COLLEGE OF BUSINESS EDUCATION

DODOMA CAMPUS

***Student Name*:**	**EMMANUEL FREDY SHANI**
***Registration Number*:**	**03.8853.01.02.2023**
***Course*:**	**BIT II**
***Lecturer*:**	**ATUPELE CAIRO MWAITETE**
***Subject*:**	**PROGRAMMING IN JAVA**
***Subject Code*:**	**ITU07312**
***Nature of Assignment*:**	**INDIVIDUAL ASSIGNMENT**
***Submission* :**	**28 JAN 2025**

REPORT

Title: Petrol Price Comparison Application

Theme: Digital Solutions for Everyday Challenges in Tanzania

1. Introduction

The **Petrol Price Comparison Application** is a Java-based standalone application that provides a graphical user interface (GUI) for users to manage and compare petrol prices across different gas stations. The application is developed using **Swing** for the GUI and implements file handling to store and retrieve data on petrol prices.

Rising fuel prices and varying costs across petroleum stations have made it essential for consumers to find affordable options quickly. **Petrol Price Comparison Application** allows users to check fuel prices at different stations and compare them to make informed decisions. This app aims to reduce the hassle of locating cost-effective fueling options, saving both time and money.

2. Features Implemented (Functionalities)

The application allows users to perform the following actions:

2.1 **Add Petrol Prices**

The app allows users to select a location and displays the fuel prices for nearby petroleum stations. Information displayed includes: Station Name, Fuel Type (e.g., petrol, diesel) and Price per Liter

2.2 **View Prices**

Displays a list of all stored petrol prices

2.3 **Search for a Gas Station**

Users can search for petrol prices by entering the gas station's name.

2.4 **Edit Petrol Prices**

Users can update the price of petrol for a specific gas station.

2.5 **Delete a Gas Station Entry**

Users can remove a gas station and its price from the list.

2.6 **Save Changes**

Ensures that the updated price list is stored in a text file for persistence

2.7 Exit

Users can exit the application at any time through the main menu, which provides a clean exit message.

3. Implementation Details

a. Graphical User Interface (GUI)

- Developed using **JFrame, JTextField, JTextArea, JLabel, JButton**, and **JOptionPane** for user interactions.
- A **FlowLayout** is used for arranging components within the main window.

- Buttons handle user actions such as adding, viewing, editing, and deleting gas station entries.

b. Data Handling

- The application maintains a list of **GasStation** objects to store petrol prices.
- Uses **ArrayList** to manage gas station data dynamically.

c. File Persistence

- **Loading Data:**
  - Reads stored petrol prices from gas_stations.txt using **BufferedReader**.
  - Parses each line into GasStation objects.
- **Saving Data:**
  - Writes the updated list of petrol prices back to gas_stations.txt using **BufferedWriter**.

