Criterion C- Development

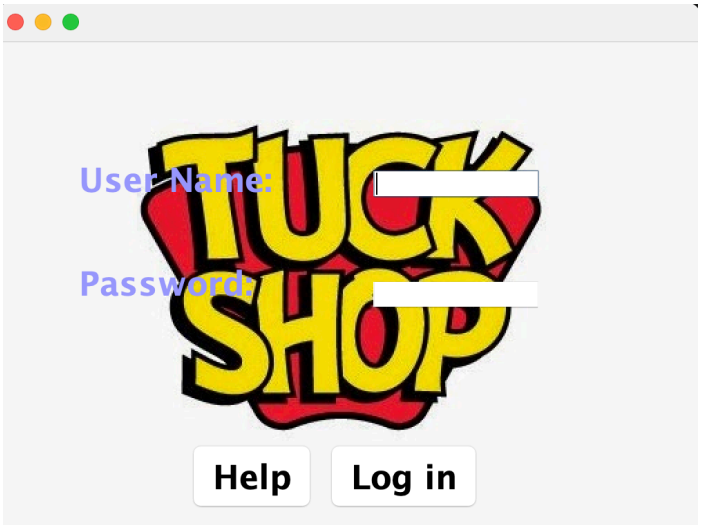
Table of Contents

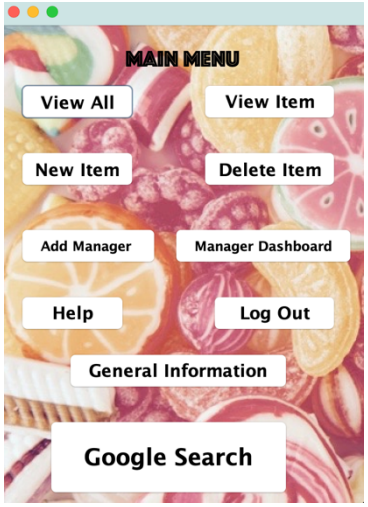
1. Creating GUI in NetBeans 2
2. Button Group and Checkbox for Selection Login 4
3. GUI in Python- Visual Studio Code 5
4. Random number generator in python 6
5. Accessing Email with smtplib in python 7
6. Creation of a LAN and File Sharing 8
7. Creation of Folder's based on customer ID. 9
8. Saving of bill in PDF in the customers folders 10
9. 2-Factor Authentication 11
10. Connection between Java and Python 12
11. Print Functionality 13
12. Exporting JTable into PDF 14
13. Adding date with JCalender 15
14. Opening PDF HELP file based on button click 16
15. Opening website based on button click. 17
16. Backend database on NetBeans 18
17. Insert and Search operations using SQL in the database. 21
18. Exportable Database 23
19. Exporting a JAR File 24
20. User Defined Methods 25
21. Encapsulation 26
22. Object-Oriented Database 27
23. Inheritance 28
24. Polymorphism 29
25. Using additional Java Libraries 30
26. Using Mathematical Functions 31
27. Parsing classes of different types into each other 32
28. Usage of Data structure: Array Lists and Dictionary 33
29. Try Catch loops 34
30. Conditional Statements 35
31. Populating Combo Box using data from database 36
32. Graphical representations using JFreeChart 37

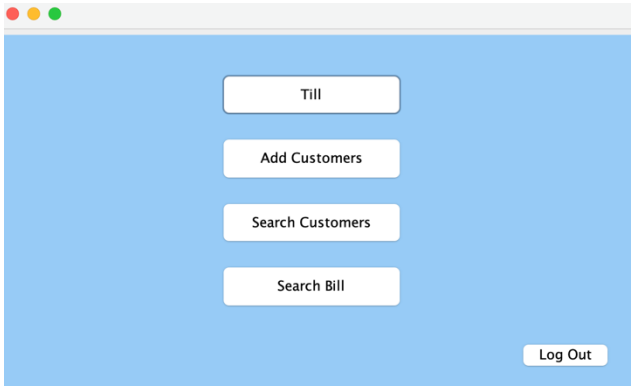
Bibliography 38

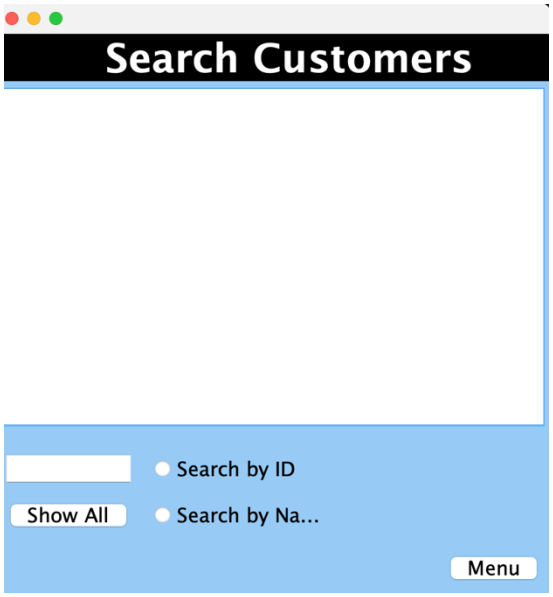
1. Creating GUI in NetBeans

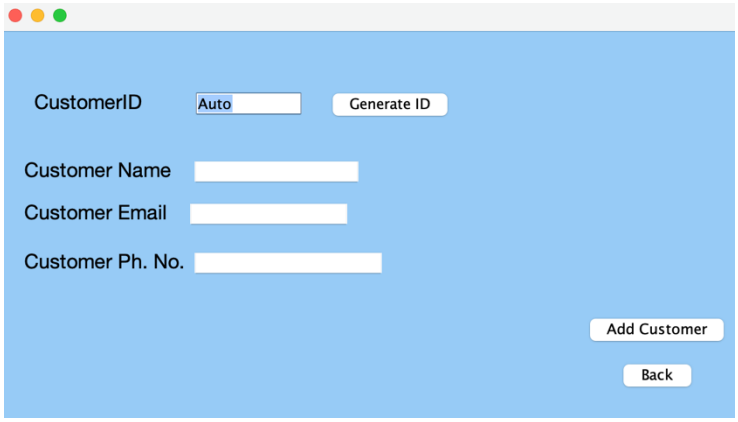
Purpose	Success Criterion
To make the product's user interface simple to utilize for the consumers.	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20

