**ВИСОКА ТЕХНИЧКА ШКОЛА СТРУКОВНИХ СТУДИЈА У НОВОМ САДУ**

**ИСПИТНИ ПРОЈЕКАТ**

**ПРЕДМЕТ: ЈАВА ПРОГРАМИРАЊЕ**

Студент: Професор:

Павле Вукајловић МИТ18/19 Петар Васиљевић

**Питања:**

# **Пројекат**

# **Кратак опис:**

Урађена је апликација за радњу рачунара и рачунарских компоненти. База садржи две табеле, артикле (ид, назив, категорија артикла и цена) и администратори (име, лозинка, и роле). У зависности од тога који корисник се улогује, дозвољене су одређене операције над базом. Администратор може потпуно да манипулише базом, док запослени у првом прозору може само да учита податке у табелу и обрише податке из табеле, али може такође и да отвори администраторски панел кликом на дугме „измена“ јер је то тражено у пројектном задатку. Пројекат садржи логин форму, унос, преглед, измену и брисање података из TableView-a, измену и брисање после клика на дугме „измена“.

# Код:

# Administrator/AdministratorController

public class AdministratorController implements Initializable{  
  
 @FXML  
 private TextField naziv;  
 @FXML  
 private TextField kat;  
 @FXML  
 private TextField cena;  
 @FXML  
 private TableView<ArtikliData> artiklitable;  
 @FXML  
 private TableColumn<ArtikliData, String> idcolumn;  
 @FXML  
 private TableColumn<ArtikliData, String> nazivcolumn;  
 @FXML  
 private TableColumn<ArtikliData, String> katcolumn;  
 @FXML  
 private TableColumn<ArtikliData, String> cenacolumn;  
 @FXML  
 private Button del;  
  
 private dbConnection dc;  
 private ObservableList <ArtikliData> data;  
 private FXMLUtils Fx = new FXMLUtils();  
  
 private String sql = "SELECT \* FROM artikli ";  
  
 public void initialize(URL url, ResourceBundle rb){  
 this.dc = new dbConnection();  
 }  
  
 @FXML  
 private void loadArtikliData() throws SQLException {  
 Fx.loadArtikliData(artiklitable,idcolumn, nazivcolumn, katcolumn, cenacolumn, data, sql);  
 }  
 @FXML  
 private void dodajArtikl(ActionEvent event) throws SQLException{  
 Fx.dodajArtikl(naziv, kat, cena);  
 }  
 @FXML  
 private void brisiArtikl(ActionEvent event) throws SQLException {  
 Fx.brisiArtikl(artiklitable,del);  
 }  
 @FXML  
 private void izmeniArtikl(ActionEvent event) throws SQLException{  
 Fx.izmeniArtikl(artiklitable, naziv, kat, cena);  
 }  
  
 @FXML  
 private void obrisiFormu(ActionEvent event){  
 Fx.obrisiFormu(naziv, kat, cena);  
 }  
  
}

# Administrator/Admin.fxml

<?xml version="1.0" encoding="UTF-8"?>  
  
<?import javafx.geometry.\*?>  
<?import javafx.scene.control.\*?>  
<?import javafx.scene.layout.\*?>  
  
