit;

?>
```

Home.php

```
<?php
    session_start();
    //session validation
    require("process/BaseUrl.php");
    $url = url();
    $ssn_id = session_id();
    $ssn=md5("smartcab");
    if(isset($_SESSION[$ssn]) && isset($_SESSION['unique_id']) && isset($_SESSION['id']))
    {
        //retrive user email to validate session
        include("process/config.php");
        $id = mysqli_real_Escape_string($con,$_SESSION['id']);
        $querySsn = "select * from `member` where `id`=$id";
        $querySsn_run = mysqli_query($con,$querySsn);
        $querySsn_row = mysqli_fetch_assoc($querySsn_run);
        mysqli_close($con);

        $email = $querySsn_row['email'];
        $email_valid = md5($email);
        if($_SESSION['unique_id']==$ssn_id && $_SESSION[$ssn]==$email_valid)
        {
?>
<!DOCTYPE html>
<html lang="en">

<head>

    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <meta name="description" content="">
    <meta name="author" content="Techsperia">
    <link rel="shortcut icon" type="image/png" href="images/favicon.png"
/>

    <title>Smart cab | home</title>

    <!-- Bootstrap Core CSS -->
    <link href="bower_components/bootstrap/dist/css/bootstrap.min.css"
rel="stylesheet">

    <!-- MetisMenu CSS -->
    <link href="bower_components/metisMenu/dist/metisMenu.min.css"
rel="stylesheet">
```

```
    <!-- Timeline CSS -->
    <link href="css/timeline.css" rel="stylesheet">

    <!-- Custom CSS -->
    <link href="css/sb-admin-2.css" rel="stylesheet">

    <!-- Morris Charts CSS -->
    <link href="bower_components/morrisjs/morris.css" rel="stylesheet">

    <!-- Custom Fonts -->
    <link href="bower_components/font-awesome/css/font-awesome.min.css" rel="stylesheet" type="text/css">
</head>

<body>

    <div id="wrapper">
        <!-- Navigation -->
        <?php
            include("nav.php");
        ?>
        <!-- /.navigation -->
        <div id="page-wrapper">
```

```
<div class="row">
    <div class="col-lg-12" style="margin-top:-30px">
        <h1 class="page-header">Dashboard</h1>
    </div>
    <!-- /.col-lg-12 -->
</div>
<!-- /.row -->
<div class="row">
    <div class="col-lg-3 col-md-6">
        <div class="panel panel-primary">
            <div class="panel-heading">
                <div class="row">
                    <div class="col-xs-3">
                        <i class="fa fa-taxi fa-5x"></i>
                    </div>
                    <div class="col-xs-9 text-right">
                        <div class="medium">
                        <?php
                            include("process/config.php");
                                                                $fare = "SELECT
SUM(`eprice`) AS `price` FROM `rides` where `fid` IN (SELECT `id`
FROM `fare` WHERE `cid`=2)";
                                                                $run =
mysqli_query($con,$fare);
                            $row = mysqli_fetch_assoc($run);
```

```
                                                    echo "<h4>&#8377; ".round($row['price'])."</h4>";
                                    ?>
                                        </div>
                                    <div>Uber</div>
                                </div>
                            </div>
                        </div>
                        <a href="uber-rides.php">
                            <div class="panel-footer">
                                <span class="pull-left">View Rides</span>
                                <span class="pull-right"><i class="fa fa-arrow-circle-right"></i></span>
                                <div class="clearfix"></div>
                            </div>
                        </a>
                    </div>
                </div>
                <div class="col-lg-3 col-md-6">
                    <div class="panel panel-green">
                        <div class="panel-heading">
                            <div class="row">
                                <div class="col-xs-3">
                                    <i class="fa fa-taxi fa-5x"></i>
                                </div>
                                <div class="col-xs-9 text-right">
                                    <div class="huge">
                                    <?php
                                        $fare = "SELECT SUM(`eprice`) AS `price` FROM
`rides` where `fid` IN (SELECT `id` FROM `fare` WHERE `cid`=1)";
                                        $run = mysqli_query($con,$fare);
                                        $row = mysqli_fetch_assoc($run);
                                        echo "<h4>&#8377;
".round($row['price'])."</h4>";
                                    ?>
                                    </div>
                                    <div>Ola cab</div>
                                </div>
                            </div>
                        </div>
                        <a href="ola-rides.php">
                            <div class="panel-footer">
                                <span class="pull-left">View Rides</span>
                                <span class="pull-right"><i class="fa fa-arrow-circle-
right"></i></span>
                                <div class="clearfix"></div>
                            </div>
                        </a>
                    </div>
```

```
</div>
<div class="col-lg-3 col-md-6">
    <div class="panel panel-yellow">
        <div class="panel-heading">
            <div class="row">
                <div class="col-xs-3">
                    <i class="fa fa-taxi fa-5x"></i>
                </div>
                <div class="col-xs-9 text-right">
                    <div class="huge">
                    <?php
                        $fare = "SELECT SUM(`eprice`) AS `price` FROM `rides` where `fid` IN (SELECT `id` FROM `fare` WHERE `cid`=3)";
                        $run = mysqli_query($con,$fare);
                        $row = mysqli_fetch_assoc($run);
                        echo "<h4>&#8377; ".round($row['price'])."</h4>";
                    ?>
                    </div>
                    <div>TABcab</div>
                </div>
            </div>
        </div>
        <a href="tabcab-rides.php">
```

```
            <div class="panel-footer">
                <span class="pull-left">View Rides</span>
                <span class="pull-right"><i class="fa fa-arrow-circle-right"></i></span>
                <div class="clearfix"></div>
            </div>
        </a>
    </div>
</div>
<div class="col-lg-3 col-md-6">
    <div class="panel panel-red">
        <div class="panel-heading">
            <div class="row">
                <div class="col-xs-3">
                    <i class="fa fa-taxi fa-5x"></i>
                </div>
                <div class="col-xs-9 text-right">
                    <div class="huge">
                    <?php
                        $fare = "SELECT SUM(`eprice`) AS `price` FROM `rides`";
                        $run = mysqli_query($con,$fare);
                        $row = mysqli_fetch_assoc($run);
                        echo "<h4>&#8377; ".round($row['price'])."</h4>";
```

```
                              ?>
                              </div>
                              <div>All cab</div>
                            </div>
                          </div>
                        </div>
                        <a href="all-rides.php">
                          <div class="panel-footer">
                            <span class="pull-left">View Rides</span>
                            <span class="pull-right"><i class="fa fa-arrow-circle-
right"></i></span>
                            <div class="clearfix"></div>
                          </div>
                        </a>
                      </div>
                    </div>
                  </div> <!-- /#row -->
                </div><!-- /#page-wrapper -->
              </div><!-- /#wrapper -->

              <!-- jQuery -->
              <script src="bower_components/jquery/dist/jquery.min.js"></script>

              <!-- Bootstrap Core JavaScript -->
```

```
    <script
src="bower_components/bootstrap/dist/js/bootstrap.min.js"></script>
 <!-- Metis Menu Plugin JavaScript -->
    <script
src="bower_components/metisMenu/dist/metisMenu.min.js"></script>


    <!-- Custom Theme JavaScript -->
    <script src="js/sb-admin-2.js"></script>
</body>
</html>
<?php
        }
        else
        {
            session_unset();
            session_destroy();
            echo "<script>window.location.href='$url'</script>";
        }
    }
    else
    {
        echo "<script>window.location.href='$url'</script>";
    }
?>
```

Index.php