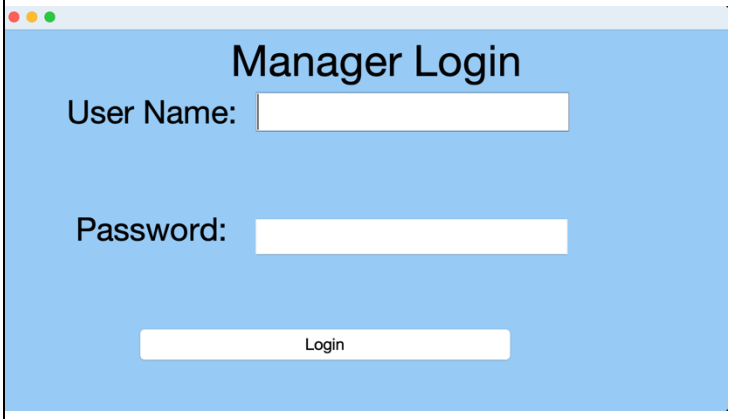


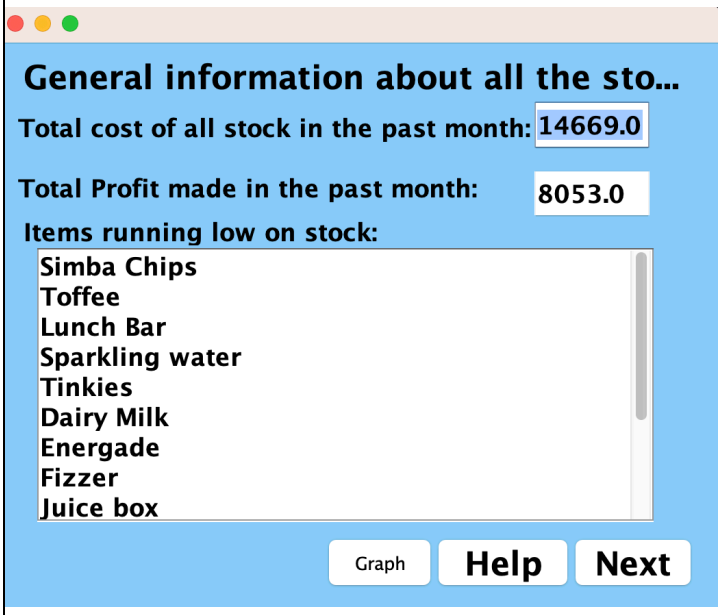


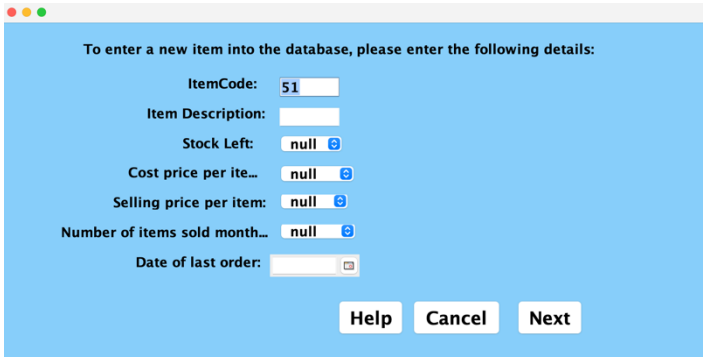


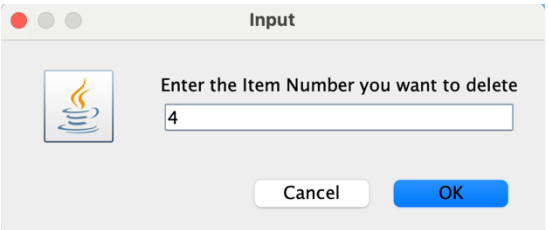


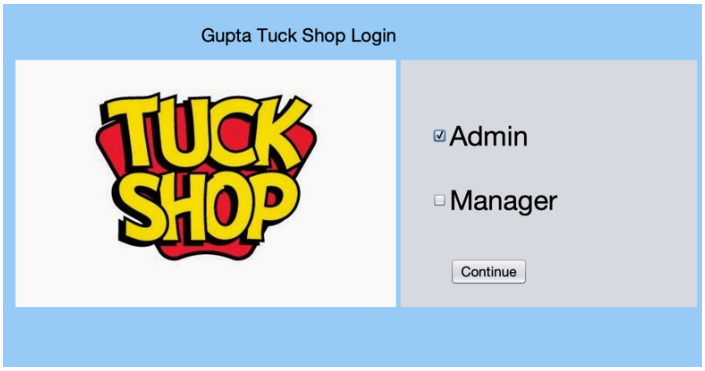


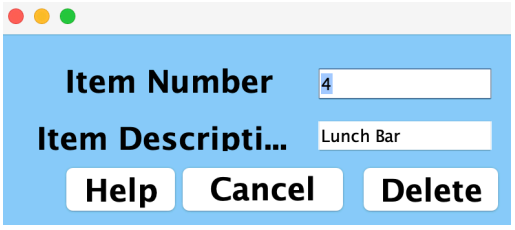


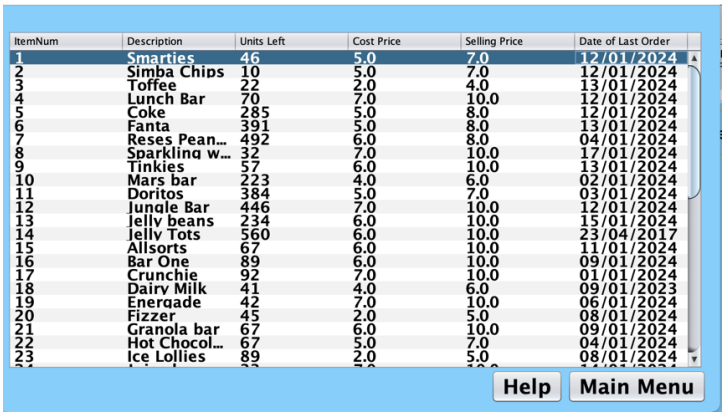








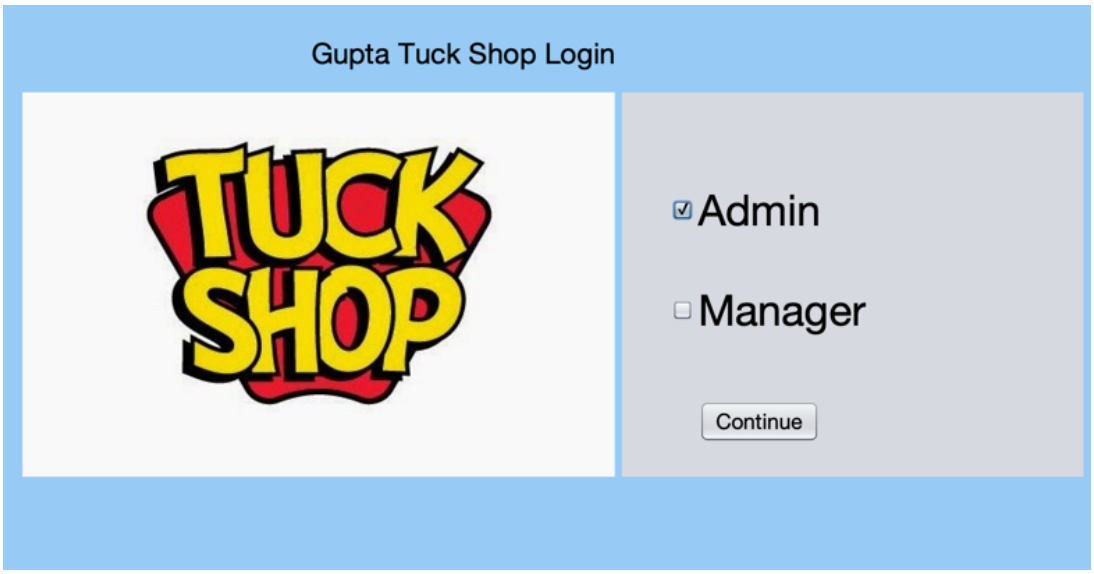


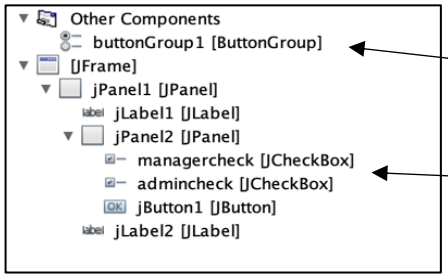


ItemNum	Description	Units Left	Cost Price	Selling Price	Date of Last Order
1	Smarties	46	5.0	7.0	12/01/2024
2	Simba Chips	10	5.0	7.0	12/01/2024
3	Toffee	22	2.0	4.0	13/01/2024
4	Lunch Bar	70	7.0	10.0	12/01/2024
5	Coke	285	5.0	8.0	12/01/2024
6	Fanta	391	5.0	8.0	13/01/2024
7	Reses Pean...	492	6.0	8.0	04/01/2024
8	Sparkling w...	32	7.0	10.0	17/01/2024
9	Tinkies	57	6.0	10.0	13/01/2024
10	Mars bar	223	4.0	6.0	02/01/2024
11	Doritos	384	5.0	7.0	03/01/2024
12	Jungle Bar	446	7.0	10.0	12/01/2024
13	Jelly beans	234	6.0	10.0	15/01/2024
14	Jelly Tots	560	6.0	10.0	23/04/2017
15	Allsorts	67	6.0	10.0	11/01/2024
16	Bar One	89	6.0	10.0	09/01/2024
17	Crunchie	92	7.0	10.0	01/01/2024
18	Dairy Milk	41	4.0	6.0	09/01/2023
19	Energade	42	7.0	10.0	06/01/2024
20	Fizzer	45	2.0	5.0	08/01/2024
21	Granola bar	67	6.0	10.0	09/01/2024
22	Hot Chocol...	67	5.0	7.0	04/01/2024
23	Ice Lollies	89	2.0	5.0	08/01/2024

2. Button Group and Checkbox for Selection Login

Purpose	Success Criterion
Both the Admin and the Manager will have a different logins as Admin will have more features which the manager will be restricted from.	13





Originally, both the checkboxes were clickable and 2 different frames were opening based on the selected login. To resolve this, I added a button group which enables the user to only select 1 of the 2 checkboxes.

I also added an if condition which allows only 1 window to open based on the checkbox and sets the other window to be invisible.

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    if(managercheck.isSelected()){
    this.setVisible (false); // Sets this frame to be invisible
        JFrameLogIn p = null;
        p = new JFrameLogIn();
        p.setVisible(true);
    // TODO add your handling code here:
    }
    else if(admincheck.isSelected()){
        this.setVisible (false); // Sets this frame to be invisible
        ManagerLogin p = new ManagerLogin();
        p.setVisible(true);
    }
    // TODO add your handling code here:
}
```

3. GUI in Python- Visual Studio Code

Purpose	Success Criterion
To implement OTP, we require certain tools which are available in Python but not Java.	8

```python
import random
import smtplib
from tkinter import *
```

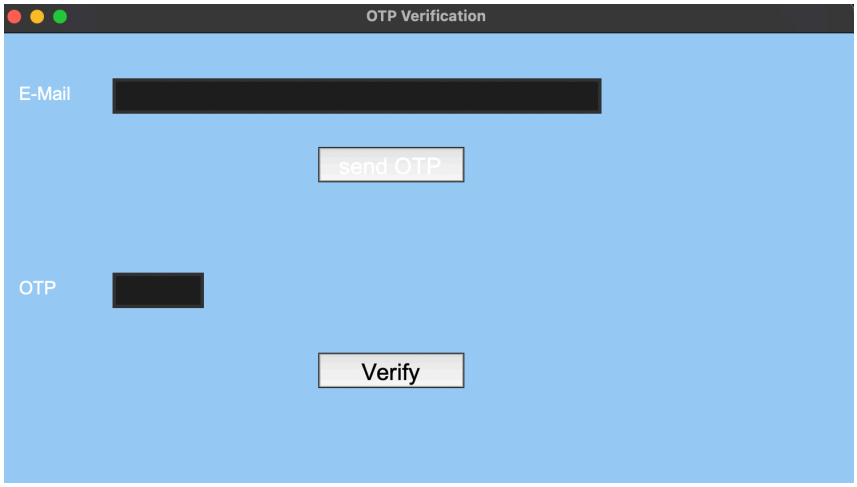
Importing the Tkinter Library for constructing the basic GUI of my application.

```python
otp = Tk()
otp.title('OTP Verification')
otp.geometry('750x400')
otp.config(bg='#97ccf6')
```

Giving the Tkinter block a title, size, and background color.

```python
mailMsg = Label(otp, text="E-Mail", justify=LEFT, bg='#97ccf6', font=("Arial", 16))
mailMsg.place(x=15, y=40)

receiverMail = Entry(otp, text='mail.gmail.com', width=35, font=("Arial", 20), borderwidth=0)
receiverMail.place(x=100, y=40)
receiverM = StringVar()

sendOTP = Button(otp, text="send OTP", width=8, height=1, font=("Arial", 20), borderwidth=0, bg="#AA5656", fg="white", command=connectingSendEr)
sendOTP.place(x=280, y=100)

otpMsg = Label(otp, text="OTP", bg='#97ccf6', font=('Arial', 16))
otpMsg.place(x=15, y=210)

codeEntry = Entry(otp, width=6, font=("Arial", 20), borderwidth=0)
codeEntry.place(x=100, y=210)

verify = Button(otp, text="Verify", width=8, height=1, font=("Arial", 20), borderwidth=0, bg="#AA5656", fg="black",
                command=checkOTP)
verify.place(x=280, y=280)

otp.mainloop()
```

Adding all the GUI elements, which includes buttons, labels, and text inputs.



4. Random number generator in python

Purpose	Success Criterion
To generate an OTP we require a random number generator.	8

```python
import random
import smtplib
from tkinter import *


def generateOTP():
    randomCode = ''.join(str(random.randint(0, 9)) for i in range(6))
    return randomCode


sender = [redacted]
password = [redacted]
code = generateOTP()


def connectingSender():
    receiver = receiverMail.get()
    server = smtplib.SMTP('smtp.gmail.com', 587)
    server.starttls()
    server.login(sender, password)
    sendingMail(receiver, server)


def sendingMail(receiver, server):
    msg = 'Hello! \n This is your OTP is ' + code
    server.sendmail(sender, receiver, msg)
    server.quit()
```

Imported the random library.

Defined a function to generate a 6- digit random code.

Defined a function to send a mail with the random code which has been generated.

5. Accessing Email with smtplib in python

Purpose	Success Criterion
To access email and sending an email, we require the smtplib library which is available in python.	8

```python
import random
import smtplib
from tkinter import *


def generateOTP():
    randomCode = ''.join(str(random.randint(0, 9)) for i in range(6))
    return randomCode


sender = [redacted]
password = [redacted]
code = generateOTP()


def connectingSender():
    receiver = receiverMail.get()
    server = smtplib.SMTP('smtp.gmail.com', 587)
    server.starttls()
    server.login(sender, password)
    sendingMail(receiver, server)


