```
public class Ecommerce {

import javafx.application.Application;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.layout.GridPane;
import javafx.scene.layout.Pane;
import javafx.stage.Stage;
import java.io.IOException;

public class Ecommerce extends Application {
    private final int height = 500, width = 400, headerLine = 50;
    ProductList productList = new ProductList();
    Pane bodyPane;
    Customer loggedInCustomer = null;
    Button signInButton = new Button("Sign In");
    Label welcomeLabel = new Label("Welcome Customer");

    private GridPane headerBar() {
        GridPane header = new GridPane();
        TextField searchBar = new TextField();
        Button searchButton = new Button("Search");

        searchButton.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent actionEvent) {
                bodyPane.getChildren().clear();
                bodyPane.getChildren().add(productList.getAllProducts());
            }
        });

        signInButton.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent actionEvent) {
                bodyPane.getChildren().clear();
                bodyPane.getChildren().add(loginPage());
            }
        });
        header.setHgap(10);
        header.add(searchBar, 0, 0);
        header.add(searchButton, 1, 0);
        header.add(signInButton, 2, 0);
        return header;
    }

    private void showMessage(String message) {
        Dialog<String> dialog = new Dialog<String>();
        // Setting the title
        dialog.setTitle("Orders");
        ButtonType type = new ButtonType("Ok", ButtonBar.ButtonData.OK_DONE);
        // Setting the content of the dialog
        dialog.setContentText(message);
        // Adding buttons to the dialog pane
        dialog.getDialogPane().getButtonTypes().add(type);
        dialog.showAndWait();
    }

    private GridPane loginPage() {
        Label userLabel = new Label("User Name");
        Label passLabel = new Label("Password");
        TextField userName = new TextField();
        userName.setPromptText("Enter User Name");
        PasswordField password = new PasswordField();
        password.setPromptText("Enter password");
        Button loginButton = new Button("Login");
        Label messageLabel = new Label("Please Login");
        loginButton.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent actionEvent) {
                String user = userName.getText();
                String pass = password.getText();
                loggedInCustomer = Login.customerLogin(user, pass);
                if (loggedInCustomer != null) {
                    welcomeLabel.setText("Welcome" + loggedInCustomer.getName());
                    messageLabel.setText("Login Successfull!!");
                } else {
                    messageLabel.setText("Login Failed");
                }
            }
        });
        GridPane loginPane = new GridPane();
        loginPane.setHgap(10);
        loginPane.setVgap(10);
        loginPane.setTranslateX(10);
        loginPane.add(userLabel, 0, 0);
        loginPane.add(userName, 1, 0);
        loginPane.add(passLabel, 0, 1);
        loginPane.add(password, 1, 1);
        loginPane.add(loginButton, 0, 2);
        loginPane.add(messageLabel, 1, 2);
        return loginPane;
    }

    private GridPane footerBar() {
        Button buyNowButton = new Button("Buy Now");
        buyNowButton.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent actionEvent) {
                Product product = productList.getSelectedProduct();
                boolean orderStatus = false;
                if (product != null && loggedInCustomer != null) {
                    orderStatus = Order.placeOrder(loggedInCustomer, product);
                }
                if (orderStatus == true) {
                    showMessage("Order placed");
                } else {

                }

            }
        });
        GridPane footer = new GridPane();
        footer.setTranslateY(headerLine + height);
        footer.add(buyNowButton, 0, 1);

        return footer;
    }

    private Pane createContent() {
        Pane root = new Pane();
        root.setPrefSize(width, 2 * headerLine + height);

        bodyPane = new Pane();
        bodyPane.setPrefSize(width, height);
        bodyPane.setTranslateX(50);
        bodyPane.setTranslateY(headerLine);
        bodyPane.getChildren().add(loginPage());
        root.getChildren().addAll(headerBar(),
                // loginPage(),
                // productList.getAllProducts()
                bodyPane, footerBar());
        return root;
    }
    @Override
    public void start(Stage stage) throws IOException {
        // FXMLLoader fxmlLoader = new
        // FXMLLoader(HelloApplication.class.getResource("hello-view.fxml"));
        Scene scene = new Scene(createContent());
        stage.setTitle("Ecommerce!");
        stage.setScene(scene);
        stage.show();
    }

    public static void main(String[] args) {
        launch();
    }
}}

public class Customer{
package com.example.demo;

public class Customer {
    int id;
    String name;

    public Customer(int id, String name, String email) {
        this.id = id;
        this.name = name;
        this.email = email;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    String email;

}

}

public class Product {
    package com.example.demo;

import javafx.beans.property.SimpleDoubleProperty;
import javafx.beans.property.SimpleIntegerProperty;
import javafx.beans.property.SimpleStringProperty;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;

import java.sql.ResultSet;
```

```
public class Product {
    private SimpleIntegerProperty id;

    private SimpleStringProperty name;
    private SimpleDoubleProperty price;

    public int getId() {
        return id.get();
    }

    public String getName() {
        return name.get();
    }