```
<?php
   session_start();
   //session validation
   require("process/BaseUrl.php");
   $url = url();
   $ssn_id = session_id();
   $ssn=md5("smartcab");
   if(isset($_SESSION[$ssn]) && isset($_SESSION['unique_id']) &&
isset($_SESSION['id']))
   {
      //retrive user email to validate session
      include("process/config.php");
      $id = mysqli_real_escape_string($con,$_SESSION['id']);
      $querySsn = "select * from `member` where `id`=$id";
      $querySsn_run = mysqli_query($con,$querySsn);
      $querySsn_row = mysqli_fetch_assoc($querySsn_run);
      mysqli_close($con);


      $email = $querySsn_row['email'];
      $email_valid = md5($email);
      if($_SESSION['unique_id']==$ssn_id &&
$_SESSION[$ssn]==$email_valid)
```

```
        {
            echo "<script>window.location.href='$url/home.php'</script>";
        }
        else
        {
            session_unset();
            session_destroy();
        }
    }
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <meta name="description" content="">
    <meta name="author" content="Smartcab">
    <link rel="shortcut icon"  href="images/smartcab.svg" />
    <title>Smart Cab | Login</title>
    <!-- Bootstrap Core CSS -->
    <link href="bower_components/bootstrap/dist/css/bootstrap.min.css"
rel="stylesheet">
```

```
<!-- MetisMenu CSS -->
<link href="bower_components/metisMenu/dist/metisMenu.min.css" rel="stylesheet">
<!-- Custom CSS -->
<link href="css/sb-admin-2.css" rel="stylesheet">
<!-- Custom Fonts -->
<link href="bower_components/font-awesome/css/font-awesome.min.css" rel="stylesheet" type="text/css">
<style>
.error
{
    width:100%;
    margin-bottom:10px;
    font-size:13px;
    padding-left:10px !important;
    background-color:rgb(213, 50, 50);
    color:#ffffff !important;
}
.error_data
{
    padding : 3px;
}
a
{
        text-decoration:none !important;
        color:rgba(0,0,0,0.7) !important;
    }
    a:focus
    {
        outline: none !important;
        outline-offset: none !important;
    }
    .neptune_header
    {
        text-align:center;
        font-size: 500%;
        margin-top: 5%;
        margin-bottom: -3%;
        color:rgba(0,0,0,0.7);
    }
    </style>
</head>
<body background="images/bg.jpg" style="background-size:100%;" >
    <div class="container">
        <div class="row">
         <div class="neptune_header"><img src="images/smartcab.svg"
height="70px" style="margin-bottom: 20px;"/><a style="margin-
left:5px;" href="<?=$url?>">Smart Cab</a></div>
```

```
</div>
<div class="row">
    <div class="col-md-4 col-md-offset-4">
        <div class="login-panel panel panel-default" style="background-color:rgba(0,0,0,0.1); border-color:rgba(0,0,0,0.3);">
            <div class="panel-heading" style="background-color:rgba(0,0,0,0.1); color:rgba(0,0,0,0.9); border-color:rgba(0,0,0,0.3);">
                <h3 class="panel-title">Please Sign In</h3>
            </div>
            <div class="panel-body">
                <form id="form_login">
                    <fieldset>
                        <div class="form-group">
                            <input class="form-control" placeholder="User Name" name="uname" type="email" autofocus required>
                        </div>
                        <div class="form-group">
                            <input class="form-control" placeholder="password" name="password" type="password" required>
                        </div>
                        <!-- Change this to a button or input when using this as a form -->
                        <div class="error_data"></div>
                    </fieldset>
                    <input type="submit" value="Login" id="submit_login" class="btn btn-lg btn-success btn-block">
                    </form>
                </div>
            </div>
        </div>
    </div>
</div>
<!-- jQuery -->
<script src="bower_components/jquery/dist/jquery.min.js"></script>
<!-- Bootstrap Core JavaScript -->
<script
src="bower_components/bootstrap/dist/js/bootstrap.min.js"></script>
<!-- Metis Menu Plugin JavaScript -->
<script
src="bower_components/metisMenu/dist/metisMenu.min.js"></script>
<!-- Custom Theme JavaScript -->
<script src="js/sb-admin-2.js"></script>
<script src="js/login.js"></script>
</body>
</html>
```

Nav.php

```
<?php
    //session validation
    $ssn=md5("smartcab");
    if(isset($_SESSION[$ssn]) && isset($_SESSION['unique_id']) &&
isset($_SESSION['id']))
    {
        $ssn_id = session_id();

        //retrive user email to validate session
        include("process/config.php");
        $id = mysqli_real_Escape_string($con,trim($_SESSION['id']));
        $queryNavSsn = "select * from `member` where `id`=$id";
        $queryNavSsn_run = mysqli_query($con,$queryNavSsn);
        $queryNavSsn_row = mysqli_fetch_assoc($queryNavSsn_run);
        mysqli_close($con);

        $email = $queryNavSsn_row['email'];
        $email = md5($email);
        if($_SESSION['unique_id']==$ssn_id && $_SESSION[$ssn]==$email)
        {
?>
<!-- Navigation -->
```

```
        <nav class="navbar navbar-default navbar-static-top"
role="navigation" style="margin-bottom: 0">

            <div class="navbar-header">

                <button type="button" class="navbar-toggle" data-
toggle="collapse" data-target=".navbar-collapse">

                    <span class="sr-only">Toggle navigation</span>

                    <span class="icon-bar"></span>

                    <span class="icon-bar"></span>

                    <span class="icon-bar"></span>

                </button>

                <a class="navbar-brand" href="<?=$url;?>">Smart Cab</a>

            </div>

            <ul class="nav navbar-top-links navbar-right">



                <!-- /.dropdown -->

                <?=$queryNavSsn_row['name'];?>

                <li class="dropdown">

                    <a class="dropdown-toggle" data-toggle="dropdown"
href="javascript:void(0)">

                        <i class="fa fa-user fa-fw"></i>  <i class="fa fa-caret-
down"></i>

                    </a>

                    <ul class="dropdown-menu dropdown-user">

                        <li><a href="#"><i class="fa fa-user fa-fw"></i> User
Profile</a>
```

```
            </li>
            <li><a href="#"><i class="fa fa-gear fa-fw"></i> Settings</a>
            </li>
            <li class="divider"></li>
            <li><a href="<?=$url;?>/logout.php"><i class="fa fa-sign-out fa-fw"></i> Logout</a>
            </li>
          </ul>
          <!-- /.dropdown-user -->
        </li>
      </ul>
      <!-- /.navbar-top-links -->
      <div class="navbar-default sidebar" role="navigation">
        <div class="sidebar-nav navbar-collapse">
          <ul class="nav" id="side-menu">
            <li class="sidebar-search" style="padding-top:3px;padding-bottom:3px;">
              <div class="input-group custom-search-form">
                <form method="get" action="machine.php">
                <input type="text" class="form-control" style="width:80%;" name="id" placeholder="Search...">
                <span class="input-group-btn">
                     <button class="btn btn-default" type="submit">
                       <i class="fa fa-search"></i>
```

```
                  </button>
        </span>
        </form>
    </div><!-- /input-group -->
</li>
<li>
    <a href="javascript:void(0)"><i class="fa fa-edit fa-fw"></i> Uber <span class="fa arrow"></span></a>
     <ul class="nav nav-second-level">
        <li>
            <a href="uber-rate.php"> <i class="fa fa-edit fa-fw"></i> Update Rate</a>
        </li>
        <li>
            <a href="uber-rides.php"> <i class="fa fa-table fa-fw"></i> View Rides</a>
        </li>
    </ul>
</li>
<li>
    <a href="javascript:void(0)"><i class="fa fa-edit fa-fw"></i> Ola cab<span class="fa arrow"></span></a>
    <ul class="nav nav-second-level">
        <li>
```

