Pizza Invoicing Software

By Vanshaj Pahwa(40615003118)

ABOUT

- This project is written in Java programming language. It is designed to use in Pizza Restaurants to generate invoices for each order.
- It uses Java Swing for displaying a GUI(Graphical User Interface) and MySQL connected through a JDBC driver to hold records of the customers, employees and the products.
- Invoices generated can be given as a receipt to the customer.

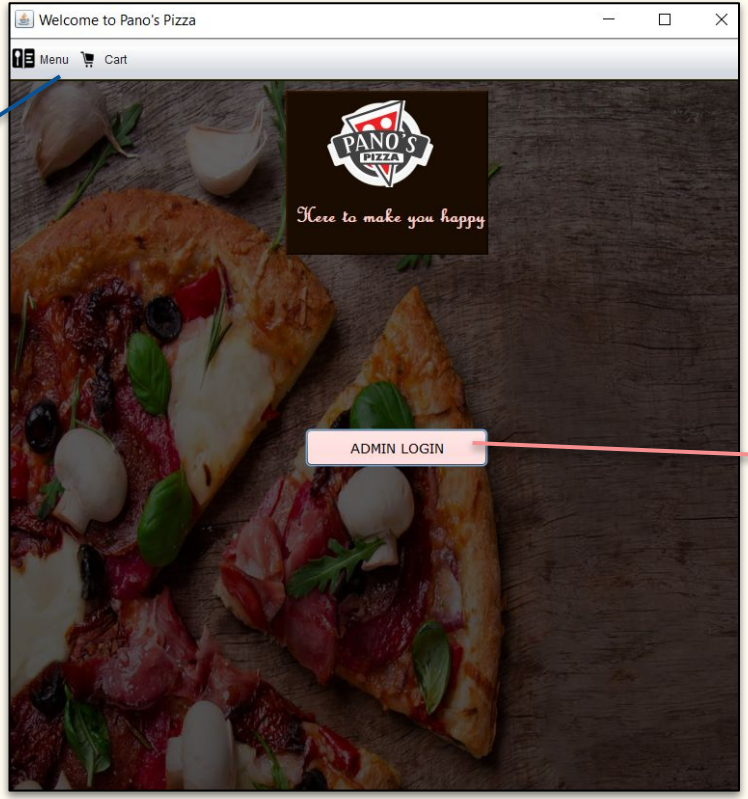
Technology Used

1. **JAVA** : Java is a class-based, object-oriented programming language that is designed to produce softwares on multiple platforms.
2. **Swing** : Java Swing is a part of Java Foundation Classes (JFC) that is used to create *window-based applications*. It is built on the top of AWT (Abstract Windowing Toolkit) API and entirely written in java.
3. **JDBC** : JDBC stands for Java Database Connectivity. JDBC is a Java API(Application Programming Interface) used to connect and execute the query with the database.
4. **MySQL** : MySQL is an open-source relational database management system (RDBMS) based on Structured Query Language (SQL). MySQL is used to add, access, and process data stored in a database.

Structure of the Project

- Welcome Screen consists of **View Menu** , **Cart** and **ADMIN LOGIN.**
- New users can register and existing users can login to proceed further in the application.
- Logged in users can add the required pizza to the cart in **Order Now** Screen
- After checking out, invoice is generated.
- Employee’s information can be managed in the **ADMIN WINDOW.**