    public Double getPrice() {
        return price.get();
    }

    public Product(int _id, String _name, Double _price) {
        this.id = new SimpleIntegerProperty(_id);
        this.name = new SimpleStringProperty(_name);
        this.price = new SimpleDoubleProperty(_price);
    }

    public static ObservableList<Product> getAllProducts() {
        String query = "SELECT * from product";
        return getProducts(query);
    }

    public static ObservableList<Product> getProducts(String query) {
        DataBaseConnection dbConn = new DataBaseConnection();
        ResultSet rs = dbConn.getQueryTable(query);
        ObservableList<Product> result = FXCollections.observableArrayList();
        try {
            if (rs != null) {
                while (rs.next()) {
                    result.add(new Product(rs.getInt("product_id"), rs.getString("product_name"),
                            rs.getDouble("price")));
                }
            }
        } catch (Exception e) {
            e.printStackTrace();
        }

        return result;
    }
}

}

public class ProductList {
    package com.example.demo;

import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.scene.layout.Pane;

public class ProductList {

        public TableView<Product> productTable;

        public Pane getAllProducts() {
            TableColumn id = new TableColumn("Product_ID");
            id.setCellValueFactory(new PropertyValueFactory<>("id"));

            TableColumn name = new TableColumn("Product_Name");
            name.setCellValueFactory(new PropertyValueFactory<>("name"));

            TableColumn price = new TableColumn("Product_Price");
            price.setCellValueFactory(new PropertyValueFactory<>("price"));

            ObservableList<Product> data = FXCollections.observableArrayList();
            data.addAll(new Product(1, "Laptop", 9000.0),
                    new Product(2, "Laptop_2", 9500.0));

            ObservableList<Product> productList = Product.getAllProducts();

            productTable = new TableView<>();
            productTable.setItems(productList);
            productTable.getColumns().addAll(id, name, price);

            Pane tablePane = new Pane();
            tablePane.getChildren().add(productTable);
            return tablePane;
        }

        public Product getSelectedProduct() {
            return productTable.getSelectionModel().getSelectedItem();
        }
    }

}

public class Order {

    public class Order {
        public static boolean placeOrder(Customer customer, Product product) {
            try {
                String placeOrder = "INSERT INTO orders(customer_id,product_id,status) VALUES(" +
customer.getId() + ","
                        + product.getId() + ",'Ordered')";
                DataBaseConnection dbConn = new DataBaseConnection();
                return dbConn.insertUpdate(placeOrder);
            } catch (Exception e) {
                e.printStackTrace();
            }
            return false;
        }
    }

}

public class DataBase Connection
{
package com.example.demo;
import java.sql.*;

public class DataBaseConnection {
    String dbUrl = "jdbc:mysql://localhost:3306/ecommerce";
    String username = "root";
    String password = "Voda$1796";

    private Statement getStatement() {
        try {
            Connection conn = DriverManager.getConnection(dbUrl, username, password);
            return conn.createStatement();
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    public ResultSet getQueryTable(String query) {
        Statement statement = getStatement();
        try {
            assert statement != null;
            return statement.executeQuery(query);
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return null;
    }

    public boolean insertUpdate(String query) {
        Statement statement = getStatement();
        try {
            statement.executeUpdate(query);
            return true;
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return false;
    }

    public static void main(String[] args) {
        String query = "SELECT * from product";
        DataBaseConnection dbConn = new DataBaseConnection();
        ResultSet rs = dbConn.getQueryTable(query);
        if (rs != null) {
            System.out.println("Connected to Database");
        }
    }

}

}

public class Login {
package com.example.demo;

import java.math.BigInteger;
import java.nio.charset.StandardCharsets;
import java.security.MessageDigest;
import java.sql.ResultSet;

public class Login {
    private byte[] getSha(String input) {
        try {
            MessageDigest md = MessageDigest.getInstance("SHA-256");
            return md.digest(input.getBytes(StandardCharsets.UTF_8));
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    private String getEncryptedPassword(String password) {
        try {
            BigInteger num = new BigInteger(1, getSha(password));
            StringBuilder hexString = new StringBuilder();
            return hexString.toString();
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    public static Customer customerLogin(String userEmail, String passWord) {
        // SELECT * FROM customer WHERE email='shan@gmail.com' AND password='1234';
        String query = "SELECT * FROM customer WHERE email='" + userEmail + "' and password='" +
passWord + "'";
        // System.out.println(query);
        DataBaseConnection dbConn = new DataBaseConnection();
        try {
            ResultSet rs = dbConn.getQueryTable(query);
            if (rs != null && rs.next()) {
                return new Customer(rs.getInt("customer_id"), rs.getString("name"),
rs.getString("email"));
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    public static void main(String[] args) {
        System.out.println(customerLogin("shantanu@gmail.com", "1234"));

    }
}

}
```