A Report

On

Mess Management System

JP_2324_TD1_16

Report Prepared By:

Krupa Bhalsod - 92310104009

Class 4TD1

Guided By:

Dr. Tarannum Bloch Marwadi University

Mess Management System

Objective

The Mess Management System aims to modernize and streamline the operations of institutional mess facilities through the implementation of a comprehensive digital platform. The primary objective of the system is to automate various administrative tasks, including user registration, login authentication, menu display, attendance tracking, and feedback submission. By replacing manual record-keeping processes with automated solutions, the system seeks to enhance efficiency and reduce the burden on administrative staff. Additionally, the system prioritizes user experience by providing intuitive interfaces and streamlined processes, ensuring that users can easily access information about menus, mark their attendance, and submit feedback. Moreover, the system aims to facilitate better communication between mess administrators and users, fostering transparency, responsiveness, and accountability. Through data collection and analysis, the system enables informed decision-making, allowing administrators to gain insights into user preferences, attendance trends, and areas for improvement. Ultimately, the Mess Management System aims to optimize resource allocation, minimize wastage, and improve overall operational efficiency in institutional mess facilities.

Technology Used in Project Development

Programming Language: Java

- Provides robustness, portability, and scalability.
- Enables cross-platform compatibility.

Database Management System: MySQL

- Offers reliability, performance, and scalability.
- Handles data related to user registration, attendance records, menu details, and feedback submissions.

Graphical User Interface (GUI): Java Swing

- Lightweight framework for building graphical applications.
- Provides a rich set of components and APIs for intuitive interfaces.

Database Connectivity: JDBC (Java Database Connectivity)

- Establishes connections with the MySQL database.
- Facilitates efficient retrieval and manipulation of data.

Security Measures:

- Implements secure authentication mechanisms using encryption techniques.

Functional Modules

Class Overview:

- **Name: Registration**
- **Description:** This class represents the registration frame of the Mess Management System application. It provides functionality for users to register by entering their username, password, email, and GR number.
- **Extends: JFrame**
- **Implements: ActionListener**

Class Variables:

1. **JButton submitBtn**: Button for submitting the registration data.
2. **JButton loginBtn**: Button for navigating to the login frame.
3. **JLabel usrLbl**: Label for displaying "Enter Username:".
4. **JLabel pwdLbl**: Label for displaying "Enter Password:".
5. **JLabel mailLbl**: Label for displaying "Enter Email:".
6. **JLabel grLbl**: Label for displaying "Enter GR number:".
7. **JTextField usrTxf**: Text field for entering the username.
8. **JTextField pwdTxf**: Text field for entering the password.
9. **JTextField mailTxf**: Text field for entering the email.
10. **JTextField grTxf**: Text field for entering the GR number.
11. **Connection con**: Connection object for database connection.
12. **Statement stmt**: Statement object for executing SQL queries.
13. **ResultSet rs**: Result set object for storing query results.
14. **PreparedStatement ps**: Prepared statement object for executing parameterized SQL queries.
15. **String usrData**: String variable to store the username data.
16. **String pwdData**: String variable to store the password data.
17. **String mailData**: String variable to store the email data.
18. **String grData**: String variable to store the GR number data.

Class Methods:

1. **Registration()**: Constructor method responsible for initializing the registration frame, UI components, and event listeners.
2. **insertData()**: Method to insert registration data into the database. It retrieves user input from text fields and executes an SQL insert query.
3. **actionPerformed(ActionEvent e)**: Method to handle button click events. It implements the **ActionListener** interface and performs actions based on the source of the event.
4. **main(String[] args)**: Main method to instantiate the **Registration** class and create an instance of the registration frame.

Class Functionality:

- The **Registration** class provides a graphical user interface (GUI) for users to register by entering their credentials.
- It establishes a connection to the database and inserts the registration data upon submission.
- The class extends **JFrame** to create a window for user interaction and implements **ActionListener** to handle button click events.

Class Overview:

- **Name: Login**
- **Description:** This class represents the login frame of the Mess Management System application. It allows users to authenticate by entering their username and password.
- **Extends: JFrame**
- **Implements: ActionListener**

Class Variables:

1. **JButton submitBtn**: Button for submitting the login credentials.
2. **JLabel usrLbl**: Label for displaying "Enter Username:".
3. **JLabel pwdLbl**: Label for displaying "Enter Password:".
4. **JLabel authenticateLbl**: Label for displaying authentication status.
5. **JTextField usrTxf**: Text field for entering the username.
6. **JTextField pwdTxf**: Text field for entering the password.

Class Methods:

1. **Login()**: Constructor method responsible for initializing the login frame, UI components, and event listeners.
2. **authenticate()**: Method to authenticate user credentials by querying the database. It retrieves username and password input from text fields and executes an SQL select query.
3. **actionPerformed(ActionEvent ae)**: Method to handle button click events. It implements the **ActionListener** interface and performs actions based on the source of the event.

Class Functionality:

- The **Login** class provides a GUI for users to log in by entering their username and password.
- It connects to the database and authenticates user credentials upon submission.
- Upon successful authentication, the user is redirected to the homepage of the application.

Class Overview:

- **Name: HomePage**
- **Description:** This class represents the homepage frame of the Mess Management System application. It serves as the main interface for users after successful login, providing access to different functionalities of the system.
- **Extends: JFrame**
- **Implements: ActionListener**

Class Variables:

1. **JButton showMenuPageBtn**: Button for navigating to the menu page.
2. **JButton attendencePageBtn**: Button for navigating to the attendance page.
3. **JButton feedbackPageBtn**: Button for navigating to the feedback page.

Class Methods:

1. **HomePage()**: Constructor method responsible for initializing the homepage frame, UI components, and event listeners.
2. **actionPerformed(ActionEvent e)**: Method to handle button click events. It implements the **ActionListener** interface and performs actions based on the source of the event.

Class Functionality:

- The **HomePage** class provides a GUI for users to navigate to different functionalities of the Mess Management System application.
- It includes buttons to access the menu page, attendance page, and feedback page.
- Each button is associated with an action listener to handle user interactions and perform the corresponding navigation.

Class Overview:

- **Name: Menu**
- **Description:** This class represents the menu page of the Mess Management System application. It displays the menu for different weekdays and allows users to navigate to other functionalities.
- **Extends: JFrame**
- **Implements: ActionListener**

Class Variables:

1. **JComboBox<String> weekdayComboBox**: Dropdown menu to select weekdays.
2. **JTextArea resultArea**: Area to display the menu for the selected weekday.
3. **JButton homeBtn**: Button to navigate back to the homepage.

Class Methods:

1. **Menu()**: Constructor method responsible for initializing the menu frame, UI components, and event listeners.
2. **actionPerformed(ActionEvent e)**: Method to handle button click events. It implements the **ActionListener** interface and performs actions based on the source of the event.
3. **displayMenu()**: Method to display the menu for the selected weekday.
4. **getMenuFromDatabase(String weekday)**: Method to retrieve the menu from the database based on the selected weekday.

Class Functionality:

- The **Menu** class provides a GUI for users to view the menu for different weekdays.
- It includes a dropdown menu to select weekdays and a text area to display the corresponding menu.
- Users can navigate back to the homepage using the home button.
- The class retrieves menu data from the database and dynamically updates the UI based on the selected weekday.

Class Overview:

- **Name: Attendance**
- **Description:** This class represents the attendance page of the Mess Management System application. It allows users to mark their attendance for each day of the week.
- **Extends: JFrame**

Class Variables:

1. **DateFormat dateFormat**: Object to format dates.
2. **JButton submitButton**: Button to submit attendance.
3. **JButton homeBtn**: Button to navigate back to the homepage.
4. **JLabel usrLabel**: Label for username input.
5. **JTextField usrTxf**: Text field for entering the username.
6. **int checkBoxY**: Y-coordinate for positioning checkboxes dynamically.

Class Methods:

1. **Attendance()**: Constructor method responsible for initializing the attendance frame, UI components, and event listeners.
2. **displayWeekCheckBoxes(Date startDate)**: Method to display checkboxes for each day of the week.
3. **markAttendance()**: Method to mark attendance based on the selected checkboxes and insert attendance records into the database.
4. **actionPerformed(ActionEvent ae)**: Method to handle button click events.

Class Functionality:

- The **Attendance** class provides a GUI for users to mark their attendance for each day of the week.
- It dynamically generates checkboxes for each day, allowing users to select whether they are present or absent.
- Users can enter their username and submit their attendance.
- Attendance records are inserted into the database upon submission.