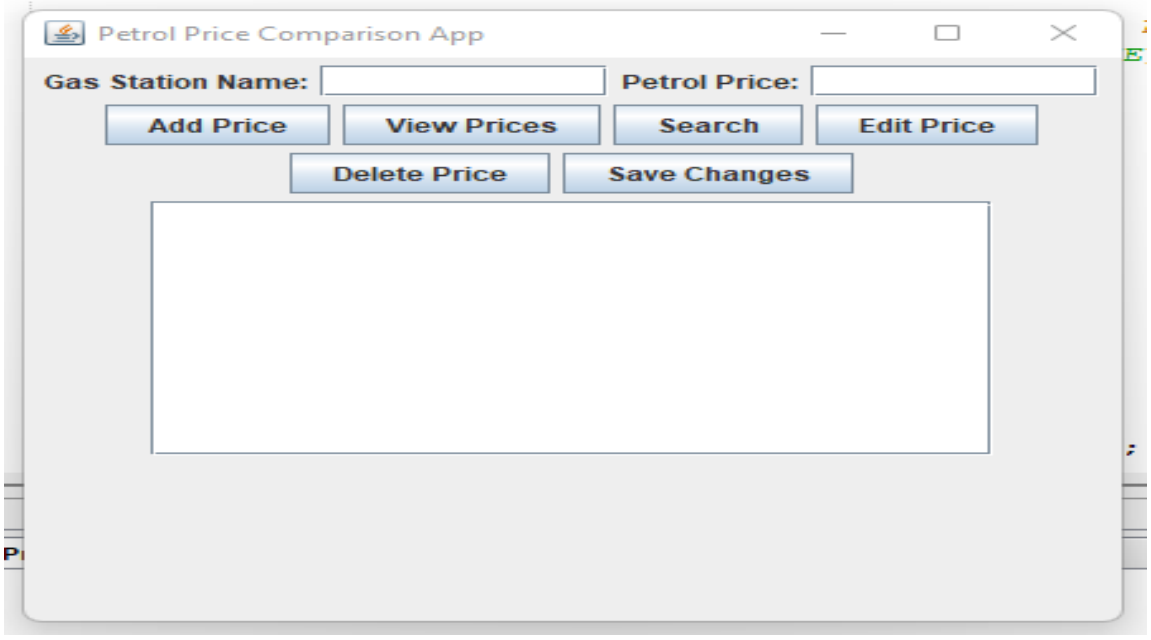
4. **Characteristics**

- **User-Friendly Interface:** Simple and intuitive design for ease of use.
- **Data Persistence:** Ensures petrol prices are saved and retrieved
- **Efficient Searching & Editing:** Allows users to search for specific gas stations and modify petrol prices efficiently.
- **Error Handling & Validation:** Prevents invalid inputs and provides user feedback through dialog messages.

5. Project Screenshots

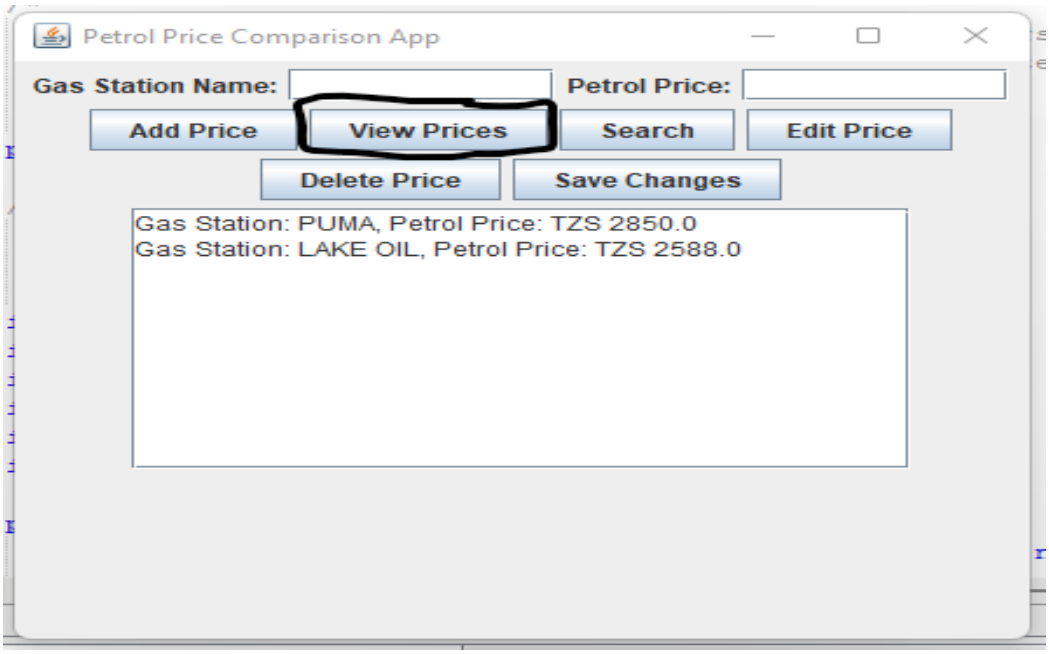
5.1 Welcome Screen

The initial interface displays a menu where users can choose to check fuel prices, compare stations, or add a station to their favorites.



