CONNECTION MANAGER.JAVA

package com.uniq;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

public class ConnectionManager {

static Connection conn = null;

static String url = "jdbc:mysql://localhost:3306/qrcode";

public static Connection getConnection() {

try {

Class.forName("com.mysql.jdbc.Driver");

String username = "root";

String password = "root";

conn = DriverManager.getConnection(url, username, password);

} catch (ClassNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (SQLException e) {

e.printStackTrace();

}

return conn;

}

}

LOGIN CHECKER.JAVA

package com.uniq;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import java.sql.\*;

import java.util.logging.Level;

import java.util.logging.Logger;

public class LoginChecker extends HttpServlet {

Connection con;

Statement st;

ResultSet rs;

protected void processRequest(HttpServletRequest request,

HttpServletResponse response) throws ServletException, IOException,

SQLException {

response.setContentType("text/html;charset=UTF-8");

PrintWriter out = response.getWriter();

try {

String user = request.getParameter("user");

String pass = request.getParameter("pass");

String url = "jdbc:mysql://localhost:3306/qrcode";

Class.forName("com.mysql.jdbc.Driver");

con = DriverManager.getConnection(url, "root", "root");

st = con.createStatement();

rs = st.executeQuery("select \* from userlogin");

//json\_encode(rs);

boolean flag = false;

while (rs.next()) {

String dbuser = rs.getString(1);

String dbpass = rs.getString(2);

if (user.equalsIgnoreCase(dbuser) && pass.equals(dbpass)) {

flag = true;

break;

}

}

if (flag) {

response.sendRedirect("CustomerLogs.html");

} else {

response.sendRedirect("index.jsp?result=invalid");

}

} catch (ClassNotFoundException ex) {

Logger.getLogger(LoginChecker.class.getName()).log(Level.SEVERE,

null, ex);

} finally {

out.close();

}

}

private void json\_encode(ResultSet rs2) {

// TODO Auto-generated method stub

}

// <editor-fold defaultstate="collapsed"

// desc="HttpServlet methods. Click on the + sign on the left to edit the code.">

/\*\*

\* Handles the HTTP <code>GET</code> method.

\*

\* @param request

\* servlet request

\* @param response

\* servlet response

\* @throws ServletException

\* if a servlet-specific error occurs

\* @throws IOException

\* if an I/O error occurs

\*/

@Override

protected void doGet(HttpServletRequest request,

HttpServletResponse response) throws ServletException, IOException {

try {

processRequest(request, response);

} catch (SQLException ex) {

Logger.getLogger(LoginChecker.class.getName()).log(Level.SEVERE,

null, ex);

}

}

@Override

protected void doPost(HttpServletRequest request,

HttpServletResponse response) throws ServletException, IOException {

try {

processRequest(request, response);

} catch (SQLException ex) {

Logger.getLogger(LoginChecker.class.getName()).log(Level.SEVERE,

null, ex);

}

}

/\*\*

\* Returns a short description of the servlet.

\*

\* @return a String containing servlet description

\*/

@Override

public String getServletInfo() {

return "Short description";

}// </editor-fold>

}

PAY SERVLET.JAVA

package com.uniq;

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

/\*\*

\* Servlet implementation class payServlet

\*/

public class payServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public payServlet() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

}

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

String billno = request.getParameter("billno").toString();

System.out.println(billno);

request.getRequestDispatcher("CustomerLogs.html").forward(request, response);

}

}

QR CODE SERVLET.JAVA

package com.uniq.qrcode;

import java.io.ByteArrayOutputStream;

import java.io.IOException;

import java.io.OutputStream;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import net.glxn.qrgen.QRCode;

import net.glxn.qrgen.image.ImageType;

public class QRCodeServlet extends HttpServlet {

@Override

protected void doGet(HttpServletRequest request,

HttpServletResponse response) throws ServletException, IOException {

String qrtext = request.getParameter("qrtext");

String pname=request.getParameter("pname");

String pdate=request.getParameter("pdate");

String pquants=request.getParameter("pquants");

String pprice=request.getParameter("pprice");

String result = pname+";"+pdate+";"+pquants+";"+pprice;

ByteArrayOutputStream out = QRCode.from(qrtext).to(

ImageType.PNG).stream();

response.setContentType("image/png");

response.setContentLength(out.size());

OutputStream outStream = response.getOutputStream();

outStream.write(out.toByteArray());

outStream.flush();

outStream.close();

