SOURCE CODE FOR THE SYSTEM

Home Screen

|  |  |
| --- | --- |
| packagecom.example.hp.ecommerce; |  |
|  |  |
|  | importandroid.content.Intent; |
|  | importandroid.os.Bundle; |
|  | importandroid.support.annotation.NonNull; |
|  | importandroid.support.design.widget.FloatingActionButton; |
|  | importandroid.support.design.widget.Snackbar; |
|  | importandroid.support.v7.widget.LinearLayoutManager; |
|  | importandroid.support.v7.widget.RecyclerView; |
|  | importandroid.util.Log; |
|  | importandroid.view.LayoutInflater; |
|  | importandroid.view.View; |
|  | importandroid.support.design.widget.NavigationView; |
|  | importandroid.support.v4.view.GravityCompat; |
|  | importandroid.support.v4.widget.DrawerLayout; |
|  | importandroid.support.v7.app.ActionBarDrawerToggle; |
|  | importandroid.support.v7.app.AppCompatActivity; |
|  | importandroid.support.v7.widget.Toolbar; |
|  | importandroid.view.Menu; |
|  | importandroid.view.MenuItem; |
|  | importandroid.view.ViewGroup; |
|  | importandroid.widget.TextView; |
|  |  |
|  | importcom.example.hp.ecommerce.ViewHolder.ProductViewHolder; |
|  | importcom.example.hp.ecommerce.Model.Products; |
|  | importcom.example.hp.ecommerce.Prevalent.Prevalent; |
|  | importcom.firebase.ui.database.FirebaseRecyclerAdapter; |
|  | importcom.firebase.ui.database.FirebaseRecyclerOptions; |
|  | importcom.google.firebase.database.DatabaseReference; |
|  | importcom.google.firebase.database.FirebaseDatabase; |
|  | importcom.squareup.picasso.Picasso; |
|  |  |
|  | importde.hdodenhof.circleimageview.CircleImageView; |
|  | importio.paperdb.Paper; |
|  |  |
|  | publicclassHomeextendsAppCompatActivity |
|  | implementsNavigationView.OnNavigationItemSelectedListener |
|  | { |
|  |  |
|  | privateDatabaseReferenceProductsRef; |
|  | privateRecyclerViewrecyclerView; |
|  | RecyclerView.LayoutManagerlayoutManager; |
|  |  |
|  | @Override |
|  | protectedvoidonCreate(BundlesavedInstanceState) { |
|  | super.onCreate(savedInstanceState); |
|  | setContentView(R.layout.activity\_home); |
|  |  |
|  | ProductsRef=FirebaseDatabase.getInstance().getReference().child("Products"); |
|  | Log.d("----prodRef",ProductsRef.toString()); |
|  |  |
|  | Paper.init(this); |
|  |  |
|  | Toolbartoolbar =findViewById(R.id.toolbar); |
|  | toolbar.setTitle("Home"); |
|  | setSupportActionBar(toolbar); |
|  |  |
|  |  |
|  | FloatingActionButtonfab =findViewById(R.id.fab); |
|  | fab.setOnClickListener(newView.OnClickListener() { |
|  | @Override |
|  | publicvoidonClick(Viewview) |
|  | { |
|  | Intentintent =newIntent(Home.this,CartActivity.class); |
|  | startActivity(intent); |
|  | } |
|  | }); |
|  |  |
|  | DrawerLayoutdrawer =findViewById(R.id.drawer\_layout); |
|  | ActionBarDrawerToggletoggle =newActionBarDrawerToggle( |
|  | this, drawer, toolbar, R.string.navigation\_drawer\_open, R.string.navigation\_drawer\_close); |
|  | drawer.addDrawerListener(toggle); |
|  | toggle.syncState(); |
|  |  |
|  |  |
|  | NavigationViewnavigationView =findViewById(R.id.nav\_view); |
|  | navigationView.setNavigationItemSelectedListener(this); |
|  |  |
|  | ViewheaderView =navigationView.getHeaderView(0); |
|  | TextViewuserNameTextView =headerView.findViewById(R.id.user\_profile\_name); |
|  | CircleImageViewprofileImageView =headerView.findViewById(R.id.user\_profile\_image); |
|  |  |
|  | userNameTextView.setText(Prevalent.currentonlineUser.getName()); |