```
            <a href="ola-rate.php"> <i class="fa fa-edit fa-fw"></i> Update Rate</a>
        </li>
        <li>
            <a href="ola-rides.php"> <i class="fa fa-table fa-fw"></i> View Rides</a>
        </li>
    </ul>
</li>
<li>
    <a href="javascript:void(0)"><i class="fa fa-edit fa-fw"></i> TABcab <span class="fa arrow"></span></a>
    <ul class="nav nav-second-level">
        <li>
            <a href="tabcab-rate.php"> <i class="fa fa-edit fa-fw"></i> Update Rate</a>
        </li>
        <li>
            <a href="tabcab-rides.php"> <i class="fa fa-table fa-fw"></i> View Rides</a>
        </li>
    </ul>
</li>
<li>
    <a href="javascript:void(0)"><i class="fa fa-edit fa-fw"></i> Users <span class="fa arrow"></span></a>
```

```
                        <ul class="nav nav-second-level">
                            <li>
                                <a href="all-users.php"> <i class="fa fa-edit fa-
fw"></i> View all</a>
                            </li>
                        </ul>
                    </ul>
                </div>
                <!-- /.sidebar-collapse -->
            </div>
            <!-- /.navbar-static-side -->
        </nav> <!-- /.navigation -->
<?php
        }
        else
        {
            session_unset();
            session_destroy();
            echo "<script>window.location.href='$url'</script>";
    else
    {
        echo "<script>window.location.href='$url'</script>";
    }
?>
```

getRate.php

```
<?php
    session_start();
    //session validation
    require("BaseUrl.php");
    $url = url();
    $ssn_id = session_id();
    $ssn=md5("smartcab");
    if(isset($_SESSION[$ssn]) && isset($_SESSION['unique_id']) &&
isset($_SESSION['id']))
    {
        //retrive user email to validate session
        include("config.php");
        $id = mysqli_real_Escape_string($con,$_SESSION['id']);
        $querySsn = "select * from `member` where `id`=$id";
        $querySsn_run = mysqli_query($con,$querySsn);
        $querySsn_row = mysqli_fetch_assoc($querySsn_run);
        mysqli_close($con);


        $email = $querySsn_row['email'];
        $email_valid = md5($email);
        if($_SESSION['unique_id']==$ssn_id &&
$_SESSION[$ssn]==$email_valid)
```

```
{
   $resJson = ['hasError'=>0];
  if(!isset($_POST['id']) || !isset($_POST['car'])){
      $resJson['hasError']++;
      $resJson['msg'] = 'Field is not set!';
      print_r(json_encode($resJson));
      exit;
   }
   if(empty($_POST['id']) || empty($_POST['car'])){
      $resJson['hasError']++;
      $resJson['msg'] = 'Field value cannot be empty!';
      print_r(json_encode($resJson));
      exit;
   }
   if(strlen(trim($_POST['id'])) <= 0){
      $resJson['hasError']++;
      $resJson['msg'] = 'Company ID cannot be white space!';
      print_r(json_encode($resJson));
      exit;
   }
   if(strlen(trim($_POST['car'])) <= 0){
      $resJson['hasError']++;
      $resJson['msg'] = 'Types of car must be selected!';
      print_r(json_encode($resJson));
```

```
            exit;
         }

        include("config.php");
        $id=mysqli_real_escape_string($con,$_POST['id']);
        $car=mysqli_real_escape_string($con,$_POST['car']);

        $query = "SELECT *  FROM `fare` WHERE `cid`=$id AND
`tcid`=$car";
         $run = mysqli_query($con,$query);
         if(mysqli_num_rows($run) != 1){
            $resJson['hasError']++;
            $resJson['msg'] = 'Fare does not exist!';
            print_r(json_encode($resJson));
            mysqli_close($con);
            exit;
         }

         $row = mysqli_fetch_assoc($run);
         mysqli_close($con);
         $resJson['price'] = $row['price'];
         $resJson['date'] = $row['date'];
         print_r(json_encode($resJson));
         exit;
        }
        else
        {
            session_unset();
            session_destroy();
            echo "<script>window.location.href='$url'</script>";
        }
    }
    else
    {
        echo "<script>window.location.href='$url'</script>";
    }
?>
```

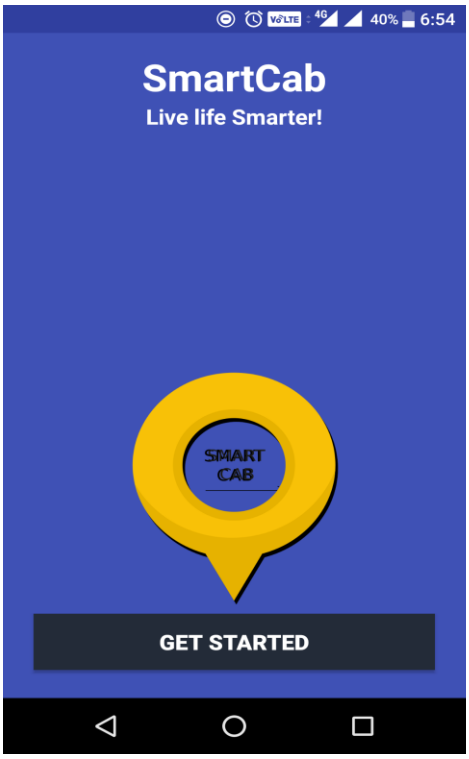
updateRate.php

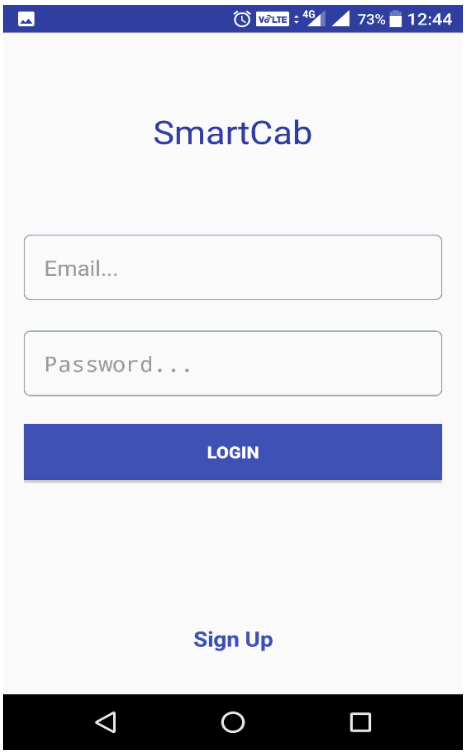
```
<?php
    session_start();
    //session validation
    require("BaseUrl.php");
    $url = url();
    $ssn_id = session_id();
    $ssn=md5("smartcab");
    if(isset($_SESSION[$ssn]) && isset($_SESSION['unique_id']) &&
isset($_SESSION['id']))
    {
        //retrive user email to validate session
        include("config.php");
        $id = mysqli_real_Escape_string($con,$_SESSION['id']);
        $querySsn = "select * from `member` where `id`=$id";
        $querySsn_run = mysqli_query($con,$querySsn);
        $querySsn_row = mysqli_fetch_assoc($querySsn_run);
        mysqli_close($con);