def sendingMail(receiver, server):
    msg = 'Hello! \n This is your OTP is ' + code
    server.sendmail(sender, receiver, msg)
    server.quit()
```

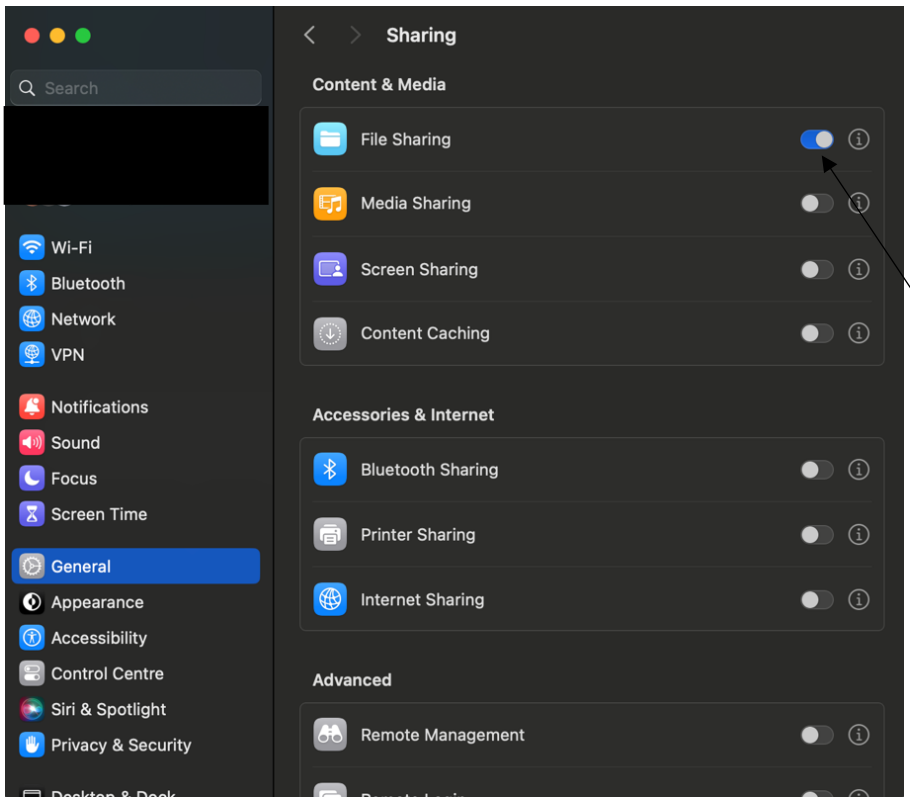
Imported the library- smtplib (Simple Mail Transfer Protocol)

I added the Gmail id and password which I used to send the passwords to.

Defined a function to send a mail to the email id with the One-time password. The library helps to connect to the server and send a mail to the user.

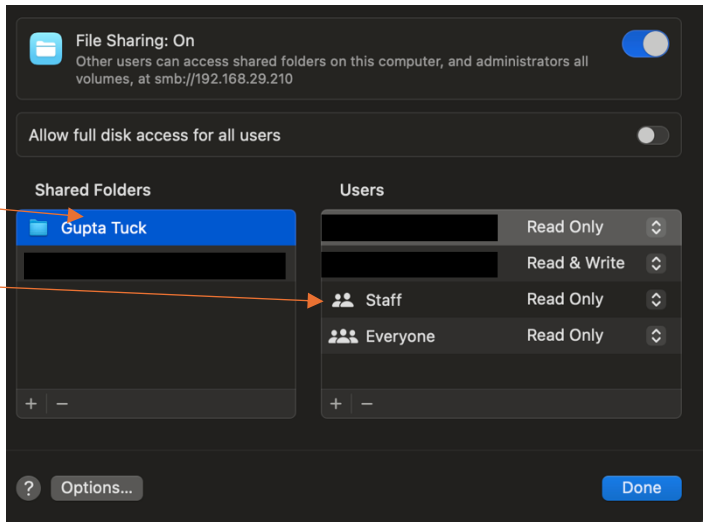
6. Creation of a LAN and File Sharing

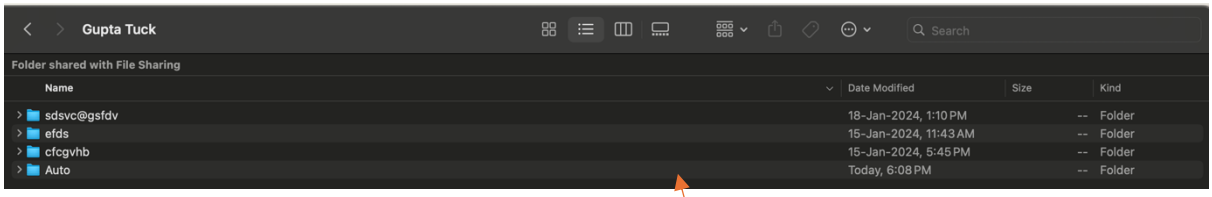
Purpose	Success Criterion
To make the sharing database of customer bills easy.	15



To create a LAN First I went to system preferences and enabled file sharing feature from general settings so that I could share the customers and their bills with all PCs in the LAN.

I added the folder which I wanted to share with others in the LAN, named Gupta Tuck and I added all the contacts who I want to give access too.

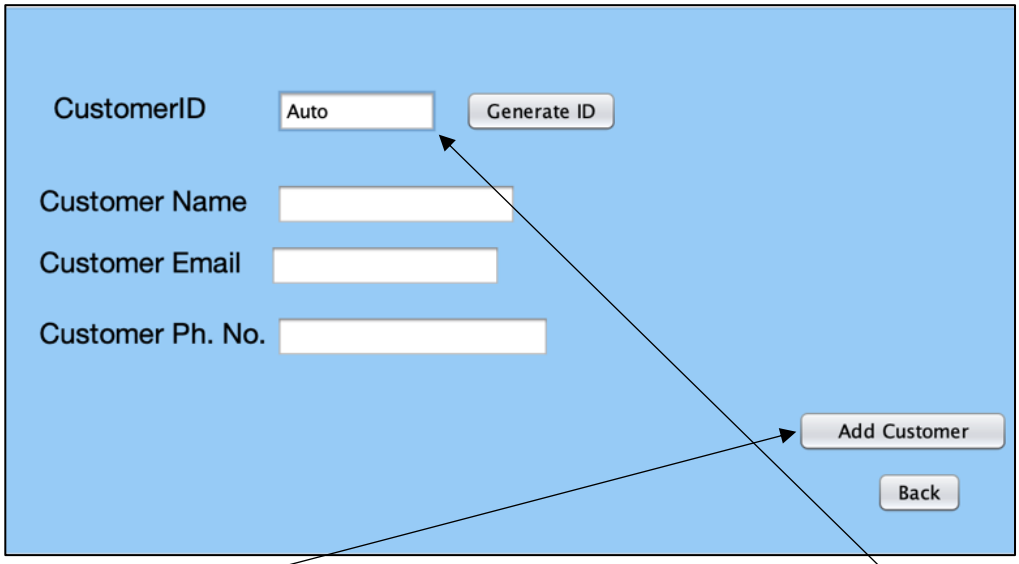




This is a screenshot from another laptop which has the folder shared from my laptop with all the customers.

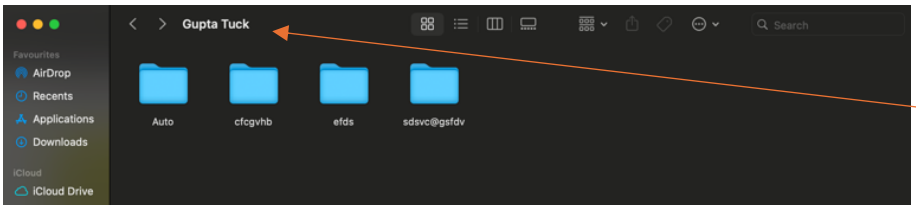
7. Creation of Folder's based on customer ID.

Purpose	Success Criterion
Each customer can have their own folders for saving their bills.	17



When the Add customer button is clicked, the system will take the CustomerID from text input t1 and create a new folder in the folder named "Gupta Tuck" and that folder will contain every customer bill for future returns.

```
String t = tID.getText();
String t1= "/Users/vanshchawla/Downloads/Gupta Tuck/";
String dictionary = t1.concat(t);
File f =new File(dictionary);
JOptionPane.showMessageDialog(null, dictionary);


f.mkdirs();
```



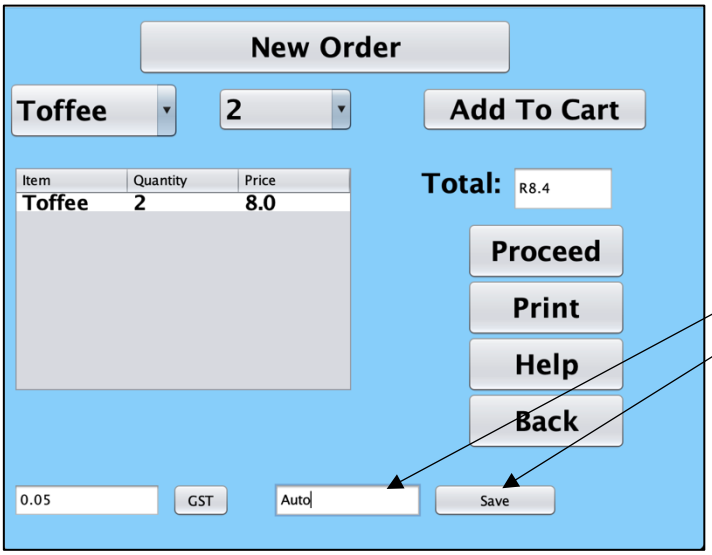
This is the folder.

8. Saving of bill in PDF in the customers folders

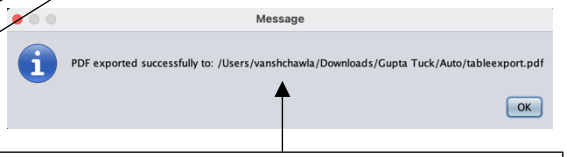
Purpose	Success Criterion
The company can keep records of all sold items which will later come in handy if a customer wants to return an order.	18

```
    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

    try {
        String d= jTextField1.getText();
        String d1= "/Users/vanshchawla/Downloads/Gupta Tuck/";
        String d2 = d1.concat(d);
        String filePath = d2+"/tableexport.pdf";
        JTableToPDF(jTableItems, filePath);
        JOptionPane.showMessageDialog(null, "PDF exported successfully to: " + filePath);
    } catch (Exception ex) {
        Logger.getLogger(JFrameTill.class.getName()).log(Level.SEVERE, null, ex);
        JOptionPane.showMessageDialog(null, "Error exporting PDF");
    }
}
```

This is the file path.

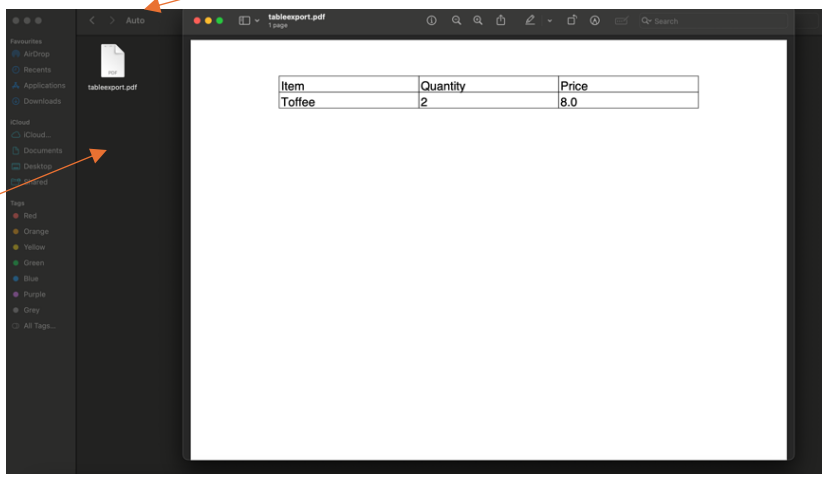


When the button Save is clicked, the Bill's PDF will get saved in the folder based on the text input value and the customer id.



This is a dialogue box which will confirm that the bill is saved in the customers folder.

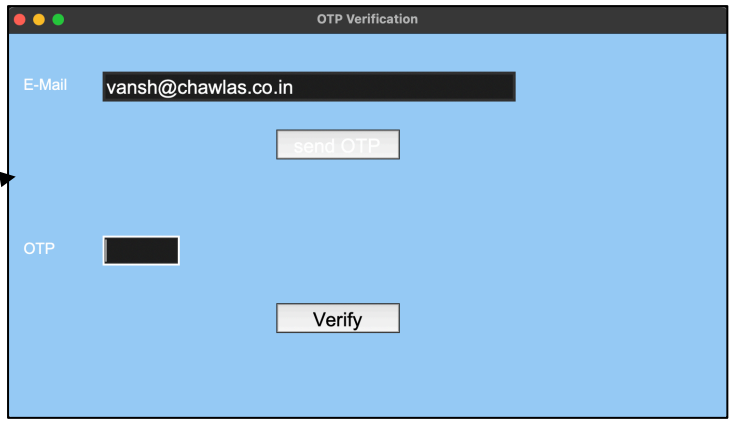
This Bill got saved in the Auto Folder with the same items which were in the cart.



9. 2-Factor Authentication

Purpose	Success Criterion
To improve security of the application.	8

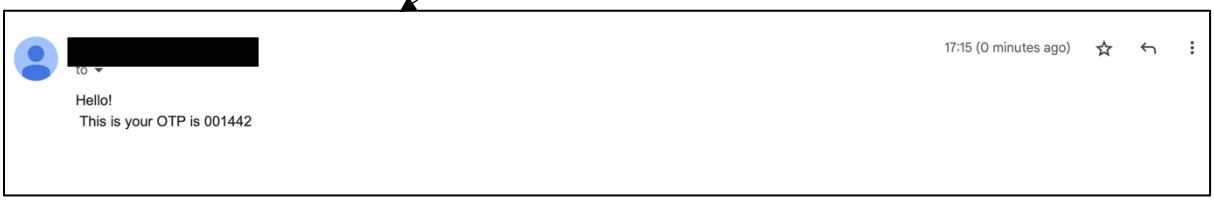
This screen opens when the admin wants to log in for 2 factor authentication.



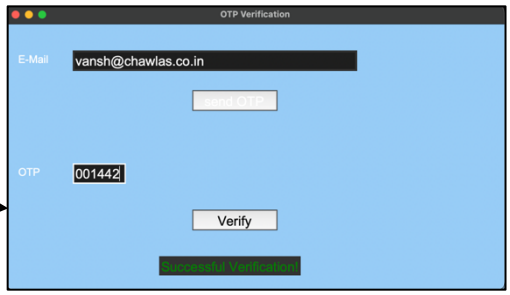

```python
def sendingMail(receiver, server):
    msg = 'Hello! \n This is your OTP is ' + code
    server.sendmail(sender, receiver, msg)
    server.quit()
```

With the smtp library the code gets sent to the message next to the msg variable.

This is the mail which gets sent.



The OTP gets confirmed with the generated OTP by the random library.