|  | Picasso.get().load(Prevalent.currentonlineUser.getImage()).placeholder(R.drawable.profile).into(profileImageView); |
|  |  |
|  | recyclerView =findViewById(R.id.recycler\_menu); |
|  | recyclerView.setHasFixedSize(true); |
|  | layoutManager =newLinearLayoutManager(this); |
|  | recyclerView.setLayoutManager(layoutManager); |
|  |  |
|  | } |
|  |  |
|  | @Override |
|  | protectedvoidonStart() |
|  | { |
|  | super.onStart(); |
|  |  |
|  | FirebaseRecyclerOptions<Products>options = |
|  | newFirebaseRecyclerOptions.Builder<Products>() |
|  | .setQuery(ProductsRef, Products.class) |
|  | .build(); |
|  |  |
|  | FirebaseRecyclerAdapter<Products, ProductViewHolder>adapter = |
|  | newFirebaseRecyclerAdapter<Products, ProductViewHolder>(options) { |
|  | @Override |
|  | protectedvoidonBindViewHolder(@NonNullProductViewHolderholder, intposition, @NonNullfinalProductsmodel) |
|  | { |
|  |  |
|  | holder.txtProductName.setText(model.getPname()); |
|  | holder.txtProductDescription.setText(model.getDescription()); |
|  | holder.txtProductPrice.setText("Price: "+model.getPrice()+"$"); |
|  | Picasso.get().load(model.getImage()).into(holder.imageView); |
|  |  |
|  | holder.itemView.setOnClickListener(newView.OnClickListener() { |
|  | @Override |
|  | publicvoidonClick(Viewv) { |
|  | Intentintent =newIntent(Home.this, ProductDetailsActivity.class); |
|  | intent.putExtra("pid",model.getPid()); |
|  | startActivity(intent); |
|  | } |
|  | }); |
|  |  |
|  | } |
|  |  |
|  | @NonNull |
|  | @Override |
|  | publicProductViewHolderonCreateViewHolder(@NonNullViewGroupparent, inti) |
|  | { |
|  | Viewview =LayoutInflater.from(parent.getContext()).inflate(R.layout.product\_items\_layout,parent,false); |
|  | ProductViewHolderholder =newProductViewHolder(view); |
|  | returnholder; |
|  | } |
|  | }; |
|  | recyclerView.setAdapter(adapter); |
|  | adapter.startListening(); |
|  |  |
|  | } |
|  |  |
|  | @Override |
|  | publicvoidonBackPressed() { |
|  | DrawerLayoutdrawer =findViewById(R.id.drawer\_layout); |
|  | if(drawer.isDrawerOpen(GravityCompat.START)) { |
|  | drawer.closeDrawer(GravityCompat.START); |
|  | } else{ |
|  | super.onBackPressed(); |
|  | } |
|  | } |
|  |  |
|  | @Override |
|  | publicbooleanonCreateOptionsMenu(Menumenu) { |
|  | //Inflate the menu; this adds items to the action bar if it is present. |
|  | getMenuInflater().inflate(R.menu.home, menu); |
|  | returntrue; |
|  | } |
|  |  |
|  | @Override |
|  | publicbooleanonOptionsItemSelected(MenuItemitem) { |
|  | //Handle action bar item clicks here. The action bar will |
|  |  |
|  | intid =item.getItemId(); |
|  |  |
|  |  |
|  | returnsuper.onOptionsItemSelected(item); |
|  | } |
|  |  |
|  | @SuppressWarnings("StatementWithEmptyBody") |
|  | @Override |
|  | publicbooleanonNavigationItemSelected(MenuItemitem) { |
|  | //Handle navigation view item clicks here. |
|  | intid =item.getItemId(); |
|  |  |
|  | if(id ==R.id.nav\_cart) |
|  | { |
|  | Intentintent =newIntent(Home.this,CartActivity.class); |
|  | startActivity(intent); |
|  | } |
|  | elseif(id ==R.id.nav\_search) |
|  | { |
|  | Intentintent =newIntent(Home.this,SearchProductsActivity.class); |
|  | startActivity(intent); |
|  | } |
|  |  |
|  | elseif(id ==R.id.nav\_settings) |
|  | { |
|  | Intentintent =newIntent(Home.this, SettingsActivity.class); |
|  | startActivity(intent); |
|  | } |
|  | elseif(id ==R.id.nav\_logout) |
|  | { |
|  | Paper.book().destroy(); |