Welcome Screen



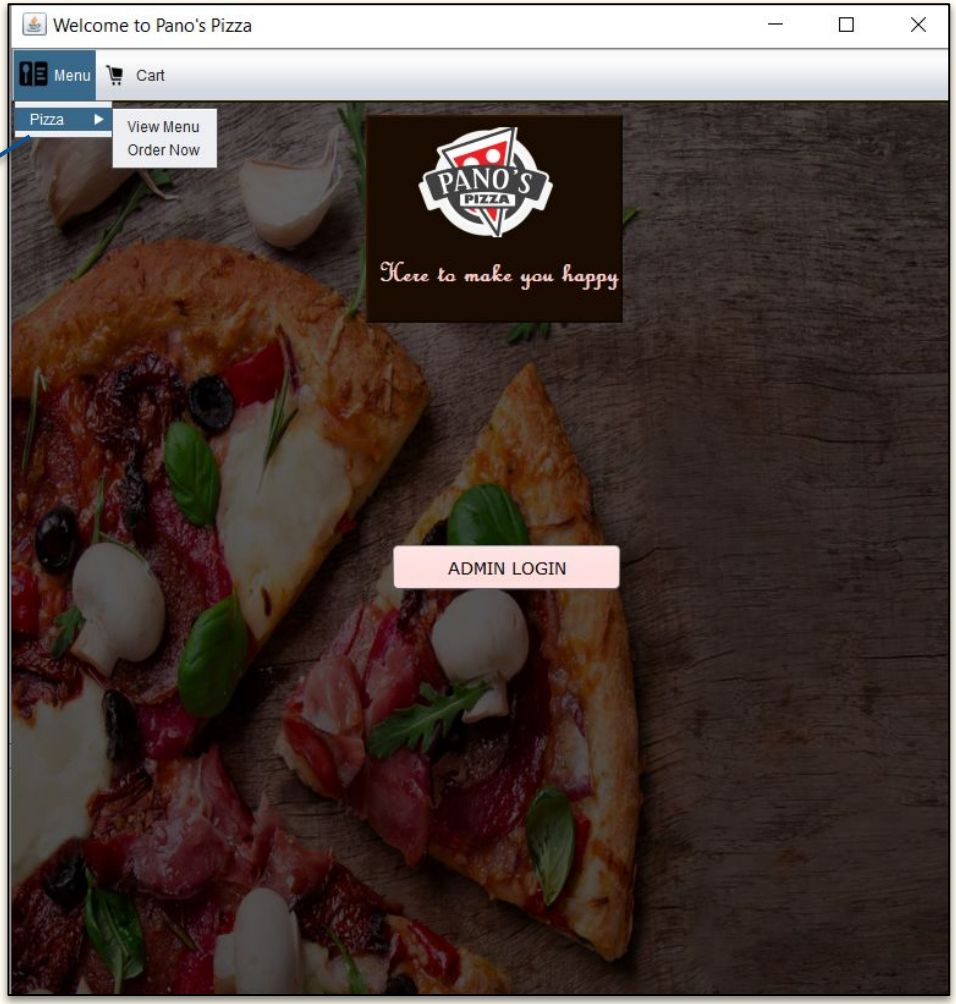
Menu Bar :

It has 2 options: Menu & Cart, user will be redirected to billing screen by clicking on 'View Cart'

Admin login:

This button will load Admin's Screen.

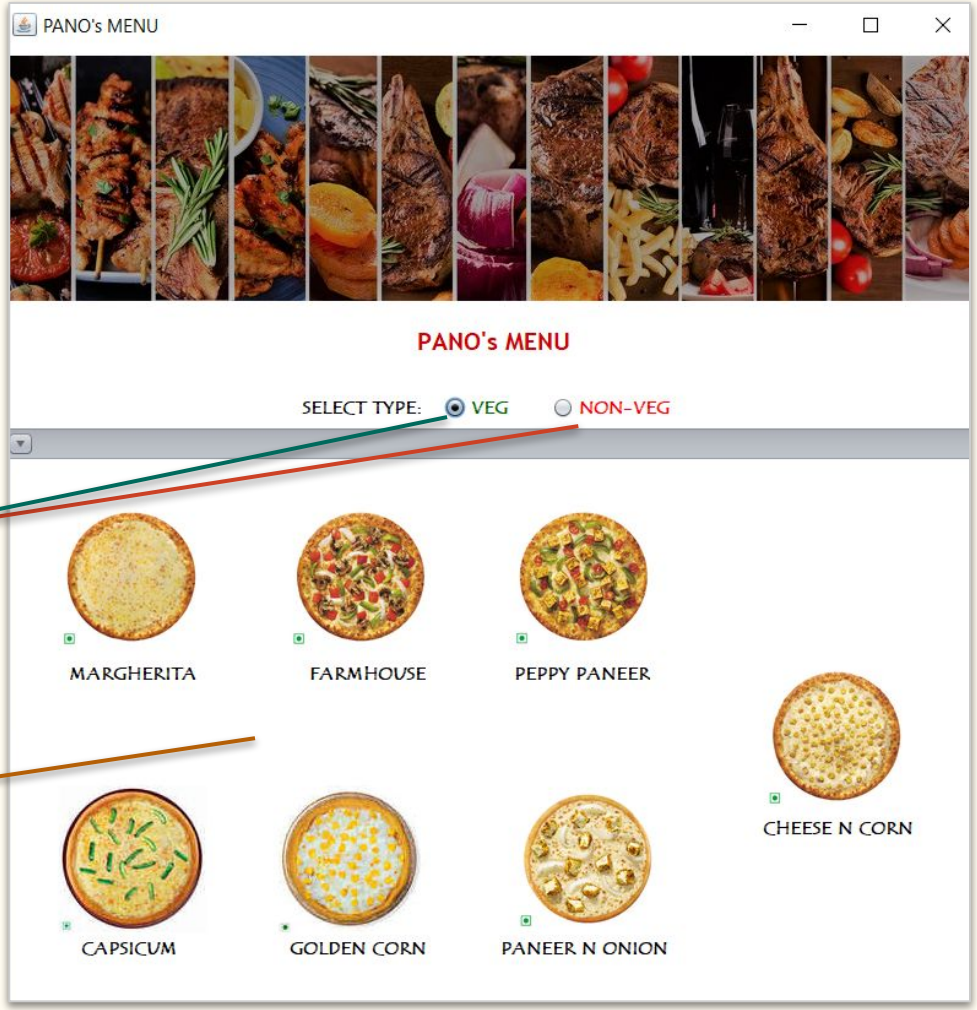
1. **<u>View Menu</u>** : Click to see what's on the restaurant menu?
2. **<u>Order Now</u>** : Click to select type, size and customisation of pizza.