Class Overview:

- **Name: Feedback**
- **Description:** This class represents the feedback submission page of the Mess Management System application. It allows users to submit their feedback along with their username.
- **Extends: JFrame**

Class Variables:

1. **JButton submitbtn**: Button to submit feedback.
2. **JButton homeBtn**: Button to navigate back to the homepage.
3. **JLabel feedbackLbl**: Label for feedback input.
4. **JTextArea feedbackTextArea**: Text area for entering feedback.
5. **JLabel usrLabel**: Label for username input.
6. **JTextField usrTxf**: Text field for entering the username.

Class Methods:

1. **Feedback()**: Constructor method responsible for initializing the feedback frame, UI components, and event listeners.
2. **insertFeedback()**: Method to insert feedback into the database along with the username.
3. **actionPerformed(ActionEvent e)**: Method to handle button click events.

Class Functionality:

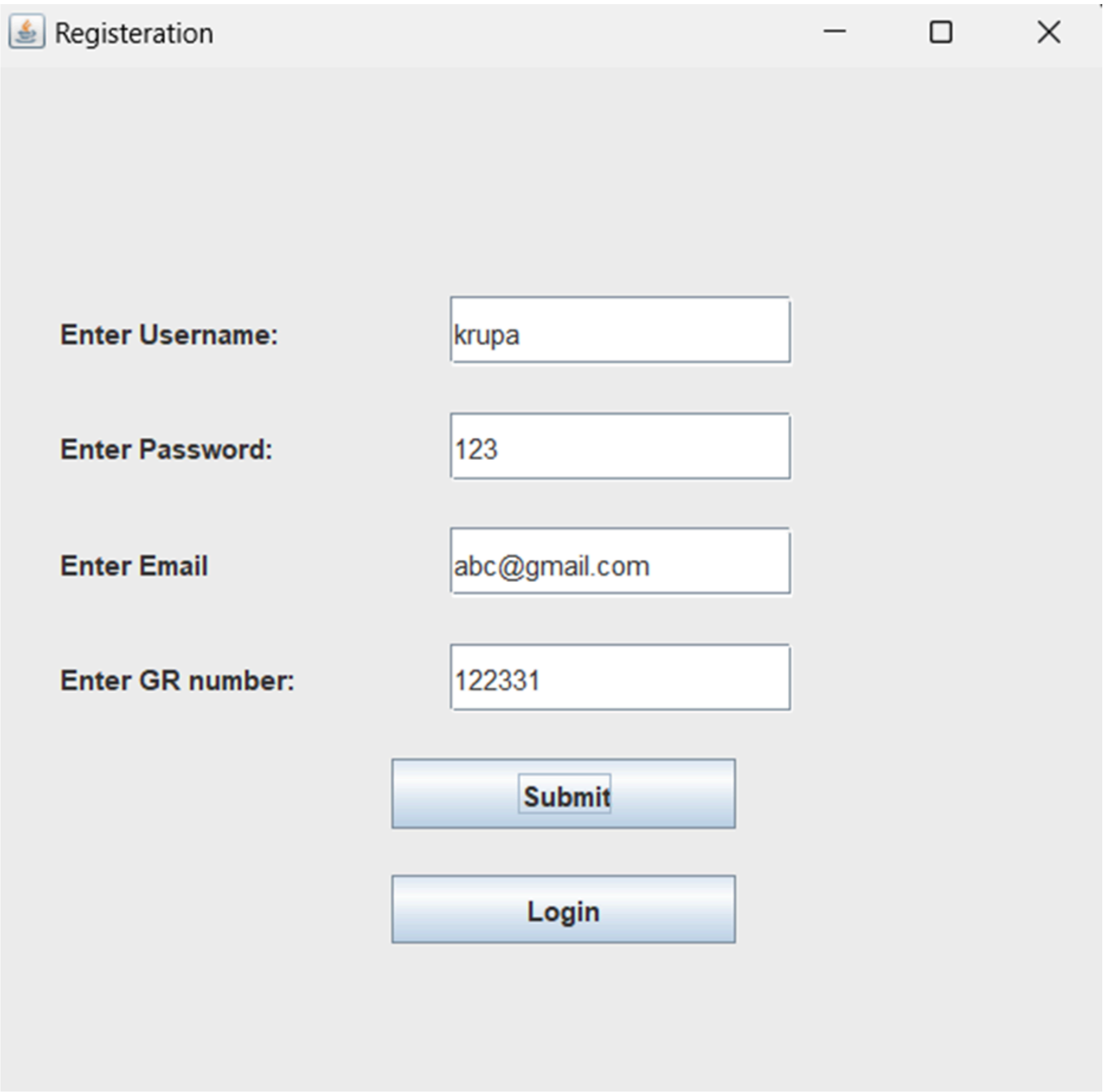
- The **Feedback** class provides a GUI for users to submit their feedback along with their username.
- Users can enter their feedback in a text area and their username in a text field.
- Feedback records are inserted into the database upon submission.
- Users can navigate back to the homepage using the home button.

Advantages

1. **Efficient Resource Management**: The Mess Management System optimizes the allocation of resources such as food ingredients, utensils, and staff. By automating processes like menu planning, attendance tracking, and feedback collection, it ensures that resources are utilized effectively, minimizing waste and reducing operational costs.
2. **Improved User Experience**: The system enhances the overall dining experience for users by providing them with convenient features such as online registration, menu browsing, and feedback submission. Users can easily access information about meal schedules, menus, and provide feedback, resulting in a seamless and user-friendly experience.
3. **Enhanced Communication**: The system facilitates better communication between mess administrators, staff, and users. Through features like real-time notifications, announcements, and feedback analysis, stakeholders can stay informed and address issues promptly. This fosters a collaborative environment and improves transparency within the mess management system.
4. **Data-Driven Decision Making**: By collecting and analyzing data on user preferences, attendance patterns, and feedback, the system enables administrators to make informed decisions. They can identify trends, anticipate demand, and tailor menus and services to meet the diverse needs of users. This data-driven approach enhances the quality of service and ensures customer satisfaction.
5. **Streamlined Operations**: The automation of routine tasks such as attendance tracking, menu planning, and feedback management streamlines operations and reduces administrative burden. Mess staff can focus on delivering high-quality meals and services without being bogged down by manual paperwork and repetitive tasks, leading to increased productivity and efficiency.
6. **Scalability and Adaptability**: The Mess Management System is designed to be scalable and adaptable to different environments and requirements. Whether it's a small hostel mess or a large institutional cafeteria, the system can be customized to suit the specific needs and preferences of the users. Additionally, it can accommodate future expansions and modifications with ease, ensuring long-term viability and sustainability.

Screenshots

1. First we will run Registration.java file which will result in the frame shown below to be visible. We will then enter the credentials.