|  | Intentintent =newIntent(Home.this,MainActivity.class); |
|  | intent.addFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK|Intent.FLAG\_ACTIVITY\_CLEAR\_TASK); |
|  | startActivity(intent); |
|  | } |
|  |  |
|  | DrawerLayoutdrawer =(DrawerLayout) findViewById(R.id.drawer\_layout); |
|  | drawer.closeDrawer(GravityCompat.START); |
|  | returntrue; |
|  | } |
|  | } |

Signing Up a User

|  |  |
| --- | --- |
| publicclassUsers{ |  |
|  |  |
|  | privateStringname,phone,password,image, address; |
|  |  |
|  | publicUsers(){} |
|  |  |
|  |  |
|  | publicUsers(Stringname, Stringphone, Stringpassword, Stringimage, Stringaddress) { |
|  | this.name =name; |
|  | this.phone =phone; |
|  | this.password =password; |
|  | this.image =image; |
|  | this.address =address; |
|  | } |
|  |  |
|  | publicStringgetName() { |
|  | returnname; |
|  | } |
|  |  |
|  | publicvoidsetName(Stringname) { |
|  | this.name =name; |
|  | } |
|  |  |
|  | publicStringgetPhone() { |
|  | returnphone; |
|  | } |
|  |  |
|  | publicvoidsetPhone(Stringphone) { |
|  | this.phone =phone; |
|  | } |
|  |  |
|  | publicStringgetPassword() { |
|  | returnpassword; |
|  | } |
|  |  |
|  | publicvoidsetPassword(Stringpassword) { |
|  | this.password =password; |
|  | } |
|  |  |
|  | publicStringgetImage() { |
|  | returnimage; |
|  | } |
|  |  |
|  | publicvoidsetImage(Stringimage) { |
|  | this.image =image; |
|  | } |
|  |  |
|  | publicStringgetAddress() { |
|  | returnaddress; |
|  | } |
|  |  |
|  | publicvoidsetAddress(Stringaddress) { |
|  | this.address =address; |
|  | } |
|  | } |

Signing In By the User

|  |  |
| --- | --- |
| importandroid.app.ProgressDialog; |  |
|  | importandroid.content.Intent; |
|  | importandroid.support.annotation.NonNull; |
|  | importandroid.support.v7.app.AppCompatActivity; |
|  | importandroid.os.Bundle; |
|  | importandroid.text.TextUtils; |
|  | importandroid.view.View; |
|  | importandroid.widget.Button; |
|  | importandroid.widget.Toast; |
|  |  |
|  | importcom.example.hp.ecommerce.Model.Users; |
|  | importcom.example.hp.ecommerce.Prevalent.Prevalent; |
|  | importcom.google.firebase.database.DataSnapshot; |
|  | importcom.google.firebase.database.DatabaseError; |
|  | importcom.google.firebase.database.DatabaseReference; |
|  | importcom.google.firebase.database.FirebaseDatabase; |
|  | importcom.google.firebase.database.ValueEventListener; |
|  |  |
|  | importio.paperdb.Paper; |
|  |  |
|  | publicclassMainActivityextendsAppCompatActivity{ |
|  |  |
|  | privateButtonjoinnowBtn, loginbtn; |
|  | ProgressDialogprogressDialog; |
|  |  |
|  | @Override |
|  | protectedvoidonCreate(BundlesavedInstanceState) { |
|  | super.onCreate(savedInstanceState); |
|  | setContentView(R.layout.activity\_main); |
|  |  |
|  | joinnowBtn =findViewById(R.id.main\_join\_now\_btn); |
|  | loginbtn =findViewById(R.id.main\_login\_btn); |
|  | Paper.init(this); |
|  | progressDialog =newProgressDialog(this); |
|  |  |
|  |  |
|  | loginbtn.setOnClickListener(newView.OnClickListener() { |
|  | @Override |
|  | publicvoidonClick(Viewv) { |
|  | Intentintent =newIntent(MainActivity.this,LoginActivity.class); |
|  | startActivity(intent); |
|  | } |
|  | }); |
|  |  |
|  |  |
|  | joinnowBtn.setOnClickListener(newView.OnClickListener() { |
|  | @Override |
|  | publicvoidonClick(Viewv) { |
|  | Intentintent =newIntent(MainActivity.this,RegisterActivity.class); |