<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="672.0" prefWidth="854.0" xmlns="http://javafx.com/javafx/10.0.2-internal" xmlns:fx="http://javafx.com/fxml/1" fx:controller="Administrator.AdministratorController">  
 <children>  
 <HBox layoutX="461.0" layoutY="34.0" prefHeight="604.0" prefWidth="314.0">  
 <children>  
 <TableView fx:id="artiklitable" prefHeight="654.0" prefWidth="318.0">  
 <columns>  
 <TableColumn fx:id="idcolumn" prefWidth="75.0" text="ID" />  
 <TableColumn fx:id="nazivcolumn" prefWidth="75.0" text="NAZIV" />  
 <TableColumn fx:id="katcolumn" prefWidth="92.0" text="KATEGORIJA" />  
 <TableColumn fx:id="cenacolumn" prefWidth="71.0" text="CENA" />  
 </columns>  
 </TableView>  
 </children>  
 </HBox>  
 <VBox layoutX="53.0" layoutY="34.0" prefHeight="604.0" prefWidth="374.0">  
 <children>  
 <Label prefHeight="17.0" prefWidth="402.0" text="Unesite informacije artikla">  
 <VBox.margin>  
 <Insets />  
 </VBox.margin>  
 <padding>  
 <Insets left="50.0" />  
 </padding>  
 </Label>  
 <TextField fx:id="naziv" promptText="Naziv">  
 <VBox.margin>  
 <Insets left="15.0" right="15.0" top="15.0" />  
 </VBox.margin>  
 </TextField>  
 <TextField fx:id="kat" promptText="Kategorija">  
 <VBox.margin>  
 <Insets left="15.0" right="15.0" top="15.0" />  
 </VBox.margin>  
 </TextField>  
 <TextField fx:id="cena" promptText="Cena">  
 <VBox.margin>  
 <Insets left="15.0" right="15.0" top="15.0" />  
 </VBox.margin>  
 </TextField>  
 <Button fx:id="dodaj" mnemonicParsing="false" onAction="#dodajArtikl" prefHeight="25.0" prefWidth="314.0" text="Dodaj artikl">  
 <VBox.margin>  
 <Insets left="15.0" top="15.0" />  
 </VBox.margin>  
 </Button>  
 <Button mnemonicParsing="false" onAction="#obrisiFormu" prefHeight="25.0" prefWidth="314.0" text="Obriši podatke forme">  
 <VBox.margin>  
 <Insets left="15.0" top="15.0" />  
 </VBox.margin>  
 </Button>  
 <Button mnemonicParsing="false" onAction="#loadArtikliData" prefHeight="25.0" prefWidth="314.0" text="Učitaj podatke">  
 <VBox.margin>  
 <Insets left="15.0" top="15.0" />  
 </VBox.margin>  
 </Button>  
 <Button mnemonicParsing="false" onAction="#izmeniArtikl" prefHeight="25.0" prefWidth="314.0" text="Izmena ">  
 <VBox.margin>  
 <Insets left="15.0" top="15.0" />  
 </VBox.margin>  
 </Button>  
 <Button fx:id="del" mnemonicParsing="false" onAction="#brisiArtikl" prefHeight="25.0" prefWidth="314.0" text="Brisanje ">  
 <VBox.margin>  
 <Insets left="15.0" top="15.0" />  
 </VBox.margin>  
 </Button>  
 </children>  
 <padding>  
 <Insets left="15.0" right="15.0" />  
 </padding>  
 </VBox>  
 </children>  
</AnchorPane>

# dbUtil/dbConnection

public class dbConnection {  
 private static final String *USERNAME* = "root";  
 private static final String *PASSWORD* = "";  
 private static final String *CONN* = "jdbc:mysql://localhost/prodavnica";  
 private static final String *SCONN* = "jdbc:sqlite:login.sqlite";  
  
 public static Connection getConnection()  
 throws SQLException {  
 try {  
 Class.*forName*("com.mysql.cj.jdbc.Driver");  
  
 return DriverManager.*getConnection*(*CONN*,*USERNAME*, *PASSWORD*);  
  
 } catch (ClassNotFoundException | SQLException ex) {  
 ex.printStackTrace();  
 }  
 return null;  
 }

# Login/LoginController

public class LoginController implements Initializable {  
  
 LoginModel loginModel = new LoginModel();  
 @FXML  
 private Label statusbaze;  
 @FXML  
 private TextField ime;  
 @FXML  
 private PasswordField lozinka;  
 @FXML  
 private ComboBox<option> izbor;  
 @FXML  
 private Button logindugme;  
 @FXML  
 private Label isLoginstatus;  
  