```python
def checkOTP():
    if code == codeEntry.get():
        accept = Label(otp, text='Successful Verification!', fg='green', font=('Arial', 20))
        accept.place(x=230, y=350)
    else:
        refuse = Label(otp, text='Unsuccessful Verification!', fg='red', font=('Arial', 20))
        refuse.place(x=230, y=350)
```

This checks if the OTP is correct and proceeds to display a message based on whether the verification is successful.

10. Connection between Java and Python

Purpose	Success Criterion
To connect the OTP verification (made on python) with the rest of the application (made on java)	8

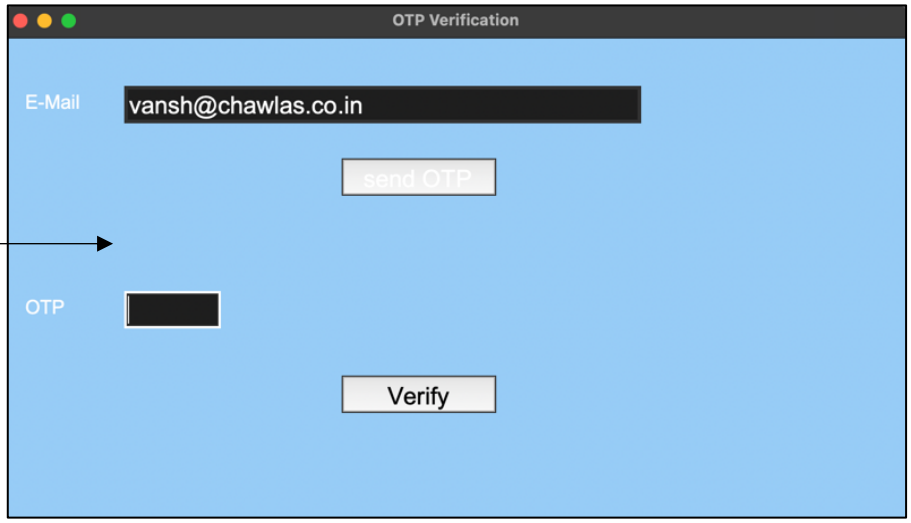
```java
public static void main(String[] args) {
        JFrame frame = new JFrame("Java Tkinter Launcher");
        JButton openTkinterButton = new JButton("Open Tkinter Script");

        openTkinterButton.addActionListener((ActionEvent e) -> {
            openTkinterWindow();
        });

        frame.getContentPane().add(openTkinterButton);
        frame.setSize(600, 400);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }

    private static void openTkinterWindow() {
        try {
            // Specify the path to your Python script
            String pythonScriptPath = "/Users/vanshchawla/Downloads/otpmain/final.py";

            // Run the Python script using ProcessBuilder
            ProcessBuilder processBuilder = new ProcessBuilder("python3", pythonScriptPath);
            Process process = processBuilder.start();

            // Wait for the process to finish
            int exitCode = process.waitFor();

            // Print the exit code for debugging
            System.out.println("Python script exit code: " + exitCode);
        } catch (IOException | InterruptedException ex) {
        }
    }
```

This creates an external frame based on my Tkinter root geometry.

This is the path of the python file.

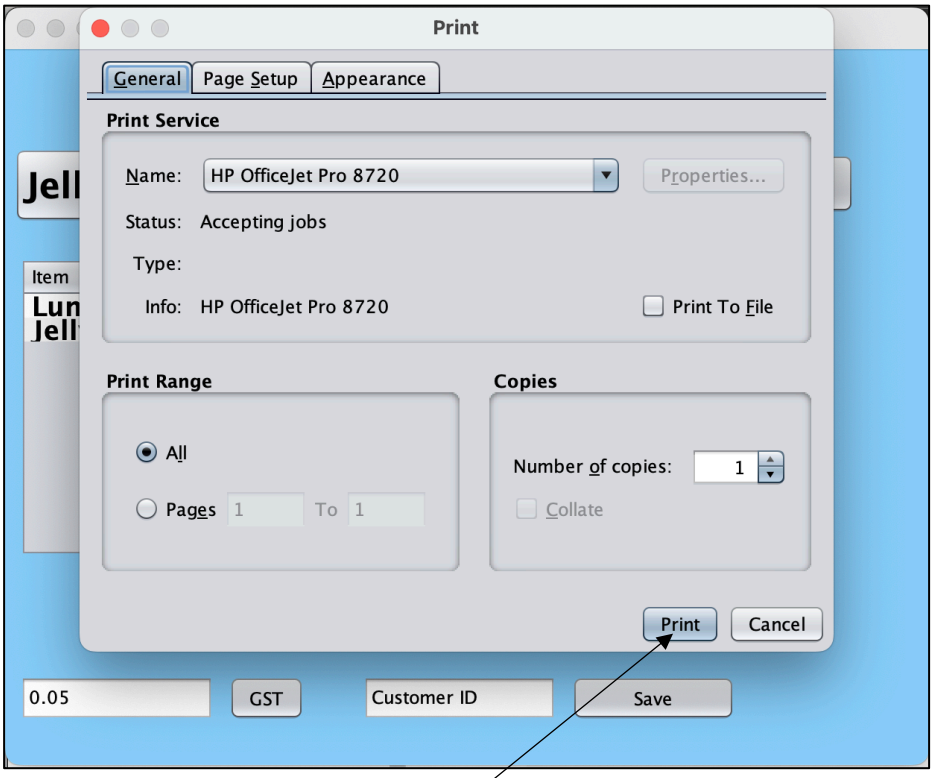
This defines that this is a python file.

This is the python file which gets opened once the button gets clicked.



11. Print Functionality

Purpose	Success Criterion
For the manager/admin to give the customer a printout of their bill.	3

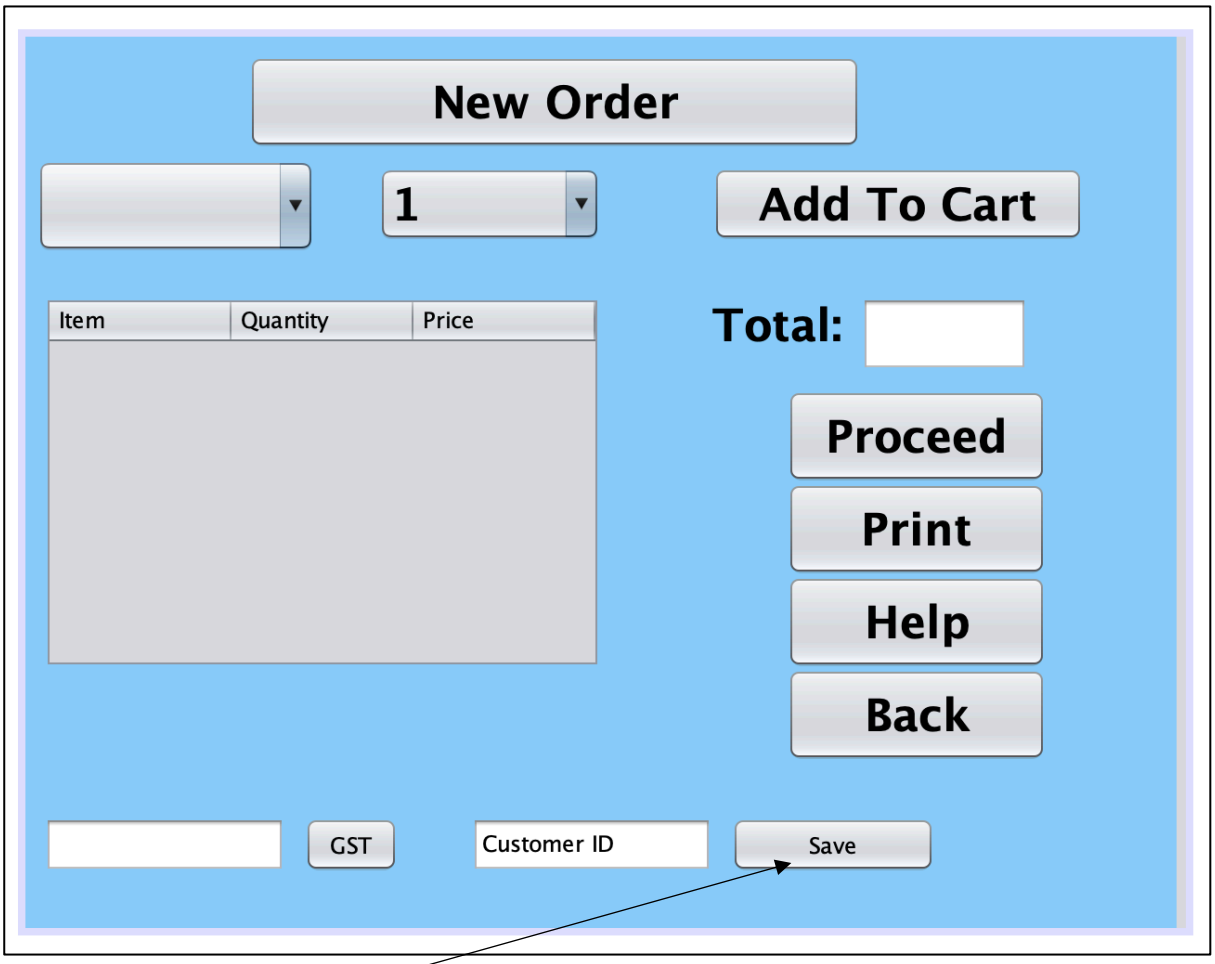


The print table function enables the table to get exported as PDF and prints the table. I have added a footer and header to the page which consists of the page number.

```
private void jButtonPrintActionPerformed(java.awt.event.ActionEvent evt) {
        try{
          MessageFormat header = new MessageFormat ("Page Header"); // Specifies the header of the page
          MessageFormat footer = new MessageFormat ("Page (1, number,integer)"); // specifies the footer of the page
          jTableItems.print (JTable.PrintMode.FIT_WIDTH, header, footer); //prints the table

      }
      catch (Exception e ){

      }
}
```

12. Exporting JTable into PDF

Purpose	Success Criterion
The bill gets converted to PDF so that it can be saved in the customers folder.	18

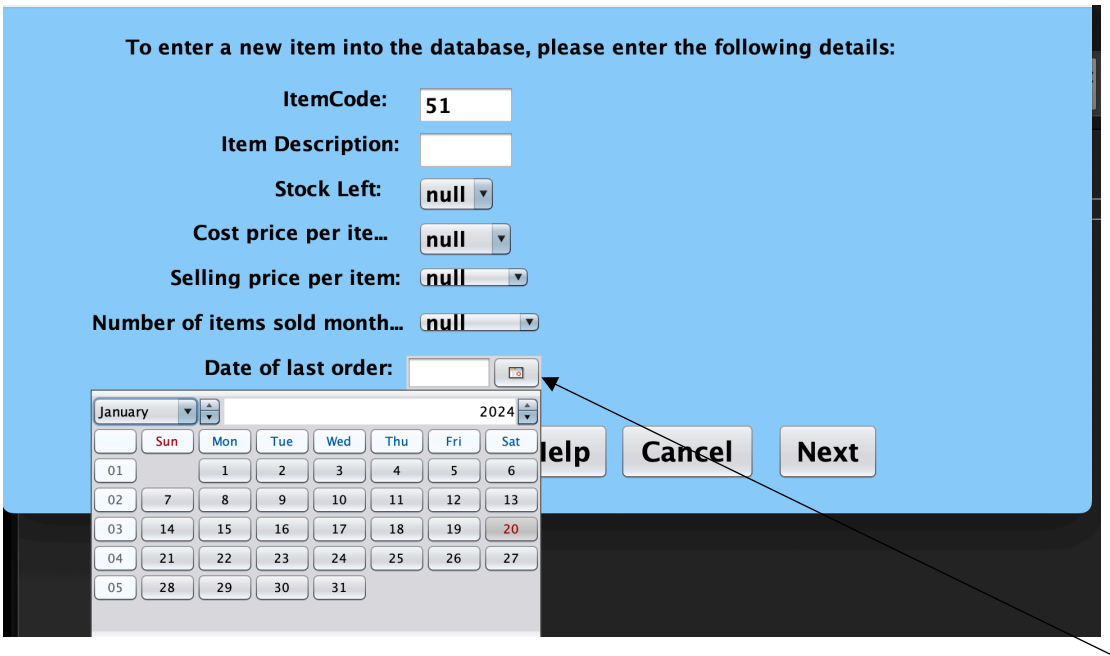


Once the save button is clicked, the jTableItems convert's itself into a PDF and saved to the file path mentioned in String d1.

```
    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

    try {
        String d= jTextField1.getText();
        String d1= "/Users/vanshchawla/Downloads/Gupta Tuck/";
        String d2 = d1.concat(d);
        String filePath = d2+"/tableexport.pdf";
        JTableToPDF(jTableItems, filePath);
        JOptionPane.showMessageDialog(null, "PDF exported successfully to: " + filePath);
    } catch (Exception ex) {
        Logger.getLogger(JFrameTill.class.getName()).log(Level.SEVERE, null, ex);
        JOptionPane.showMessageDialog(null, "Error exporting PDF");
    }
}
```

13. Adding date with JCalender

Purpose	Success Criterion
To add the last date the stocks got updated	4



This gets todays date from the calendar and stores it into the variable.

With this button, you can select the date of last order.