        $email = $querySsn_row['email'];
        $email_valid = md5($email);
        if($_SESSION['unique_id']==$ssn_id &&
$_SESSION[$ssn]==$email_valid)
```

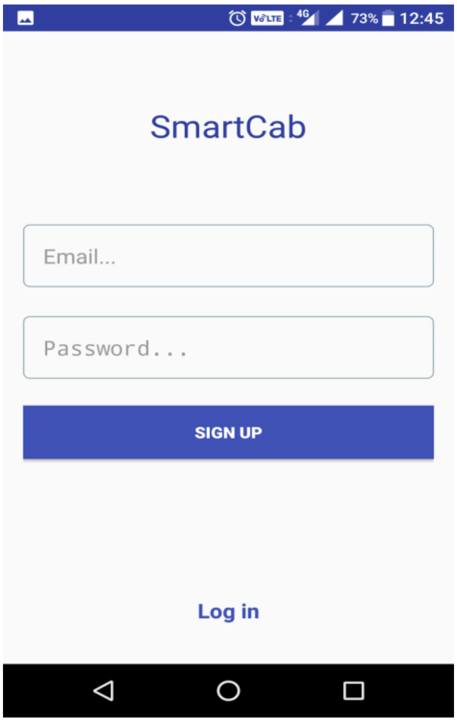
```
        {
           $resJson = ['hasError'=>0];
          if(!isset($_POST['id']) || !isset($_POST['price']) ||
!isset($_POST['car'])){
              $resJson['hasError']++;
              $resJson['msg'] = 'Field is not set!';
              print_r(json_encode($resJson));
              exit;
           }
           if(empty($_POST['id']) || empty($_POST['price'])  ||
empty($_POST['car'])){
              $resJson['hasError']++;
              $resJson['msg'] = 'Field value cannot be empty!';
              print_r(json_encode($resJson));
              exit;
           }
           if(strlen(trim($_POST['id'])) <= 0){
              $resJson['hasError']++;
              $resJson['msg'] = 'ID cannot be white space!';
              print_r(json_encode($resJson));
              exit;
           }
           if(strlen(trim($_POST['price'])) <= 0){
              $resJson['hasError']++;
        $resJson['msg'] = 'Price cannot be white space!';
        print_r(json_encode($resJson));
        exit;
      }
      if(strlen(trim($_POST['car'])) <= 0){
        $resJson['hasError']++;
        $resJson['msg'] = 'Type of cr must be selected!';
        print_r(json_encode($resJson));
        exit;
      }

     include("config.php");
     $id = mysqli_real_escape_string($con,$_POST['id']);
     $price = mysqli_real_escape_string($con,$_POST['price']);
      $car = mysqli_real_escape_string($con,$_POST['car']);

     $checkQuery = "SELECT `id`  FROM `fare` WHERE `cid`=$id
AND `tcid`=$car";
      $checkQuery_run = mysqli_query($con,$checkQuery);
      if(mysqli_num_rows($checkQuery_run) != 1){
        $resJson['hasError']++;
        $resJson['msg'] = 'Fare does not exist!';
        print_r(json_encode($resJson));
        mysqli_close($con);
        exit;
    }

    $updateQuery = "UPDATE `fare` SET
`price`=ROUND($price,1),`date`=now() WHERE `cid`=$id AND
`tcid`=$car";
    if(!mysqli_query($con,$updateQuery)){
        $resJson['hasError']++;
        $resJson['msg'] = mysqli_msg($con);
        print_r(json_encode($resJson));
        mysqli_close($con);
        exit;
    }

    $query = "SELECT `date`,`price` FROM `fare` WHERE `cid`=$id
AND `tcid`=$car";
    $run = mysqli_query($con,$query);
    $row = mysqli_fetch_assoc($run);
    $resJson['price'] = $row['price'];
    $resJson['date'] = $row['date'];
    mysqli_close($con);
    $resJson['msg'] = 'Fare Rate successfully updated!';