 @Override  
 public void initialize( URL url, ResourceBundle rb){  
 if(this.loginModel.isDatabaseConnected()) {  
 this.statusbaze.setText("Povezano na bazu");  
 }  
 else{  
 this.statusbaze.setText("Nije povezano na bazu");  
 }  
 this.izbor.setItems(FXCollections.*observableArrayList*(option.*values*()));  
 }  
  
 @FXML  
 public void login(ActionEvent event){  
 try{  
 if(this.loginModel.isLogin(this.ime.getText(), this.lozinka.getText(), ((option)this.izbor.getValue()).toString().toLowerCase()))  
 {  
 Stage stage = (Stage) this.logindugme.getScene().getWindow();  
 stage.close();  
  
 switch (((option) this.izbor.getValue()).toString()) {  
 case "Administrator":  
 administratorLogin();  
 break;  
 case "Zaposleni":  
 zaposleniLogin();  
 break;  
 }  
 }  
 else{  
 this.isLoginstatus.setText("Pogrešan unos");  
 }  
 }  
 catch(Exception localEx){  
 }  
 }  
  
 public void zaposleniLogin() {  
 try {  
 Stage userstage = new Stage();  
 FXMLLoader loader = new FXMLLoader();  
 Pane root = (Pane) loader.load(getClass().getResource("/Zaposleni/Zaposleni.fxml").openStream());  
 ZaposleniController zaposleniController = (ZaposleniController) loader.getController();  
 Scene scene = new Scene(root);  
 userstage.setScene(scene);  
 userstage.setTitle("Zaposleni Dashboard");  
 userstage.setResizable(false);  
 userstage.show();  
 } catch (IOException ex) {  
 ex.printStackTrace();  
 }  
 }  
 public void administratorLogin(){  
 try {  
 Stage adminStage = new Stage();  
 FXMLLoader administratorLoader = new FXMLLoader();  
 Pane administratorroot = (Pane)administratorLoader.load(getClass().getResource("/Administrator/Admin.fxml").openStream());  
 AdministratorController administratorController = (AdministratorController)administratorLoader.getController();  
 Scene adminscene = new Scene(administratorroot);  
 adminStage.setScene(adminscene);  
 adminStage.setTitle(("Administrator Dashboard"));  
 adminStage.setResizable(false);  
 adminStage.show();  
 }  
  
 catch (IOException e){  
 e.printStackTrace();  
 }  
 }  
  
}

# Login/LoginModel

public class LoginModel {  
 Connection connection;  
  
 public LoginModel(){  
 try{  
 this.connection = dbConnection.*getConnection*();   
 }  
 catch  
 (SQLException ex){  
 ex.printStackTrace();  
 }  
 if(this.connection == null){  
 System.*exit*(1);  
 }  
 }  
 public boolean isDatabaseConnected(){  
 return this.connection != null;  
 }  
  
 public boolean isLogin(String ime, String lozinka, String opt) throws Exception{  
 PreparedStatement pr = null;  
 ResultSet rs = null;  
 String sql = "SELECT \* from administrator where ime = ? and lozinka = ? and role = ?";  
 try {  
 pr = this.connection.prepareStatement(sql);  
 pr.setString(1, ime);  
 pr.setString(2, lozinka);  
 pr.setString(3, opt);  
 rs = pr.executeQuery();  
  
 boolean boll1;  
 if (rs.next()) {  
 return true;  
 }  
 return false;  
 }  
 catch (SQLException ex ) {  
 return false;  
 }  
 finally{  
 pr.close();  
 rs.close();  
 }  
  
 }  
}

# Login/Option

public enum option {  
 *Administrator*, *Zaposleni*;  
  
 private option(){ }  
  
 public String value(){  
  
 return name();  
 }  
 public static option fromvalue(String v)  
 {  
 return *valueOf*(v);  
 }