```
String description = TextFieldItemDescription.getText(); // Gets and stores the description from the
int StockLeft = Integer.parseInt(jComboBoxStockLeft.getSelectedItem().toString()); // Stores what th
double CostPrice = Double.parseDouble(jComboBoxCostPrice.getSelectedItem().toString()); // Stores wh
double SellingPrice = Double.parseDouble(jComboBoxSellingPrice.getSelectedItem().toString()); // Sto
int SoldMonthly = Integer.parseInt(jComboBoxItemsSold.getSelectedItem().toString());// Stores what t
SimpleDateFormat dcn = new SimpleDateFormat("dd/MM/yyyy");// sets the format of the date
String LastOrder = dcn.format(jDateChooserLastOrder.getDate());// Gets the date from the calendar an
arr.InsertItem(description, StockLeft, CostPrice, SellingPrice, SoldMonthly, LastOrder); // creates
JOptionPane.showMessageDialog(null, "The item has been added"); // A message is displayed to the use
```

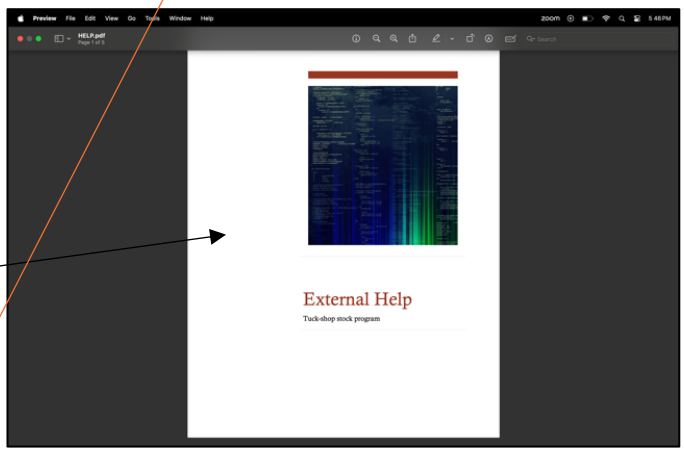
This will make sure that the date is set and stored in a “dd/mm/yyyy” format.

14. Opening PDF HELP file based on button click

Purpose	Success Criterion
To assist the manager, while facing a technical problem	7

Clicking on this button will open an external Help File for the admin which will guide them about the various features of the application.

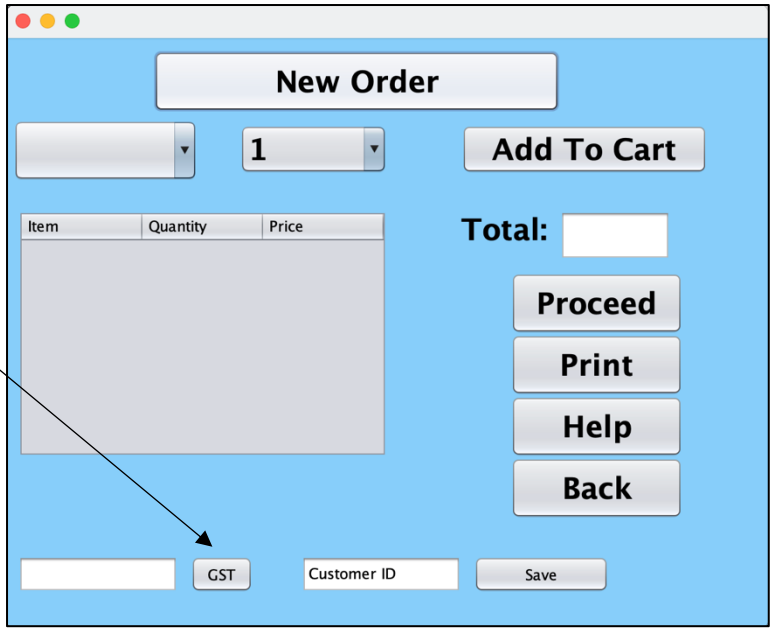
This is the help file which opens when the button gets clicked.




```
private void jButtonHelpActionPerformed(java.awt.event.ActionEvent evt) {
    if (Desktop.isDesktopSupported()) {
        try {
            File myFile = new File("HELP.pdf"); //Looks for the Help file
            Desktop.getDesktop().open(myFile); //Opens up the help pdf file

        } catch (IOException ex) {
        }

    }
}
```

15. Opening website based on button click.

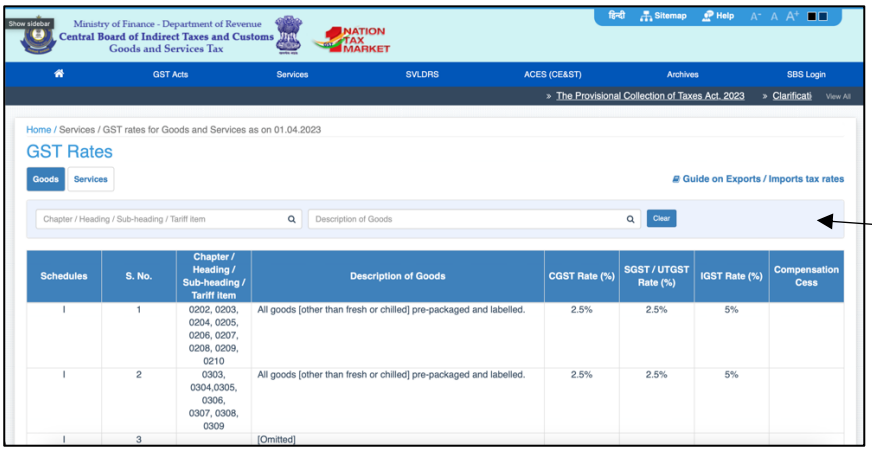
Purpose	Success Criterion
To add GST to the total.	9



By clicking on this button, the manager can access the most recent GST posted by the government.

This is the URL of the website which is supposed to open once the button gets clicked.

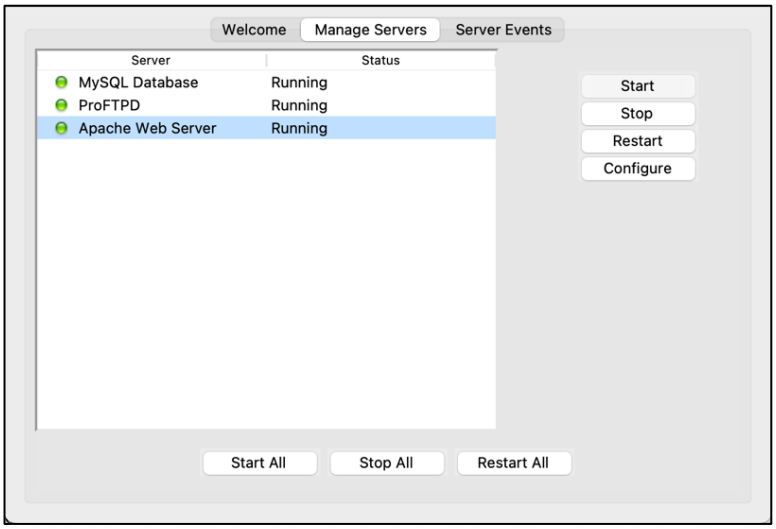
```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
if (Desktop.isDesktopSupported()) {

    //making a desktop object
    Desktop desktop = Desktop.getDesktop();
    try {
        URI uri = new URI("https://cbic-gst.gov.in/gst-goods-services-rates.html");
        desktop.browse(uri);
    } catch (IOException excp) {
        excp.printStackTrace();
    } catch (URISyntaxException excp) {
        excp.printStackTrace();
    }
  }
// TODO add your handling code here:
}
```



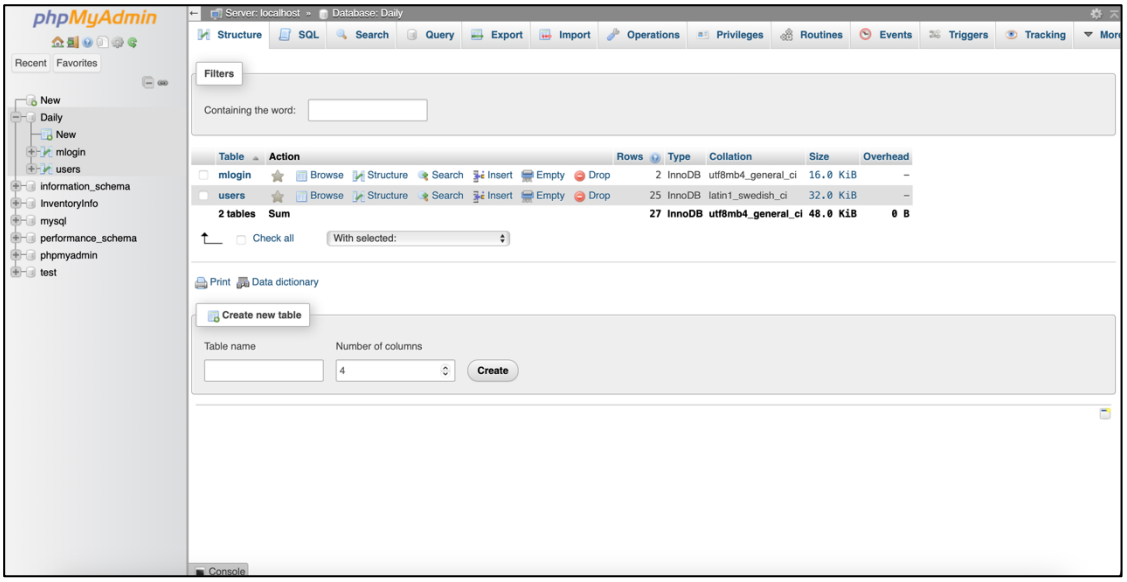
This is the government website which opens with the most recent GST.

16. Backend database on NetBeans

Purpose	Success Criterion
To make sure all entries are saved on a database.	5

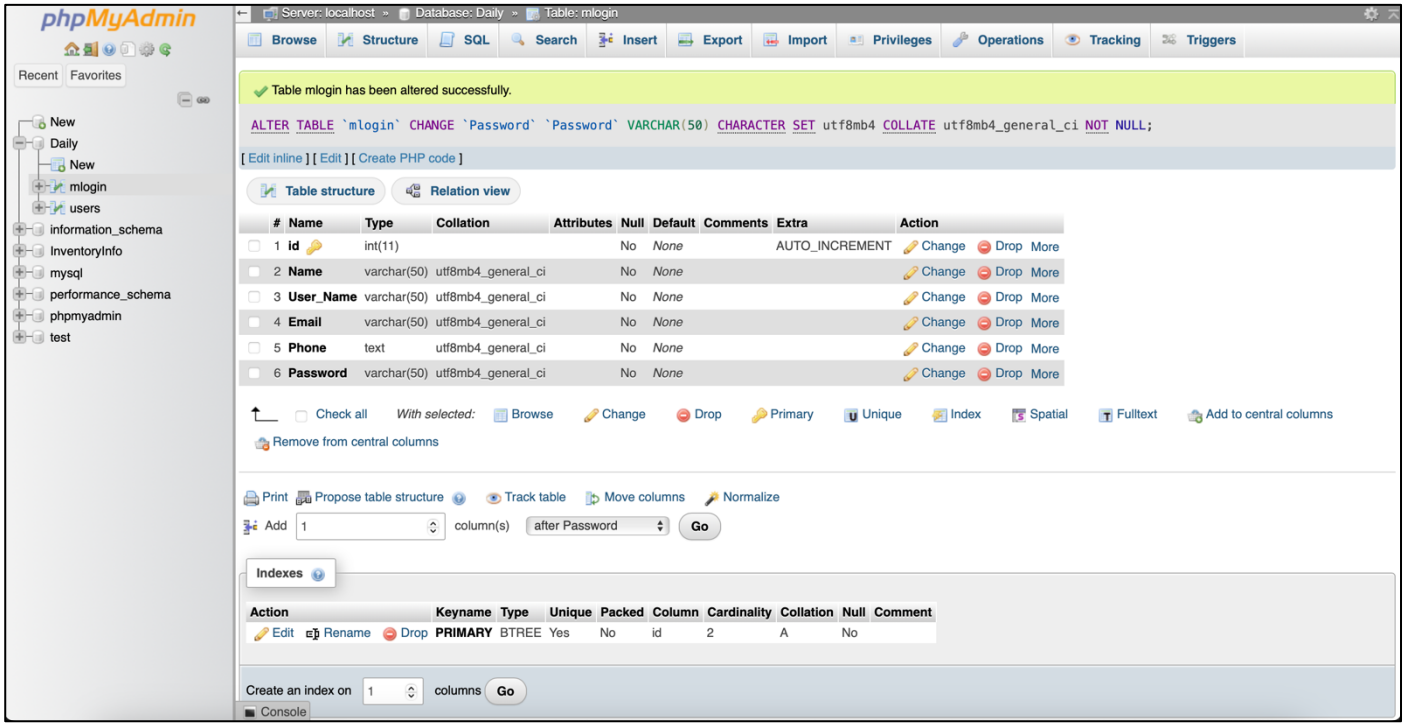


Manager OSX (XAMPP Control Panel) permits the usage of modules like MySQL and Apache.

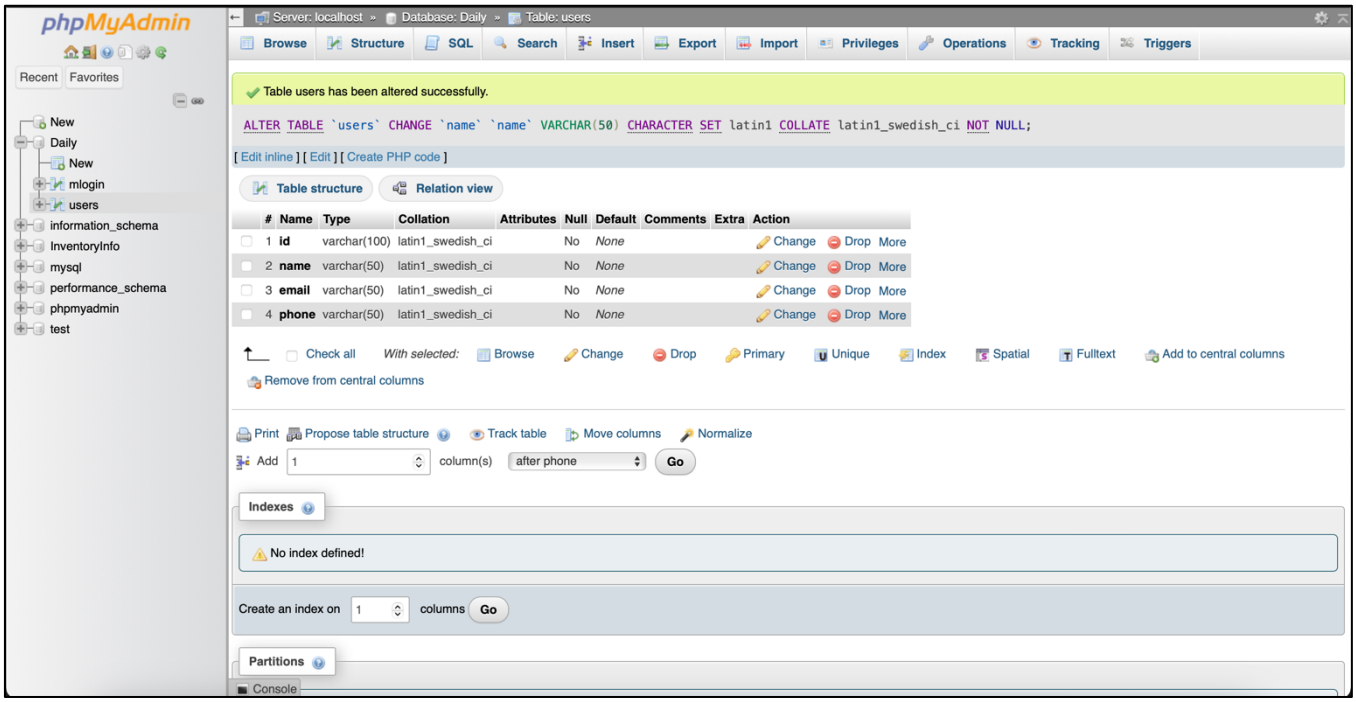


Complete List of tables which I have made on phpMyAdmin.

mlogin Table



users Table



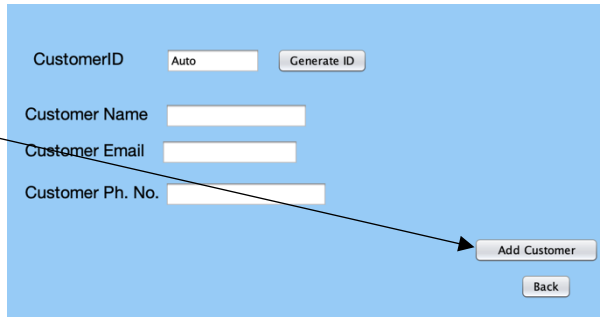
I created a connection between my backend and my database, with the name Database name Daily.