    print_r(json_encode($resJson));
    exit;
```

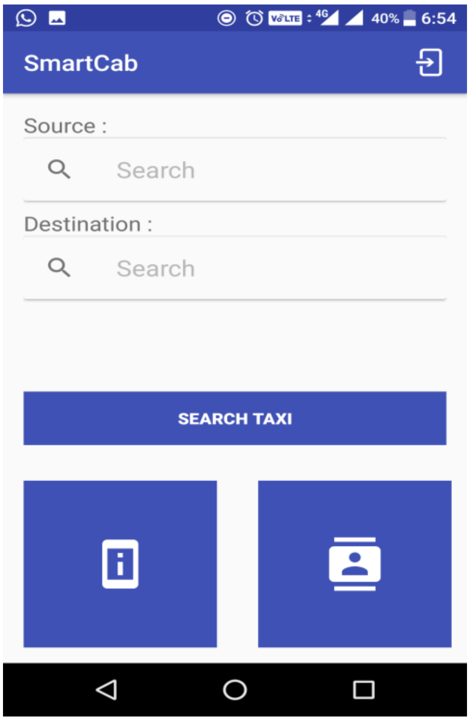
```
        }
        else
        {
            session_unset();
            session_destroy();
            echo "<script>window.location.href='$url'</script>";
        }
    }
    else
    {
        echo "<script>window.location.href='$url'</script>";
    }
?>
```

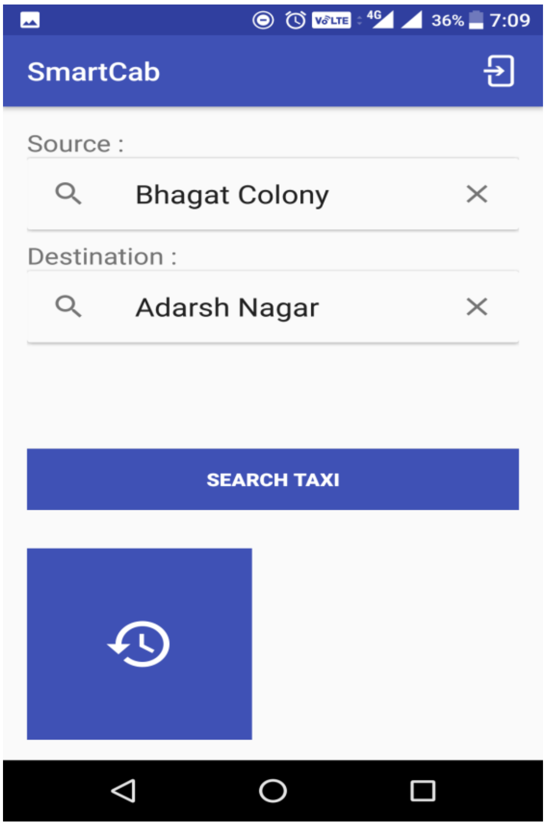
SCREENSHOTS


SmartCab
Live life Smarter!
SMART CAB
GET STARTED


SmartCab
Email...
Password...
LOGIN
Sign Up


SmartCab
Email...
Password...
SIGN UP
Log in


SmartCab
Source :
Search
Destination :
Search
SEARCH TAXI

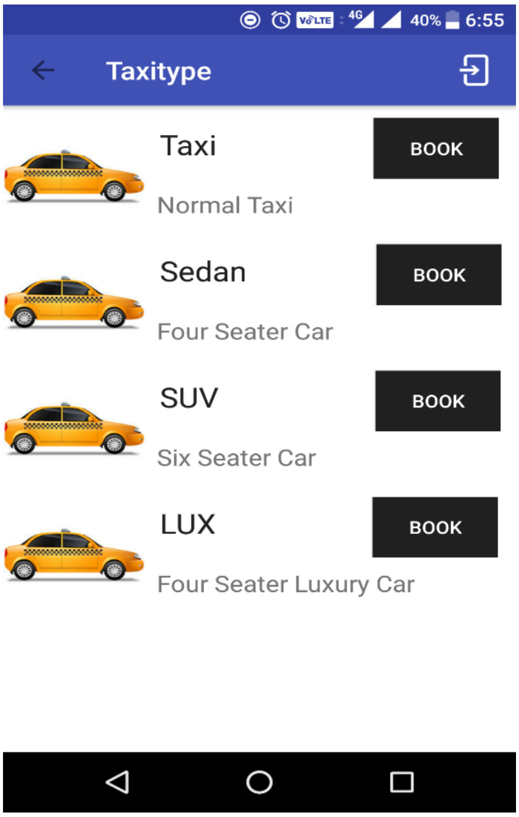


Source :

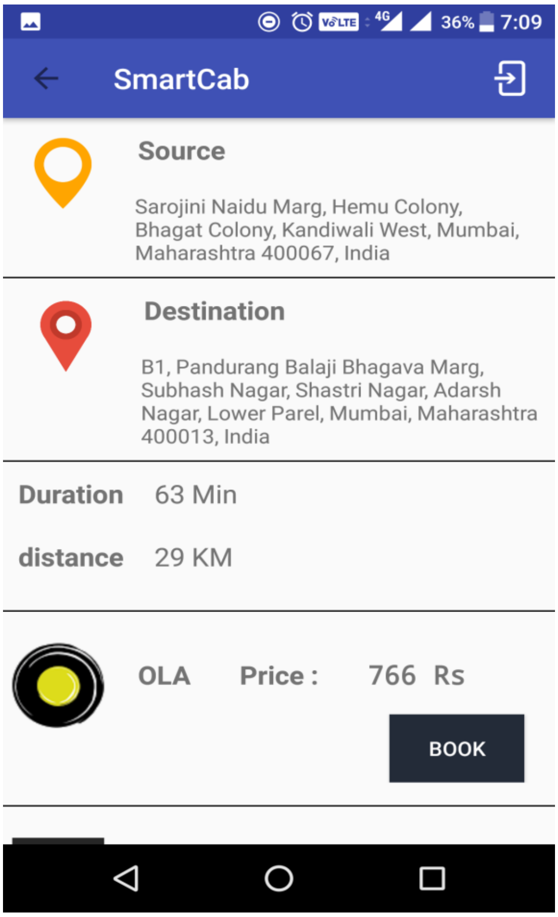
Bhagat Colony

Destination :

Adarsh Nagar



Taxi Normal Taxi	BOOK
Sedan Four Seater Car	BOOK
SUV Six Seater Car	BOOK
LUX Four Seater Luxury Car	BOOK



Source

Sarojini Naidu Marg, Hemu Colony, Bhagat Colony, Kandiwali West, Mumbai, Maharashtra 400067, India

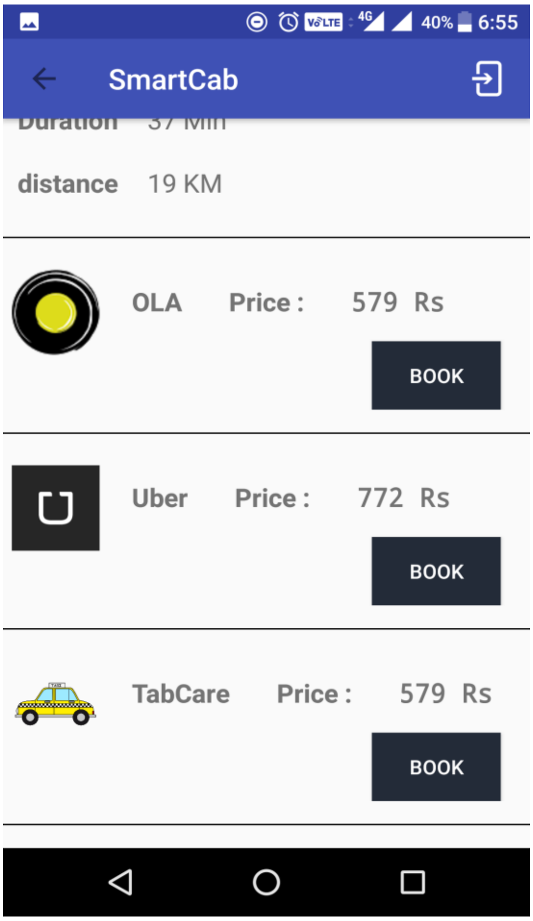
Destination

B1, Pandurang Balaji Bhagava Marg, Subhash Nagar, Shastri Nagar, Adarsh Nagar, Lower Parel, Mumbai, Maharashtra 400013, India

Duration 63 Min

distance 29 KM

OLA Price : 766 Rs



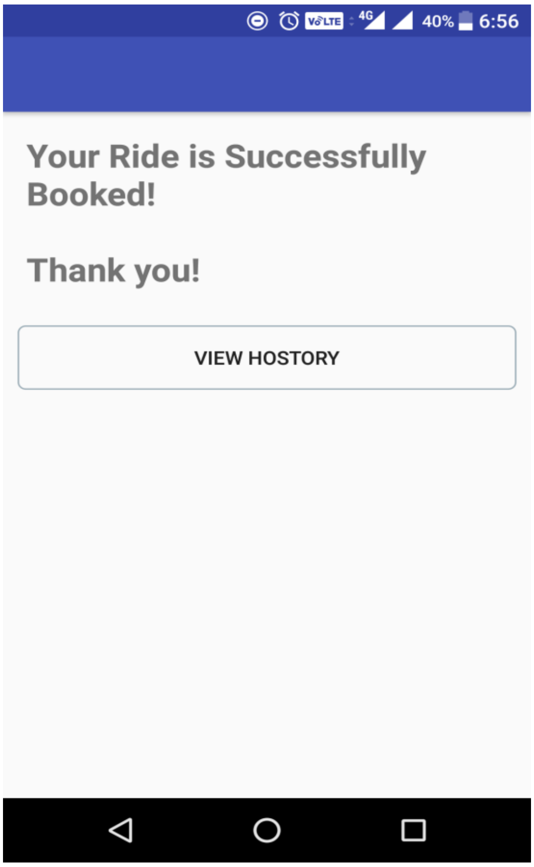
Duration 37 Min

distance 19 KM

OLA Price : 579 Rs

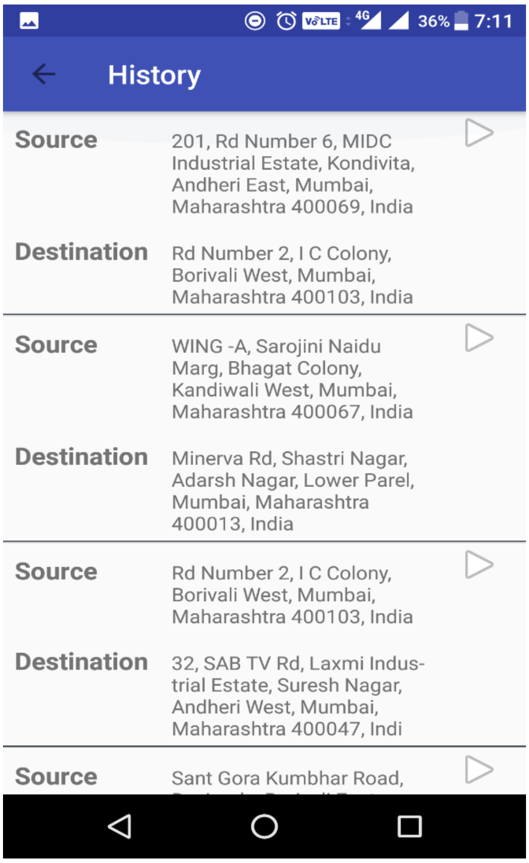
Uber Price : 772 Rs

TabCare Price : 579 Rs



Your Ride is Successfully Booked!

Thank you!



