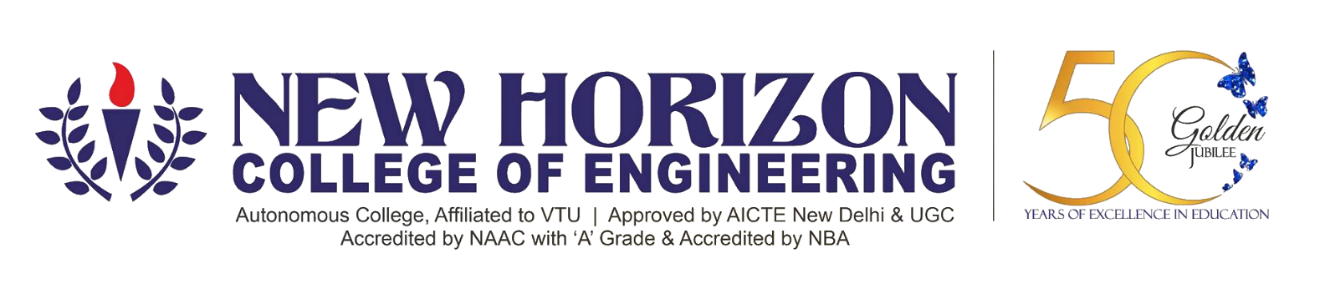
****

**A MINI PROJECT**

**REPORT**

*for*

*Mini Project in JAVA (19CSE48)*

**FINANCIAL AND PRODUCTION OPERATING SOFTWARE**

*Submitted by*

**Jashwanth M S**

**USN: 1NH19CS066,**

**Semester-Section: 4-B**

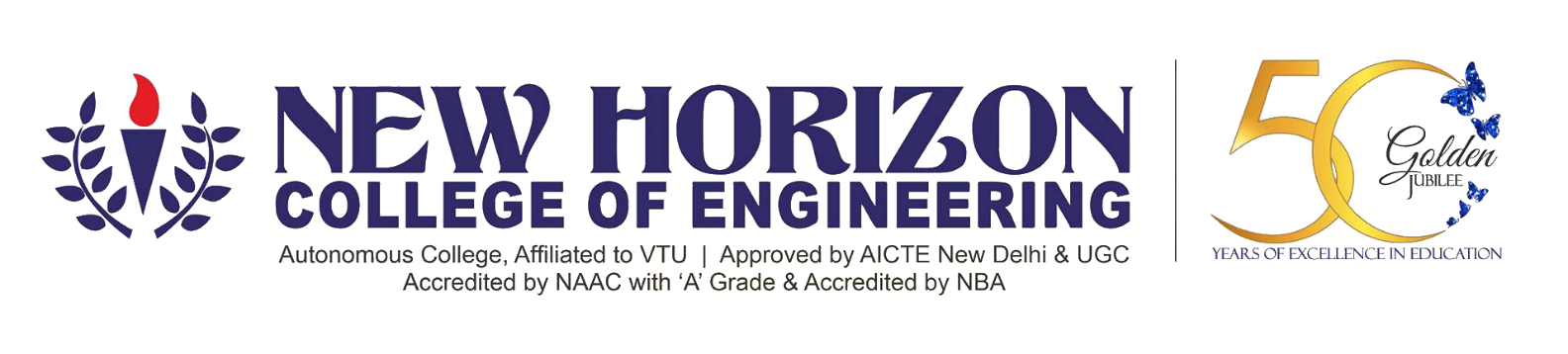
*In partial fulfillment for the award of*