# Login/Main

public class Main extends Application {  
  
 public void start(Stage stage) throws Exception{  
 Parent root = (Parent) FXMLLoader.*load*(getClass().getResource("login.fxml"));  
 Scene scene = new Scene(root);  
 stage.setScene(scene);  
 stage.setTitle("Sistem baze podataka radnje");  
 stage.show();  
 }  
 public static void main(String [] args){  
 *launch*(args);  
 }  
}

# Login/login.fxml

<?xml version="1.0" encoding="UTF-8"?>  
  
<?import javafx.scene.control.\*?>  
<?import javafx.scene.layout.\*?>  
<?import javafx.scene.text.\*?>  
  
<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="350.0" prefWidth="350.0" xmlns="http://javafx.com/javafx/10.0.2-internal" xmlns:fx="http://javafx.com/fxml/1" fx:controller="Login.LoginController">  
 <children>  
 <Label layoutX="77.0" layoutY="295.0" text="Status baze:" />  
 <Label fx:id="statusbaze" layoutX="147.0" layoutY="295.0" prefHeight="17.0" prefWidth="119.0" />  
 <TextField fx:id="ime" layoutX="77.0" layoutY="92.0" prefHeight="25.0" prefWidth="197.0" promptText="ime" />  
 <Label layoutX="164.0" layoutY="66.0" text="Ime:" />  
 <Label layoutX="154.0" layoutY="128.0" text="Lozinka:" />  
 <PasswordField fx:id="lozinka" layoutX="77.0" layoutY="156.0" prefHeight="25.0" prefWidth="197.0" promptText="lozinka" />  
 <ComboBox fx:id="izbor" layoutX="77.0" layoutY="200.0" prefHeight="25.0" prefWidth="197.0" promptText="Administrator/Zaposleni" />  
 <Button fx:id="logindugme" layoutX="77.0" layoutY="249.0" mnemonicParsing="false" onAction="#login" prefHeight="25.0" prefWidth="197.0" text="Uloguj se" />  
 <Label fx:id="isLoginstatus" layoutX="124.0" layoutY="20.0" prefHeight="30.0" prefWidth="104.0" text="Ulogujte se">  
 <font>  
 <Font size="20.0" />  
 </font></Label>  
 </children>  
</AnchorPane>

# Utils/ArtikliData

public class ArtikliData {  
  
 private final StringProperty ID;  
 private final StringProperty naziv;  
 private final StringProperty kategorija;  
 private final StringProperty cena;  
  
 public ArtikliData(String id, String naziv, String kategorija, String cena ){  
  
 this.ID = new SimpleStringProperty(id);  
 this.naziv = new SimpleStringProperty(naziv);  
 this.kategorija= new SimpleStringProperty(kategorija);  
 this.cena = new SimpleStringProperty(cena);  
  
 }  
 public String getID() {  
 return ID.get();  
 }  
  
 public StringProperty IDProperty() {  
 return ID;  
 }  
  
 public void setID(String ID) {  
 this.ID.set(ID);  
 }  
  
 public String getNaziv() {  
 return naziv.get();  
 }  
  
 public StringProperty nazivProperty() {  
 return naziv;  
 }  
  
 public void setNaziv(String naziv) {  
 this.naziv.set(naziv);  
 }  
  
 public String getCena() {  
 return cena.get();  
 }  
  
 public StringProperty cenaProperty() {  
 return cena;  
 }  
  
 public void setCena(String cena) {  
 this.cena.set(cena);  
 }  
  
 public String getKategorija() {  
 return kategorija.get();  
 }  
  
 public StringProperty kategorijaProperty() {  
 return kategorija;  
 }  
  
 public void setKategorija(String kategorija) {  
 this.kategorija.set(kategorija);  
 }

# Utils/FXMLUtils

public class FXMLUtils {  
  
  
 public void loadArtikliData(TableView<ArtikliData> artik, TableColumn<ArtikliData, String> idc, TableColumn<ArtikliData, String> nac,  
 TableColumn<ArtikliData, String> katc, TableColumn<ArtikliData, String> cenc, ObservableList<ArtikliData> data, String sql) throws SQLException {  
 try {  
 Connection conn = dbConnection.*getConnection*();  
 data = FXCollections.*observableArrayList*();  
  