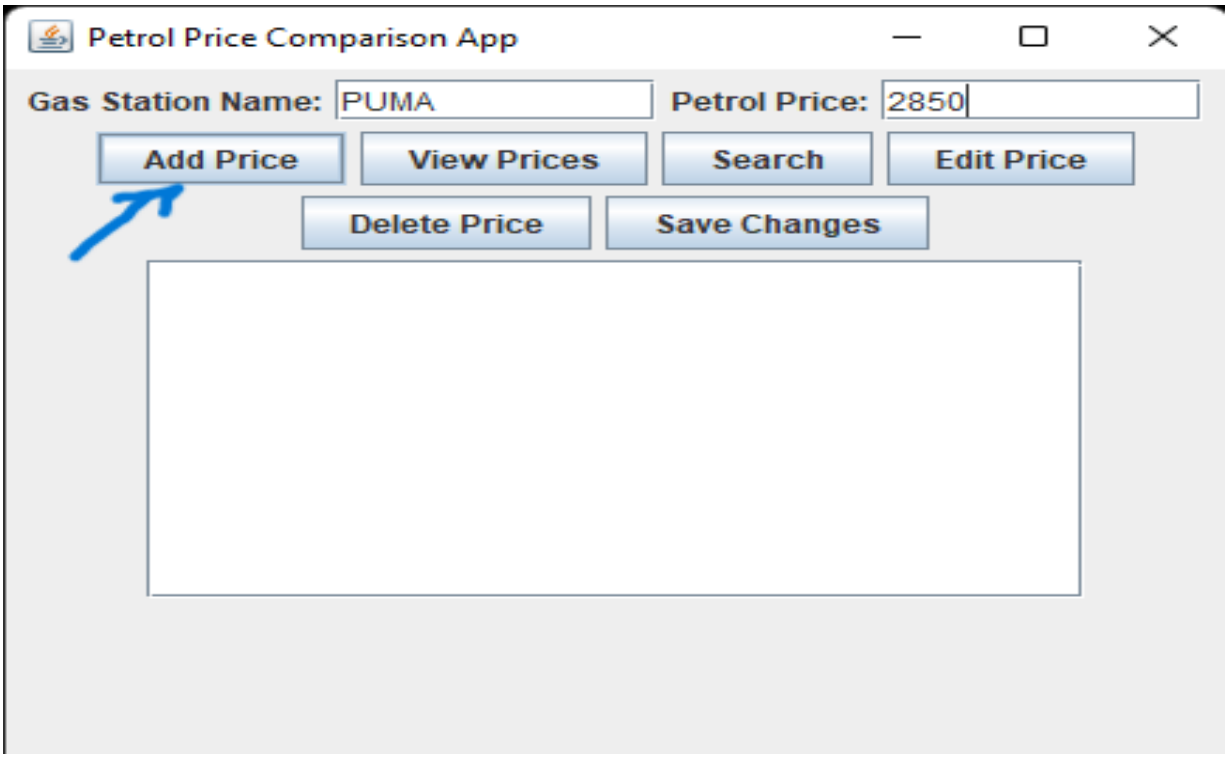
5.2 Check Fuel Prices

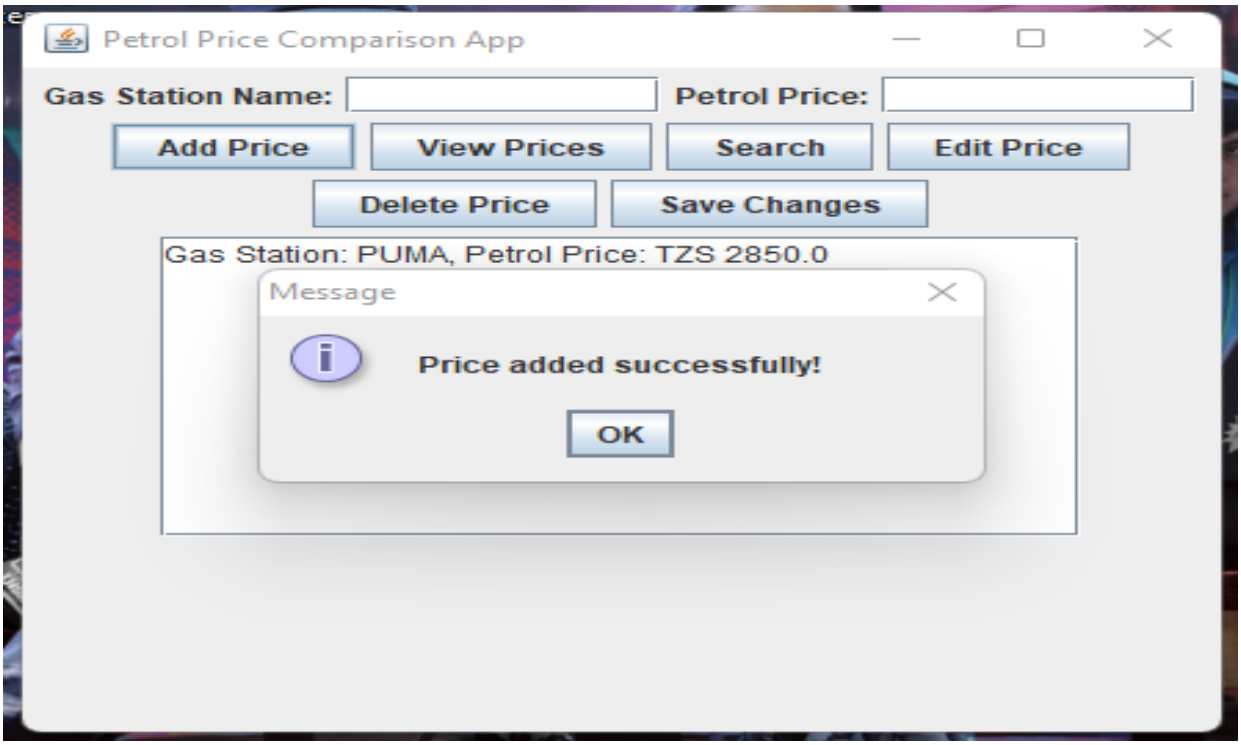
Users can view fuel prices for petroleum stations based on their selected location.