*the degree of*

**Bachelor of Engineering**

*in*

**COMPUTER SCIENCE AND ENGINEERING**

***Certificate***

*This is to certify that the mini project work titled*

**FINANCIAL AND PRODUCTION OPERATING SOFTWARE**

*Submitted in partial fulfillment of the degree of*

*Bachelor of Engineering in*

*Computer Science and Engineering by*

**Jashwanth M S**

USN: 1NH19CS066

*DURING*

*EVEN SEMESTER 2020-2021*

*for*

*COURSE CODE: 19CSE48*

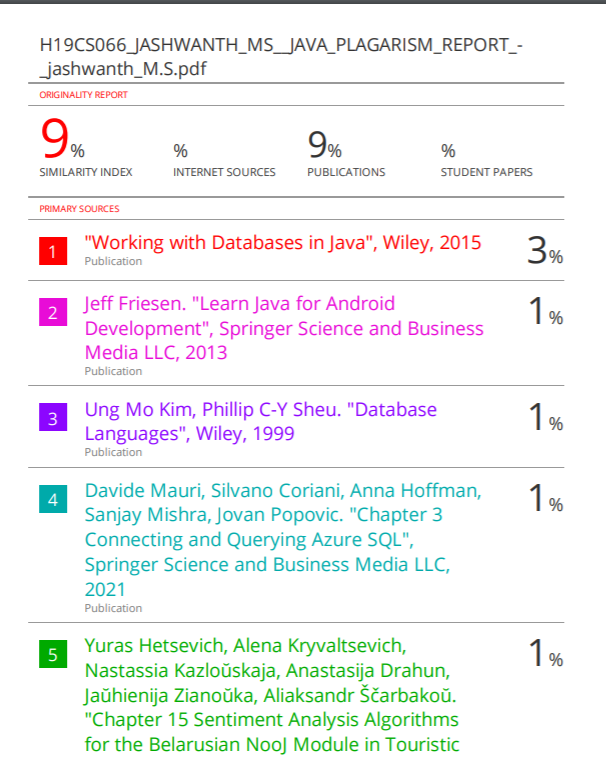
Signature of Reviewer Signature of HOD

SEMESTER END EXAMINATION

*Name of the Examiner Signature with date*

1. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**PLAGARISM CERTIFICATE**

**ABSTRACT**

# Financial and production operating software is the one that helps a person To build his business in the right way possible, Nowadays if a person wants to build an industry He doesn’t have any idea about many things like how to start, where to start, hiring employees, How other competitors are working .along with these things he won't have any idea about financial management how much salary to be paid for employees .to give a complete idea of business management we require software so that it can manage all of this chaos and helps a person to build his business in the right way possible .one of those kinds of software is Financial and production operating software. which helps in managing the whole business of person efficiently so that we can help him to bring productivity

# Keywords: MySQL, Database, Front-End, Backend, Java, OOPS.

# ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be impossible without the mention of the people who made it possible, whose constant guidance and encouragement crowned our efforts with success.

I have great pleasure in expressing gratitude to **Dr. Mohan Manghnani**, Chairman of New Horizon Educational Institutions for providing the necessary infrastructure and creating a good environment.

I take this opportunity to express my profound gratitude to **Dr. Manjunatha,** Principal, New Horizon College of Engineering, for his constant support and encouragement.

I am grateful to **Dr. Amarjeet Singh,** Dean - Academics, for his unfailing encouragement and suggestions, given to me in the course of my project work.

I would also like to thank **Dr. B. Rajalakshmi**, Professor, and Head, Department of Computer Science and Engineering, for her constant support.

I also express my gratitude to **Dr. Thirukkumaran R**, Department of Computer Science and Engineering, my project guide, for constantly monitoring the development of the project and setting up precise deadlines. His valuable suggestions were the motivating factors in completing the work.