Registeration

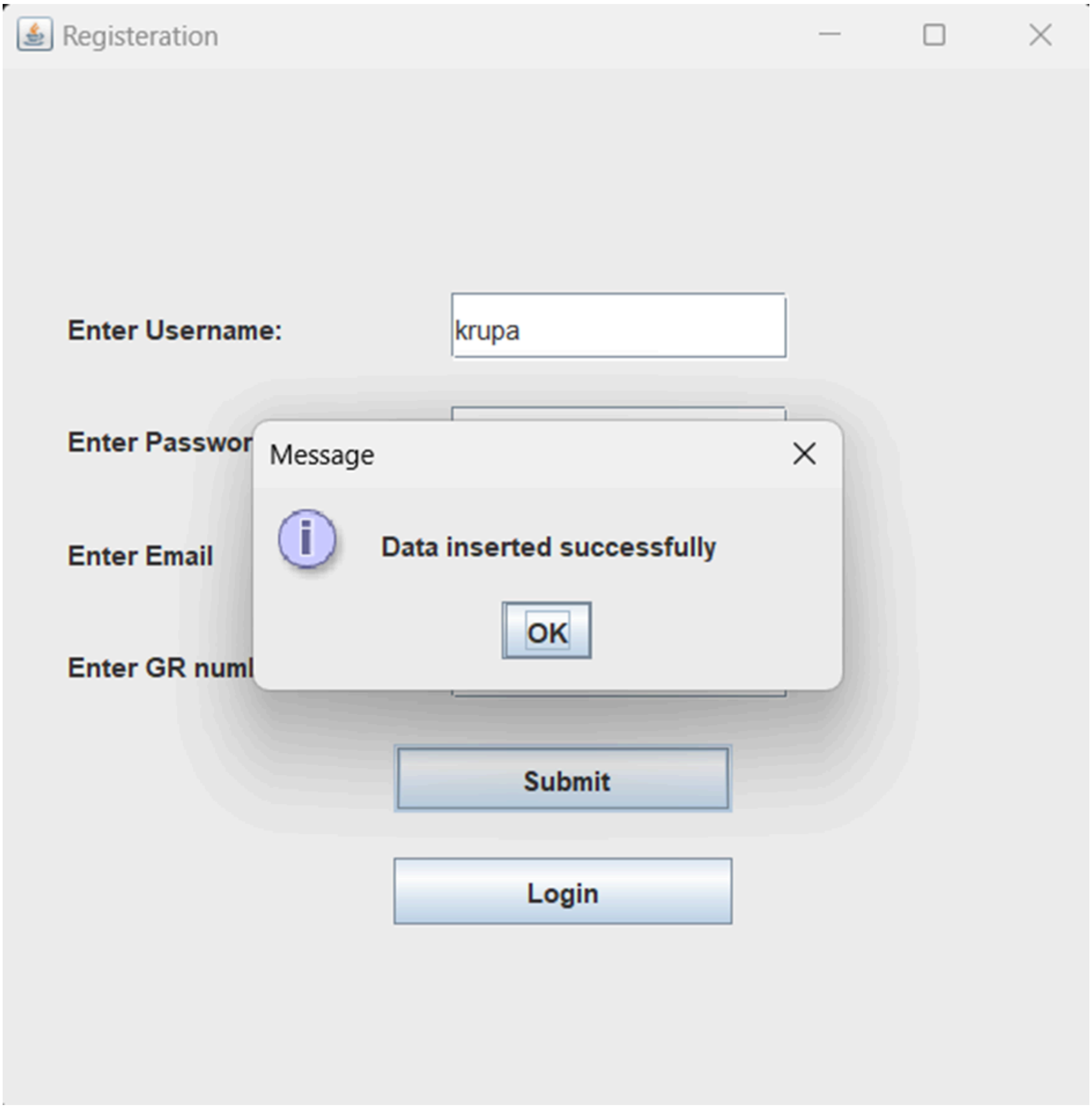
Enter Username: krupa

Enter Password: 123

Enter Email abc@gmail.com

Enter GR number: 122331

2. After clicking the submit button we will see a message “Data inserted successfully”. Then we will be directed to the Login frame. We can also see it being reflected in the database.