5.3 Add Favorite Station and Price

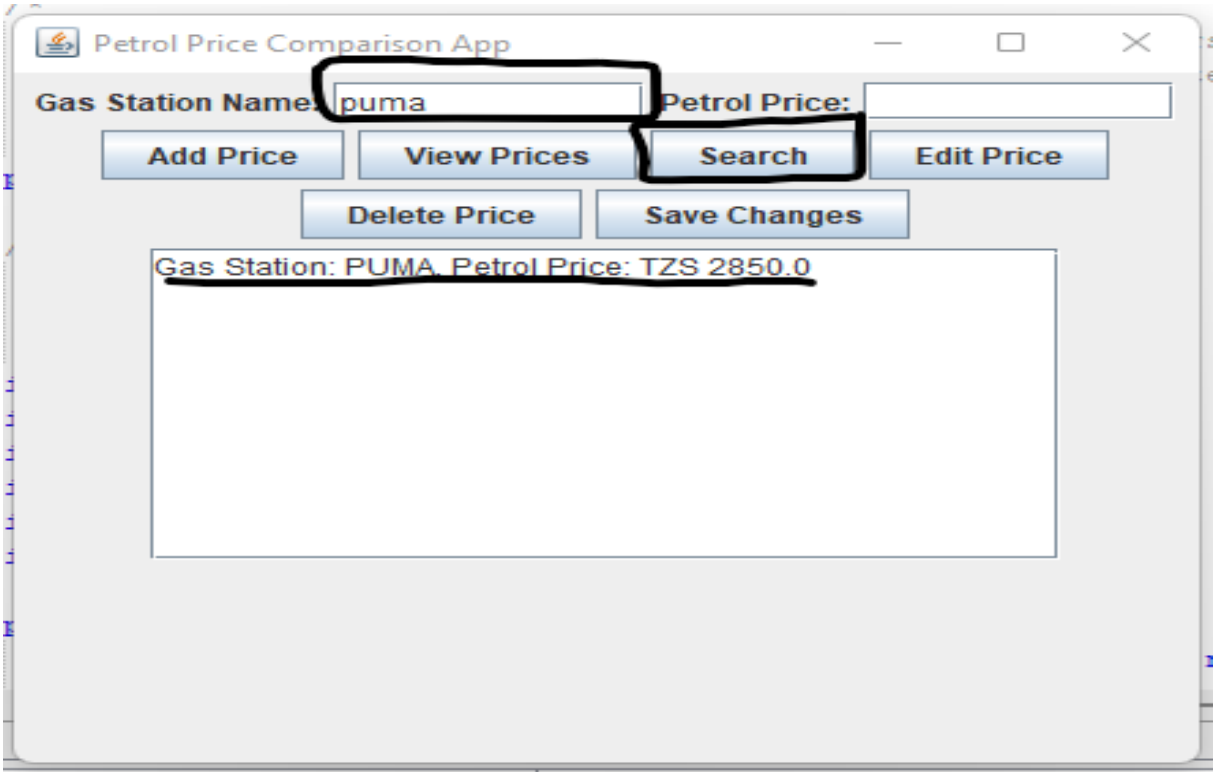
Users can save their most frequently visited stations for quick access.