}

}

IP ADDRESS.JAVA

**package** com.example.qrcode;

**public** **class** Ipaddress {

**public** **static** **final** String *URL* = "http://192.168.43.111:8080/QrcodeServer/ParameterServlet";

}

LOGIN.JAVA

package com.example.qrcode;

import java.io.BufferedReader;

import java.io.InputStream;

import java.io.InputStreamReader;

import java.util.ArrayList;

import java.util.List;

import org.apache.http.HttpResponse;

import org.apache.http.NameValuePair;

import org.apache.http.client.HttpClient;

import org.apache.http.client.entity.UrlEncodedFormEntity;

import org.apache.http.client.methods.HttpPost;

import org.apache.http.impl.client.DefaultHttpClient;

import org.apache.http.message.BasicNameValuePair;

import android.app.Activity;

import android.app.ProgressDialog;

import android.content.Intent;

import android.content.SharedPreferences;

import android.os.Bundle;

import android.os.Handler;

import android.os.Message;

import android.text.TextUtils;

import android.util.Log;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.Button;

import android.widget.EditText;

import android.widget.TextView;

import android.widget.Toast;

public class Login extends Activity {

EditText email, phoneno;

Button login;

TextView register;

String mail, pswd;

private ProgressDialog pd;

private Thread thread;

HttpClient httpClient;

HttpPost httpPost;

HttpResponse httpResponse;

private InputStream stream = null;

private StringBuilder stringBuilder;

private String error\_response;

SharedPreferences sharedPref;

SharedPreferences.Editor editor;

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.login1);

email = (EditText) findViewById(R.id.email);

phoneno = (EditText) findViewById(R.id.password);

login = (Button) findViewById(R.id.btnLogin);

register = (TextView) findViewById(R.id.link\_to\_register);

// Create object of SharedPreferences.

sharedPref = getSharedPreferences("mypref", 0);

// now get Editor

editor = sharedPref.edit();

register.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View v) {

// TODO Auto-generated method stub

Intent i = new Intent(Login.this, Registration.class);

startActivity(i);

}

});

}

public void Login(View v) {

// put your value

editor.putString("name", email.getText().toString());

editor.putString("phonenumber", phoneno.getText().toString());

// commits your edits

editor.commit();

if (TextUtils.isEmpty(email.getText().toString())

|| TextUtils.isEmpty(phoneno.getText().toString())) {

Toast.makeText(Login.this, "Email or Password field is empty",

Toast.LENGTH\_SHORT).show();

} else {

pd = ProgressDialog.show(Login.this, "", "Loggin you in...", false,

false);

thread = new Thread() {

public void run() {

try {

httpClient = new DefaultHttpClient();

httpPost = new HttpPost(Ipaddress.URL);

List<NameValuePair> nameValuePair = new ArrayList<NameValuePair>();

nameValuePair.add(new BasicNameValuePair("flag", "3"));

nameValuePair.add(new BasicNameValuePair("Email", email

.getText().toString()));

nameValuePair.add(new BasicNameValuePair("Phoneno",

phoneno.getText().toString()));

httpPost.setEntity(new UrlEncodedFormEntity(

nameValuePair));

httpResponse = httpClient.execute(httpPost);

InputStream inputStream = httpResponse.getEntity()

.getContent();

InputStreamReader inputStreamReader = new InputStreamReader(

inputStream);

BufferedReader bufferedReader = new BufferedReader(

inputStreamReader);

stringBuilder = new StringBuilder();

String bufferedStrChunk = null;

while ((bufferedStrChunk = bufferedReader.readLine()) != null) {

stringBuilder.append(bufferedStrChunk);

}

System.out

.println("Login :" + stringBuilder.toString());

if (stringBuilder.toString().trim().equals("yes")) {

handler.sendEmptyMessage(1);

} else {

handler.sendEmptyMessage(2);

}

} catch (Exception e) {

handler.sendEmptyMessage(3);

}

}

};

thread.start();

}

}

private Handler handler = new Handler() {

public void handleMessage(Message msg) {

switch (msg.what) {

case 1:

pd.dismiss();

Log.i("login response", "login success");

startActivity(new Intent(Login.this, MainActivity.class));

break;

case 2:

pd.dismiss();

Toast.makeText(Login.this,

"Your username and password is wrong !",

Toast.LENGTH\_LONG).show();

break;

case 3:

pd.dismiss();

Toast.makeText(Login.this,

"Please check your internet connection or URL!",

Toast.LENGTH\_LONG).show();

break;