Registeration

Enter Username: krupa

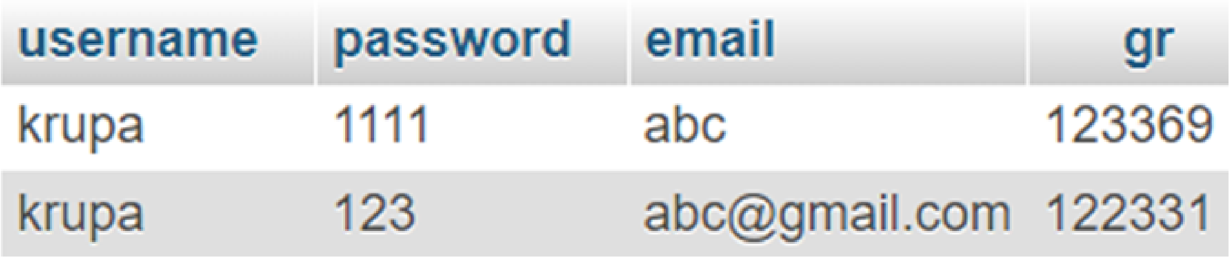
Enter Passwor

Enter Email

Enter GR numl

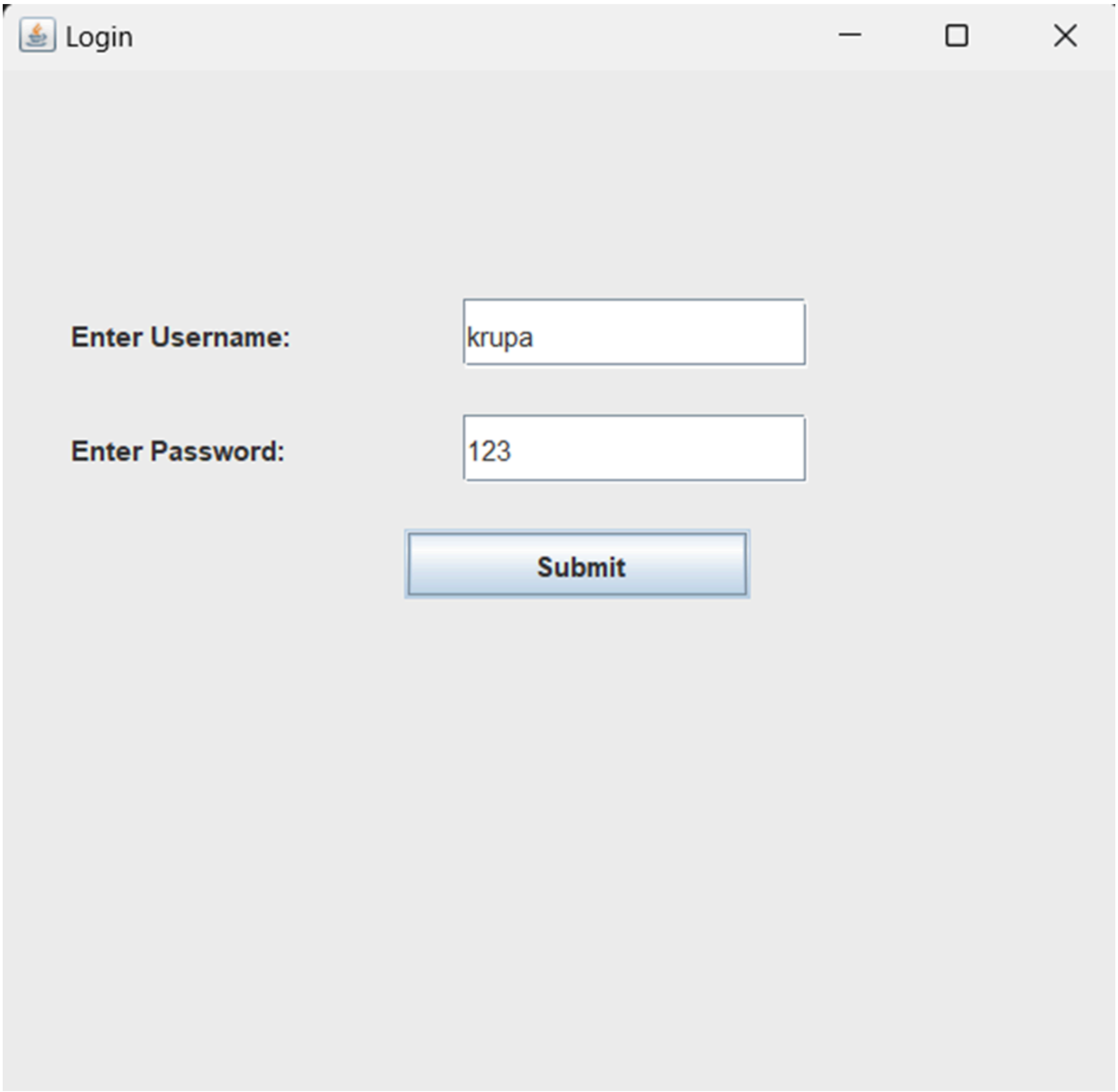
Message

Data inserted successfully



username	password	email	gr
krupa	1111	abc	123369
krupa	123	abc@gmail.com	122331

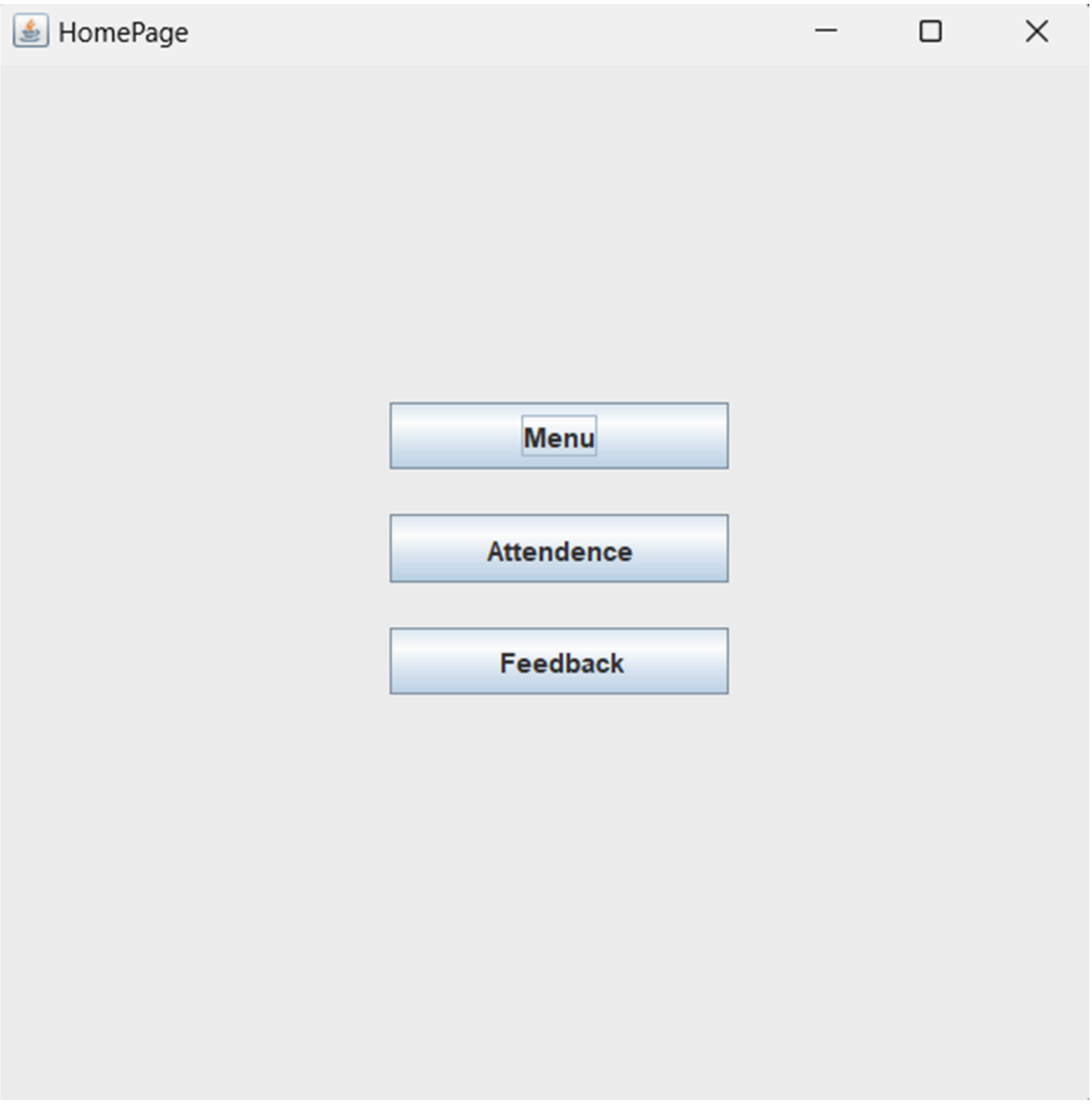
3. Then in the Login Frame insert your credentials, and then we will be directed to the HomePage frame.