1. View Menu

Radio buttons to select the type of Pizza.

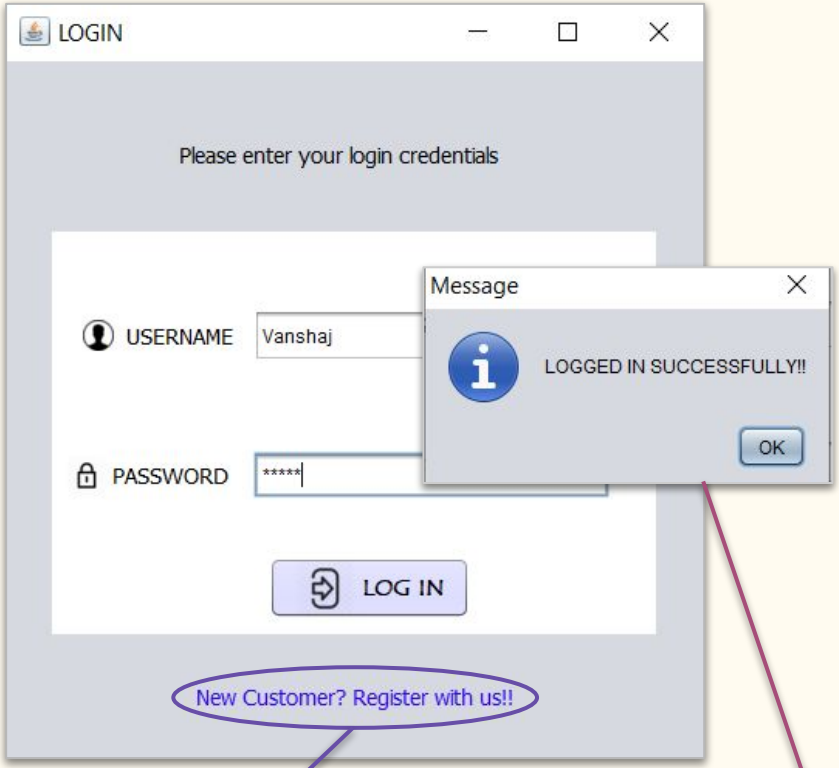
Internal frame showing Veg Pizzas

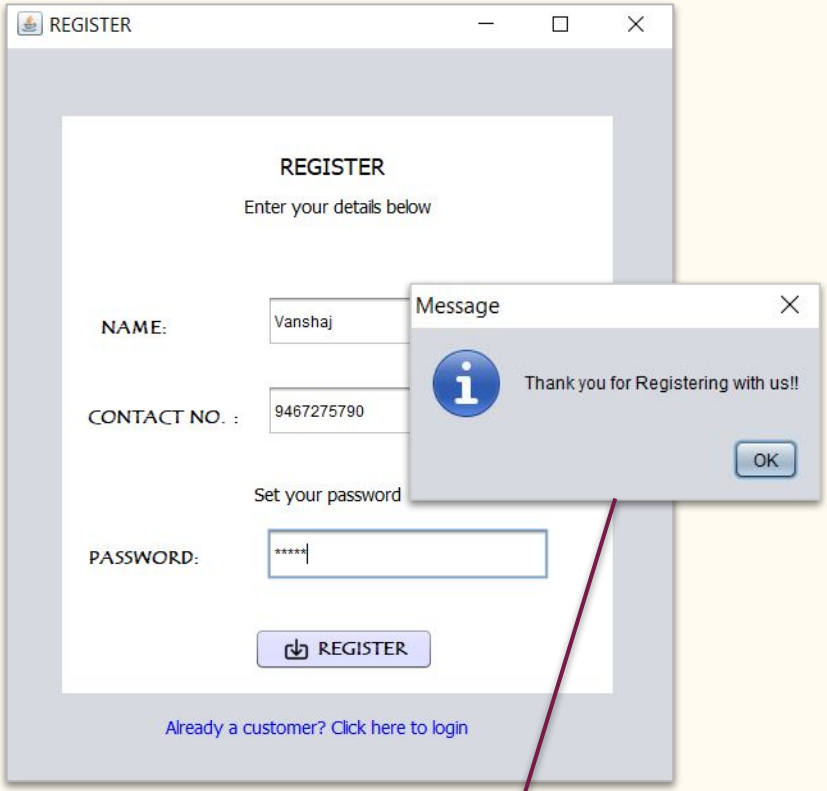


2. Order Now

- Before adding Pizza to the cart, the users who are visiting the application for the first time will have to provide their Name, Contact No. and Password in order