VISHWAKARMA INSTITUTE OF INFORMATION TECHNOLOGY

Pune

Department Of Information Technology

Topic :- Restaurant Management System

Name	Roll No.	PRN No.
Vedant Kokate	233027	22211513
Rajlakshmi Kolate	233028	22210878
Vedant Patil	233045	22210536

Batch :- C2

Class :-Second Year

INTRODUCTION

A Restaurant Management System (RMS) is a comprehensive software solution designed to streamline and optimize various operations within a restaurant, ranging from order processing and inventory management to customer service and financial tracking. It acts as a centralized platform to efficiently manage all facets of a restaurant's daily activities, enhancing overall productivity and customer satisfaction. Here's a detailed introduction for your report:

1. Overview:

A Restaurant Management System integrates different processes and functionalities essential for smooth restaurant operations. It typically includes modules for order taking, kitchen management, inventory control, employee scheduling, billing, customer relationship management (CRM), and analytics.

2. Key Components:

Point of Sale (POS): The core of the RMS, where orders are placed, processed, and payments are accepted. It includes features for menu customization, order modifications, and billing.

Inventory Management: Tracks ingredients, supplies, and stock levels in real-time, helping in minimizing waste and optimizing inventory orders.

Table Management: Manages table reservations, seating arrangements, and optimizing table turnover for efficient service.

Employee Management: Handles staff schedules, payroll, performance tracking, and training.

Reporting and Analytics: Generates reports on sales, inventory usage, customer preferences, and other key metrics, aiding in decision-making and strategy planning.

3. Functionalities:

Order Processing: Allows seamless order placement, modification, and fulfillment, reducing errors and enhancing efficiency.

Menu Management: Enables easy menu updates, including adding or removing items, changing prices, and displaying special offers.

Customer Relationship Management: Stores customer data for personalized service, loyalty programs, and targeted marketing campaigns.

Kitchen Display Systems: Streamlines communication between the kitchen and front-of-house, improving order accuracy and speed.

Billing and Payment Processing: Offers various payment options and simplifies billing processes for both customers and staff.

4. Benefits:

Enhanced Efficiency: Automates tasks, reduces manual errors, and speeds up processes, resulting in faster service and happier customers.

Improved Accuracy: Minimizes order mistakes, inventory discrepancies, and billing errors.

Data-Driven Decision Making: Provides insights into sales trends, popular dishes, peak hours, and other crucial data to make informed decisions.

Cost Control: Optimizes inventory levels, reduces wastage, and helps in identifying cost-saving opportunities.

Enhanced Customer Experience: Enables personalized service, efficient table management, and quicker service, leading to higher customer satisfaction.

5. Future Trends:

Integration of AI and Machine Learning for predictive analytics.

Contactless ordering and payment systems for a more hygienic experience.

Enhanced mobile app integration for reservations, loyalty programs, and ordering.

CONCEPTS USED :-

MySQL:

Data Management: MySQL is used to create and manage the database that stores crucial information for the RMS. It stores data such as menu items, customer details, order information, reservation schedules, employee records, and more. PHP interacts with the MySQL database to retrieve or update information as needed, ensuring the system always has access to the latest data.

Integration:

These technologies work together seamlessly in the RMS. For example, when a customer places an order through the system, HTML provides the structure of the order form, CSS styles it for an attractive layout, JavaScript validates the inputs, PHP processes the order information on the server, and MySQL stores the order details in the database.

Overall Functionality:

The RMS combines these technologies to offer functionalities like:

Menu management and customization (HTML/CSS for display, PHP/MySQL for storage and updates).

Order processing and management (JavaScript for dynamic interfaces, PHP/MySQL for data handling).

Reservation systems (HTML/CSS for forms, PHP/MySQL for handling reservations).

Employee management (PHP/MySQL for storing employee records, PHP for user authentication).

Reporting and analytics (PHP/MySQL for data retrieval and analysis).