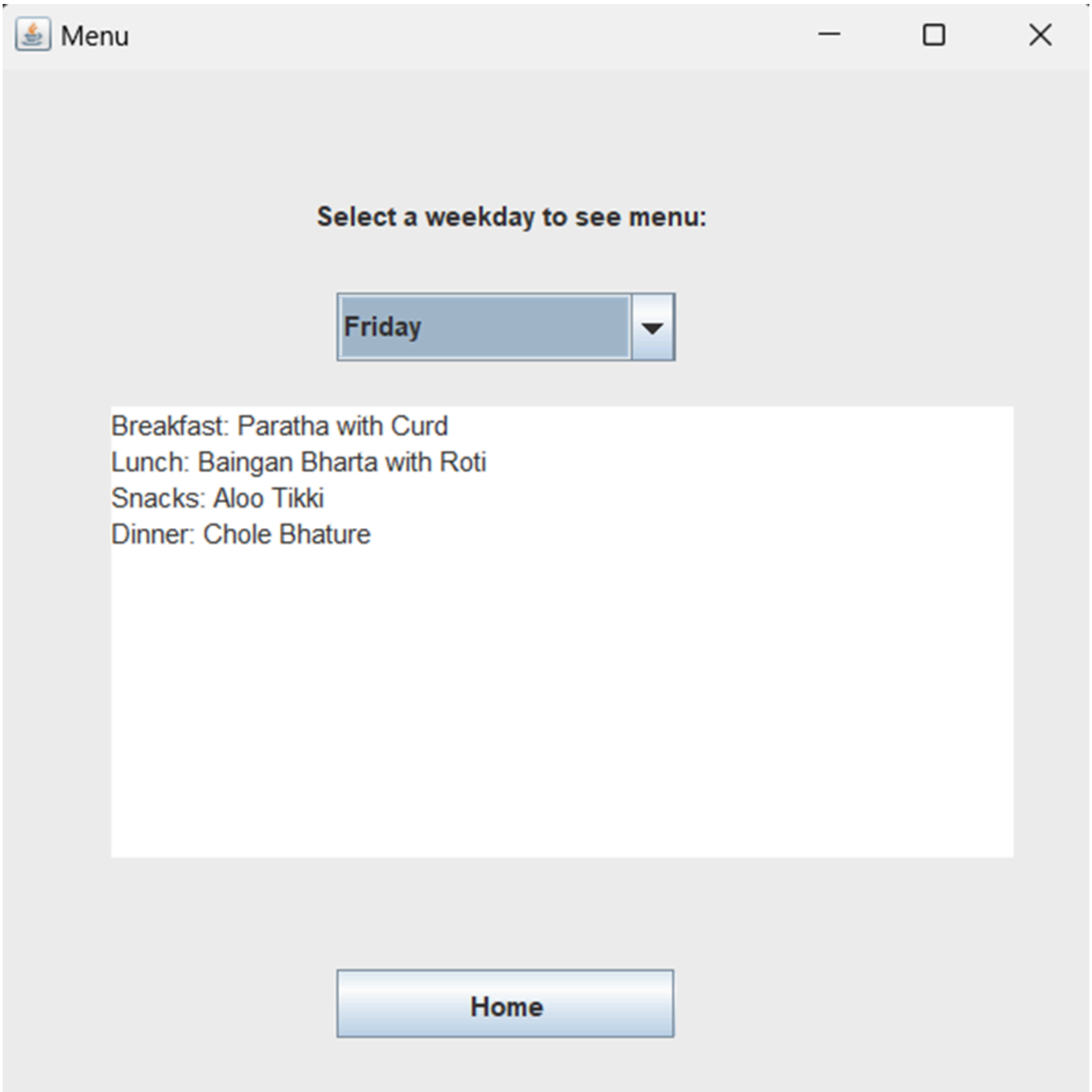
Enter Username: krupa

Enter Password: 123

4. In the HomePage frame we will see three buttons: "Menu" , "Attendance" and "Feedback".

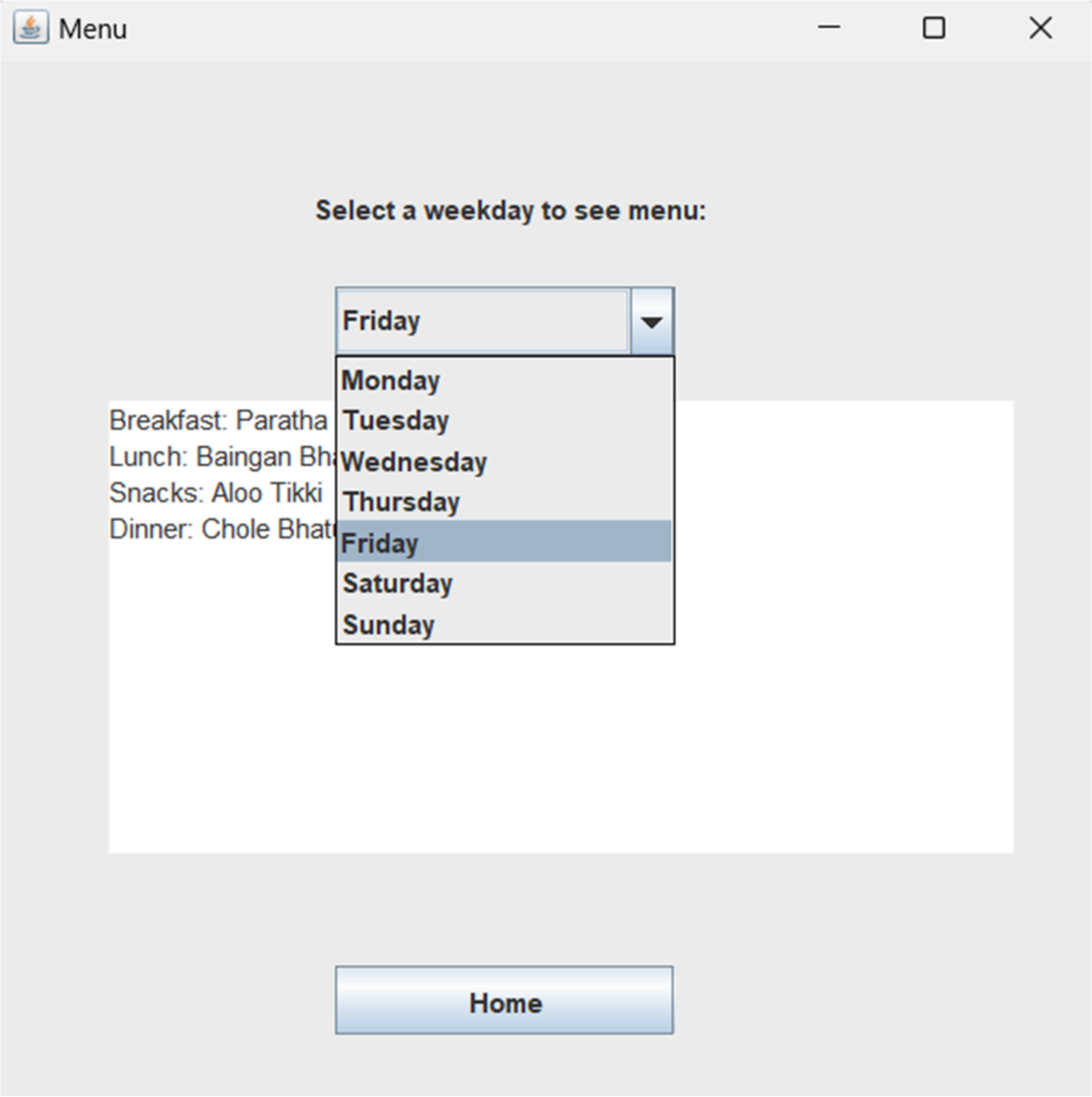


5. If you press the “Menu” button, you will see a Menu frame showing the menu for that day of the week (by default the date will be taken from your device).



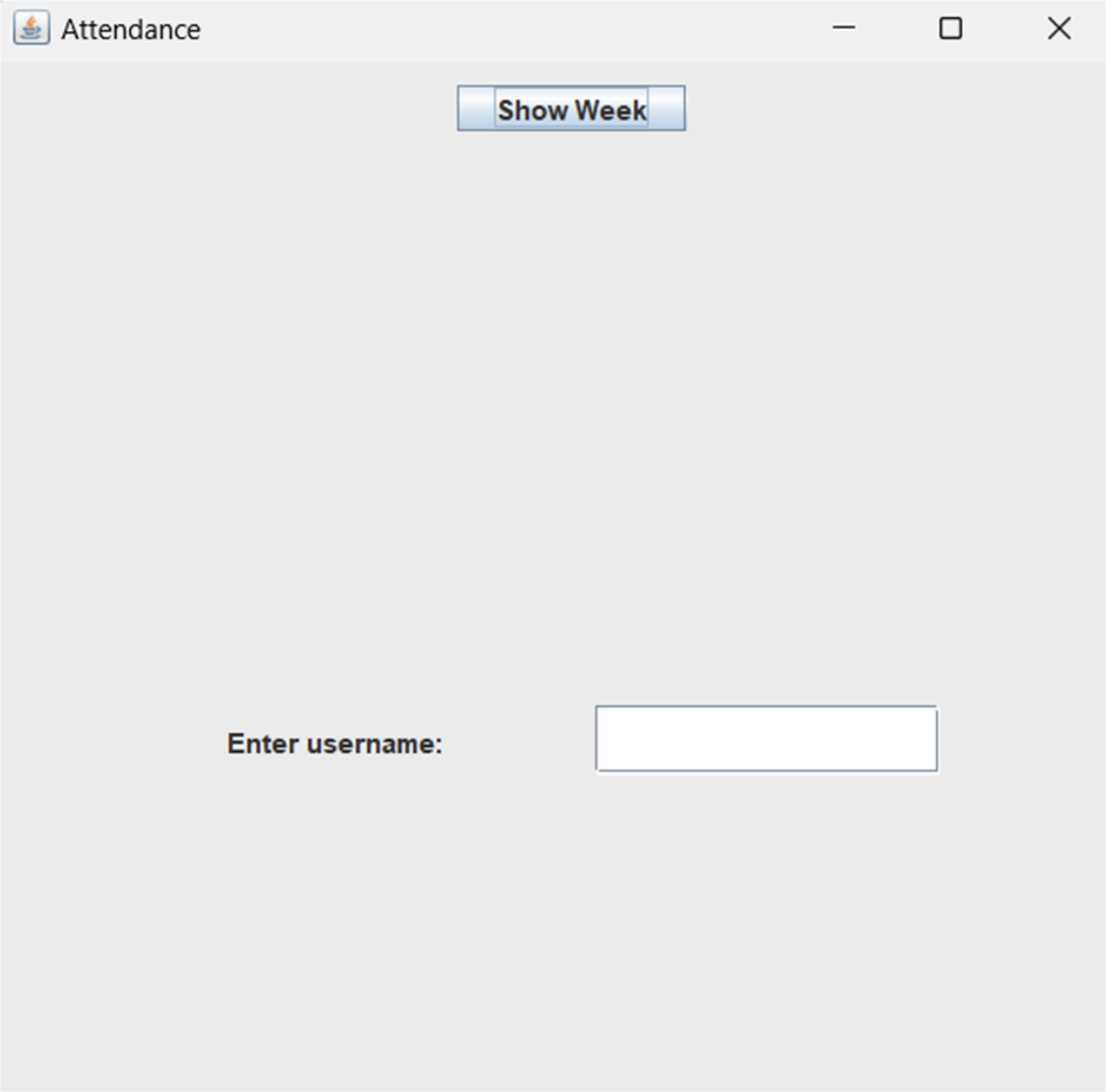
Breakfast: Paratha with Curd
Lunch: Baingan Bharta with Roti
Snacks: Aloo Tikki
Dinner: Chole Bhature

You can also see a Combo box which shows the drop down list showing all days of the week. You can select a day and the menu of that day will be shown to you.