to authenticate them to grant access to the further process.
  - Register form is connected to MySQL by JDBC to store the details of the user in database.

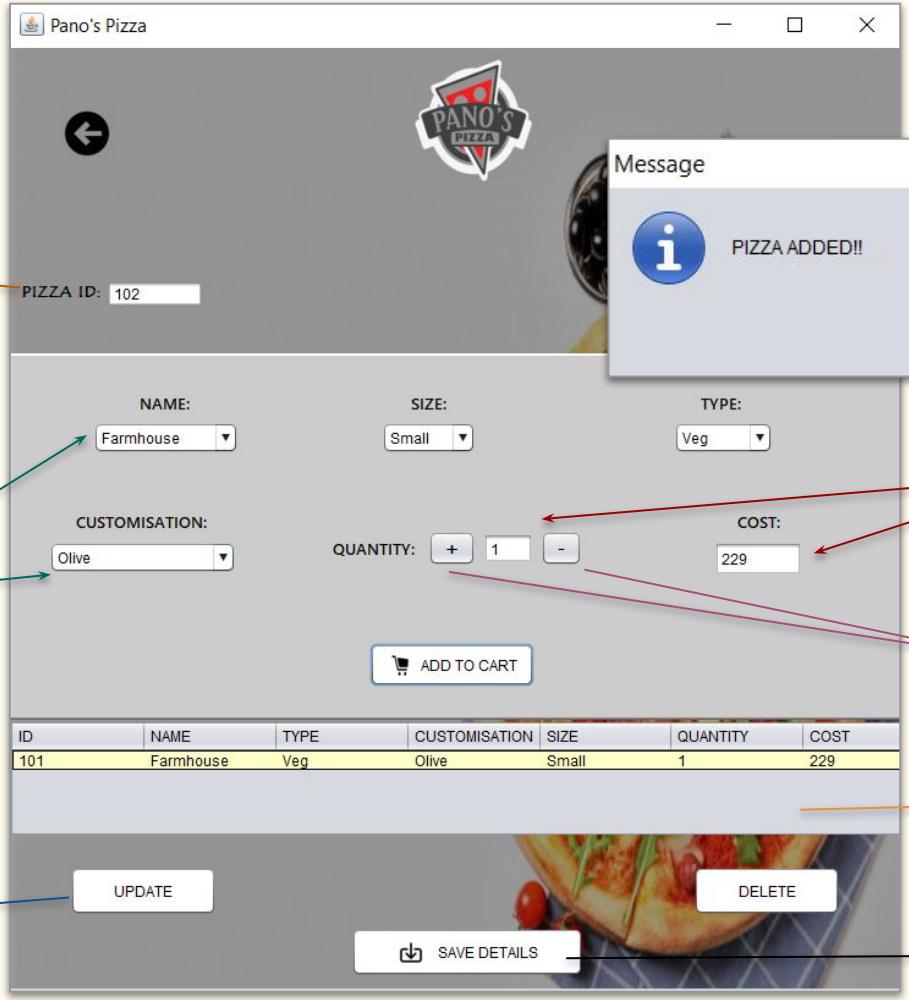
- Any user having an account can ‘LOG IN’ by entering their Username and Password.
  If they match user then successfully logged into the application.