 ResultSet rs = conn.createStatement().executeQuery(sql);  
 while (rs.next()) {  
 data.add(new ArtikliData(rs.getString(1), rs.getString(2), rs.getString(3), rs.getString(4)));  
 }  
 } catch (SQLException e) {  
 System.*err*.println("Error" + e);  
 }  
 idc.setCellValueFactory(new PropertyValueFactory<ArtikliData, String>("ID"));  
 nac.setCellValueFactory(new PropertyValueFactory<ArtikliData, String>("naziv"));  
 katc.setCellValueFactory(new PropertyValueFactory<ArtikliData, String>("kategorija"));  
 cenc.setCellValueFactory(new PropertyValueFactory<ArtikliData, String>("cena"));  
 artik.setItems(null);  
 artik.setItems(data);  
 }  
  
  
 public void dodajArtikl(TextField naz, TextField kat, TextField cen) throws SQLException {  
 String sqlartikl = "INSERT INTO artikli ( naziv, kategorija, cena) VALUES (?,?,?)";  
 try {  
 Connection conn = dbConnection.*getConnection*();  
 PreparedStatement statement = conn.prepareStatement(sqlartikl);  
  
 statement.setString(1, naz.getText());  
 statement.setString(2, kat.getText());  
 statement.setString(3, cen.getText());  
 statement.execute();  
 conn.close();  
 } catch (SQLException e) {  
 e.printStackTrace();  
 }  
 }  
  
 public void brisiArtikl(TableView<ArtikliData> artik, Button del) throws SQLException {  
 String sqlartikl = "DELETE FROM artikli WHERE id= ?";  
 try {  
 Connection conn = dbConnection.*getConnection*();  
 PreparedStatement statement = conn.prepareStatement(sqlartikl);  
 del.disableProperty().bind(Bindings.*isEmpty*(artik.getSelectionModel().getSelectedItems()));  
 ArtikliData artikl = artik.getSelectionModel().getSelectedItem();  
 String id = artikl.getID();  
 statement.setString(1, id);  
 statement.executeUpdate();  
 statement.execute();  
 conn.close();  
 } catch (SQLException e) {  
 e.printStackTrace();  
 }  
 }  
 public void izmeniArtikl(TableView<ArtikliData> artik,TextField naz, TextField kat, TextField cen ) throws SQLException{  
 String sqlartikl = "UPDATE artikli SET naziv=?, kategorija=?, cena=? WHERE id=? ";  
 try{  
 Connection conn = dbConnection.*getConnection*();  
 PreparedStatement statement = conn.prepareStatement(sqlartikl);  
 ArtikliData artikl = artik.getSelectionModel().getSelectedItem();  
 String id = artikl.getID();  
 statement.setString(1,naz.getText());  
 statement.setString(2,kat.getText());  
 statement.setString(3,cen.getText());  
 statement.setString(4, id);  
 statement.execute();  
 conn.close();  
 }  
 catch (SQLException e){  
 e.printStackTrace();  
 }  
 }  
 public void obrisiFormu(TextField naz, TextField kat, TextField cen){//brisanje forme  
 naz.setText("");  
 kat.setText("");  
 cen.setText("");  
 }  
}

# Zaposleni/ZaposleniControler

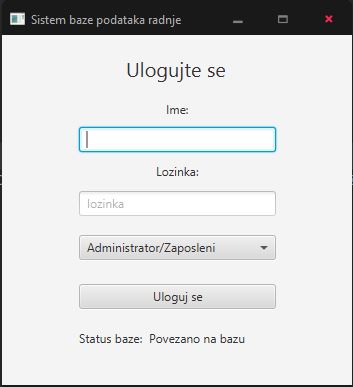
public class ZaposleniController implements Initializable {  
  