}}};}

MAIN ACTIVITY.JAVA

package com.example.qrcode;

import java.util.ArrayList;

import java.util.List;

import org.apache.http.HttpResponse;

import org.apache.http.NameValuePair;

import org.apache.http.client.HttpClient;

import org.apache.http.client.entity.UrlEncodedFormEntity;

import org.apache.http.client.methods.HttpPost;

import org.apache.http.impl.client.DefaultHttpClient;

import org.apache.http.message.BasicNameValuePair;

import android.app.Activity;

import android.app.AlertDialog;

import android.content.DialogInterface;

import android.content.Intent;

import android.content.SharedPreferences;

import android.os.Bundle;

import android.preference.PreferenceManager;

import android.util.Log;

import android.view.LayoutInflater;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.Button;

import android.widget.EditText;

import android.widget.TextView;

import android.widget.Toast;

public class MainActivity extends Activity {

HttpClient httpClient;

HttpPost httpPost;

HttpResponse httpResponse;

double uprice;

String s;

String email;

ArrayList<String> arg;

ArrayList<String> myArrayList = new ArrayList<String>();

ArrayList<String> myPriceList = new ArrayList<String>();

ArrayList<String> myQuantsList = new ArrayList<String>();

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

SharedPreferences sharedPref = getSharedPreferences("mypref", 0);

email = sharedPref.getString("name", "");

Log.i("EmailID", "" + email);

Button scanBtn = (Button) findViewById(R.id.btnScan);

scanBtn.setOnClickListener(new OnClickListener() {

public void onClick(View v) {

// TODO Auto-generated method stub

try {

Intent intent = new Intent(

"com.google.zxing.client.android.SCAN");

intent.putExtra("SCAN\_MODE", "QR\_CODE\_MODE,PRODUCT\_MODE");

startActivityForResult(intent, 0);

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

Toast.makeText(getApplicationContext(), "ERROR:" + e, 1)

.show();

}

}

});

}

// In the same activity you’ll need the following to retrieve the results:

public void onActivityResult(int requestCode, int resultCode, Intent intent) {

if (requestCode == 0) {

if (resultCode == RESULT\_OK) {

final String result = intent.getStringExtra("SCAN\_RESULT");

arg = new ArrayList<String>();

for (String ob : result.split(":")) {

arg.add(ob.trim());

}

Log.d("Arraylist", "" + arg);

Toast.makeText(getBaseContext(), "name : " + arg.get(1),

Toast.LENGTH\_LONG).show();

AlertDialog.Builder alert = new AlertDialog.Builder(

MainActivity.this);

LayoutInflater inflater = MainActivity.this.getLayoutInflater();

// this is what I did to added the layout to the alert dialog

View layout = inflater.inflate(R.layout.productdetail, null);

alert.setView(layout);

final TextView productName = (TextView) layout

.findViewById(R.id.textView1);

final TextView actualprice = (TextView) layout

.findViewById(R.id.textView2);

final TextView updatedprice = (TextView) layout

.findViewById(R.id.textView3);

final TextView expirydate = (TextView) layout

.findViewById(R.id.textView5);

final EditText usernameInput = (EditText) layout

.findViewById(R.id.quantity);

productName.setText(arg.get(0));

actualprice.setText("Product price : " + arg.get(3));

expirydate.setText("Expiry Date : " + arg.get(1));

alert.setPositiveButton("OK",

new DialogInterface.OnClickListener() {

public void onClick(DialogInterface dialog,

int whichButton) {

Log.i("quantity", ""

+ usernameInput.getText().toString());

s = usernameInput.getText().toString();

double i = Double.parseDouble(s);

int i1 = Integer.parseInt(arg.get(3));

uprice = i \* i1;

Log.i("updated price", "" + uprice);

Log.i("Check", "working");

Productname pna = new Productname();

pna.setPname(arg.get(0));

pna.setPprice("" + uprice);

pna.setPquants(s);

Log.i("NewData", "" + pna.getPname());

myArrayList.add(pna.getPname());

myPriceList.add(pna.getPprice());

myQuantsList.add(pna.getPquants());

}

});

alert.setNegativeButton("CANCEL",

new DialogInterface.OnClickListener() {

public void onClick(DialogInterface dialog,

int whichButton) {

// Canceled.

dialog.cancel();

}

});

alert.show();

}

} else if (resultCode == RESULT\_CANCELED) {

Toast.makeText(this, "Scan cancelled...", Toast.LENGTH\_LONG).show();

}

}

public void viewproducts(View v) {

SharedPreferences sPrefs = PreferenceManager

.getDefaultSharedPreferences(this);

SharedPreferences.Editor sEdit = sPrefs.edit();

for (int i = 0; i < myArrayList.size(); i++) {

sEdit.putString("pname" + i, myArrayList.get(i));

}

for (int i = 0; i < myPriceList.size(); i++) {

sEdit.putString("price" + i, myPriceList.get(i));

Log.i("price list", "" + myPriceList.get(i));