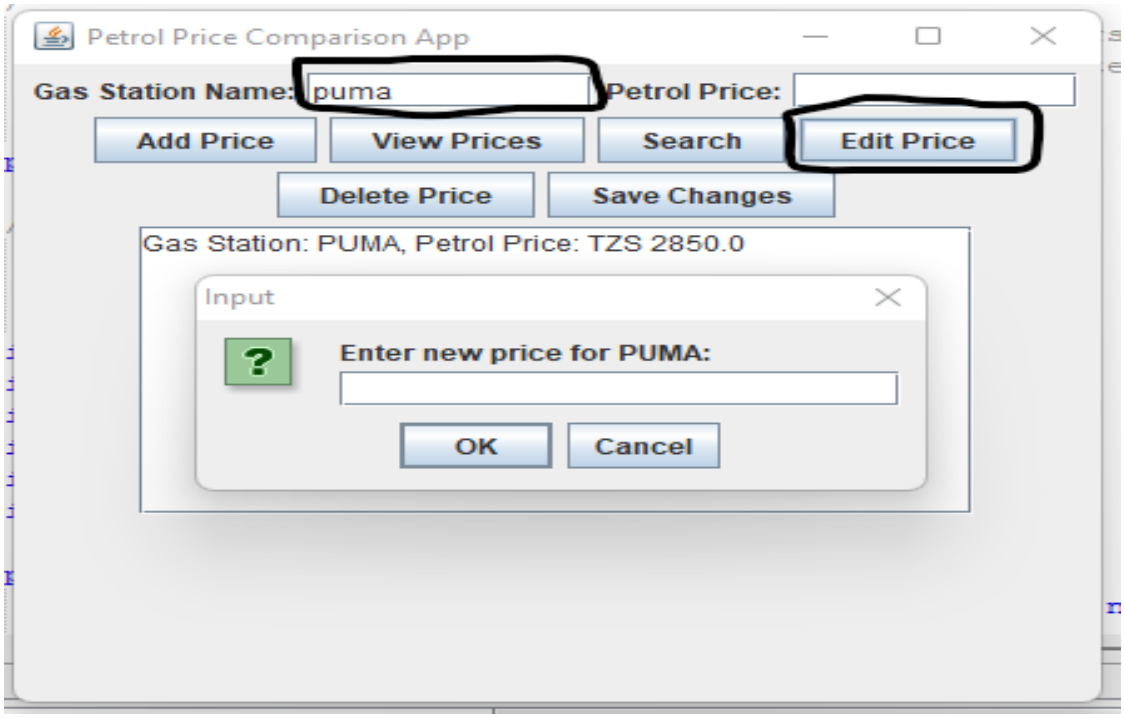
5.4 **Search for a Gas Station**

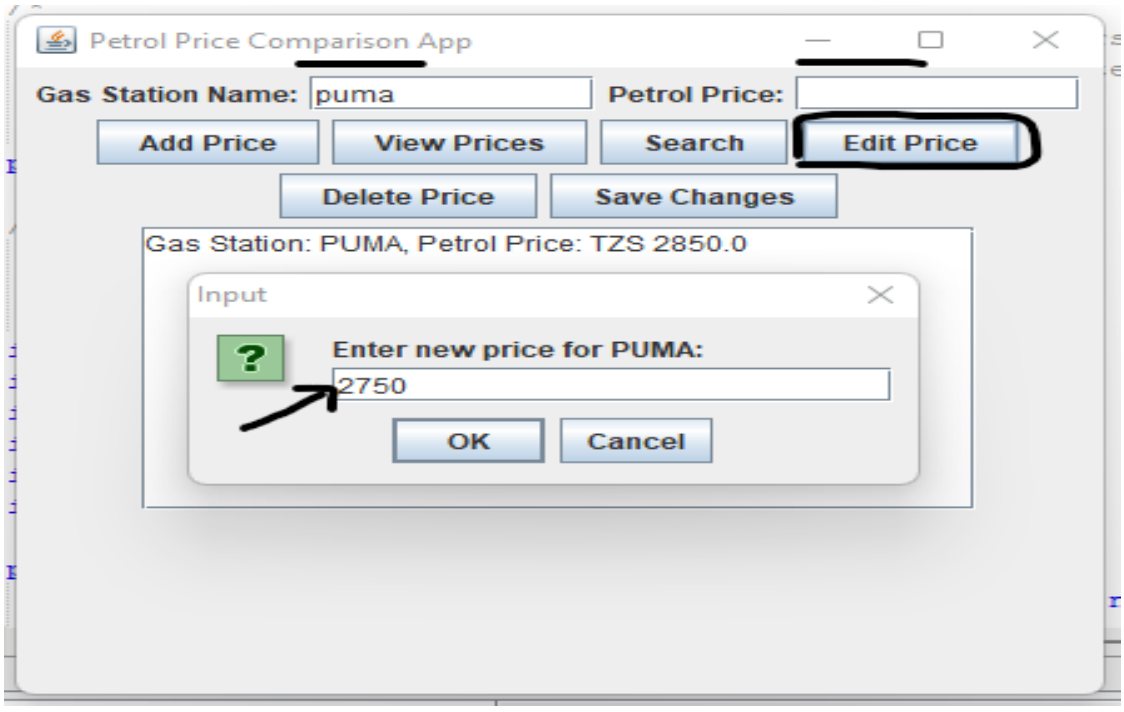
Users can search for petrol prices by entering the gas station's name.



5.5 **Edit Petrol Prices**

Users can update the price of petrol for a specific gas station.