**Jashwanth M S**

**USN: 1NH19CS066**

**CONTENTS**

**ABSTRACT II**

**ACKNOWLEDGMENT III**

**LIST OF FIGURES VI**

1. **INTRODUCTION** 
   1. PROBLEM DEFINITION **1**
   2. OBJECTIVES **1**
   3. METHODOLOGY TO BE FOLLOWED **1**
   4. EXPECTED OUTCOMES **2**
2. **FUNDAMENTALS OF JAVA PROGRAMMING 3**
   1. INTRODUCTION 3
   2. CLASS 3
   3. OBJECT 4
   4. INHERITANCE 4
   5. POLYMORPHISM 4
   6. ABSTRACT CLASS 4
   7. JAVA PACKAGES 5
   8. EXCEPTION HANDLING 5
   9. THREADS 6
   10. I/O BASICS 6
3. **REQUIREMENTS AND SPECIFICATION**
   1. HARDWARE REQUIREMENTS **22**
   2. SOFTWARE REQUIREMENTS  **22**
4. **DESIGN**
   1. DESIGN GOALS **27**
   2. FLOWCHART/AGLORITHM  **30**
5. **IMPLEMENTATION** 
   1. MODULE 1 FUNCTIONALITY **47**
   2. MODULE 2 FUNCTIONALITY  **49**
   3. MODULE 3 FUNCTIONALITY **51**
   4. MODULE 4 FUNCTIONALITY **53**
6. **RESULTS**

|  |  |  |
| --- | --- | --- |
| **6.1** | Login Page | **26** |
| **6.2** | Authentication Details | **26** |
| **6.3** | Backend -Users Data Base | **27** |
| **6.4** | Backend-Product Data Base | **27** |
| **6.5** | Frontend-Product Details | **27** |
| **6.6** | Backend-Places Data Base | **28** |
| **6.7** | Frontend-Places Details | **28** |
| **6.8** | Backend-Resources Data Base | **28** |
| **6.9** | Frontend- Resources Details | **29** |
| **6.10** | production-Management | **29** |
| **6.11** | Financial-Management | **29** |
| **6. 12** | Bill-Generation | **30** |

1. **CONCLUSION 31**

**REFERENCES 32**

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No** | **Figure Description** | **Page No** |
| 1 | Login Page | 26 |
| 2 | Authentication Details | 26 |
| 3 | Backend -Users Data Base | 27 |
| 4 | Backend-Product Data Base | 27 |
| 5 | Frontend-Product Details | 27 |
| 6 | Backend-Places Data Base | 28 |
| 7 | Frontend-Places Details | 28 |
| 8 | Backend-Resources Data Base | 28 |
| 9 | Frontend- Resources Details | 29 |
| 10 | production-Management | 29 |
| 11 | Financial-Management | 29 |
| 12 | Bill-Generation | 30 |
|  |  |  |
|  |  |  |

**CHAPTER 1**

**INTRODUCTION**

**1.1 PROBLEM DEFINITION**

* To develop a Computerized operating software which will be help full in managing the business of the person based on his area of interest
* The software should be able to understand the requirement of users

And to help in the best way possible to grow his business

* The software should be able to handle the back-end data which is stored in the database and to perform operations on that data
* Processed data from the backend should be displayed to the user so that he can get information about that particular thing and he can perform operations on that data
* A proper database management system in the backend like MySQL

To arrange data in an organized manner

**1.2 OBJECTIVES**

* To understand and realize the real-world project, understand the problems associated with developing it, and explore the appropriate solution to it.
* To understand the application development in the desktop application-based system.
* To identify Java language components and how they work together in applications.
* To analyze and apply the JDBC concepts and create applications based on the database.
* To learn how to use exception handling in Java applications.
* The specific objective of my project is to help a person to grow his businesses in an efficient way possible

**1.3 METHODOLOGY TO BE FOLLOWED**

First to create a login page after getting authentication from the login page further process is continued

The proper database management system is connected to the software to process the backend data efficiently

Using concepts of classes objects and constructors along with exception handling Input-output operations are carried out

**1.4 EXPECTED OUTCOMES**

* Understood the application development in desktop application-based systems.
* Analyzed and applied the JDBC concepts and linked the database and Java UI.
* The system is made user-friendly and convenient to use.

**CHAPTER 2**

**FUNDAMENTALS OF JAVA PROGRAMMING**

**2.1** **Financial and production operating software**

Financial and production operating software uses Concepts of Object-oriented programming using java as a programming language along with that it also uses Java Database connection using MySQL as a database

**2.2** **CLASS**

A Class is like an object constructor or a "blueprint" for creating objects. In **OOPs**,

a **class** is an extensible program-code-template for creating objects, providing initial

values for state (member variables) and implementations of behavior (member functions

or methods)

**2.3** **OBJECT**

An entity that has a state and behavior is known as an object e.g., chair, bike, marker, pen,

table, car, etc. It can be physical or logical (tangible and intangible). An example of an

intangible object is the banking system.