Link to load register form

Message Dialog Box showing the user has successfully registered and logged into the application.



This unique ID updates on every order

Message Dialog Box showing pizza added to the cart

Combo boxes used for selecting name, size, type and customisation of the pizza

Text Fields showing quantity and the cost of the pizza.

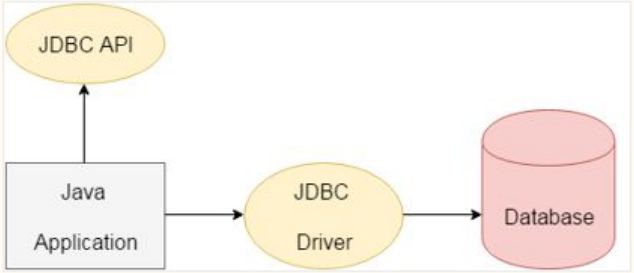
This will increment and decrement the particular quantity of the order

Cart showing the pizza added by the customer

'UPDATE' & 'DELETE' buttons used to update and delete the selected pizza from the cart.

Save details to proceed to the billing screen

How JDBC work?



JDBC stands for Java Database Connectivity. JDBC is a Java API that is used connect and execute the query with the database.

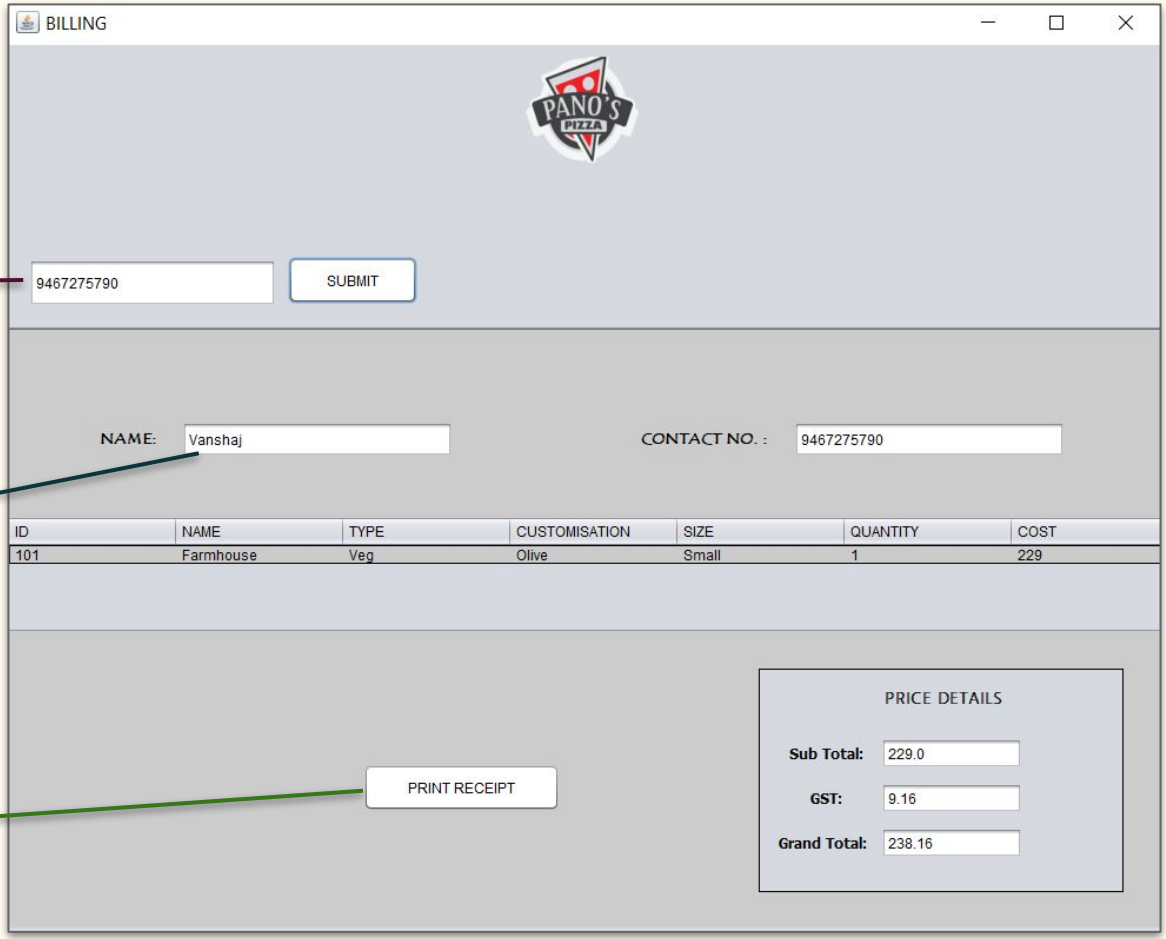
In this project, the pizza added to the cart is connected to the database by the JDBC-ODBC bridge driver. The cost of the pizza will be updated according to it's customisation through the data stored in database. JDBC will execute the 'UPDATE' and 'DELETE' queries to update and delete the data respectively.