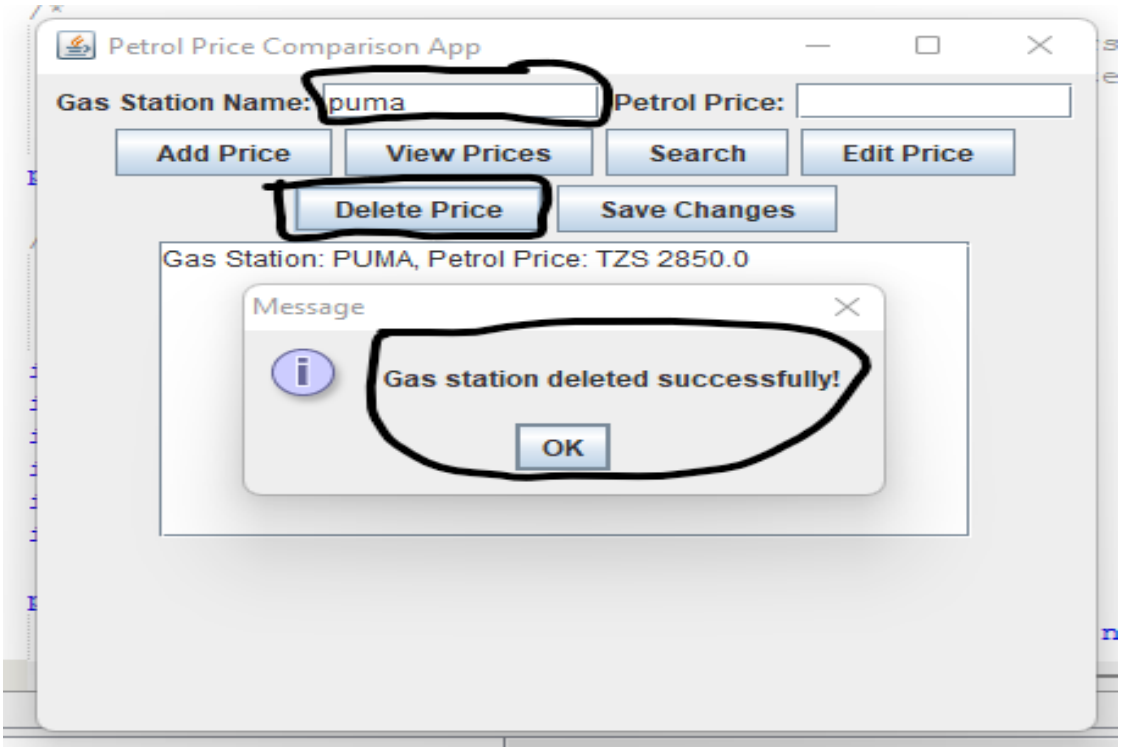




5.6 **Delete a Gas Station Entry**

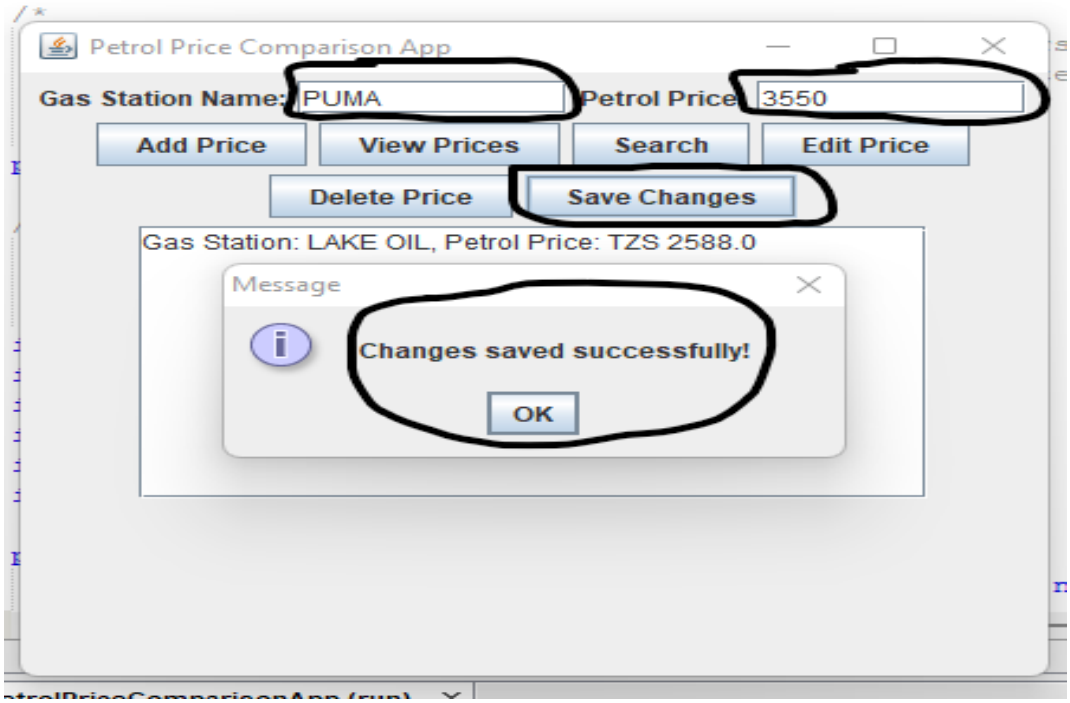
Users can remove a gas station and its price from the list

.