An object has three characteristics:

o **State:** represents the data of an object.

o **Behavior:** represents the functionality of an object such as deposit, withdrawal, etc.

o **Identity:** An object identity is typically implemented via a unique ID. The value

of the ID is not visible to the external user. However, it is used internally by the

JVM to identify each object uniquely.

**2.4 INHERITANCE**

Inheritance in Java is a mechanism in which one object acquires all the properties and

behaviors of a parent object. It is an important part of OOPs. Inheritance represents

the IS-A relationship which is also known as a *parent-child* relationship.

Terms used in Inheritance

o Class**:** A class is a group of objects which have common properties. It is a

template or blueprint from which objects are created.

o Sub Class/Child Class: A subclass is a class that inherits the other class. It is also

called a derived class, extended class, or child class.

o Super Class/Parent Class: Superclass is the class from where a subclass inherits

the features. It is also called a base class or a parent class.

o Reusability: As the name specifies, reusability is a mechanism that facilitates

you to reuse the fields and methods of the existing class when you create a new

class. You can use the same fields and methods already defined in the previous

class.

**2.5 POLYMORPHISM**

Polymorphism in Java is a concept by which we can perform a single action in different

ways. There are two types of polymorphism in Java: compile-time polymorphism and

runtime polymorphism. We can perform polymorphism in java by method overloading

and method overriding.

**2.6 ABSTRACT CLASS**

Data **abstraction** is the process of hiding certain details and showing only essential

information to the user. **Abstraction** can be achieved with either **abstract** classes or

interfaces

The abstract keyword is a non-access modifier, used for classes and methods:

· **Abstract class:** is a restricted class that cannot be used to create objects (to access

it, it must be inherited from another class).

· **Abstract method:** can only be used in an abstract class, and it does not have a

body. The body is provided by the subclass (inherited from).

**2.7 JAVA PACKAGES**

A java package is a group of similar types of classes, interfaces, and sub-packages. Package in java can be categorized in two forms, built-in package, and user-defined package. There are many built-in packages.

**2.8 EXCEPTION HANDLING**

The Exception Handling in Java is one of the powerful mechanisms to handle the runtime

errors so that normal flow of the application can be maintained.

The core advantage of exception handling is to maintain the normal flow of the

application. An exception normally disrupts the normal flow of the application that is why

we use exception handling.

**2.9 THREADS**

A thread, in the context of Java, is the path followed when executing a program. All Java programs have at least one thread, known as the main thread, which is created by the Java Virtual Machine (JVM) at the program’s start when the main() method is invoked with the main thread.

In Java, creating a thread is accomplished by implementing an interface and extending a class. Every Java thread is created and controlled by java. lang.Thread class.

A single-threaded application has only one thread and can handle only one task at a time. To handle multiple tasks in parallel, multi-threading is used: multiple threads are created, each performing a different task.

**2.10 I/O BASICS**

Java uses the concept of a stream to make I/O operations fast. The java.io package contains all the classes required for input and output operations. We can perform file handling in Java by Java I/O API.

**CHAPTER 3**

**REQUIREMENT SPECIFICATION**

**3.1** **HARDWARE SPECIFICATIONS**

• Processor - Intel Core i5

• Speed - 1.8 GHz

• RAM - 256 MB

• Hard Disk - 10 GB

**3.2 SOFTWARE SPECIFICATIONS**

• Operating System - Any JDK installed OS

• Database - MySQL

**CHAPTER 4**

**DESIGN**

**4.1 DESIGN GOALS**

Financial and production operating software is designed in such a way that it should be useful to a beginner in the industry who is unaware of the financial and production operations of the industry

This software is designed using concepts of OOPS using java as a programming language as a part of the front-end and to maintain an efficient database we have used MySql as a database which is implemented using SQL queries