}

for (int i = 0; i < myQuantsList.size(); i++) {

sEdit.putString("Quantity" + i, myQuantsList.get(i));

}

sEdit.putInt("size", myArrayList.size());

sEdit.putInt("size1", myPriceList.size());

sEdit.putInt("size2", myQuantsList.size());

sEdit.commit();

Intent i = new Intent(MainActivity.this, Productlist.class);

startActivity(i);

}

}

MODEL.JAVA

**package** com.example.qrcode;

**public** **class** Model {

**private** String name;

**private** String price;

**private** String quants;

**private** **boolean** selected;

**public** Model(String name, String price, String quants, **boolean** selected) {

**super**();

**this**.name = name;

**this**.price = price;

**this**.quants = quants;

**this**.selected = selected;

}

**public** Model() {

// **TODO** Auto-generated constructor stub

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** **boolean** isSelected() {

**return** selected;

}

**public** **void** setSelected(**boolean** selected) {

**this**.selected = selected;

}

**public** String getPrice() {

**return** price;

}

**public** **void** setPrice(String price) {

**this**.price = price;

}

**public** String getQuants() {

**return** quants;

}

**public** **void** setQuants(String quants) {

**this**.quants = quants;

}

}

PRODUCTLIST.JAVA

package com.example.qrcode;

import java.math.BigDecimal;

import java.util.ArrayList;

import java.util.List;

import java.util.Random;

import org.apache.http.HttpResponse;

import org.apache.http.NameValuePair;

import org.apache.http.client.HttpClient;

import org.apache.http.client.entity.UrlEncodedFormEntity;

import org.apache.http.client.methods.HttpPost;

import org.apache.http.impl.client.DefaultHttpClient;

import org.apache.http.message.BasicNameValuePair;

import org.json.JSONException;

import android.app.Activity;

import android.app.ProgressDialog;

import android.content.Context;

import android.content.Intent;

import android.content.SharedPreferences;

import android.os.Bundle;

import android.preference.PreferenceManager;

import android.telephony.gsm.SmsManager;

import android.util.Log;

import android.view.LayoutInflater;

import android.view.Menu;

import android.view.View;

import android.view.View.OnClickListener;

import android.view.ViewGroup;

import android.widget.ArrayAdapter;

import android.widget.CheckBox;

import android.widget.ListView;

import android.widget.TextView;

import android.widget.Toast;

import com.paypal.android.sdk.payments.PayPalPayment;

import com.paypal.android.sdk.payments.PayPalService;

import com.paypal.android.sdk.payments.PaymentActivity;

import com.paypal.android.sdk.payments.PaymentConfirmation;

public class Productlist extends Activity {

private ListView list\_products;

private ProgressDialog pd;

private CrimeListAdapter crimeListAdapter = null;

double sum = 0;

SharedPreferences sharedPref;

String email, billno;

ArrayList<String> al = new ArrayList<String>();

ArrayList<Model> productlist = new ArrayList<Model>();

Model model;

String j, k;

ArrayList<String> array;

ArrayList<String> as = new ArrayList<String>();

ArrayList<String> myArrayList = new ArrayList<String>();

ArrayList<String> myQuantsList = new ArrayList<String>();

HttpClient httpClient;

HttpPost httpPost;

HttpResponse httpResponse;

String ownerphno = "9790363948", ph;

int i1;

// set to PaymentActivity.ENVIRONMENT\_PRODUCTION to move real money.

// set to PaymentActivity.ENVIRONMENT\_SANDBOX to use your test credentials

// from https://developer.paypal.com

// set to PaymentActivity.ENVIRONMENT\_NO\_NETWORK to kick the tires without

// communicating to PayPal's servers.

private static final String CONFIG\_ENVIRONMENT = PaymentActivity.ENVIRONMENT\_SANDBOX;

// note that these credentials will differ between live & sandbox

// environments.

private static final String CONFIG\_CLIENT\_ID = "ATheHhDM0UADhwiJ5coE\_jXQNFCU6TQeex6Gz7phoTH8ffz4SoQrHjVznaqR";

// when testing in sandbox, this is likely the -facilitator email address.

private static final String CONFIG\_RECEIVER\_EMAIL = "nalendran.m-facilitator@gmail.com";

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.productlist);

Intent intent = new Intent(this, PayPalService.class);

intent.putExtra(PaymentActivity.EXTRA\_PAYPAL\_ENVIRONMENT,

CONFIG\_ENVIRONMENT);

intent.putExtra(PaymentActivity.EXTRA\_CLIENT\_ID, CONFIG\_CLIENT\_ID);

intent.putExtra(PaymentActivity.EXTRA\_RECEIVER\_EMAIL,

CONFIG\_RECEIVER\_EMAIL);

startService(intent);