Source 201, Rd Number 6, MIDC Industrial Estate, Kondivita, Andheri East, Mumbai, Maharashtra 400069, India

Destination Rd Number 2, I C Colony, Borivali West, Mumbai, Maharashtra 400103, India

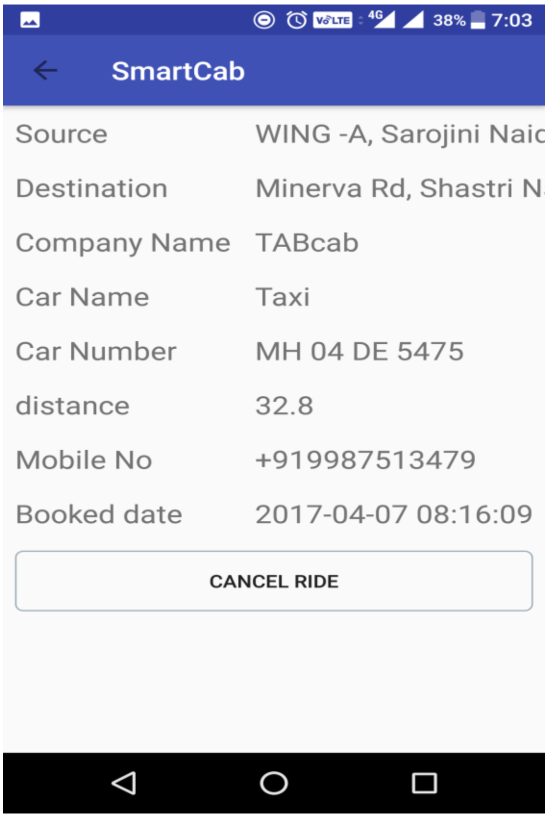
Source WING -A, Sarojini Naidu Marg, Bhagat Colony, Kandiwali West, Mumbai, Maharashtra 400067, India

Destination Minerva Rd, Shastri Nagar, Adarsh Nagar, Lower Parel, Mumbai, Maharashtra 400013, India

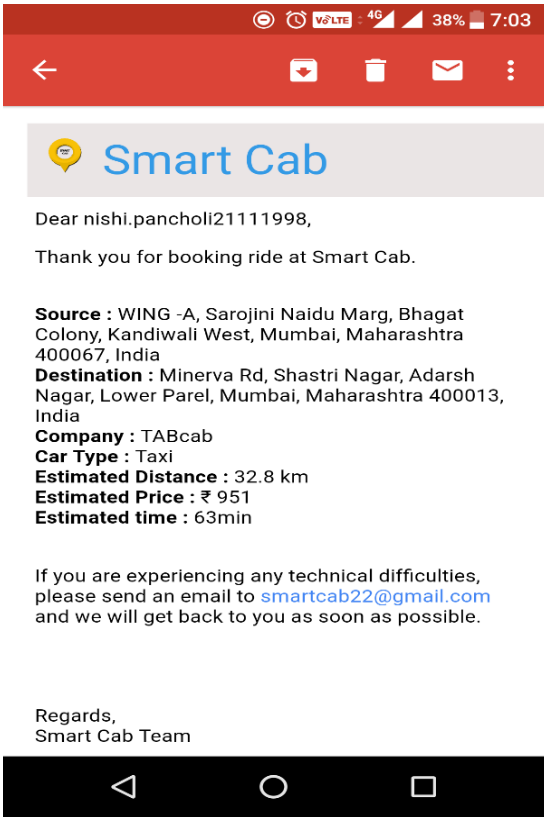
Source Rd Number 2, I C Colony, Borivali West, Mumbai, Maharashtra 400103, India

Destination 32, SAB TV Rd, Laxmi Industrial Estate, Suresh Nagar, Andheri West, Mumbai, Maharashtra 400047, Indi

Source Sant Gora Kumbhar Road,



Source	WING -A, Sarojini Naic
Destination	Minerva Rd, Shastri N
Company Name	TABcab
Car Name	Taxi
Car Number	MH 04 DE 5475
distance	32.8
Mobile No	+919987513479
Booked date	2017-04-07 08:16:09



Smart Cab

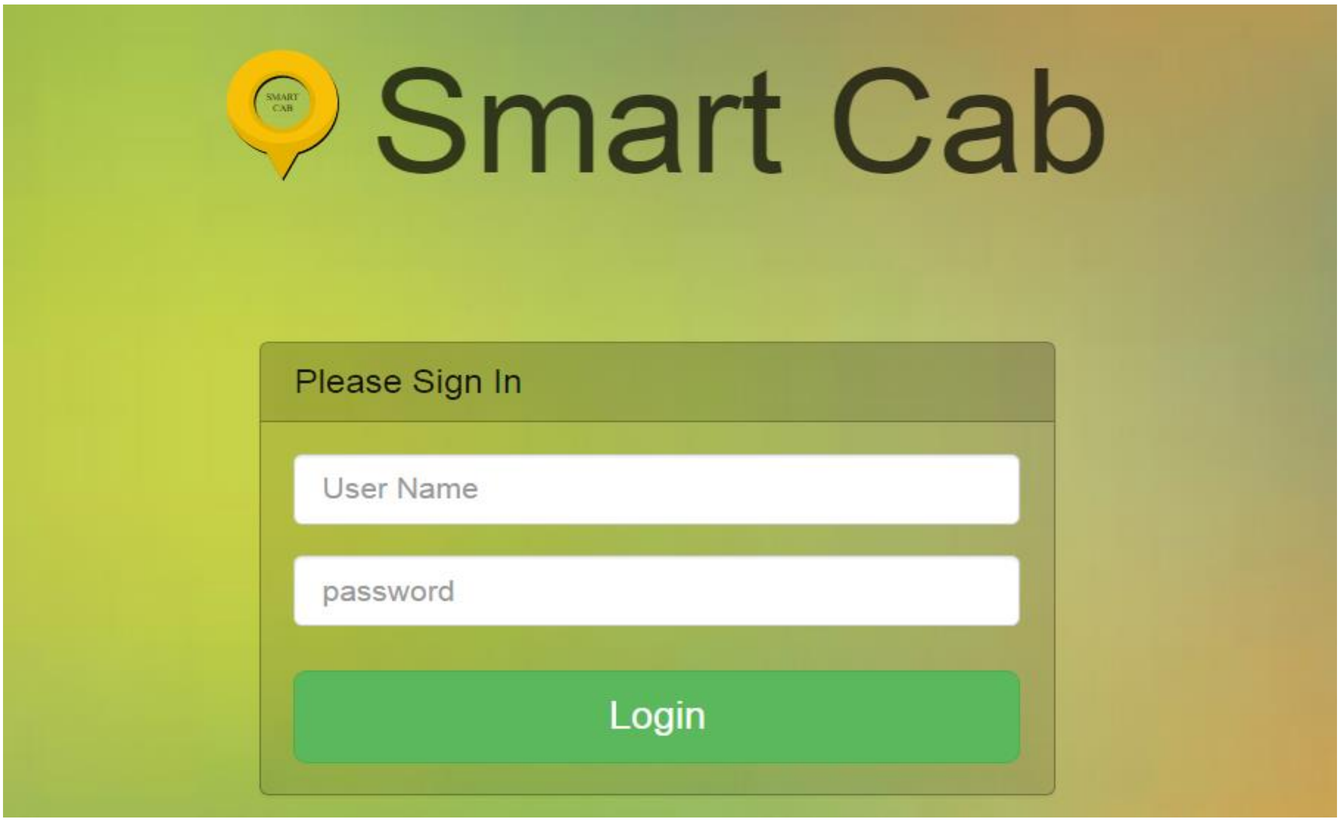
Dear nishi.pancholi21111998,

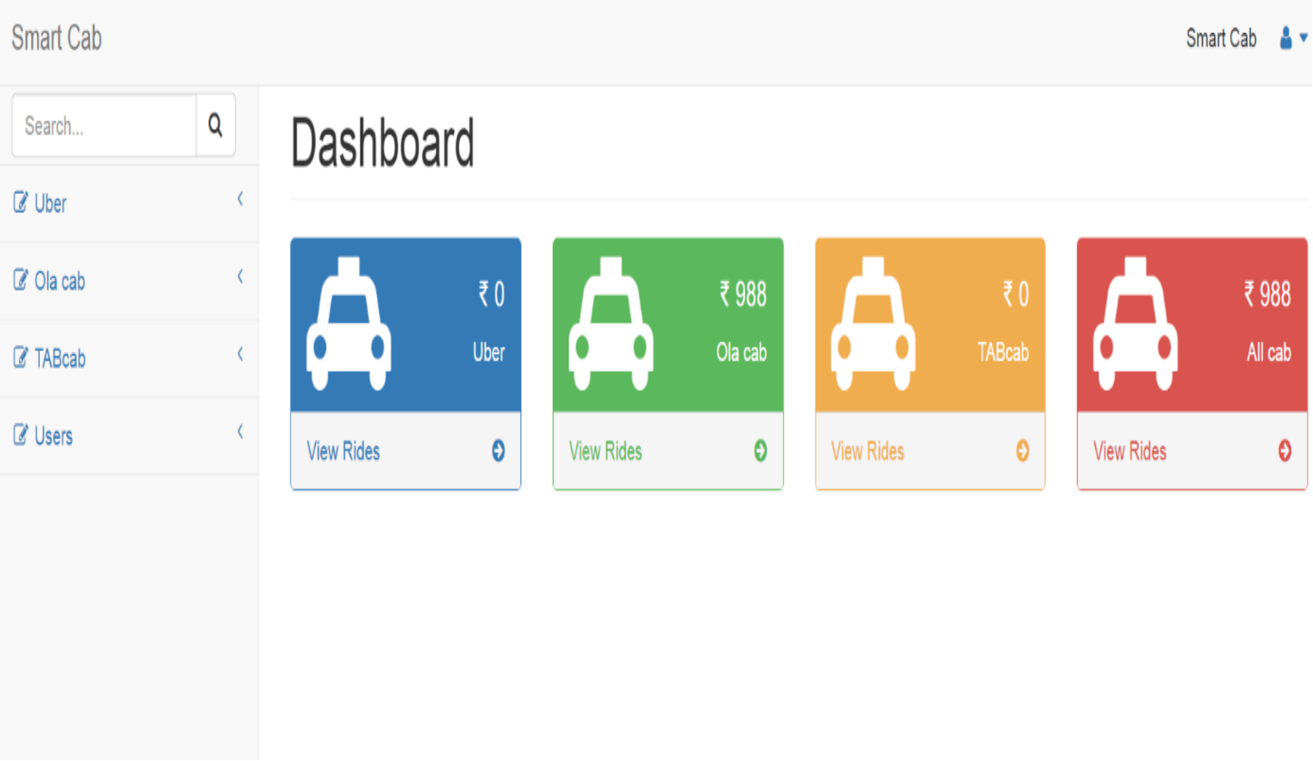
Thank you for booking ride at Smart Cab.

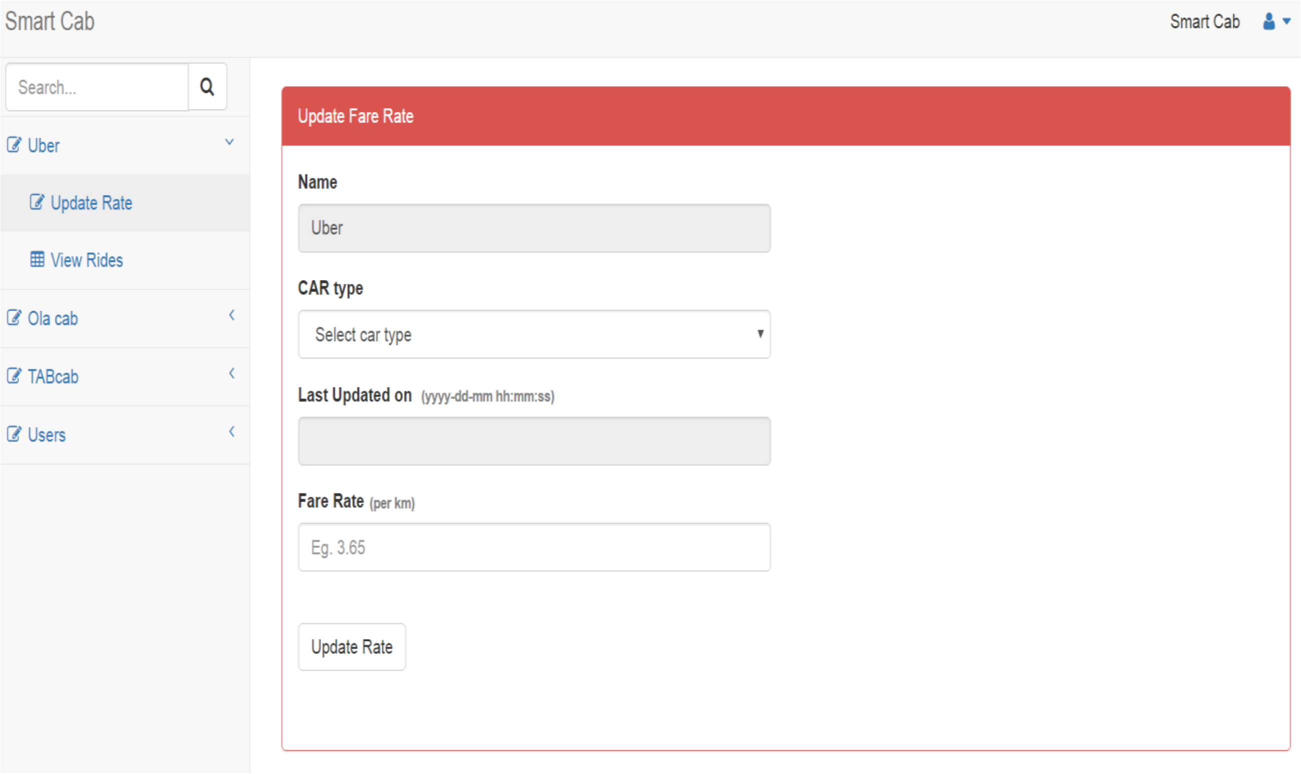
Source : WING -A, Sarojini Naidu Marg, Bhagat Colony, Kandiwali West, Mumbai, Maharashtra 400067, India
Destination : Minerva Rd, Shastri Nagar, Adarsh Nagar, Lower Parel, Mumbai, Maharashtra 400013, India
Company : TABcab
Car Type : Taxi
Estimated Distance : 32.8 km
Estimated Price : ₹ 951
Estimated time : 63min

If you are experiencing any technical difficulties, please send an email to smartcab22@gmail.com and we will get back to you as soon as possible.