```
public class MyConnection {

    public static Connection getConnection(){
        Connection con = null;
        try {
            Class.forName("com.mysql.jdbc.Driver");
            con = DriverManager.getConnection("jdbc:mysql://localhost/Daily","root","");
        } catch (Exception ex) {
            System.out.println(ex.getMessage());
        }

        return con;
    }
} //Connecting the application to the local host database
```

17. Insert and Search operations using SQL in the database.

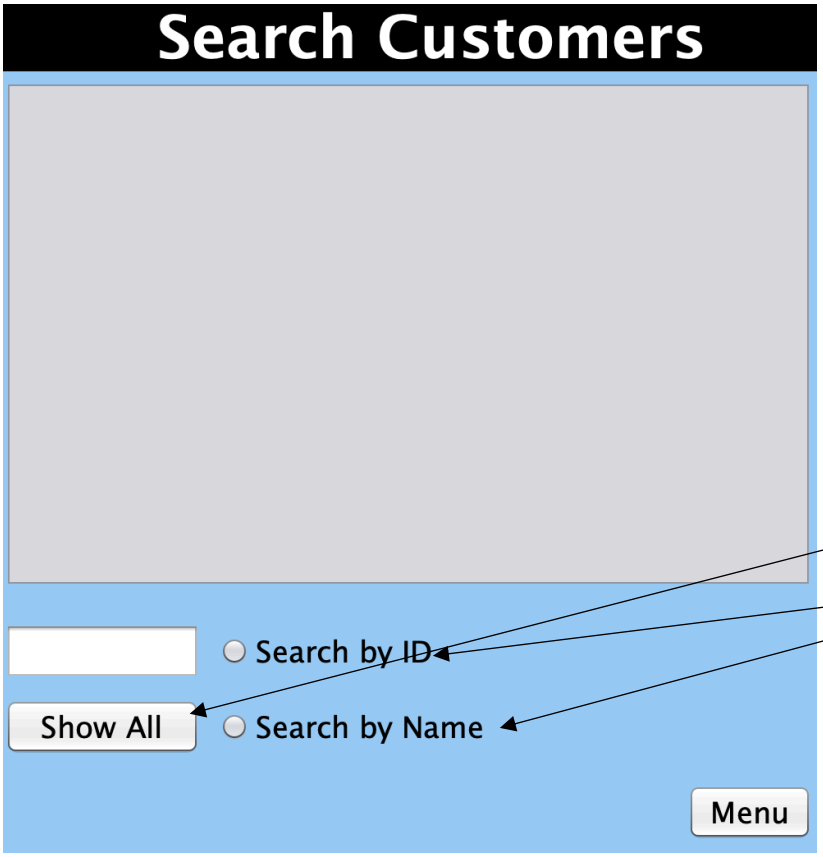
Purpose	Success Criterion
To add customers and search for them to get their ID if they forget.	5,6,11

Once the Add customer button is clicked, all the text will be extracted from the text fields, and it will be saved in the database as it is under the columns.




```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

    JOptionPane.showMessageDialog(null,"Inserting a Customer");
    Connection con = Backend.MyConnection.getConnection();
    PreparedStatement ps;
    try {

        ps = con.prepareStatement("INSERT INTO `users`(`id`,`name`,`email`,`phone`) VALUES (?,?,?,?)");//Query stat

        ps.setString(1, tID.getText());

        ps.setString(2, tN.getText());

        ps.setString(3, tE.getText());

        ps.setString(4, tP.getText());

        ps.executeUpdate();
        JOptionPane.showMessageDialog(null, "Welcome " + tN.getText()+"to Customer Database of Gupta Tuck Shop" );

    } catch (Exception ex) {

        Logger.getLogger(JFrameCustomers.class.getName()).log(Level.SEVERE, null, ex);

    }
    String t = tID.getText();
    String t1= "/Users/vanshchawla/Downloads/Gupta Tuck/";
    String dictionary = t1.concat(t);
    File f =new File(dictionary);
     JOptionPane.showMessageDialog(null, dictionary);


    f.mkdirs();
```

A Dialogue box will pop up with the customer's name and send through a welcome message.



Once the show all button is clicked and based on either ID or Name is selected, the entry will get searched in the database.

This is for searching with ID.

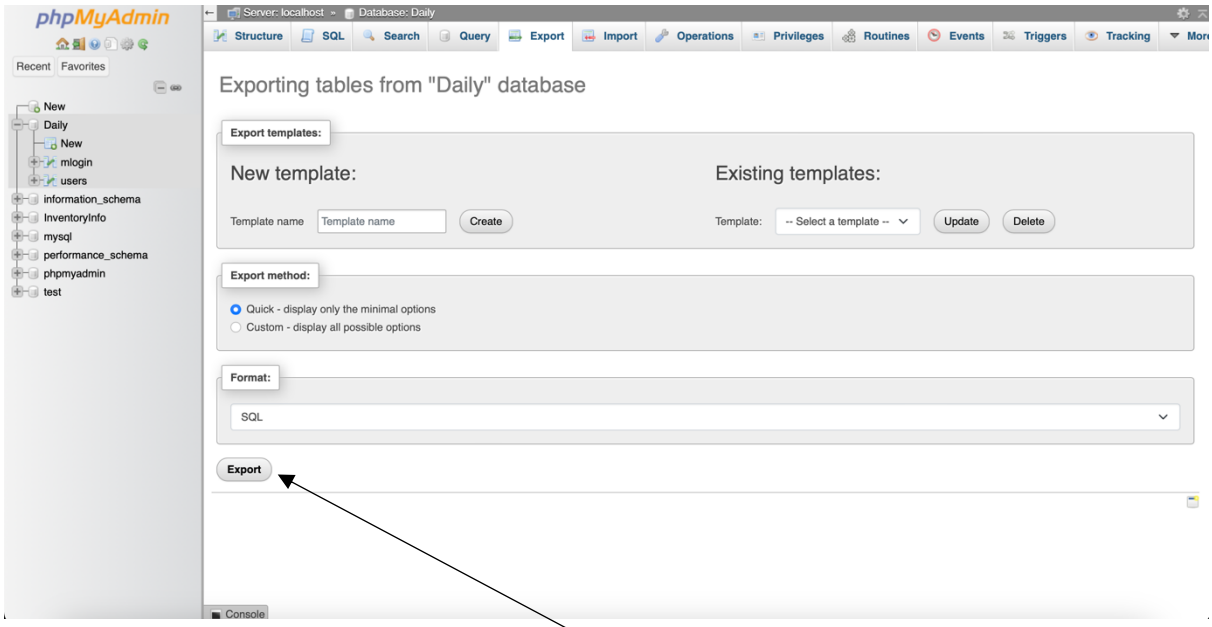
```
private void FillMatchID(){
 String sql =  "SELECT id, name, phone, email FROM  users where id = ?";
try{
    PreparedStatement pst = con.prepareStatement(sql);

    pst.setString(1, tMatchID.getText());

   rs = pst.executeQuery();
   ShowTable.setModel(DbUtils.resultSetToTableModel(rs));