5.7 **Save Changes**

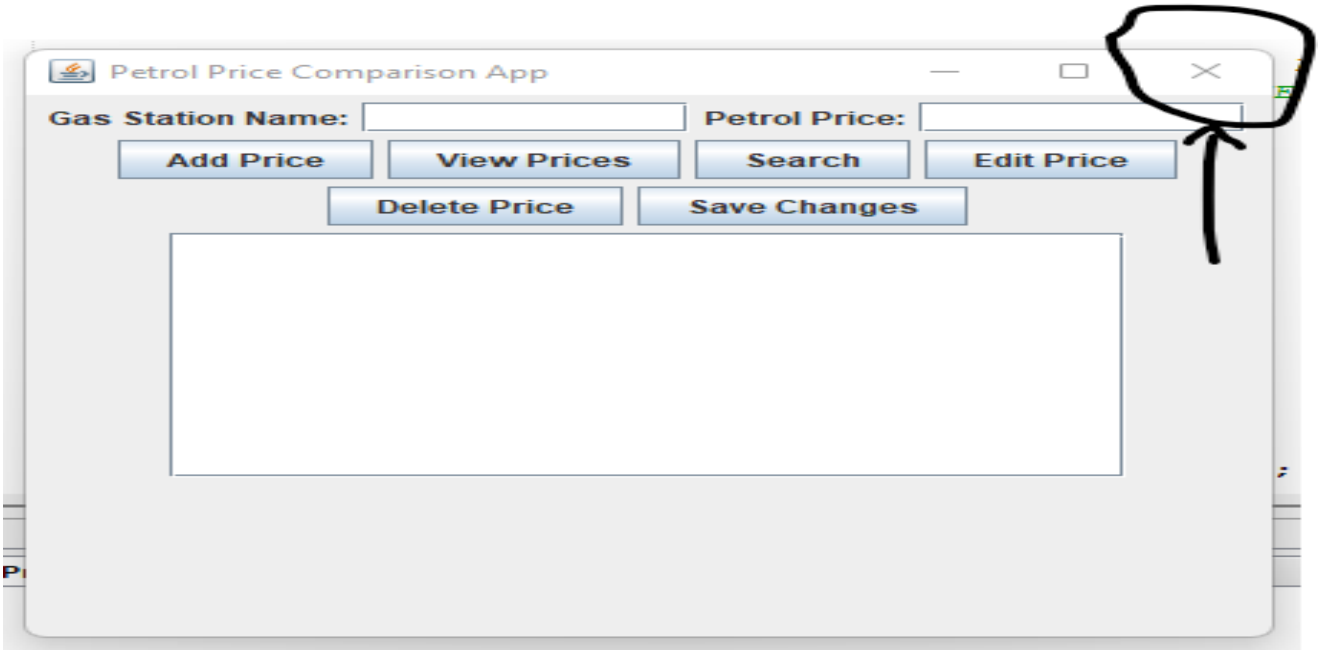
Ensures that the updated price list is stored in a text file for persistence



5.8 **Exit Screen**

Users can cleanly exit the application, with a friendly goodbye message.

(These screenshots are placeholders that would ideally be replaced with actual screenshots from the console output.)



6. Challenges Faced

6.1 Simulating Real-Time Fuel Prices

Problem: Integrating real-time price updates from petroleum stations was not feasible in a basic console application.

Solution: Predefined data for a fixed set of stations was used to simulate the functionality.

6.2 User Input Validation

Problem: Users might enter invalid or misspelled station names or locations.

Solution: The app checks user input against a predefined list of stations and provides error messages for invalid entries.

6.3 Interface Simplicity

Problem: The console-based interface limits the user experience and makes accessing large datasets cumbersome.

Solution: A simple menu-based approach was implemented to ensure ease of navigation and usability.

6.4 Feature Expansion

Problem: Adding features like notifications for price drops or integrating a map interface required advanced APIs and GUI elements.

Solution: These were deferred to future development phases.

7. Future Work

7.1 Real-Time Price Updates

Integrate APIs to fetch real-time fuel prices from petroleum stations.

7.2 Enhanced Interface

Develop a graphical user interface (GUI) using JavaFX or convert the application into a mobile app.

7.3 Location-Based Search

Incorporate GPS functionality to suggest the nearest and cheapest stations.

7.4 Price Drop Notifications

Implement push notifications to alert users of price reductions at their favorite stations.

8. Recommendations

This project demonstrates the potential of simple digital tools in addressing everyday challenges. With further enhancements, such as real-time data integration and a mobile interface, this app could become a vital tool for vehicle owners across Tanzania and beyond.

9. Conclusion

The Petroleum Station Price App provides a practical solution to a common problem faced by drivers in Tanzania. By enabling users to compare fuel prices across stations, it empowers them to make informed decisions and save money. It provides a **user-friendly interface**, maintains **persistent data storage**, and offers essential CRUD (Create, Read, Update, Delete) functionalities for managing fuel prices. The application's simple design ensures accessibility for users, even those with limited technical expertise.