|  | startActivity(intent); |
|  | } |
|  | }); |
|  |  |
|  | StringUserPhoneKey=Paper.book().read(Prevalent.UserPhoneKey); |
|  | StringUserPasswordKey=Paper.book().read(Prevalent.UserPasswordKey); |
|  | if(UserPhoneKey!=""&&UserPasswordKey!="") |
|  | { |
|  | if(!TextUtils.isEmpty(UserPhoneKey) &&!TextUtils.isEmpty(UserPasswordKey)) |
|  | { |
|  | AllowAccess(UserPhoneKey,UserPasswordKey); |
|  | progressDialog.setTitle("Already Logged In"); |
|  | progressDialog.setMessage("Please Wait.."); |
|  | progressDialog.setCanceledOnTouchOutside(false); |
|  | progressDialog.show(); |
|  |  |
|  | } |
|  | } |
|  |  |
|  |  |
|  | } |
|  |  |
|  | privatevoidAllowAccess(finalStringphone,finalStringpassword) |
|  | { |
|  | finalDatabaseReferenceRootRef; |
|  | finalStringparentdbName ="Users"; |
|  | RootRef=FirebaseDatabase.getInstance().getReference(); |
|  | RootRef.addListenerForSingleValueEvent(newValueEventListener() { |
|  | @Override |
|  | publicvoidonDataChange(@NonNullDataSnapshotdataSnapshot) { |
|  | if(dataSnapshot.child(parentdbName).child(phone).exists()) |
|  | { |
|  | UsersuserData =dataSnapshot.child(parentdbName).child(phone).getValue(Users.class); |
|  | if(userData.getPhone().equals(phone)) |
|  | { |
|  | if(userData.getPassword().equals(password)) |
|  | { |
|  | Toast.makeText(MainActivity.this, "Already Logged in", Toast.LENGTH\_SHORT).show(); |
|  | progressDialog.dismiss(); |
|  | Intentintent =newIntent(MainActivity.this, Home.class); |
|  | Prevalent.currentonlineUser =userData; |
|  | startActivity(intent); |
|  | } else{ |
|  | Toast.makeText(MainActivity.this, "Password is Incorrect", Toast.LENGTH\_SHORT).show(); |
|  | progressDialog.dismiss(); |
|  | } |
|  | } |
|  |  |
|  | } else{ |
|  | progressDialog.dismiss(); |
|  | Toast.makeText(MainActivity.this, "Login Failed", Toast.LENGTH\_SHORT).show(); |
|  |  |
|  | } |
|  | } |
|  |  |
|  | @Override |
|  | publicvoidonCancelled(@NonNullDatabaseErrordatabaseError) { |
|  |  |
|  | } |
|  | }); |
|  | } |
|  | } |

Confirming Order

|  |  |
| --- | --- |
| importandroid.content.Intent; |  |
|  | importandroid.support.annotation.NonNull; |
|  | importandroid.support.v7.app.AppCompatActivity; |
|  | importandroid.os.Bundle; |
|  | importandroid.text.TextUtils; |
|  | importandroid.util.Log; |
|  | importandroid.view.View; |
|  | importandroid.view.ViewGroup; |
|  | importandroid.widget.Button; |
|  | importandroid.widget.EditText; |
|  | importandroid.widget.TextView; |
|  | importandroid.widget.Toast; |
|  |  |
|  | importcom.example.hp.ecommerce.Model.Cart; |
|  | importcom.example.hp.ecommerce.Model.Products; |
|  | importcom.example.hp.ecommerce.Prevalent.Prevalent; |
|  | importcom.example.hp.ecommerce.ViewHolder.CartViewHolder; |
|  | importcom.firebase.ui.database.FirebaseRecyclerAdapter; |
|  | importcom.firebase.ui.database.FirebaseRecyclerOptions; |
|  | importcom.google.android.gms.tasks.OnCompleteListener; |
|  | importcom.google.android.gms.tasks.Task; |
|  | importcom.google.firebase.database.DataSnapshot; |
|  | importcom.google.firebase.database.DatabaseError; |
|  | importcom.google.firebase.database.DatabaseReference; |
|  | importcom.google.firebase.database.FirebaseDatabase; |
|  | importcom.google.firebase.database.ValueEventListener; |
|  |  |
|  | importjava.text.SimpleDateFormat; |
|  | importjava.util.ArrayList; |
|  | importjava.util.Calendar; |
|  | importjava.util.HashMap; |
|  |  |