    }
    catch (Exception e){
        JOptionPane.showMessageDialog(null, e);
    }

}
```

This is for searching with Name.

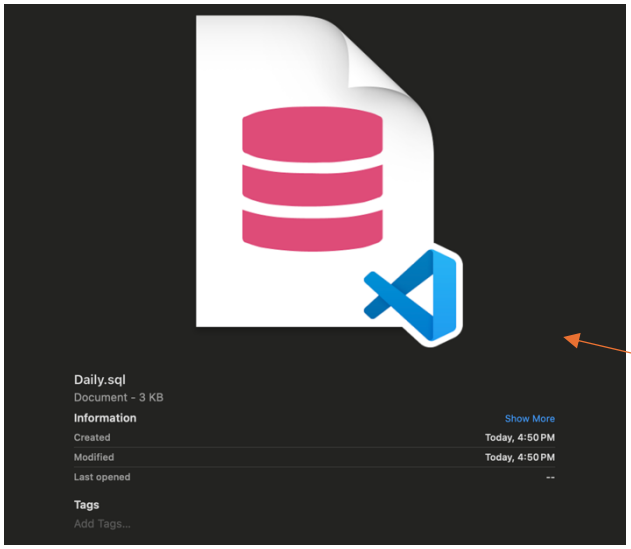
```
private void FillMatchID(String sname){
    String sql =  "SELECT name, email, phone FROM  users where name = ?";
   try{
       PreparedStatement pst = con.prepareStatement(sql);

  pst.setString(1, tMatchID.getText());

       rs = pst.executeQuery();
      ShowTable.setModel(DbUtils.resultSetToTableModel(rs));

       }
       catch (Exception e){
           JOptionPane.showMessageDialog(null, e);
       }
```

18. Exportable Database

Purpose	Success Criterion
To make the application easily portable.	15



Creating an exportable database makes the sharing of the application much easier.

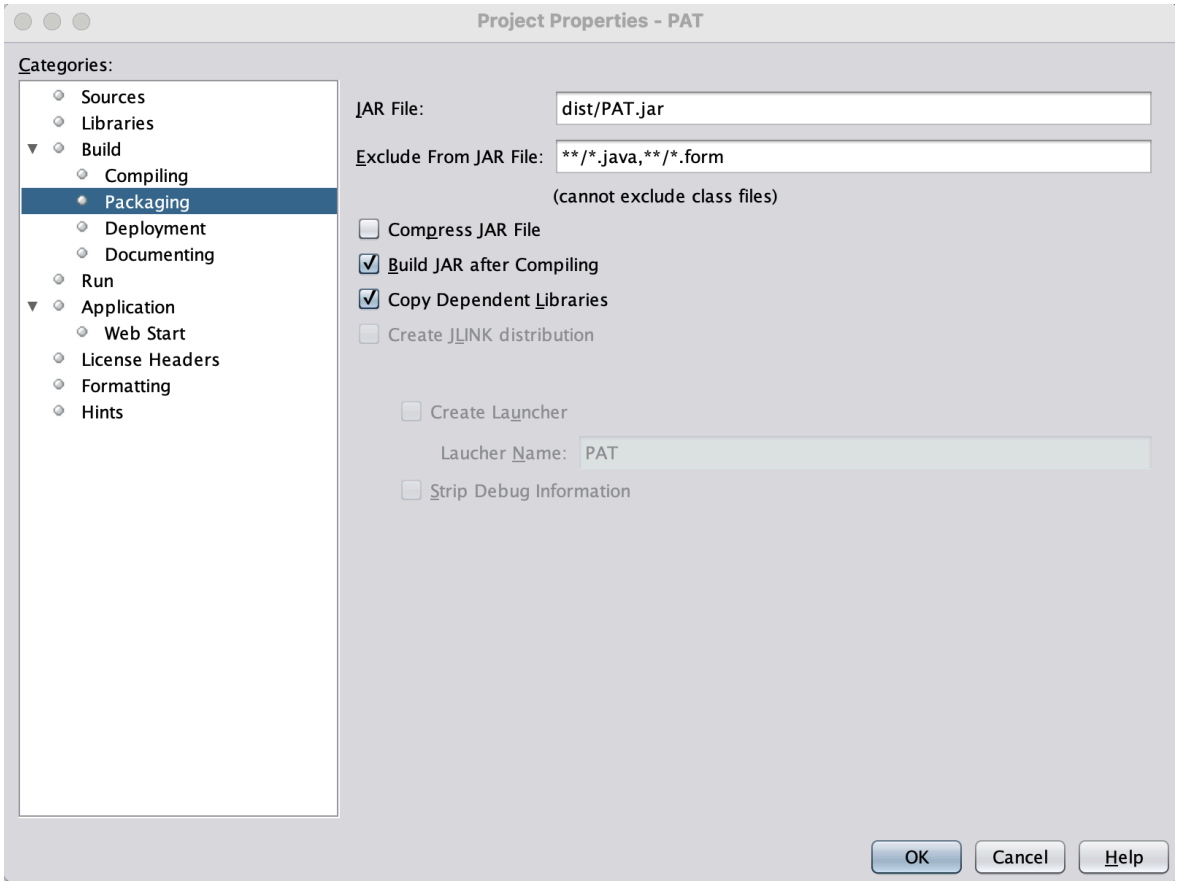


This database is saved in my system dictionary.

19. Exporting a JAR File

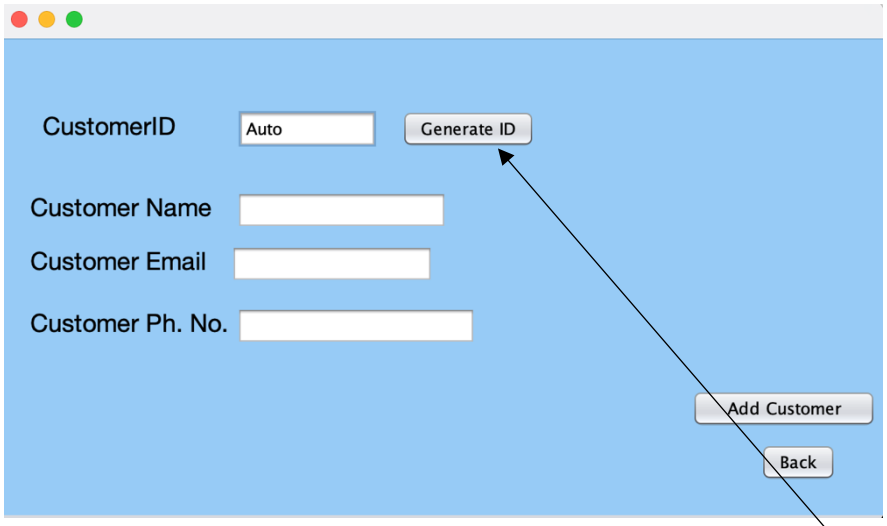
Purpose	Success Criterion
To make the application easily portable.	15

Creation of a JAR file will allow easy sharing of the application on different laptops in the LAN for all the managers.



20. User Defined Methods

Purpose	Success Criterion
To give each customer a unique ID.	5



Called the function generateID() when bID is clicked, and the method creates a unique id for the user which is a mix of the text input from the customer name, email and phone number.

```
public void generateID()
{
    String A = tN.getText();
        String A1 = A.substring(1,2);
        String C = tE.getText();
        String X = A1+C;
        tID.setText(X);
}
    private void bIDActionPerformed(java.awt.event.ActionEvent evt) {

        generateID();
        // TODO add your handling code here:
    }
```

21. Encapsulation

Purpose	Success Criterion
Encapsulation hides the complexity of item data management, promoting data integrity and controlled access within the ItemArray class.	2

```java
public class ItemArray {

    private Item[] arr = new Item[1000]; // An array of the type Item to store all the items and their details
    private int count = 0; // A counter to keep track of the number of elements in the array

    public ItemArray() throws FileNotFoundException { // A Constructor that reads in the text file with all the data about the i
        int ItemNumb = 0; // An integer to store the item num of the items
        String description = ""; // A string to store the description of the item
        int StockLeft = 0; // An integer to store the number of units left of the item
        double CostPrice = 0; // A double to store the cost price of the object
        double SellingPrice = 0; // a double to store the selling price of the object
        int SoldMonthly = 0; // an integer to store the number of units sold monthly
        String LastOrder = ""; // a string to store the date of last order of the objecy
        Scanner scFile = new Scanner(new File("Items.txt")); // Creating a scanner object to scan the text file with the details
        while (scFile.hasNext()) { // A while loop till the end of the file
            String line = scFile.nextLine(); // A variable to break up the file into lines
            Scanner scLine = new Scanner(line).useDelimiter("#"); // A scanner object to store the line, and use the delimeter #
            ItemNumb = scLine.nextInt(); // Extracts the Item number from the text file and stores it
            description = scLine.next(); // Stores the description from the text file
            StockLeft = scLine.nextInt(); // Stores the number of units left from the text file
            CostPrice = scLine.nextDouble(); // Stores the cost price from the text file
            SellingPrice = scLine.nextDouble(); // Stores the selling price from the text file
            SoldMonthly = scLine.nextInt(); // Stores the number of units sold monthly from the text file
            LastOrder = scLine.next(); // Stores the date of last order from the text file
            arr[count] = new Item(ItemNumb, description, StockLeft, CostPrice, SellingPrice, SoldMonthly, LastOrder); // Creates
            count++; // increments the size of count, adding an element to the array
        }

    }

    public int getCount() { // A typed method of the type integer that returns the size of count, the number of elements in the a
        return count;
    }
```

Construction of an array of Item objects by reading data from a file and encapsulates the logic of object initialization.

Returns the count of items in the ItemArray object, adhering to encapsulation principles.

22. Object-Oriented Database

Purpose	Success Criterion
An OODB streamlines persistent storage and retrieval of user credentials by directly managing UserName objects.	4

```java
public class UserName {
    private String username; // a variable to store the username
    private String password; // a variable to store the password

    public UserName (String username, String password) { // A constructor that assign the values
        this.username = username; //assigning the value of username
        this.password = password; //assigning the value of password

    }

    public boolean check () throws FileNotFoundException { // a typed method of the tpe boolean that checks to see if the username
        String tempName = ""; // A variable to store the value of name from the textfile
        String tempPassword = ""; // A variable to store the value of password from the textfile
        boolean found = false;  // Changes if the username and password is found
        Scanner scFile = new Scanner (new File ("UserNames.txt")); // opens the text file with all the usernmes and passwords
        while (scFile.hasNext()) { // checks to see if the text file has more line
            Scanner scLine = new Scanner (scFile.next()); // splits the line from the file
            scLine.useDelimiter("#"); // a delimeter is specified to break up each part of the line
            tempName = scLine.next();  // The username is extracted
            tempPassword = scLine.next();  // the password is extracted
                if (tempName.equals(username) && tempPassword.equals(password)) { // checks if the username and password matches
                    found = true; // changes the value of found
                    break; // stops the program from running
                }



        }

        return found; // returns the value of found
    }
}
```

The items from the text file are now getting saved in an Object-oriented Database. This is easier to use as all the database gets updated in the backend and there is no need of updating the text file constantly.

23. Inheritance

Purpose	Success Criterion
To inherit properties from another JFrame.	5

```java
public class JFrameTill extends javax.swing.JFrame {
    double total; // a variable to store the total of the order
    ItemArray iArr; // an object of the type ItemArray
    DefaultTableModel Item;// the table
    public JFrameTill() throws FileNotFoundException {
        this.iArr = new ItemArray (); // initiates the iArr object
        initComponents();

    }
```

The class `JFrameTill` inherits properties and behaviours from `javax.swing.JFrame` through extension, facilitating the creation of a customized JFrame for a point of sale system.

24. Polymorphism

Purpose	Success Criterion
Polymorphism is demonstrated through method overloading, where methods with the same name but different parameter lists coexist to handle different scenarios efficiently.	12

```
private void FillMatchID(){
 String sql =  "SELECT id, name, phone, email FROM  users where id = ?";
 try{
     PreparedStatement pst = con.prepareStatement(sql);

pst.setString(1, tMatchID.getText());

     rs = pst.executeQuery();
    ShowTable.setModel(DbUtils.resultSetToTableModel(rs));

     }
     catch (Exception e){
         JOptionPane.showMessageDialog(null, e);
     }

 }
private void FillMatchID(String sname){
 String sql =  "SELECT name, email, phone FROM  users where name = ?";
 try{
     PreparedStatement pst = con.prepareStatement(sql);

pst.setString(1, tMatchID.getText());

     rs = pst.executeQuery();
    ShowTable.setModel(DbUtils.resultSetToTableModel(rs));

     }
     catch (Exception e){
         JOptionPane.showMessageDialog(null, e);
     }


 }
```

Method signature indicates polymorphism as it has the same name as another method.

But this method accepts a parameter, demonstrating method.

25. Using additional Java Libraries

Purpose	Success Criterion
To make the product's user interface simple to utilize for the consumers.	1,7,20

```
import java.text.MessageFormat;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.DefaultComboBoxModel;
import javax.swing.JOptionPane;
import javax.swing.JTable;
import javax.swing.table.DefaultTableModel;
import java.io.FileOutputStream;
import com.itextpdf.text.*;
import com.itextpdf.text.pdf.*;
import java.io.IOException;
import java.net.URI;
import java.net.URISyntaxException;
import java.awt.Desktop;
```

com.itextpdf.text is the library which is required to convert a jTable into a PDF.

The URI library helps to open another website.

java.io.File and java.awt.Desktop are the library which helps in accessing the files on the laptop.