In essence, these technologies collaborate to create a comprehensive, user-friendly, and efficient system that streamlines various operations within a restaurant, from managing menus and orders to handling reservations and employee tasks

AWT :-

Absolutely, AWT (Abstract Window Toolkit) is a foundational part of Java's GUI components. It provides a set of APIs (Application Programming Interfaces) for creating and managing graphical user interfaces for Java programs. Here are some detailed aspects of AWT:

Components: AWT provides a wide range of GUI components such as buttons, text fields, labels, checkboxes, menus, etc., through classes like Button, TextField, Label, Checkbox, Menu, etc.

Layout Managers: AWT includes layout managers like FlowLayout, BorderLayout, GridLayout, etc., which help arrange components within containers effectively.

Event Handling: It offers event handling through listeners. Commonly used listeners include ActionListener, MouseListener, KeyListener, etc., to respond to user actions like mouse clicks, keypresses, etc.

Graphics and Drawing: AWT allows drawing directly onto components using the Graphics class. You can draw shapes, images, and text on components.

Platform Independence: AWT components are rendered using native components of the underlying operating system. While this helps in platform independence, it might result in different looks across different operating systems.

Heavyweight Components: AWT components are termed as heavyweight as they are directly mapped to the underlying operating system components. This can sometimes lead to performance issues and limitations in customization.

Legacy Technology: With the evolution of Java, newer GUI libraries like Swing and JavaFX have largely superseded AWT due to their enhanced features, greater flexibility, and better performance.

Integration: AWT components can be used alongside Swing components, allowing developers to leverage both libraries within the same application.

Limited Functionality: Compared to newer libraries like Swing and JavaFX, AWT provides limited functionality and customization options for modern GUI development.

While AWT laid the foundation for GUI programming in Java, it's now often used in conjunction with Swing or JavaFX for more sophisticated and modern GUI applications due to its limitations.

SOURCE CODE:-

Home Page –

```
package hotel;

public class Hotel {

   private String hotelName;
   private String hotelAddress;

        private Hotel(String hotelName,String hotelAddress)
        {
                this.hotelName=hotelName;
                this.hotelAddress=hotelAddress;
        }

        private static final Hotel INSTANCE = new Hotel("CRESCENT","HUBLI");

        public static Hotel getInstance()
        {
                return INSTANCE;
        }

}
```

LOGIN PAGE –

```
package hotel;
```

```
import java.awt.EventQueue;


import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;


import java.awt.Font;
import javax.swing.JTextField;
import javax.swing.JPasswordField;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.awt.event.ActionEvent;
import java.awt.Color;
import javax.swing.ImageIcon;

public class loginPage {


	private JFrame frame;
	private JTextField tfusername;
	private JPasswordField tfpwd;
	JLabel Ustar = new JLabel("*");
JLabel Pstar = new JLabel("*");

/**
 * Launch the application.
 */
public static void main(String[] args) {

    EventQueue.invokeLater(new Runnable() {
        Hotel hotel = Hotel.getInstance();
        public void run() {
            try {

                loginPage window = new loginPage();
                window.frame.setVisible(true);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}



public loginPage() {
    initialize();
    Ustar.setVisible(false);
    Pstar.setVisible(false);
    }


    private void initialize() {

        frame = new JFrame();
        frame.getContentPane().setFont(new Font("Times New Roman",
Font.BOLD, 23));
        frame.setBounds(50,50, 898, 624);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.getContentPane().setLayout(null);

        JLabel lblLogin = new JLabel("LOGIN ");
        lblLogin.setFont(new Font("Times New Roman", Font.BOLD, 32));
        lblLogin.setBounds(369, 190, 212, 67);
        frame.getContentPane().add(lblLogin);

        JLabel lblUsername = new JLabel("USERNAME");
        lblUsername.setFont(new Font("Times New Roman", Font.BOLD,
23));
        lblUsername.setBounds(269, 324, 155, 50);
        frame.getContentPane().add(lblUsername);

        JLabel lblPassword = new JLabel("PASSWORD");
        lblPassword.setFont(new Font("Times New Roman", Font.BOLD,
23));
        lblPassword.setBounds(280, 413, 133, 38);
        frame.getContentPane().add(lblPassword);

        tfusername = new JTextField();
        tfusername.setFont(new Font("Times New Roman", Font.BOLD,
23));
        tfusername.setBounds(489, 331, 208, 38);
        frame.getContentPane().add(tfusername);
        tfusername.setColumns(10);

        tfpwd = new JPasswordField();
        tfpwd.setFont(new Font("Times New Roman", Font.BOLD, 23));
        tfpwd.setBounds(489, 414, 208, 38);
        frame.getContentPane().add(tfpwd);

        JButton btnLogin = new JButton("LOGIN");
        btnLogin.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {

                Ustar.setVisible(false);
                Pstar.setVisible(false);
                if(tfusername.getText().equals(""))
                {
                    Ustar.setVisible(true);
                }
```