Billing Screen

‘Enter the contact number’ Text Field

This will show the existing customer details on the basis of contact number

This button will generate the invoice



Printing the receipt..

- itextpdf.jar is an open-source Java library that supports the development and conversion of PDF documents.
- **PdfWriter** class represents the Doc Writer for a PDF. This class belongs to com.itextpdf.text.pdf package. The constructor of this class accepts a string, representing the path of the file where the PDF is to be created.
- Empty document can be created by instantiating the **Document** class and to add a paragraph to the document, the **Paragraph** class is instantiated. The object to the document is added using the **add()** method.
- Table in the document is created using **PdfPTable** class and it's cells is created using **PdfPCell** class. The cells to the table are added using addCell() method.

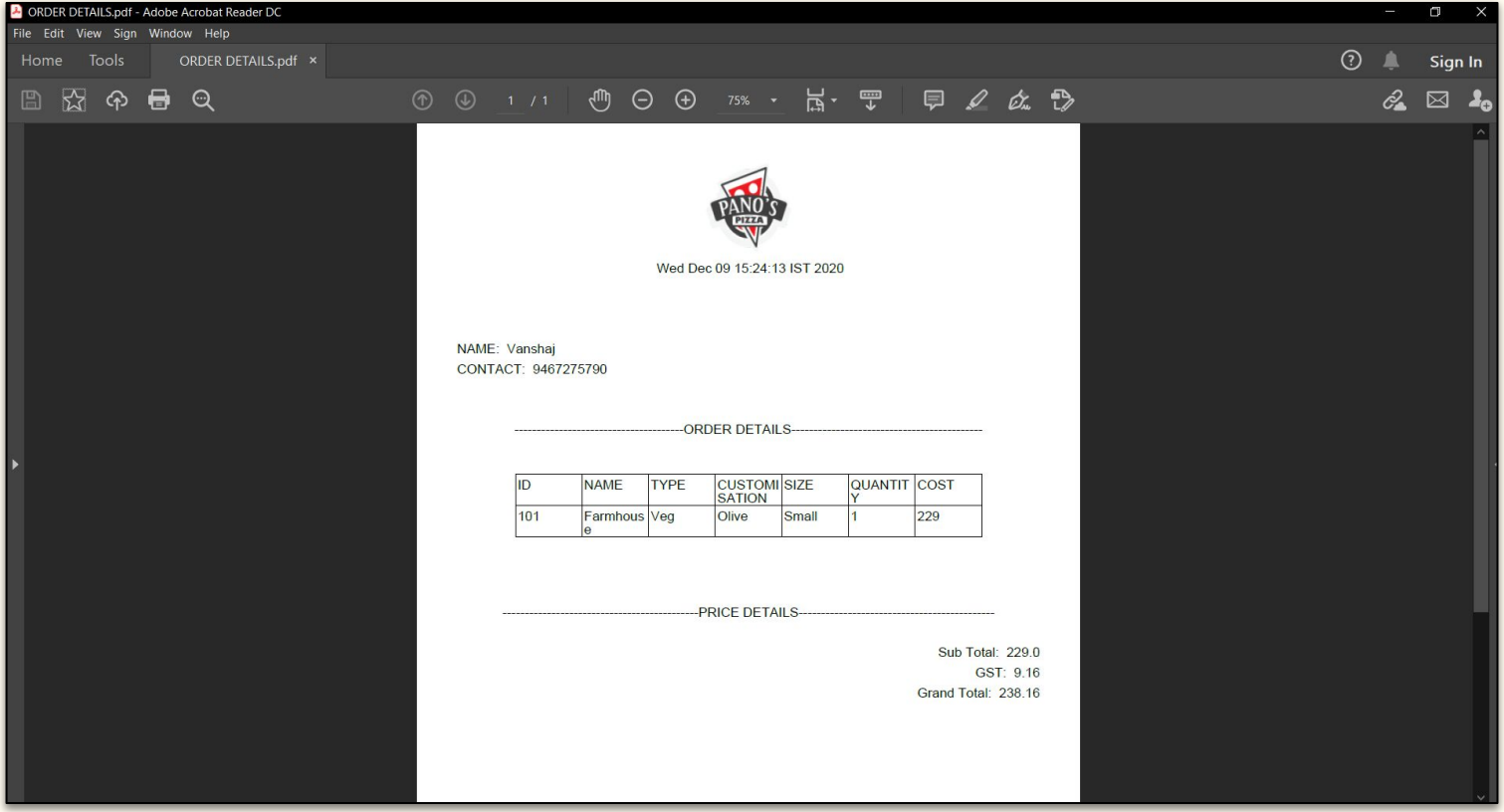
Creating a document

```
Document document = new Document();
PdfWriter.getInstance(document, new FileOutputStream("ORDER DETAILS.pdf"));
document.open();

document.addTitle("Order Details");
document.addAuthor("Vanshaj Pahwa");
document.addCreator("Vanshaj Pahwa");

String img = "C:\\Users\\vansh\\OneDrive\\Documents\\NetBeansProjects\\myfirst\\src\\myfirst\\panos_pizza_100x100.png";
Image img1 = Image.getInstance(img);
img1.scaleAbsolute(80, 80);
img1.setAlignment(Element.ALIGN_CENTER);
document.add(img1);
Paragraph paragraph = new Paragraph(" " + new java.util.Date());
paragraph.setAlignment(Element.ALIGN_CENTER);
document.add(paragraph);
paragraph = new Paragraph("\n \n \n");
paragraph.setAlignment(Element.ALIGN_LEFT);
document.add(paragraph);
paragraph = new Paragraph("NAME: " + " " + T3.getText());
paragraph.setAlignment(Element.ALIGN_LEFT);
document.add(paragraph);
paragraph = new Paragraph("CONTACT: " + " " + T4.getText());
paragraph.setAlignment(Element.ALIGN_LEFT);
document.add(paragraph);
paragraph = new Paragraph("\n \n");
paragraph.setAlignment(Element.ALIGN_LEFT);
document.add(paragraph);
```