Regards,
Smart Cab Team


Smart Cab
Please Sign In
User Name
password
Login


Smart Cab
Smart Cab
Search...
Uber
Ola cab
TABcab
Users
Dashboard
₹ 0
Uber
View Rides
₹ 988
Ola cab
View Rides
₹ 0
TABcab
View Rides
₹ 988
All cab
View Rides



Update Fare Rate

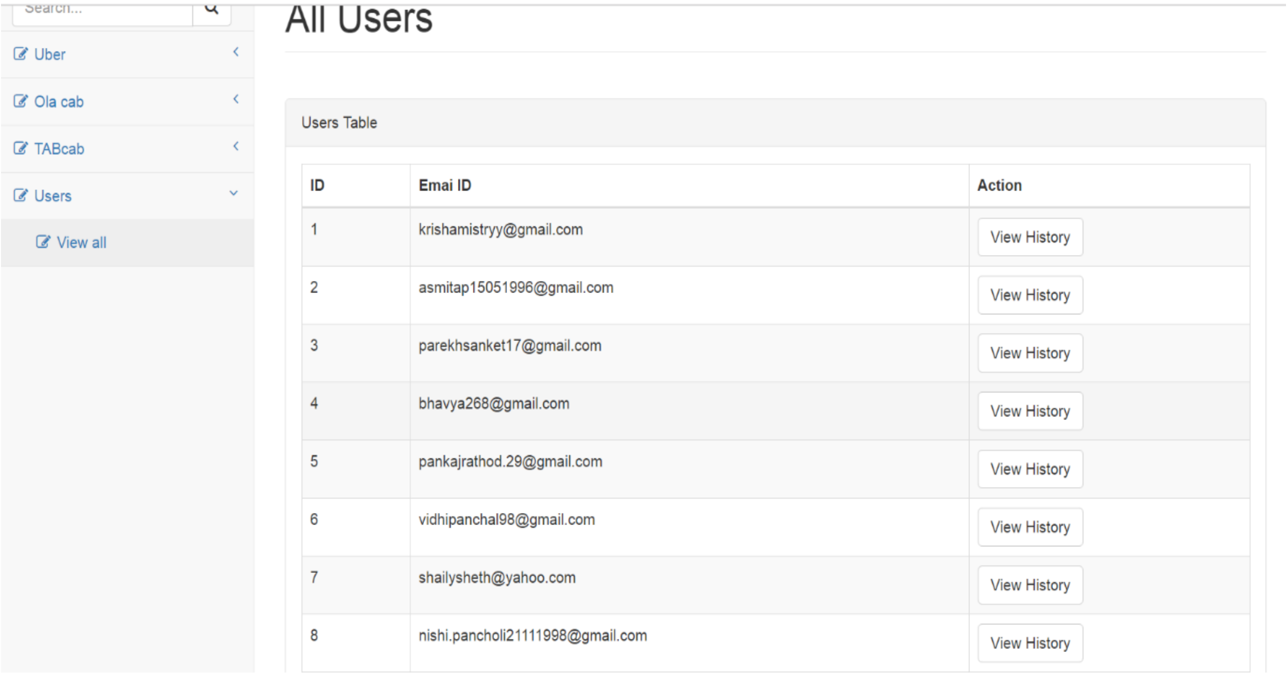
Name
Uber

CAR type
Select car type

Last Updated on (yyyy-dd-mm hh:mm:ss)

Fare Rate (per km)
Eg. 3.65

Update Rate

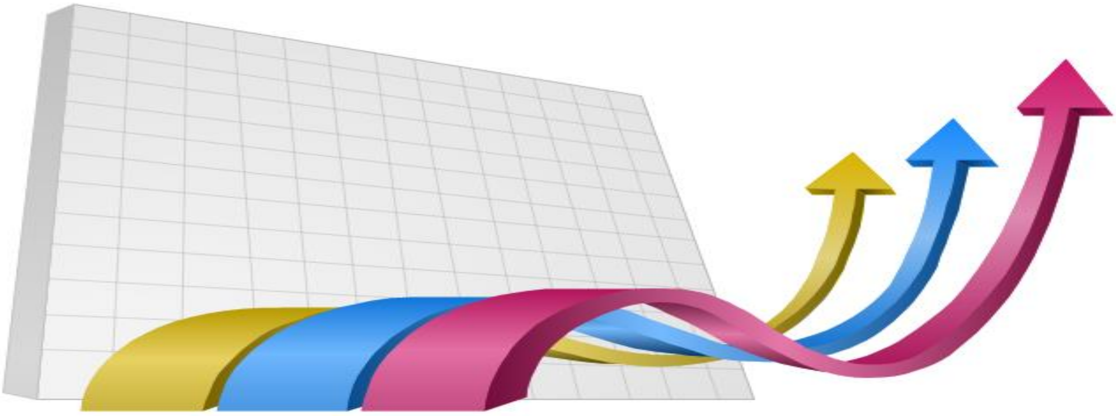


All Users

Users Table

ID	Emai ID	Action
1	krishamistryy@gmail.com	View History
2	asmitap15051996@gmail.com	View History
3	parekhsanket17@gmail.com	View History
4	bhavya268@gmail.com	View History
5	pankajrathod.29@gmail.com	View History
6	vidhipanchal98@gmail.com	View History
7	shailysheth@yahoo.com	View History
8	nishi.pancholi21111998@gmail.com	View History

TIMELINE CHART



Months	January	February	March	April

Weeks / Activities	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Research	🟦	🟦														
Analysis		🟦	🟦													
Design				🟩	🟩	🟩										
Coding						🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	
Testing							🟥		🟥		🟥		▨	▨	▨	▨

ADVANTAGES

- Users will get a clear idea about the booking of the cab
- Comparison between the different cab companies
- View history
- User-friendly
- Small in size
- Email sent to customer after booking
- Easy cancellation

LIMITATIONS

- User has to install the SmartCab application.
- No live GPS