 @FXML  
 private TableView<ArtikliData> artiklitable;  
 @FXML  
 private TableColumn<ArtikliData, String> idcolumn;  
 @FXML  
 private TableColumn<ArtikliData, String> nazivcolumn;  
 @FXML  
 private TableColumn<ArtikliData, String> katcolumn;  
 @FXML  
 private TableColumn<ArtikliData, String> cenacolumn;  
 @FXML  
 private Button del;  
 private LoginController loginController = new LoginController();  
 private FXMLUtils Fx= new FXMLUtils();  
 private dbConnection dc;  
 private ObservableList<ArtikliData> data;  
 private String sql = "SELECT \* FROM artikli ";  
  
 public void initialize(URL url, ResourceBundle rb) {  
 this.dc = new dbConnection();  
 }  
  
 @FXML  
 private void adminLogin(ActionEvent event) {  
 loginController.administratorLogin();  
 }  
  
 @FXML  
 private void loadArtikliData() throws SQLException {  
 Fx.loadArtikliData(artiklitable,idcolumn, nazivcolumn, katcolumn, cenacolumn, data, sql);  
 }  
  
 @FXML  
 private void brisiArtikl(ActionEvent event) throws SQLException {  
 Fx.brisiArtikl(artiklitable,del);  
 }  
}

# Zaposleni.fxml

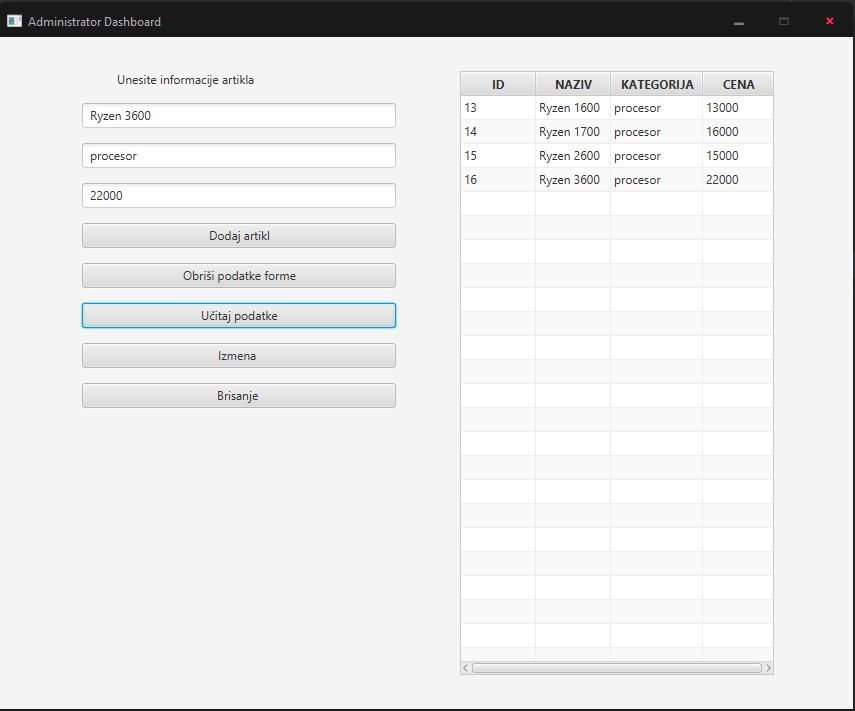
<?xml version="1.0" encoding="UTF-8"?>  
  
<?import javafx.geometry.\*?>  
<?import javafx.scene.control.\*?>  
<?import javafx.scene.layout.\*?>  
  