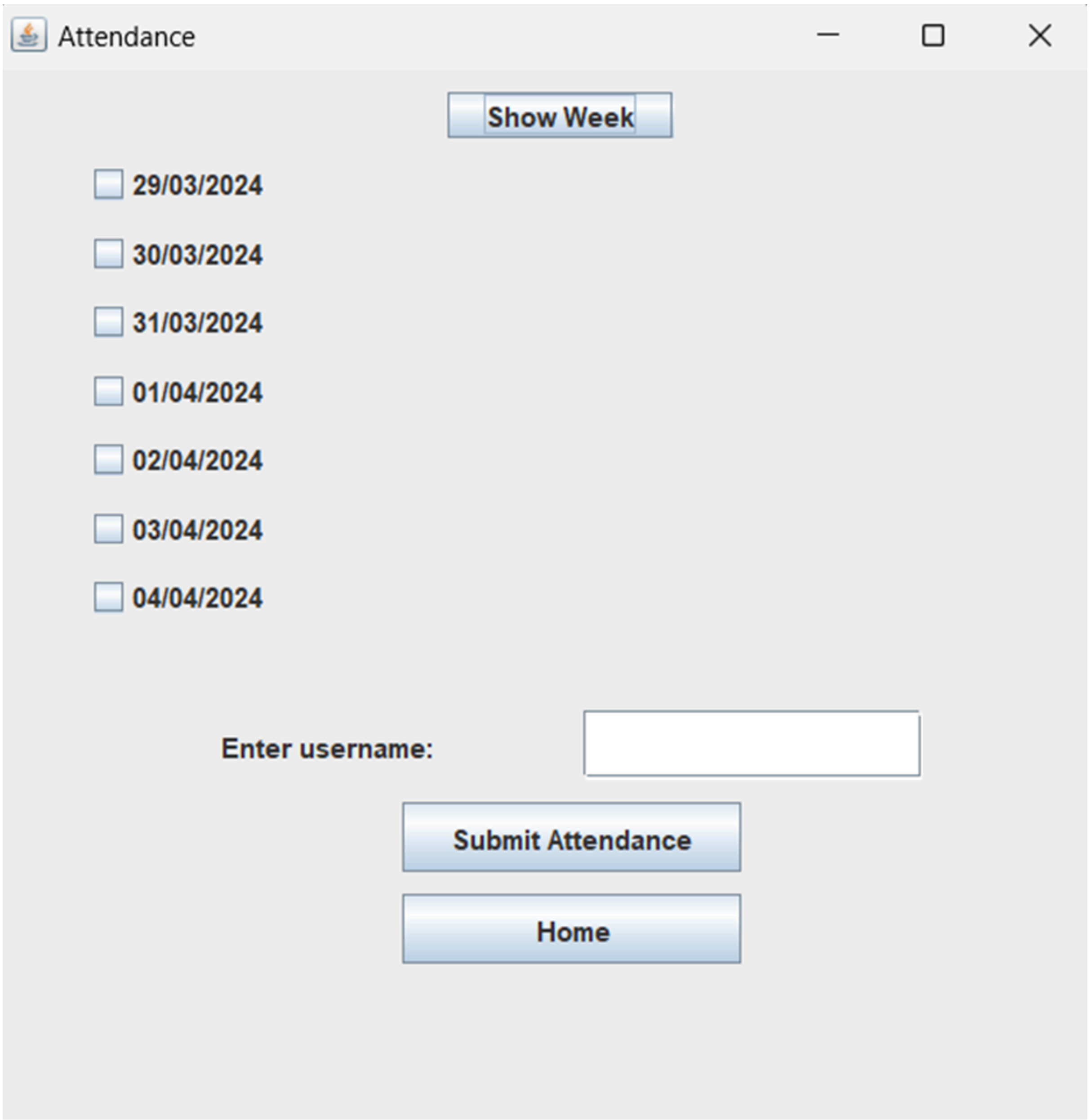


Breakfast: Paratha
Lunch: Baingan Bh
Snacks: Aloo Tikki
Dinner: Chole Bhat

6. If you press the Attendance button you will initially see only a button named “Show Week” , a label named “Enter username” and a textfield.



After pressing the “Show Week” button we will see seven checkboxes containing the current date and date of next seven days, as well as a “Submit Attendance” button as well a “Home” button.



□ 29/03/2024

□ 30/03/2024

□ 31/03/2024

□ 01/04/2024

□ 02/04/2024

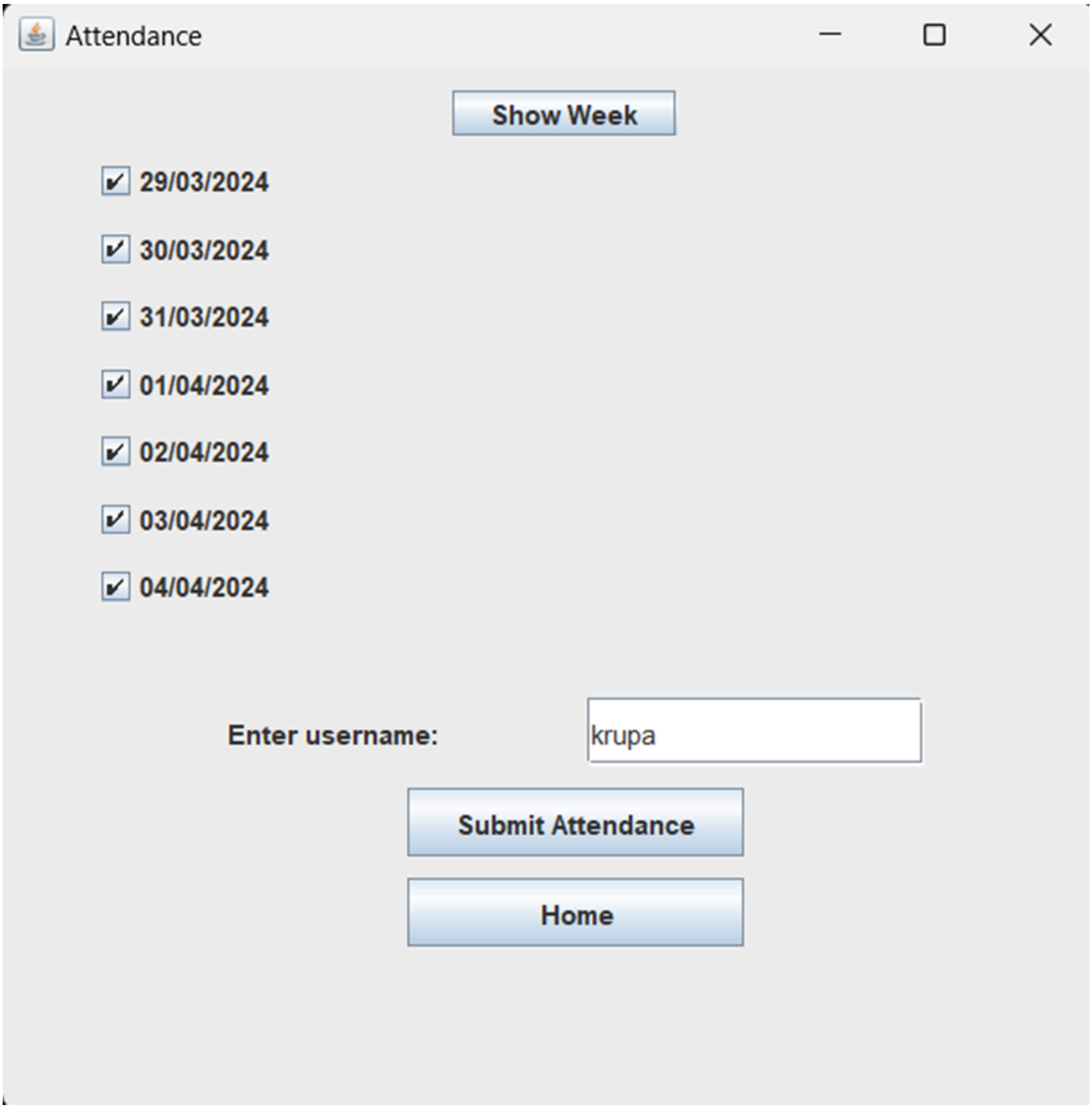
□ 03/04/2024

□ 04/04/2024

Enter username:

As you can see below we can select multiple checkboxes at once and can submit them. By doing this it will indicate that the user is going to attend a mess for the dates selected..We will also see a message "Attendance marked successfully".

We can also see it reflected in the database.