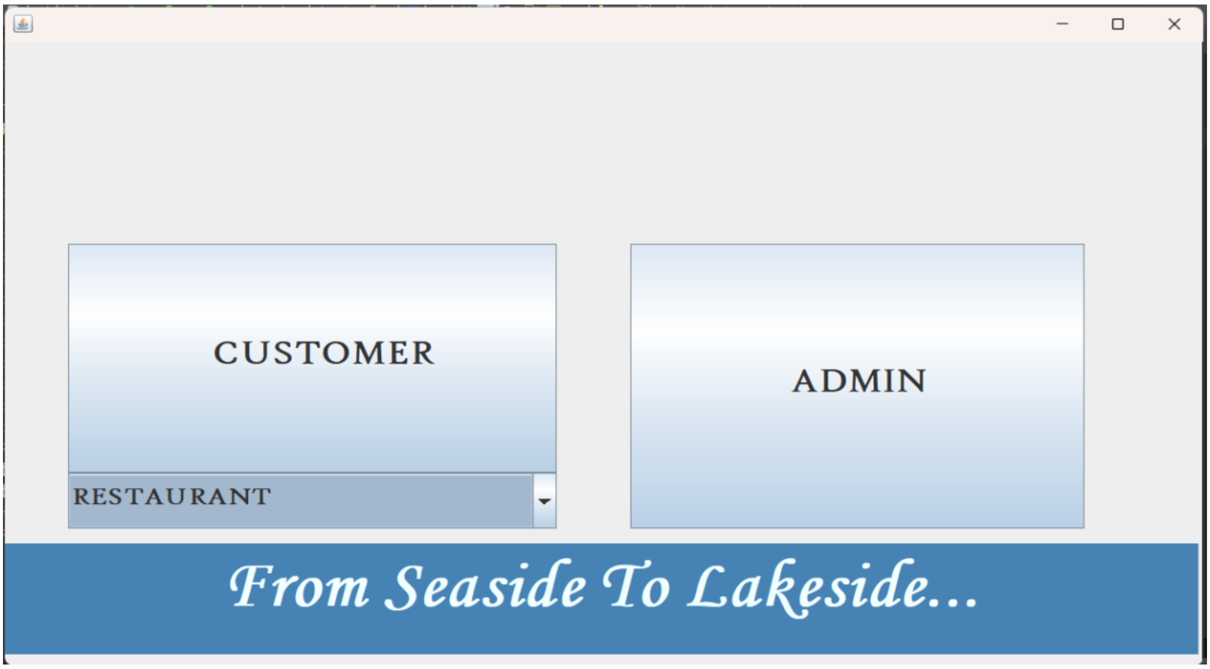
```
                 if(String.valueOf(tfpwd.getPassword()).equals(""))
                {
                     Pstar.setVisible(true);
                }
                 else
                 {
                     //String url,user,pwd;
                     GetConnection connect=new GetConnection();
                         Connection
conn=connect.getConnection();
                         try{
                             //Connection connect
=DriverManager.getConnection(url, user, pwd);
                             //System.out.println("Connection
success");
                             PreparedStatement
ps=conn.prepareStatement("SELECT * FROM login WHERE username= ? AND
password=? " );
                             ps.setString(1,
tfusername.getText());

    ps.setString(2,String.valueOf(tfpwd.getPassword()));


                             ResultSet rs=ps.executeQuery();
                             if(rs.next())
                             {
                                 secondPage sp = new
secondPage();
```

