

Rudno

IT-22002

Favourite dish project using JavaFx:

FoodItem.java

package com.example.my-favourite_dish;

public class FoodItem {

private String name;

private String description;

```
public FoodItem(String name, String description)
```

```
{
```

```
    this.name = name;
```

```
    this.description = description;
```

```
}
```

```
public String getName() {
```

```
    return name;
```

```
}
```

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

```
    this.name = name;
```

```
}
```

```
public String getDescription() {
```

```
    return description;
```

```
public void setDescription(String description) {
```

```
    this.description = description;
```

```
}
```

```
}
```

DBInfo.java

avajava\jdk\1.9\src\java

```
package com.example.my_favourite_disk;  
  
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.SQLException;  
  
public class DBInfo {  
    private static final String URL = "jdbc:mysql://  
localhost:3306/folder_db?useSSL=  
false&allowPublicKeyRetrieval=true&server  
timeZone=UTC";  
  
    private static final String USER = "AHT";  
    private static final String PASSWORD = "anpita";  
  
    public static Connection getConnection() throws  
        SQLException {  
        return DriverManager.getConnection  
            (URL, USER, PASSWORD);  
    }  
}
```


HelloApplication.java

```
package com.example.my-favourite-dish;  
import javafx.application.Application;  
import javafx.fxml.FXMLLoader;  
import javafx.scene.Stage Scene;  
import javafx.stage.Stage;
```

```
public class HelloApplication extends Application {
```

```
@Override
```

```
public void start(Stage stage) throws
```

```
{  
    Exception  
    Application {
```

```
FXMLLoader fxmlLoader = new FXMLLoader
```

```
(getClass().getResource("/com/example/
```

```
my-favourite-dish/dish-view.fxml"));
```

```
stage.setTitle("অমর প্রিয় খাদ্য");
```

```
stage.setScene(scene);
```

```
stage.show();
```

```
}
```

```
public static void main(String[] args) {
```

```
    Launch.launch(args);
```

```
}
```

```
}
```

DishController.java

```
package com.example.my_favourite_dish;
```

```
import javafx.collections.FXCollections;
```

```
import javafx.collections.ObservableList;
```

```
import javafx.fxml.FXML;
```

```
import javafx.scene.control.*;
```

```
import java.sql.*;
```

```
public class DishController {
```

```
    @FXML
```

```
    private TextField nameField;
```

```
    @FXML
```

```
    private TextField descriptionField;
```

@FXML

private TableView<Food Item> dishTable;

@FXML

private TableColumn<Food Item, String> name

Column;

@FXML

private TableColumn<Food Item, String>

descriptionColumn;

private ObservableList<Food Item, String>

descriptionColumn;

private ObservableList<Food Item> dishes =

FXCollections.observableArrayList();

private Connection connection;

@FXML

public void initialize() {

nameColumn.setCellValueFactory(new CellData


```
new javax.swing.PropertyChangeListener() {
```

```
    public void propertyChange(PropertyChangeEvent evt) {
```

```
        descriptionColumn.setCellValueFactory(new CellValueFactory<String>() {
```

```
            public String get(String key) {
```

```
                return new PropertyChangeListener() {
```

```
                    public void propertyChange(PropertyChangeEvent evt) {
```

```
                        dishTable.setItems(dishes);
```

```
                        connectToDatabase();
```

```
                        loadDishesFromDatabase();
```

```
                        dishTable.getSelectionModel().selectedItemProperty().addListener(observable, oldSelection,
```

```
                            newSelection) -> {
```

```
                                if (newSelection != null) {
```

```
                                    nameField.setText(newSelection.getName());
```

```
                                    descriptionField.setText(newSelection.getDescription());
```

```
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}
```

```
3) :
```

```
private void connectToDatabase() {
```

```
try {
```

```
connection = DriverManager.getConnection
```

```
("jdbc:mysql://localhost:3306/
```

```
("fooddb", "AhiF", "anpita");
```

```
Statement stmt = connection.createStatement
```

```
stmt.executeUpdate("CREATE TABLE IF
```

```
NOT EXISTS dishes (name VARCHAR
```

```
(255), description TEXT)");
```

```
} catch (SQLException e) {
```

```
showAlert("ডেটা বের করা", "ডেটা
```

```
বের করা সমস্যায় বরাহ হয় নি: "+
```

```
e.getMessage());
```

```
}
```



```

protected void addDish() {
    String name = nameField.getText().trim();
    if (name.isEmpty()) {
        showAlert("କ୍ଷମିତ", "ଅବଶ୍ୟକତା ଅନୁଯାୟୀ ନାମ ଦିଅନ୍ତୁ");
    }
    return;
}

```

```

try {
    Prepared Statement stmt = connection.prepareStatement(
        "INSERT INTO dishes (name, description) VALUES (?, ?)");
    stmt.setString(1, name);
    stmt.setString(2, description);
    stmt.executeUpdate();
} catch (SQLException e) {
}

```

```
showAlert("ত্রুটি", "অবশ্যই খোলা রাখুন")
```

```
খয়তি: " + e.getMessage() );
```

```
{  
    mntl() + e.getMessage() + e.getMessage()  
    return;
```

```
}
```

```
{  
    if (e.getMessage() != null) {
```

```
        dishes.add(new FoodItem(name, description));
```

```
        cleanFields();
```

```
}
```

@FXML

```
protected void updateDish() {
```

```
    FoodItem selected = dishTable.getSelectionModel
```

```
    .getSelectedItem();
```

```
{  
    if (selected == null) {
```

```
        showAlert("ত্রুটি", "অবশ্যই
```

```
        খোলা রাখুন অবশ্যই খোলা রাখুন");
```

```
        return;
```

```
}
```

```
try {
```

PreparedStatement stmt = connection.prepareStatement

Statement("UPDATE dishes ~~set~~ SET name = ?,

description = ? Where name = ?");

stmt.setString(1, nameField.getText().trim());

stmt.setString(2, descriptionField.getText().

trim());

dishTable.refresh();

)

catch (SQLException e) {

showAlert("Error", "যত্নসহ অবশর্ডে করণ

যত্নসহ:" + e.getMessage());

}

cleanFields();

@FXML

protected void deleteDish() {

FoodItem selected = dishTable.getSelectionModel

Model().getSelectedItem();


```
if(selected == null) {
```

```
    showAlert("Error", "Please select a dish")
```

```
    return;
}
```

```
PreparedStatement stmt = connection.
```

```
prepareStatement("DELETE FROM
```

```
dishes WHERE name = ?");
```

```
stmt.setString(1, selected.getName());
```

```
stmt.executeUpdate();
```

```
dishes.remove(selected);
```

```
}
```

```
catch (SQLException e) {
```

```
    showAlert("Error", "Please select a dish")
```

```
    return;
}
```

```
message();
```

```
cleanFields();
```

```
}
```

```
@FXML
```

```
public void cleanFields(){
```

```
    nameField.clean();
```

```
    descriptionField.clean();
```

```
    dishTable.getSelectionModel().cleanSelection();
```

```
}
```

```
private void showAlert(String title, String
```

```
message){
```

```
    Alert alert = new Alert(Alert.AlertType.
```

```
        INFORMATION);
```

```
    alert.setTitle(title);
```

```
    alert.setHeaderText(null);
```

```
    alert.setContentText(message);
```

```
    alert.showAndWait();
```

```
}
```

```
}
```