☑ 29/03/2024

☑ 30/03/2024

☑ 31/03/2024

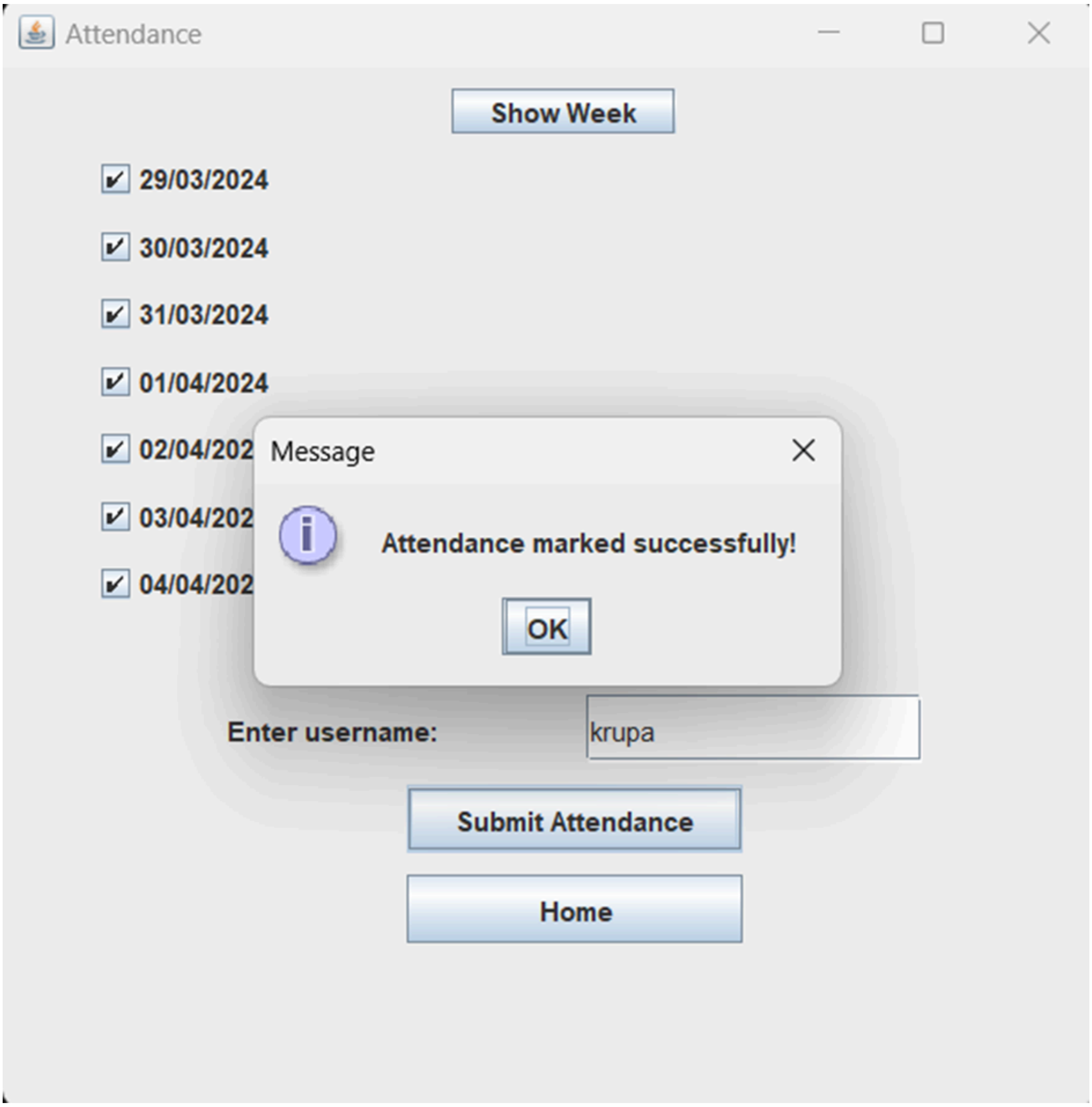
☑ 01/04/2024

☑ 02/04/2024

☑ 03/04/2024

☑ 04/04/2024

Enter username: krupa



☑ 29/03/2024

☑ 30/03/2024

☑ 31/03/2024

☑ 01/04/2024

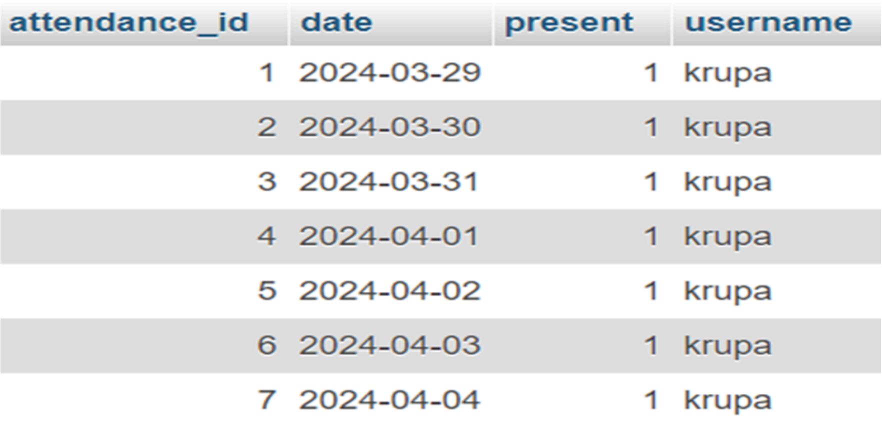
☑ 02/04/202

☑ 03/04/202

☑ 04/04/202

Attendance marked successfully!

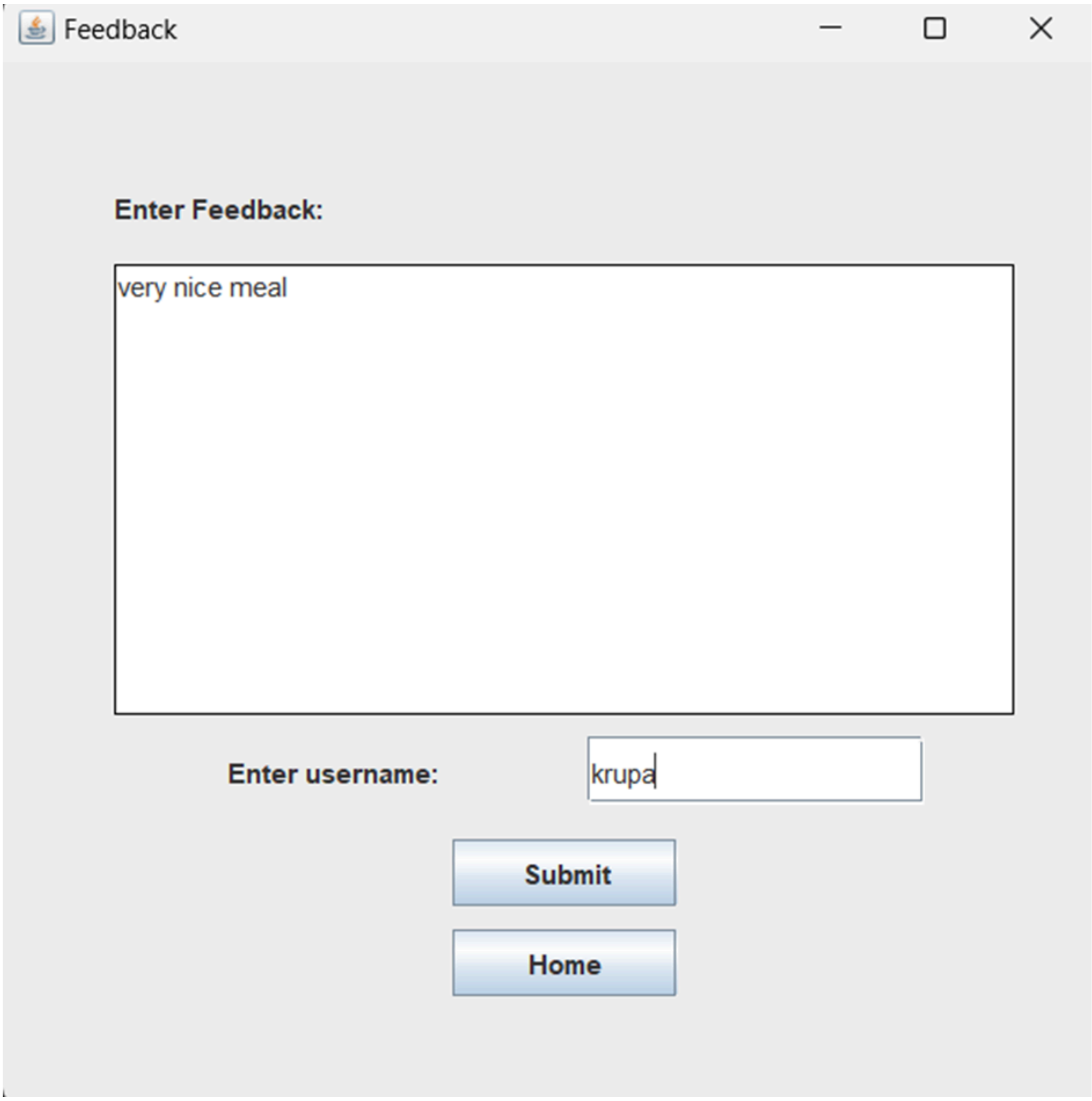
Enter username: krupa



attendance_id	date	present	username
1	2024-03-29	1	krupa
2	2024-03-30	1	krupa
3	2024-03-31	1	krupa
4	2024-04-01	1	krupa
5	2024-04-02	1	krupa
6	2024-04-03	1	krupa
7	2024-04-04	1	krupa