```
import java.awt.Desktop;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JFrame;
import org.jfree.chart.ChartFactory;
import org.jfree.chart.ChartPanel;
import org.jfree.chart.JFreeChart;
import org.jfree.chart.plot.PlotOrientation;
import org.jfree.data.category.CategoryDataset;
import org.jfree.data.category.DefaultCategoryDataset;
```

These libraries are required to import JFreeChart which generates a bar graph based on data.

26. Using Mathematical Functions

Purpose	Success Criterion
To calculate subtotal and final cost total GST.	9,19

```
private void jButtonAddItemActionPerformed(java.awt.event.ActionEvent evt) {
        GST();
}

public void GST()
{
    try {

        ItemArray it = new ItemArray(); //an object of the type ItemArray
        double saleprice = 0; // A variable to store the selling price of the items
        String item = jComboBoxItems.getSelectedItem().toString(); // Gets the description of the item selected by
        int amount = Integer.parseInt(jComboBoxUnits.getSelectedItem().toString()); // Gets the amount ordered by
        double price = it.itemprice(item, amount);//method to find out the items price using the items name,
        saleprice = amount * price;//calculating the amount of money required for the purchse
        it.updateFile(item, amount);//method to upadte the database,to the restoc list the amount of items bought
        Item.addRow(new Object[]{item, amount, price*amount});//creating new row in the table using the items name
        total = total + saleprice;//calculating final price
       double x;
       x= Double.parseDouble(jTextField2.getText());
        total = total+ (total *x);

        Paid.setText("R" + total);//displaying the total price

    } catch (FileNotFoundException ex) {
        Logger.getLogger(JFrameTill.class.getName()).log(Level.SEVERE, null, ex);
    }
}
```

To find the total price of the Bill after GST gets added on requires the use of a mathematical function where the variable total represents the sale price + the GST which got calculated based on the text input.

27. Parsing classes of different types into each other

Purpose	Success Criterion
Converting user input (strings) into numerical values (integers/doubles) facilitates accurate calculations and data handling.	9

```
ItemArray it = new ItemArray(); //an object of the type ItemArray
double saleprice = 0; // A variable to store the selling price of the items
String item = jComboBoxItems.getSelectedItem().toString(); // Gets the descrip
int amount = Integer.parseInt(jComboBoxUnits.getSelectedItem().toString()); //
double price = it.itemprice(item, amount);//method to find out the items price
saleprice = amount * price;//calculating the amount of money required for the
it.updateFile(item, amount);//method to upadte the database,to the restoc list
Item.addRow(new Object[]{item, amount, price*amount});//creating new row in th
total = total + saleprice;//calculating final price
double x;
x= Double.parseDouble(jTextField2.getText());
total = total+ (total *x);
```

Parsing a string obtained from a text field into a double.

Parsing a string obtained from the combo box into an integer.

28. Usage of Data structure: Array Lists and Dictionary

Purpose	Success Criterion
Arrays: Sequential data storage. Dictionaries: Key-value mappings.	4, 17, 18

```
 ItemArray it = new ItemArray(); //an object of the type ItemArray
 double saleprice = 0; // A variable to store the selling price of the items
 String item = jComboBoxItems.getSelectedItem().toString(); // Gets the descr
 int amount = Integer.parseInt(jComboBoxUnits.getSelectedItem().toString());
 double price = it.itemprice(item, amount);//method to find out the items pri
 saleprice = amount * price;//calculating the amount of money required for th
 it.updateFile(item, amount);//method to upadte the database,to the restoc li
 Item.addRow(new Object[]{item, amount, price*amount});//creating new row in
 total = total + saleprice;//calculating final price
double x;
x= Double.parseDouble(jTextField2.getText());
 total = total+ (total *x);

 Paid.setText("R" + total);//displaying the total price
```

This array checks the OODB for data storage and updates it based on user, admin, and manager feedback.

```
String t = tID.getText();
String t1= "/Users/vanshchawla/Downloads/Gupta Tuck/";
String dictionary = t1.concat(t);
File f =new File(dictionary);
JOptionPane.showMessageDialog(null, dictionary);


f.mkdirs();
```

The use of dictionary shows that a new folder is getting to the dictionary once a new customer is signed up.

29. Try Catch loops

Purpose	Success Criterion
Throughout the code, try-catch loops are used to capture file and user input exception faults in order to keep the program from crashing.	1, 3, 4, 5, 7, 8, 13, 15, 18

```
try {

    String sql = " SELECT * FROM mlogin WHERE User_Name=? AND Password=? ";
    pst = con.prepareCall(sql);

    pst.setString(1, un); //useranme
    pst.setString(2, ps); //password

    rs = pst.executeQuery();

    if (rs.next()){ // its true Do this

        JOptionPane.showMessageDialog(rootPane, "You are Logging....");

        this.setVisible (false); // Sets this frame to be invisible
    MMM p = new MMM();// creates a new NewItem jFramme
    p.setVisible(true); // Displays the New Item JFramE


    }else{// its false Do this
        JOptionPane.showMessageDialog(rootPane, "Login Failed....");


    }


} catch (HeadlessException | SQLException e) {

}
```

Try-Catch Loops are used throughout the code to catch exceptions and stop the system from crashing by generating instances of the relevant exception class.

30. Conditional Statements

Purpose	Success Criterion
if-else conditions check and make decisions in programming based on whether a certain condition is true or false.	13

```
if(managercheck.isSelected()){
this.setVisible (false); // Sets this frame to be invisible
    JFrameLogIn p = null;
    p = new JFrameLogIn();
    p.setVisible(true);
// TODO add your handling code here:
}
else if(admincheck.isSelected()){
    this.setVisible (false); // Sets this frame to be invisible
    ManagerLogin p = new ManagerLogin();
    p.setVisible(true);
}
```

If else conditions used here is to distinguish between the 2 logins, if the checkbox is selected, it will open the manager login or else it will open the admin login.

31. Populating Combo Box using data from database

Purpose	Success Criterion
To show all the items available in the shop.	22

```
public ItemArray() throws FileNotFoundException { // A Constructor that reads in the text file with all the data ab
    int ItemNumb = 0; // An integer to store the item num of the items
    String description = ""; // A string to store the description of the item
    int StockLeft = 0; // An integer to store the number of units left of the item
    double CostPrice = 0; // A double to store the cost price of the object
    double SellingPrice = 0; // a double to store the selling price of the object
    int SoldMonthly = 0; // an integer to store the number of units sold monthly
    String LastOrder = ""; // a string to store the date of last order of the objecy
    Scanner scFile = new Scanner(new File("Items.txt")); // Creating a scanner object to scan the text file with th
    while (scFile.hasNext()) { // A while loop till the end of the file
        String line = scFile.nextLine(); // A variable to break up the file into lines
        Scanner scLine = new Scanner(line).useDelimiter("#"); // A scanner object to store the line, and use the de
        ItemNumb = scLine.nextInt(); // Extracts the Item number from the text file and stores it
        description = scLine.next(); // Stores the description from the text file
        StockLeft = scLine.nextInt(); // Stores the number of units left from the text file
        CostPrice = scLine.nextDouble(); // Stores the cost price from the text file
        SellingPrice = scLine.nextDouble(); // Stores the selling price from the text file
        SoldMonthly = scLine.nextInt(); // Stores the number of units sold monthly from the text file
        LastOrder = scLine.next(); // Stores the date of last order from the text file
        arr[count] = new Item(ItemNumb, description, StockLeft, CostPrice, SellingPrice, SoldMonthly, LastOrder); /
        count++; // increments the size of count, adding an element to the array
    }
```

In this array, all the items and their descriptions were defined, and they populate the combo box from the OODB.

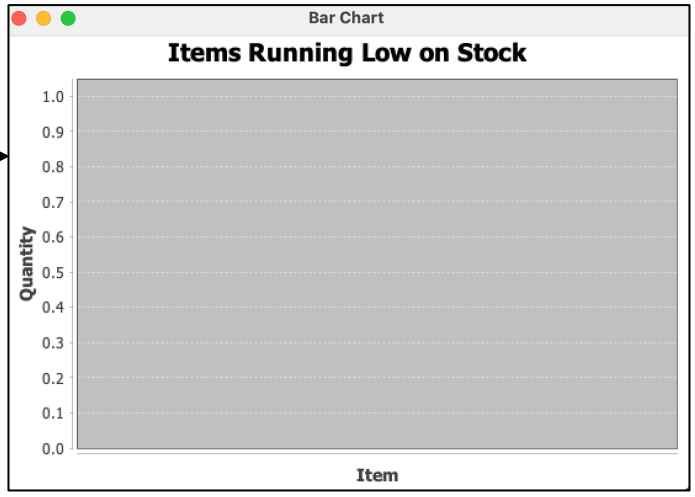
32. Graphical representations using JFreeChart

Purpose	Success Criterion
To plot a graph of all the item's running out.	1

```java
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
// Create a dataset from the text area content
        CategoryDataset dataset = createDataset();

        // Create a bar chart using JFreeChart
        JFreeChart barChart = ChartFactory.createBarChart(
                "Items Running Low on Stock", // Chart Title
                "Item", // Category Axis Label
                "Quantity", // Value Axis Label
                dataset, // Dataset
                PlotOrientation.VERTICAL,
                true, true, false
        );
        ChartPanel chartPanel = new ChartPanel(barChart);
        chartPanel.setPreferredSize(new java.awt.Dimension(560, 370));

        // Create a new JFrame to display the chart
        JFrame chartFrame = new JFrame("Bar Chart");
        chartFrame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        chartFrame.getContentPane().add(chartPanel);
        chartFrame.pack();
        chartFrame.setVisible(true);
        // TODO add your handling code here:
    }
```

This creates a graph defining its title and its axis's.

Here is a representation of the graph (The stocks were just updated, hence the quantity nothing is running low).



Word Count- 1241 Words

Bibliography

"Documentation." *FormDev Software*, www.formdev.com/jformdesigner/doc/ui/designer/button-groups/.

GfG. "Python Gui - Tkinter." *GeeksforGeeks*, GeeksforGeeks, 9 Nov. 2023, www.geeksforgeeks.org/python-gui-tkinter/.

GfG. "Random Numbers in Python." *GeeksforGeeks*, GeeksforGeeks, 5 May 2023, www.geeksforgeeks.org/random-numbers-in-python/.

GmbH, PSPDFKit. "Save PDF to Local Storage in Java: Pspdfkit SDK." *PSPDFKit*, pspdfkit.com/guides/java/save-a-document/to-local-storage/.

How to create a GUI with Java. NIIT. (n.d.). https://www.niit.com/india/knowledge-centre/create-a-GUI-with-Java

javaEd, et al. "Python GUI from Java." *Stack Overflow*, 1 Feb. 1959, stackoverflow.com/questions/16142184/python-gui-from-java.

Java Convert JTABLE Data to PDF File, www.roseindia.net/tutorial/java/swing/countJTableToPDF.html.

JFreeChart, www.jfree.org/jfreechart/. Accessed 22 Mar. 2024.

jimmyjimmy 8, et al. "How to Create a Directory in Java?" *Stack Overflow*, 1 July 1956, stackoverflow.com/questions/3634853/how-to-create-a-directory-in-java.

Malasuerte94Malasuerte94 1, et al. "Open a Link in Browser with Java Button?" *Stack Overflow*, 1 Apr. 1958, stackoverflow.com/questions/10967451/open-a-link-in-browser-with-java-button.

"Method in Java - Javatpoint." *Www.Javatpoint.Com*, www.javatpoint.com/method-in-java.

¸ Zunairah bukhariZunairah bukhari 5122 silver badges88 bronze badges, et al. "How to Open a PDF Document by Clicking a Button?" *Stack Overflow*, 1 Jan. 1963, stackoverflow.com/questions/42453720/how-to-open-a-pdf-document-by-clicking-a-button.