SharedPreferences sharedPref = getSharedPreferences("mypref", 0);

email = sharedPref.getString("name", "");

ph = sharedPref.getString("phonenumber", "");

Log.i("Email And Number", "" + email + ph);

SharedPreferences sPrefs = PreferenceManager

.getDefaultSharedPreferences(this);

ArrayList<String> myAList = new ArrayList<String>();

ArrayList<String> myPList = new ArrayList<String>();

ArrayList<String> myQList = new ArrayList<String>();

int size = sPrefs.getInt("size", 0);

int size1 = sPrefs.getInt("size1", 0);

int size2 = sPrefs.getInt("size2", 0);

for (int j = 0; j < size; j++) {

myAList.add(sPrefs.getString("pname" + j, email));

}

for (int j = 0; j < size1; j++) {

myPList.add(sPrefs.getString("price" + j, email));

}

for (int j = 0; j < size2; j++) {

myQList.add(sPrefs.getString("Quantity" + j, email));

}

Log.i("Product list", "" + myAList + myPList + myQList);

for (int i = 0, k = 0, j = 0; i < myAList.size(); i++, k++, j++) {

model = new Model(myAList.get(i), myPList.get(k), myQList.get(j),

false);

productlist.add(model);

}

crimeListAdapter = new CrimeListAdapter(this, R.layout.listofproduct,

productlist);

list\_products = (ListView) findViewById(R.id.listView1);

list\_products.setAdapter(crimeListAdapter);

crimeListAdapter.notifyDataSetChanged();

}

public class CrimeListAdapter extends ArrayAdapter<Model> {

private ArrayList<Model> productlist;

public CrimeListAdapter(Context context, int textViewResourceId,

ArrayList<Model> productlist) {

super(context, textViewResourceId, productlist);

this.productlist = new ArrayList<Model>();

this.productlist.addAll(productlist);

}

@Override

public View getView(int position, View convertView, ViewGroup parent) {

// TODO Auto-generated method stub

ViewHolder viewHolder = null;

Log.v("ConvertView", String.valueOf(position));

if (convertView == null) {

viewHolder = new ViewHolder();

convertView = LayoutInflater.from(Productlist.this).inflate(

R.layout.listofproduct, parent, false);

viewHolder.txt\_product = (TextView) convertView

.findViewById(R.id.txt\_product);

viewHolder.txt\_price = (TextView) convertView

.findViewById(R.id.txt\_price);

viewHolder.txt\_quants = (TextView) convertView

.findViewById(R.id.txt\_quants);

viewHolder.txt\_check = (CheckBox) convertView

.findViewById(R.id.checkBox1);

convertView.setTag(viewHolder);

viewHolder.txt\_check.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View v) {

// TODO Auto-generated method stub

CheckBox cb = (CheckBox) v;

Model model = (Model) cb.getTag();

model.setSelected(cb.isChecked());

}

});

} else {

viewHolder = (ViewHolder) convertView.getTag();

}

Model model = productlist.get(position);

viewHolder.txt\_product.setText(model.getName());

viewHolder.txt\_price.setText("Price: " + model.getPrice());

viewHolder.txt\_quants.setText("Quantity: " + model.getQuants());

viewHolder.txt\_check.setChecked(model.isSelected());

viewHolder.txt\_check.setTag(model);

return convertView;

}

}

private class ViewHolder {

TextView txt\_product, txt\_price, txt\_quants;

CheckBox txt\_check;

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

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

getMenuInflater().inflate(R.menu.productlist, menu);

return true;

}

@SuppressWarnings("deprecation")

public void DirectPay(View v) {

Checkproductlist();

int min = 1000;

int max = 3000;

Random r = new Random();

i1 = r.nextInt(max - min + 1) + min;

SharedPreferences sPrefs = PreferenceManager

.getDefaultSharedPreferences(this);

SharedPreferences.Editor se = sPrefs.edit();

se.putString("billno", "" + i1);

se.commit();

Log.i("RESULT\_OK", "Direct Payment");

new Thread() {

public void run() {

try {

for (int i = 0, k = 0, j = 0; i < myArrayList.size(); i++, k++, j++) {

httpClient = new DefaultHttpClient();

httpPost = new HttpPost(Ipaddress.URL);

List<NameValuePair> nameValuePair = new ArrayList<NameValuePair>();

nameValuePair.add(new BasicNameValuePair("flag", "0"));

nameValuePair.add(new BasicNameValuePair("paymenttype",

"Direct"));

nameValuePair.add(new BasicNameValuePair("status",

"UnPaid"));

nameValuePair

.add(new BasicNameValuePair("email", email));

nameValuePair.add(new BasicNameValuePair("name",

myArrayList.get(i)));