|  | import staticandroid.provider.Contacts.SettingsColumns.KEY; |
|  |  |
|  | publicclassConfirmFinalOrderActivityextendsAppCompatActivity{ |
|  |  |
|  | privateEditTextnameEt, phoneEt, addressEt, cityEt; |
|  | privateButtonconfirmOrderBtn; |
|  | privateStringtotalPrice =null; |
|  | privateTextViewamt; |
|  | privateDatabaseReferencedatabaseReference; |
|  | privateDatabaseReferenceorderRef; |
|  | privateStringTAG=getClass().getSimpleName(); |
|  |  |
|  | @Override |
|  | protectedvoidonCreate(BundlesavedInstanceState) { |
|  | super.onCreate(savedInstanceState); |
|  | setContentView(R.layout.activity\_confirm\_final\_order); |
|  |  |
|  | confirmOrderBtn =findViewById(R.id.confirm\_final\_order\_btn); |
|  | amt=findViewById(R.id.amt); |
|  |  |
|  | totalPrice =getIntent().getStringExtra("Total Price"); |
|  | amt.setText(totalPrice+"₹"); |
|  | Toast.makeText(this, "Total Price: "+totalPrice+"₹", Toast.LENGTH\_SHORT).show(); |
|  |  |
|  | confirmOrderBtn.setOnClickListener(newView.OnClickListener() { |
|  | @Override |
|  | publicvoidonClick(Viewv) |
|  | { |
|  | confirmOrder(); |
|  | } |
|  | }); |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  | } |
|  |  |
|  |  |
|  | privatevoidconfirmOrder() |
|  | { |
|  | databaseReference =FirebaseDatabase.getInstance().getReference(); |
|  | orderRef =FirebaseDatabase.getInstance().getReference() |
|  | .child("Orders").child(Prevalent.currentonlineUser.getPhone()); |
|  |  |
|  | finalDatabaseReferenceadminViewRef =FirebaseDatabase.getInstance().getReference() |
|  | .child("Cart List").child("Admin View").child(Prevalent.currentonlineUser.getPhone()); |
|  |  |
|  | finalHashMap<String, Object>orderMap =newHashMap<>(); |
|  |  |
|  | databaseReference.child("Cart List").child("Admin view").child(Prevalent.currentonlineUser |
|  | .getPhone()).child("Products").addValueEventListener(newValueEventListener() { |
|  | @Override |
|  | publicvoidonDataChange(DataSnapshotdataSnapshot) { |
|  | Iterableiterable =dataSnapshot.getChildren(); |
|  | ArrayList<Cart>cart =newArrayList<>(); |
|  | while(iterable.iterator().hasNext()){ |
|  | DataSnapshotdataSnapshot1 =(DataSnapshot) iterable.iterator().next(); |
|  | cart.add(dataSnapshot1.getValue(Cart.class)); |
|  | } |
|  |  |
|  | for(inti =0; i <cart.size(); i++){ |
|  | Log.e("Product", cart.get(i).getPid()+"-"+cart.get(i).getPname()+"-"+cart.get(i).getQuantity()); |
|  | orderMap.put("pid",cart.get(i).getPid()); |
|  | orderMap.put("pname",cart.get(i).getPname()); |
|  | orderMap.put("quantity",cart.get(i).getQuantity()); |
|  | orderMap.put("price",cart.get(i).getPrice()); |
|  |  |
|  | orderRef.child(cart.get(i).getPid()).updateChildren(orderMap) |
|  | .addOnCompleteListener(newOnCompleteListener<Void>() { |
|  | @Override |
|  | publicvoidonComplete(@NonNullTask<Void> task) { |
|  | if(task.isSuccessful()) |
|  | { |
|  | //if order confirmed empty the cart |
|  | FirebaseDatabase.getInstance().getReference().child("Cart List") |
|  | .child("User view") |
|  | .child(Prevalent.currentonlineUser.getPhone()) |
|  | .removeValue() |
|  | .addOnCompleteListener(newOnCompleteListener<Void>() { |
|  | @Override |
|  | publicvoidonComplete(@NonNullTask<Void> task) |
|  | { |
|  | if(task.isSuccessful()) |
|  | { |
|  | Toast.makeText(ConfirmFinalOrderActivity.this, |
|  | "Final Order Has Been Placed", Toast.LENGTH\_SHORT).show(); |
|  | Intentintent =newIntent(ConfirmFinalOrderActivity.this,Home.class); |