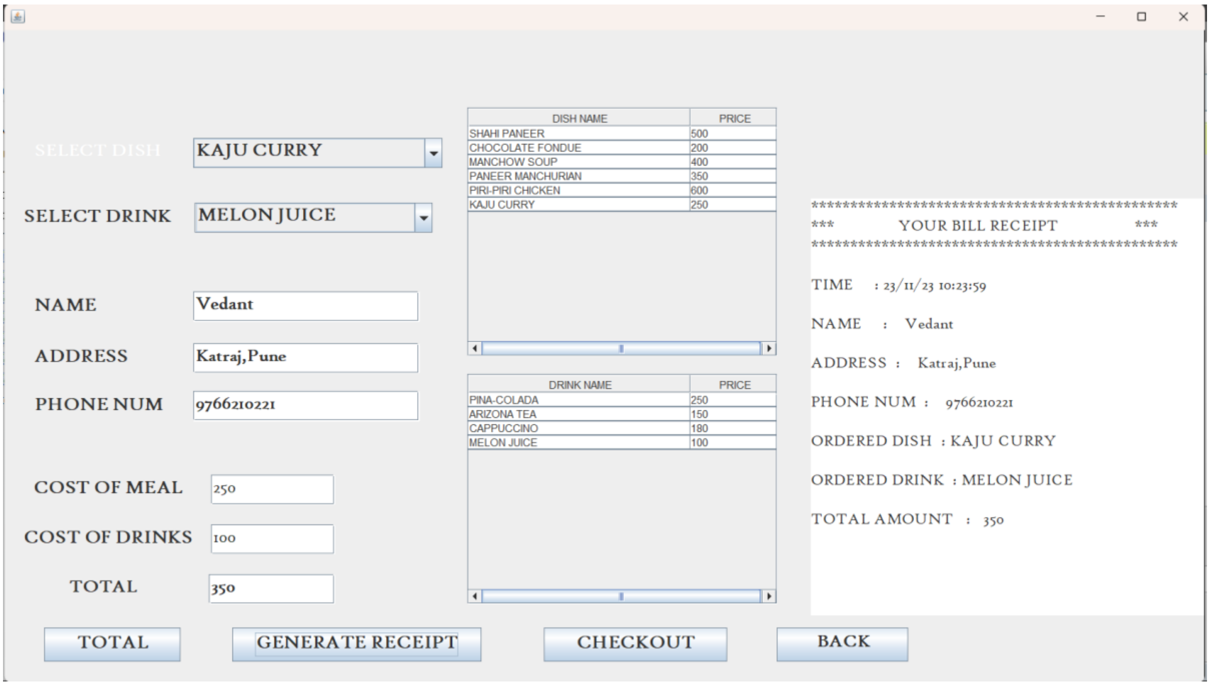
```
                                        sp.setVisible(true);
                                        sp.pack();
                                        sp.setLocationRelativeTo(null);

                                        sp.setBounds(50,50, 1015, 574);
                                        frame.setVisible(false);


                                    }
                                        else
                                JOptionPane.showMessageDialog(null,
"Error", "Please check user name / password",JOptionPane.ERROR_MESSAGE);


                            }
                                    catch(SQLException p)
                                    {
                                        p.printStackTrace();
                                    }
                    }
                }
            });
            btnLogin.setFont(new Font("Times New Roman", Font.BOLD, 23));
            btnLogin.setBounds(288, 513, 147, 50);
            frame.getContentPane().add(btnLogin);

            JButton btnCancel = new JButton("CANCEL");
```

```
        btnCancel.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                System.exit(0);
            }
        });

        btnCancel.setFont(new Font("Times New Roman", Font.BOLD,
23));
        btnCancel.setBounds(565, 513, 155, 51);
        frame.getContentPane().add(btnCancel);



        Ustar.setForeground(Color.RED);
        Ustar.setFont(new Font("Times New Roman", Font.BOLD, 28));
        Ustar.setBounds(699, 347, 46, 21);
        frame.getContentPane().add(Ustar);



        Pstar.setForeground(Color.RED);
        Pstar.setFont(new Font("Times New Roman", Font.BOLD, 28));
        Pstar.setBounds(699, 430, 46, 21);
        frame.getContentPane().add(Pstar);

        JLabel label = new JLabel("");
        label.setIcon(new ImageIcon("images\\login (8).png"));
        label.setBounds(12, 234, 314, 283);
        frame.getContentPane().add(label);
```

```
                    JLabel lblHotelManagementSystem = new JLabel("HOTEL MANAGEMENT SYSTEM");
                    lblHotelManagementSystem.setFont(new Font("Monotype Corsiva", Font.BOLD, 40));
                    lblHotelManagementSystem.setBounds(109, 72, 636, 80);
                    frame.getContentPane().add(lblHotelManagementSystem);
          }
}
```