Log.i("", "" + myArrayList.get(i));

nameValuePair.add(new BasicNameValuePair("quantity",

myQuantsList.get(j)));

Log.i("", "" + myQuantsList.get(j));

nameValuePair.add(new BasicNameValuePair("price", ""

+ as.get(k)));

Log.i("", "" + as.get(k));

nameValuePair.add(new BasicNameValuePair("Billno", ""

+ i1));

Log.i("Check", "working");

httpPost.setEntity(new UrlEncodedFormEntity(

nameValuePair));

httpResponse = httpClient.execute(httpPost);

}

} catch (Exception e) {

// handler.sendEmptyMessage(3);

Log.d("Connection status error", "Connection" + e);

}

}

}.start();

SmsManager sms = SmsManager.getDefault();

sms.sendTextMessage(ownerphno, null, "Bill no: " + i1

+ "Total Amount: " + sum, null, null);

sms.sendTextMessage(ph, null,

"Thanks for your shopping from our store receive you product based on bill no"

+ i1 + "Total Amount: " + sum, null, null);

}

public void SelectProduct(View v) {

Checkproductlist();

Toast.makeText(this, "" + sum, Toast.LENGTH\_LONG).show();

Log.i("check", "working");

PayPalPayment thingToBuy = new PayPalPayment(new BigDecimal(sum),

"USD", "Total Amount :");

Intent intent = new Intent(this, PaymentActivity.class);

intent.putExtra(PaymentActivity.EXTRA\_PAYPAL\_ENVIRONMENT,

CONFIG\_ENVIRONMENT);

intent.putExtra(PaymentActivity.EXTRA\_CLIENT\_ID, CONFIG\_CLIENT\_ID);

intent.putExtra(PaymentActivity.EXTRA\_RECEIVER\_EMAIL,

CONFIG\_RECEIVER\_EMAIL);

// It's important to repeat the clientId here so that the SDK has it if

// Android restarts your

// app midway through the payment UI flow.

intent.putExtra(PaymentActivity.EXTRA\_CLIENT\_ID,

CONFIG\_CLIENT\_ID);

intent.putExtra(PaymentActivity.EXTRA\_PAYER\_ID,

CONFIG\_RECEIVER\_EMAIL);

intent.putExtra(PaymentActivity.EXTRA\_PAYMENT, thingToBuy);

startActivityForResult(intent, 0);

Log.i("check", "working");

}

@SuppressWarnings("deprecation")

@Override

protected void onActivityResult(int requestCode, int resultCode, Intent data) {

if (resultCode == Activity.RESULT\_OK) {

PaymentConfirmation confirm = data

.getParcelableExtra(PaymentActivity.EXTRA\_RESULT\_CONFIRMATION);

if (confirm != null) {

try {

int min = 1000;

int max = 3000;

Random r = new Random();

i1 = r.nextInt(max - min + 1) + min;

SharedPreferences sPrefs = PreferenceManager

.getDefaultSharedPreferences(this);

SharedPreferences.Editor se = sPrefs.edit();

se.putString("billno", "" + i1);

se.commit();

Log.i("paymentExample", confirm.toJSONObject().toString(4));

Log.i("RESULT\_OK", "paypal");

new Thread() {

public void run() {

try {

System.out.println(myArrayList);

for (int i = 0, k = 0, j = 0; i < myArrayList

.size(); i++, k++, j++) {

Log.i("Server", "working");

httpClient = new DefaultHttpClient();

httpPost = new HttpPost(Ipaddress.URL);

List<NameValuePair> nameValuePair = new ArrayList<NameValuePair>();

nameValuePair.add(new BasicNameValuePair(

"flag", "0"));

nameValuePair.add(new BasicNameValuePair(

"paymenttype", "PayPal"));

nameValuePair.add(new BasicNameValuePair(

"status", "Paid"));

nameValuePair.add(new BasicNameValuePair(

"email", email));

nameValuePair.add(new BasicNameValuePair(

"name", myArrayList.get(i)));

Log.i("", "" + myArrayList.get(i));

nameValuePair.add(new BasicNameValuePair(

"quantity", myQuantsList.get(j)));

Log.i("", "" + myQuantsList.get(j));

nameValuePair.add(new BasicNameValuePair(

"price", "" + as.get(k)));

Log.i("", "" + as.get(k));

nameValuePair.add(new BasicNameValuePair(

"Billno", "" + i1));

httpPost.setEntity(new UrlEncodedFormEntity(

nameValuePair));

httpResponse = httpClient.execute(httpPost);