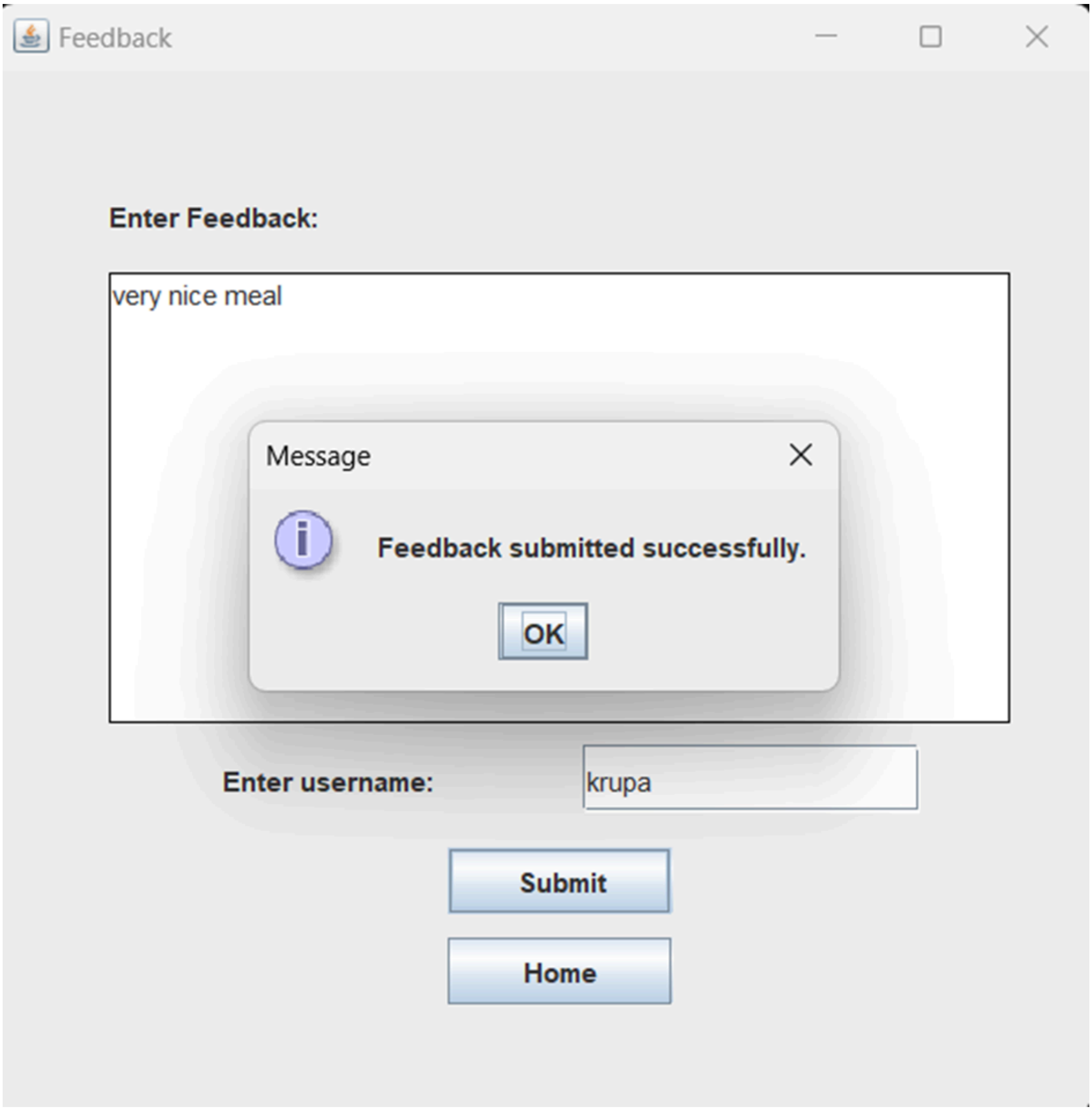
7. If you press the “Feedback” button you will be directed to the Feedback frame. Where you can enter a feedback as well your username.We can also see a message “Feedback submitted successfully.”

   You can see it being reflected in the database.



very nice meal

Enter username: krupa

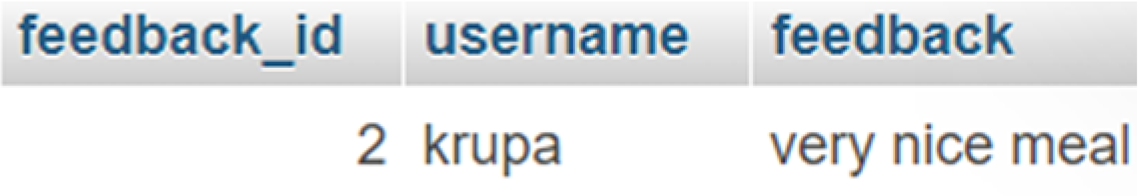


Enter Feedback:

very nice meal

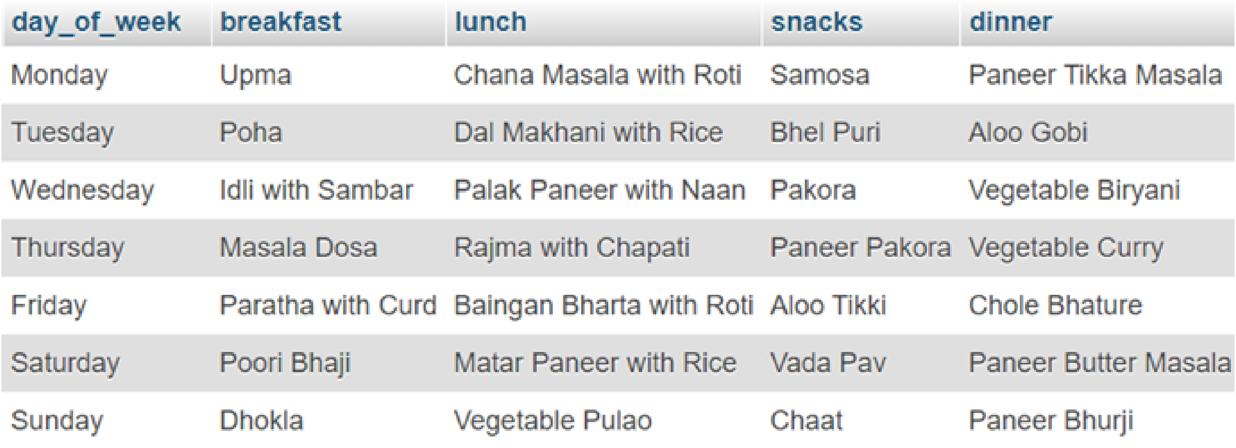
Feedback submitted successfully.

Enter username: krupa



feedback_id	username	feedback
2	krupa	very nice meal

Picture of Menu Table in database



day_of_week	breakfast	lunch	snacks	dinner
Monday	Upma	Chana Masala with Roti	Samosa	Paneer Tikka Masala
Tuesday	Poha	Dal Makhani with Rice	Bhel Puri	Aloo Gobi
Wednesday	Idli with Sambar	Palak Paneer with Naan	Pakora	Vegetable Biryani
Thursday	Masala Dosa	Rajma with Chapati	Paneer Pakora	Vegetable Curry
Friday	Paratha with Curd	Baingan Bharta with Roti	Aloo Tikki	Chole Bhature
Saturday	Poori Bhaji	Matar Paneer with Rice	Vada Pav	Paneer Butter Masala
Sunday	Dhokla	Vegetable Pulao	Chaat	Paneer Bhurji

References:

[1] Rohra, V., Sukhija, A., Lalwani, N., & Karare, A. (Year). Mess Management System Implementation. International Journal of Engineering Research & Technology, 23(24), 1-4. Retrieved from https://www.ijert.org/research/mess-management-system-implementation-IJERTCONV3IS24003.pdf

[2] OpenAI. (2022). ChatGPT (Version 3.5). [Computer software]. Retrieved from https://openai.com/chatgpt.

[3] Borhade, S., Auti, A., Bhonde, D., Gaikwad, P., & Kasab, B. S. (Year). Mess Management System Using PHP and MySQL. International Research Journal of Modernization in Engineering Technology and Science, 5(3), 50-56. DOI: https://www.doi.org/10.56726/IRJMETS34021

[4] "Java Swing Tutorial." JavaTpoint, n.d., https://www.javatpoint.com/java-swing.

[5] "Introduction to Java Swing." GeeksforGeeks, n.d., https://www.geeksforgeeks.org/introduction-to-java-swing/.