Author, image, user and restaurant details is added to the Paragraph class using add() method.

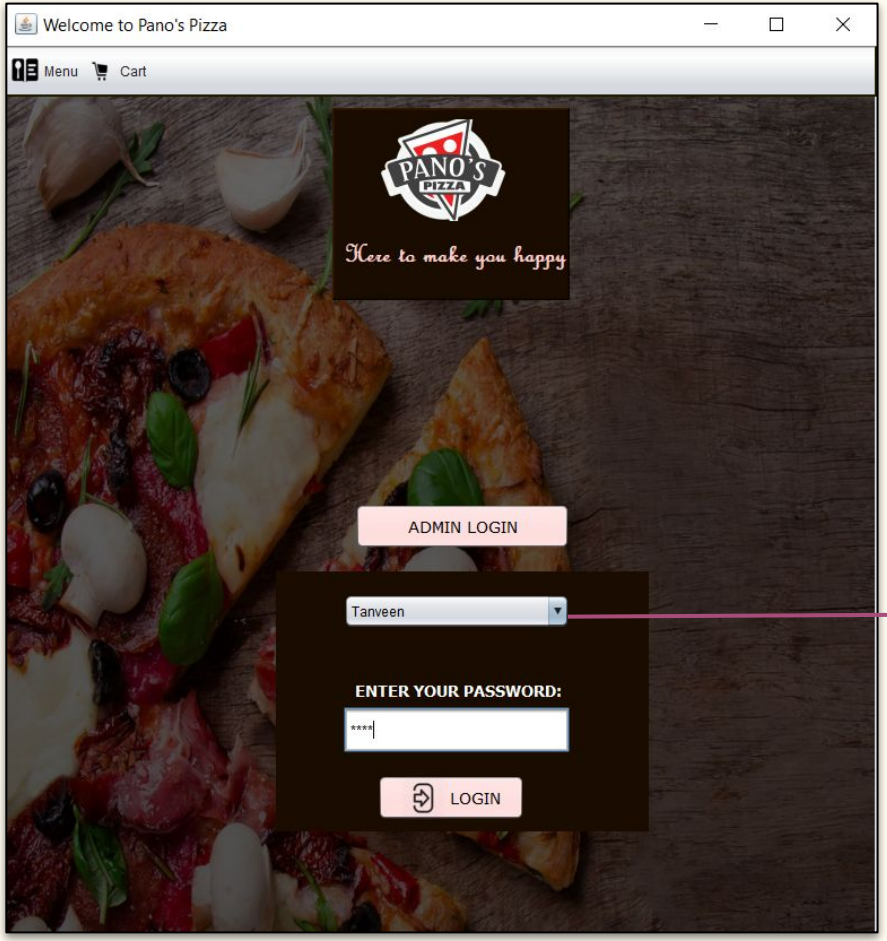
java.util.Date() is used to add the date to the document.