|  | intent.addFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK|Intent.FLAG\_ACTIVITY\_CLEAR\_TASK); |
|  | startActivity(intent); |
|  | finish(); |
|  | } |
|  | } |
|  | }); |
|  |  |
|  | } |
|  |  |
|  | } |
|  | }); |
|  |  |
|  | } |
|  | finalHashMap<String, Object>tPriceMap =newHashMap<>(); |
|  | tPriceMap.put("total\_price",totalPrice); |
|  | orderRef.updateChildren(tPriceMap); |
|  |  |
|  |  |
|  | } |
|  |  |
|  | @Override |
|  | publicvoidonCancelled(DatabaseErrordatabaseError) { |
|  |  |
|  | } |
|  | }); |
|  |  |
|  | /\*DatabaseReference orderRef = FirebaseDatabase.getInstance().getReference() |
|  | .child("Orders") |
|  | .child(Prevalent.currentonlineUser.getPhone()); |
|  | DatabaseReference adminViewRef = FirebaseDatabase.getInstance().getReference() |
|  | .child("Cart List").child("Admin View") |
|  | .child(Prevalent.currentonlineUser.getPhone()); |
|  |  |
|  | HashMap<String, Object> orderMap = new HashMap <>(); |
|  | orderMap.put("totalPrice",totalPrice); |
|  | orderMap.put("name",nameEt.getText().toString()); |
|  | orderMap.put("phone",phoneEt.getText().toString()); |
|  | orderMap.put("address",addressEt.getText().toString()); |
|  | orderMap.put("city",cityEt.getText().toString()); |
|  | orderMap.put("date",saveCurrDate); |
|  | orderMap.put("time",saveCurrTime); |
|  | orderMap.put("state","not shipped"); |
|  |  |
|  | orderRef.updateChildren(orderMap).addOnCompleteListener(new OnCompleteListener <Void>() { |
|  | @Override |
|  | public void onComplete(@NonNull Task<Void> task) |
|  | { |
|  | if(task.isSuccessful()) |
|  | { |
|  | //if order confirmed empty the cart |
|  | FirebaseDatabase.getInstance().getReference().child("Cart List") |
|  | .child("User view") |
|  | .child(Prevalent.currentonlineUser.getPhone()) |
|  | .removeValue() |
|  | .addOnCompleteListener(new OnCompleteListener <Void>() { |
|  | @Override |
|  | public void onComplete(@NonNull Task <Void> task) |
|  | { |
|  | if(task.isSuccessful()) |
|  | { |
|  | Toast.makeText(ConfirmFinalOrderActivity.this, |
|  | "Final Order Has Been Placed", Toast.LENGTH\_SHORT).show(); |
|  | Intent intent = new Intent(ConfirmFinalOrderActivity.this,Home.class); |
|  | intent.addFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK | Intent.FLAG\_ACTIVITY\_CLEAR\_TASK); |
|  | startActivity(intent); |
|  | finish(); |
|  | } |
|  | } |
|  | }); |
|  | } |
|  | } |
|  | });\*/ |
|  | }  PAYROLL SYSTEM  import javafx.application.Application;  import javafx.fxml.FXMLLoader;  import javafx.scene.Parent;  import javafx.scene.Scene;  import javafx.stage.Stage;  import java.sql.Connection;  import java.sql.DriverManager;  import java.sql.SQLException;  import javafx.scene.control.Alert;  public class Payroll extends Application {    static Connection conn = null;    @Override  public void start(Stage stage) throws Exception {    try {  Class.forName("com.mysql.jdbc.Driver");    String url = "jdbc:mysql://localhost:3306/payroll";  String username = "root";  String password = "";  try {  conn = DriverManager.getConnection(url, username, password);  } catch (SQLException e) {  Alert alert = new Alert(Alert.AlertType.ERROR);  alert.setTitle("Database");  alert.setHeaderText("Connection Error");  alert.setContentText("Cannot Connect To Database!");  alert.showAndWait();  throw new IllegalStateException("Cannot connect the database!", e);  }    } catch (ClassNotFoundException e) {  throw new IllegalStateException("Cannot find the driver in the classpath!", e);  }    Parent root = FXMLLoader.load(getClass().getResource("Payroll.fxml"));  Scene scene = new Scene(root);  stage.setScene(scene);  stage.setTitle("Payroll System");  stage.show();  }  public static void main(String[] args) {  launch(args);  }  }  PAYROLL TABLE  import javafx.beans.property.SimpleStringProperty;  import javafx.beans.property.StringProperty;  import static payroll.PayrollController.round;  public class PayrollTable {  public final SimpleStringProperty tc\_name, tc\_id, tc\_gross, tc\_basic, tc\_house\_rent, tc\_medical,  tc\_per\_day, tc\_per\_hour, tc\_over\_time, tc\_over\_time\_pay, tc\_payable;  PayrollTable(String id, String name, String gross, String overtime){  this.tc\_id = new SimpleStringProperty(id);  this.tc\_name = new SimpleStringProperty(name);  this.tc\_gross = new SimpleStringProperty(gross);  this.tc\_over\_time = new SimpleStringProperty(overtime);  double basic = (Double.parseDouble(tc\_gross.get()) / 100) \* 60; // Basic = 60% of Gross  basic = round(basic, 2);  this.tc\_basic = new SimpleStringProperty(String.valueOf(basic));  double house\_rent = (Double.parseDouble(tc\_gross.get()) / 100) \* 30; // House Rent = 30% of Gross  house\_rent = round(house\_rent, 2);  this.tc\_house\_rent = new SimpleStringProperty(String.valueOf(house\_rent));  double medical = (Double.parseDouble(tc\_gross.get()) / 100) \* 10; // Medical = 10% of Gross  medical = round(medical, 2);  this.tc\_medical = new SimpleStringProperty(String.valueOf(medical));  double per\_day = basic / 26; // Per Day = Basic / 26  per\_day = round(per\_day, 2);  this.tc\_per\_day = new SimpleStringProperty(String.valueOf(per\_day));  double per\_hour = per\_day / 8; // Per Hour = Per Day / 8  per\_hour = round(per\_hour, 2);  this.tc\_per\_hour = new SimpleStringProperty(String.valueOf(per\_hour));  double over\_time\_pay = Double.parseDouble(tc\_over\_time.get()) \* per\_hour \* 2; // Over Time Pay = Over Time \* Per Hour Pay \* 2  over\_time\_pay = round(over\_time\_pay, 2);  this.tc\_over\_time\_pay = new SimpleStringProperty(String.valueOf(over\_time\_pay));  double payable = basic + over\_time\_pay; // Payable = Basic + Over Time Pay  payable = round(payable, 2);  this.tc\_payable = new SimpleStringProperty(String.valueOf(payable));  }    public final String getId() { return tc\_id.getValue(); }  public final String getName() { return tc\_name.getValue(); }  public final String getGross() { return tc\_gross.getValue(); }  public final String getOverTime() { return tc\_over\_time.getValue(); }    public final StringProperty idProperty() { return tc\_id; }  public final StringProperty nameProperty() { return tc\_name; }  public final StringProperty grossProperty() { return tc\_gross; }  public final StringProperty basicProperty() { return tc\_basic; }  public final StringProperty house\_rentProperty() { return tc\_house\_rent; }  public final StringProperty medicalProperty() { return tc\_medical; }  public final StringProperty per\_dayProperty() { return tc\_per\_day; }  public final StringProperty per\_hourProperty() { return tc\_per\_hour; }  public final StringProperty over\_timeProperty() { return tc\_over\_time; }  public final StringProperty over\_time\_payProperty() { return tc\_over\_time\_pay; }  public final StringProperty payableProperty() { return tc\_payable; }  } |
|  |  |