<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="672.0" prefWidth="854.0" xmlns="http://javafx.com/javafx/10.0.2-internal" xmlns:fx="http://javafx.com/fxml/1" fx:controller="Zaposleni.ZaposleniController">  
 <children>  
 <HBox layoutX="471.0" layoutY="44.0" prefHeight="604.0" prefWidth="323.0">  
 <children>  
 <TableView fx:id="artiklitable" prefHeight="604.0" prefWidth="324.0">  
 <columns>  
 <TableColumn fx:id="idcolumn" prefWidth="75.0" text="ID" />  
 <TableColumn fx:id="nazivcolumn" prefWidth="75.0" text="NAZIV" />  
 <TableColumn fx:id="katcolumn" prefWidth="98.0" text="KATEGORIJA" />  
 <TableColumn fx:id="cenacolumn" prefWidth="75.0" text="CENA" />  
 </columns>  
 </TableView>  
 </children>  
 </HBox>  
 <VBox layoutX="63.0" layoutY="44.0" prefHeight="604.0" prefWidth="374.0">  
 <children>  
 <Button mnemonicParsing="false" onAction="#loadArtikliData" prefHeight="25.0" prefWidth="314.0" text="Učitaj podatke">  
 <VBox.margin>  
 <Insets left="15.0" top="15.0" />  
 </VBox.margin>  
 </Button>  
 <Button fx:id="del" mnemonicParsing="false" onAction="#brisiArtikl" prefHeight="25.0" prefWidth="314.0" text="Brisanje ">  
 <VBox.margin>  
 <Insets left="15.0" top="15.0" />  
 </VBox.margin>  
 </Button>  
 <Button mnemonicParsing="false" onAction="#adminLogin" prefHeight="25.0" prefWidth="315.0" text="Izmena/dodavanje artikala">  
 <VBox.margin>  
 <Insets left="15.0" top="15.0" />  
 </VBox.margin>  
 </Button>  
 </children>  
 <padding>  
 <Insets left="15.0" right="15.0" />  
 </padding>  
 </VBox>  
 </children>  
</AnchorPane>

# Фотографије:

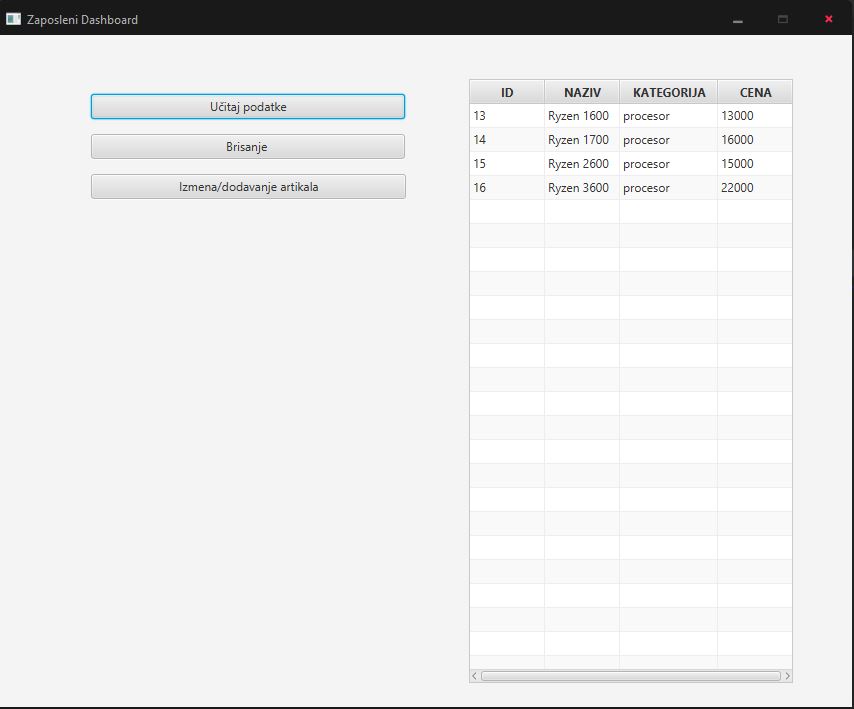
# 



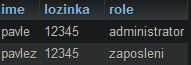
Слика 1: Логин форма



Слика 2: Администраторски панел



Слика 3:Запослени панел



Слика 4: Администратор табела базе



Слика 5: Артикли табела