}

} catch (Exception e) {

// handler.sendEmptyMessage(3);

Log.d("Connection status error", "Connection"

+ e);

}

}

}.start();

SmsManager sms = SmsManager.getDefault();

sms.sendTextMessage(ownerphno, null, "Bill no:" + i1, null,

null);

sms.sendTextMessage(ph, null,

"Thanks for your shopping from our store receive you product based on bill no"

+ i1 + "Visit Again", null, null);

startActivity(new Intent(this, Login.class));

} catch (JSONException e) {

Log.e("paymentExample",

"an extremely unlikely failure occurred: ", e);

}

}

} else if (resultCode == Activity.RESULT\_CANCELED) {

int q=0;

Log.i("RESULT\_CANCELED", "cancel payment");

Log.i("paymentExample", "The user canceled.");

System.out.println(as.size());

sum=0;

} else if (resultCode == PaymentActivity.RESULT\_PAYMENT\_INVALID) {

sum=0;

Log.i("RESULT\_PAYMENT\_INVALID", "working");

Log.i("paymentExample",

"An invalid payment was submitted. Please see the docs.");

}

}

@Override

public void onDestroy() {

stopService(new Intent(this, PayPalService.class));

super.onDestroy();

}

public void Checkproductlist() {

ArrayList<Model> productlist = crimeListAdapter.productlist;

List<Double> newList = new ArrayList<Double>();

for (int i = 0; i < productlist.size(); i++) {

Model model = productlist.get(i);

if (model.isSelected()) {

as.add(model.getPrice());

myArrayList.add(model.getName());

myQuantsList.add(model.getQuants());

Log.i("Checked product price", "" + as);

}

}

for (String myInt : as) {

newList.add(Double.valueOf(myInt));

Log.i("selected product ", "" + newList);

}

for (int j = 0; j < newList.size(); j++) {

System.out.println(newList.size());

sum = sum + newList.get(j);

Log.i("Total Amount1=", "" + sum);

}

}

}

PRODUCT NAME.JAVA

**package** com.example.qrcode;

**public** **class** Productname {

String pname;

String pprice;

String pquants;

**public** String getPquants() {

**return** pquants;

}

**public** **void** setPquants(String pquants) {

**this**.pquants = pquants;

}

**public** String getPprice() {

**return** pprice;

}

**public** **void** setPprice(String pprice) {

**this**.pprice = pprice;

}

**public** String getPname() {

**return** pname;

}

**public** **void** setPname(String pname) {

**this**.pname = pname;

}

}

REGISTRATION.JAVA

package com.example.qrcode;

import java.util.ArrayList;

import java.util.List;

import org.apache.http.HttpResponse;

import org.apache.http.NameValuePair;

import org.apache.http.client.HttpClient;

import org.apache.http.client.entity.UrlEncodedFormEntity;

import org.apache.http.client.methods.HttpPost;

import org.apache.http.impl.client.DefaultHttpClient;

import org.apache.http.message.BasicNameValuePair;

import android.app.Activity;

import android.app.ProgressDialog;

import android.content.Intent;

import android.os.Bundle;

import android.os.Handler;

import android.os.Message;

import android.util.Log;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.TextView;

import android.widget.Toast;

public class Registration extends Activity {

EditText phoneno, email, name;

Button register;

TextView loginhere;

String ph, mail, custname;

HttpClient httpClient;

HttpPost httpPost;

HttpResponse httpResponse;

private ProgressDialog pd;

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.registration);

phoneno = (EditText) findViewById(R.id.reg\_fullname);

email = (EditText) findViewById(R.id.reg\_email);

name = (EditText) findViewById(R.id.reg\_password);

loginhere = (TextView) findViewById(R.id.link\_to\_login);

}

public void Register(View v) {

ph = phoneno.getText().toString();

mail = email.getText().toString();

custname = name.getText().toString();

pd = ProgressDialog.show(Registration.this, "",

"Registering you in...", false, false);

new Thread() {

public void run() {

try {

httpClient = new DefaultHttpClient();

httpPost = new HttpPost(Ipaddress.URL);

List<NameValuePair> nameValuePair = new ArrayList<NameValuePair>();

nameValuePair.add(new BasicNameValuePair("flag", "2"));

nameValuePair.add(new BasicNameValuePair("phone", ph));

nameValuePair.add(new BasicNameValuePair("email", mail));

nameValuePair.add(new BasicNameValuePair("custname",

custname));

httpPost.setEntity(new UrlEncodedFormEntity(nameValuePair));

httpResponse = httpClient.execute(httpPost);