OUTPUT :-





SELECT DISH: KAJU CURRY

SELECT DRINK: MELON JUICE

NAME: Vedant

ADDRESS: Katraj,Pune

PHONE NUM: 9766210221

COST OF MEAL: 250

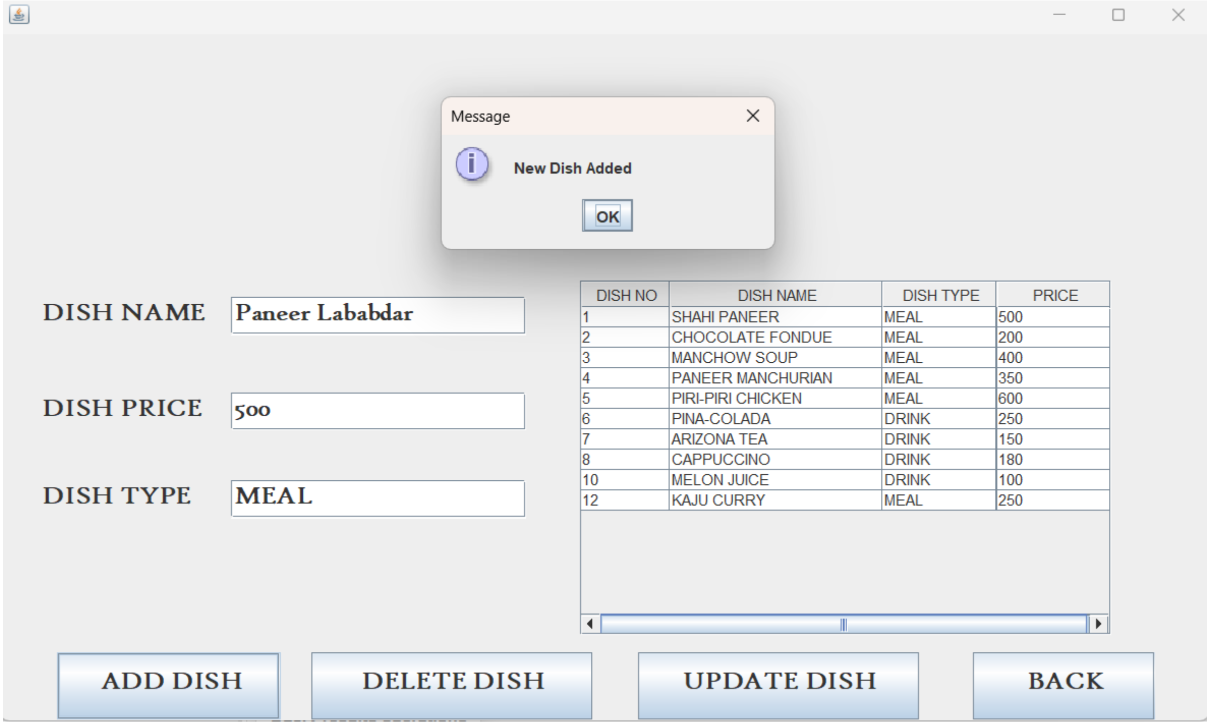
COST OF DRINKS: 100

TOTAL: 350

DISH NAME	PRICE
SHAHI PANEER	500
CHOCOLATE FONDUE	200
MANCHOW SOUP	400
PANEER MANCHURIAN	350
PIRI-PIRI CHICKEN	600
KAJU CURRY	250

DRINK NAME	PRICE
PINA-COLADA	250
ARIZONA TEA	150
CAPPUCCINO	180
MELON JUICE	100

```
*************************************************
***          YOUR BILL RECEIPT            ***
*************************************************

TIME    : 23/11/23 10:23:59

NAME    :   Vedant

ADDRESS :   Katraj,Pune

PHONE NUM :   9766210221

ORDERED DISH  : KAJU CURRY

ORDERED DRINK  : MELON JUICE

TOTAL AMOUNT   :   350
```



Message

New Dish Added

DISH NAME: Paneer Lababdar

DISH PRICE: 500

DISH TYPE: MEAL

DISH NO	DISH NAME	DISH TYPE	PRICE
1	SHAHI PANEER	MEAL	500
2	CHOCOLATE FONDUE	MEAL	200
3	MANCHOW SOUP	MEAL	400
4	PANEER MANCHURIAN	MEAL	350
5	PIRI-PIRI CHICKEN	MEAL	600
6	PINA-COLADA	DRINK	250
7	ARIZONA TEA	DRINK	150
8	CAPPUCCINO	DRINK	180
10	MELON JUICE	DRINK	100
12	KAJU CURRY	MEAL	250

CONCLUSION :-

A Restaurant Management System (RMS) is a comprehensive solution that revolutionizes the way restaurants operate. By consolidating essential tasks such as order processing, inventory management, and customer service into a centralized platform, RMS significantly enhances efficiency and productivity. This automation minimizes errors in tasks like order taking, facilitates seamless kitchen coordination, and optimizes inventory levels, ultimately reducing waste and resource mismanagement. Furthermore, the system's analytical capabilities provide valuable insights into sales trends, popular menu items, and customer preferences. These insights empower restaurant owners and managers to make informed decisions regarding menu adjustments, pricing strategies, and staffing needs, thereby contributing to improved profitability and operational success.

Customer experience lies at the heart of RMS functionality. By enabling quick service, personalized interactions, and efficient table management, these systems prioritize customer satisfaction. Modern RMS solutions leverage a combination of HTML, CSS, JavaScript, PHP, and MySQL to create user-friendly interfaces and manage complex restaurant operations. Their adaptability and scalability ensure seamless integration with evolving technological advancements and changing business requirements, accommodating menu updates and expansions. Looking ahead, future trends in RMS include integration with artificial intelligence (AI) for predictive analysis, contactless systems for hygiene and convenience, and enhanced mobile app features to cater to evolving consumer preferences.

Overall, an RMS transcends its role as a mere management tool; it serves as a strategic asset driving growth, sustainability, and customer loyalty within the competitive restaurant industry. As a pivotal technological solution, RMS represents a transformative force revolutionizing restaurant operations and positioning establishments for sustained success in the dynamic and demanding hospitality market.