Output

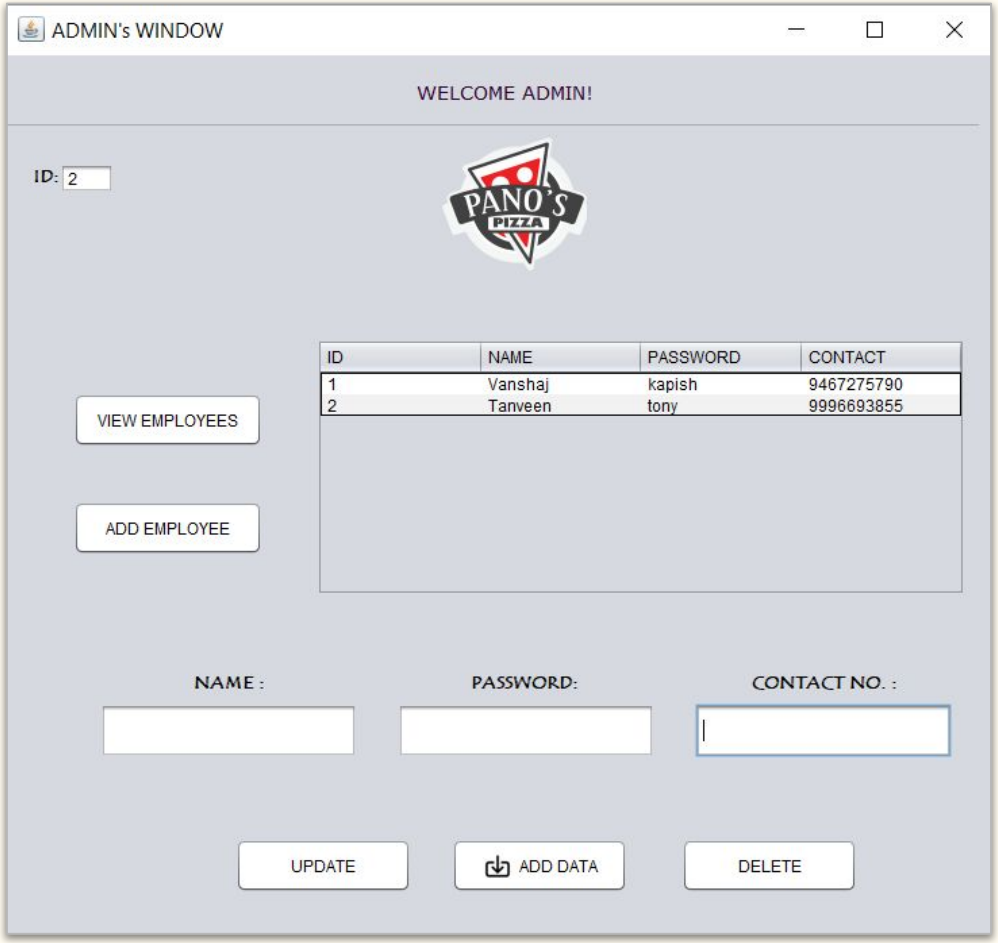


(Figure showing invoice in pdf format)

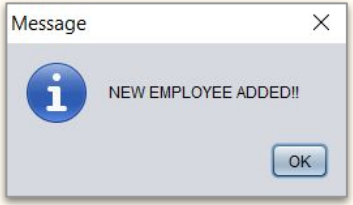
Admin's Screen



Admins can go and login to their portal from here



- Click 'VIEW EMPLOYEE' to view the Employee's data stored in database.
- 'ADD EMPLOYEE' is used to add the new employee, to do this add the data in respective fields.



(New Employee added successfully)

- 'UPDATE' and 'DELETE' buttons are used to update and delete the selected data from database respectively.

(Message Dialog Box showing Data is updated and deleted successfully)

Thank You!