- No real-time data
- User has to login.
- Requires an active internet connection.

FUTURE ENHANCEMENT

- Later it can be developed as iOS application for iPhones.
- More number of companies can be provided.
- GPS and real time data can be collaborated.
- More type of the cab and amenities can be provided.
- This app is free, updates will come as free for all users.
- First ride, common traveler offers.

CONCLUSION

SMART CAB allows users to efficiently compare fare prices of different companies. This improvement in the cab booking experience not only reduces the booking fear, but also makes sure that the cab companies deliver a great experience. For the first time, users could get a strong understanding of how much they have to pay before the ride. As people are more likely to use smart phones, this android application has a very user-friendly environment. This application also allows users to view their previous bookings. Thus, we hope that this application provides great expirence to the users and makes it easier for them to travel.

REFERENCE

1. https://enterprise.google.com/GoogleMapsAPI
2. http://www.wchotels.com/meta-search-marketing-for-hotels.html
3. www.000webhost.com
4. www.android.com/androidstudiotutorial
5. www.ftpprotocoluse.com
6. www.reference.com/mysqldatabase21346

ABOUT US

Vidhi Panchal [1498020]

vidhipanchal98@gmail.com

Nishi Pancholi [1498022]

nishi.pancholi21111998@gmail.com

Mayank Parvatia [1498024]

mparvatia66@gmail.com