Log.d("Connection success", "Connection created");

handler.sendEmptyMessage(1);

}

catch (Exception e) {

// handler.sendEmptyMessage(3);

Log.d("Connection status error", "Connection" + e);

handler.sendEmptyMessage(3);

}

}

}.start();

}

private Handler handler = new Handler() {

public void handleMessage(Message msg) {

switch (msg.what) {

case 1:

pd.dismiss();

Log.i("login response", "login success");

startActivity(new Intent(Registration.this, Login.class));

break;

case 2:

pd.dismiss();

Toast.makeText(Registration.this,

"Your username and password is wrong !",

Toast.LENGTH\_LONG).show();

break;

case 3:

pd.dismiss();

Toast.makeText(Registration.this,

"Please check your internet connection or URL!",

Toast.LENGTH\_LONG).show();

break;

}

}

};

}

CONNECTION MANAGER.JAVA

package com.child.dao;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

public class ConnectionManager {

static Connection conn = null;

static String url = "jdbc:mysql://localhost:3306/qrcode";

public static Connection getConnection() {

try {

Class.forName("com.mysql.jdbc.Driver");

String username = "root";

String password = "root";

conn = DriverManager.getConnection(url, username, password);

} catch (ClassNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (SQLException e) {

e.printStackTrace();

}

return conn;

}

}

UPDATE DB.JAVA

package com.child.dao;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import org.json.JSONArray;

import org.json.JSONException;

import org.json.JSONObject;

public class UpdateDB {

Connection conn = ConnectionManager.getConnection();

int l;

public void updateLoc(String productname, String productquantity,

String productprice, String productbillno, String email, String paymenttype, String status) {

try {

Statement st = conn.createStatement();

st.execute("INSERT INTO productdetails(productname,productquantity,productprice,Billno,Emailid,PaymentType,Status) values('"

+ productname + "','"

+ productquantity + "','" + productprice

+ "','" + productbillno

+ "','" + email

+ "','" + paymenttype

+ "','" + status

+ "')");

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

public boolean login(String email, String phone) {

boolean success = false;

try {

Statement st = conn.createStatement();

ResultSet rs = st

.executeQuery("SELECT customerphone FROM registration WHERE customeremail = '"

+ email + "'");

while (rs.next()) {

if (rs.getString("customerphone").equals(phone)) {

success = true;

} else {

success = false;

}

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return success;

}

public void Registration(String customerphone, String customeremail,

String customername) {

try {

Statement st = conn.createStatement();

st.execute("INSERT INTO registration(customerphone,customeremail,customername) values('"

+ customerphone

+ "','"

+ customeremail

+ "','"

+ customername + "')");

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

PARAMETER SERVLET.JAVA

package com.child.protect;

import com.child.dao.UpdateDB;

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

/\*\*

\* Servlet implementation class ParameterServlet

\*/

public class ParameterServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

UpdateDB updb = new UpdateDB();

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public ParameterServlet() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse

\* response)

\*/

protected void doGet(HttpServletRequest request,

HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

}

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse

\* response)

\*/

protected void doPost(HttpServletRequest request,

HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

String flag = request.getParameter("flag").toString();

int flag\_value = Integer.parseInt(flag);

if (flag\_value == 0) {

// for co-ordinates

System.out.println(request.getParameter("name").toString());

String productname = request.getParameter("name").toString();

System.out.println(request.getParameter("quantity").toString());

String productquantity = request.getParameter("quantity")

.toString();

System.out.println(request.getParameter("price").toString());

String productprice = request.getParameter("price").toString();

System.out.println(request.getParameter("Billno").toString());

String productbillno = request.getParameter("Billno").toString();

System.out.println(request.getParameter("email").toString());

String useremail = request.getParameter("email").toString();

String paymenttype = request.getParameter("paymenttype").toString();

String status = request.getParameter("status").toString();

updb.updateLoc(productname, productquantity, productprice,

productbillno, useremail,paymenttype,status);

} else if (flag\_value == 2) {

System.out.println(request.getParameter("phone").toString());

String customerphone = request.getParameter("phone").toString();

System.out.println(request.getParameter("email").toString());

String customeremail = request.getParameter("email").toString();

System.out.println(request.getParameter("custname").toString());

String customerpassword = request.getParameter("custname").toString();

updb.Registration(customerphone, customeremail, customerpassword);

} else if (flag\_value == 3) {

if (updb.login(request.getParameter("Email").toString(), request

.getParameter("Phoneno").toString())) {

response.getWriter().write("yes");

} else {

response.getWriter().write("no");

}

}

}

}