**4.2 ALGORITHM**

* Welcoming the use to the software with a welcome message
* Process of logging in depending on if a user

1. If a user is new to this software he has to register as a new user by selecting his user id and password and then undergo the process of login
2. If a user has already had a user id and password he can log in using the same user-id and password
3. If the user enters the wrong user id and password he has to undergo the process of login again and again until he enters the right password or else he has to exit from the software

* Confirming the user that he has logged in Successfully
* Requesting the user to hit the enter key to continue with the further process
* Displaying the contents which are fetched from MySQL databases here the contents are the list of products to which the software gives a service
* Asking the user to choose from the displayed items concerning what he needs the service from the software
* Depending on the choice of the user further process will continue
* If the user enters, he needs guidance for the mobile manufacturing industry then the following process takes place
  + - * 1. Displaying the list of places where he can start his manufacturing unit
        2. Asking the user to hit enter key to go for further processing of the software
        3. Displaying the list of Resources that are required for manufacturing mobile phones
        4. Asking the user to enter his cost of production
        5. Displaying some contents with respect to production of mobiles after calculating using his production budge and give him an idea of how many boxes, units can be manufactured
        6. After finishing the process of production management he has to go through the process of financial management
        7. Asking the user to hit enter key to undergo the process of Financial management
        8. Under financial management, many things like how much salary he has to pay for the employees, how much GST is applied to his product, what is the total cost of manufacturing cost per unit will be displayed along with that information about other brands selling price is also displayed
        9. Now the user has got two options either he can exit the software or else he can continue to the process of bill generation
        10. If he undergoes the process of bill generation concerning how manyunits he wants to sell a bill is generated

**CHAPTER 5**

**IMPLEMENTATION**

**5.1 MODULE 1 FUNCTIONALITY**

**5.1.1 LOGIN /REGISTER**

Under this section user has to log in with his user id and password if don’t have a user id password he has to create new and again login all his user id and password data are stored into the database for further use

**Coding part:**

LoginConstructor obj=new LoginConstructor();

if(a==1)

{

LoginConstructor obj1=new LoginConstructor();

}

class LoginConstructor

{

public static int a;

int q;

public LoginConstructor()

{

System.out.println("## Welcome ##");

System.out.println("1.NewRegistration");

System.out.println("2.To login");

Scanner sc=new Scanner(System.in);

a=sc.nextInt();

if(a!=1 && a!=2)

{

System.out.println("Invalid input");

System.exit(0);

}

if(a==1)

{

Log l1=new Log();

int i=l1.register();

if(i==0)

{

while(i!=1)

{

System.out.println("1.Reregister again 2.to exit");

Scanner sc2=new Scanner(System.in);

q=sc2.nextInt();

if(q==1)

{

int i1=l1.register();

i=i1;

}else if(q==2)

{

System.exit(0);

}

}

}

}else if(a==2)

{

Log l2=new Log();

int r=l2.login();

if(r==0)

{

while(r!=1)

{

int a,b;

System.out.println("1.Relogin again 2.to exit ");

Scanner sc2=new Scanner(System.in);

a=sc2.nextInt();

if(a==1)

{

int r1=l2.login();

r=r1;

}else if(a==2)

{

System.exit(0);

}

}

}

} else

{

System.out.println("Invalid input");

}

}

}

class Login

{

String uid=null;

String passd=null;

}

class variables{

String url="jdbc:mysql://localhost:3306/miniproject";

String username="root";

String passwd="msjashu456";

String query1="insert into login values (?,?)";

String query2="select \* from login ";

}

class Registercheck extends variables

{

String query3="select \* from login";

int flag=1;

public int rch(String s1,String s2)

{

try {

Class.forName("com.mysql.cj.jdbc.Driver");

Connection con=DriverManager.getConnection(url,username,passwd);

Statement at=con.createStatement();

ResultSet rs=at.executeQuery(query3);

while(rs.next())

{

String a1=rs.getString(1);

if(s1.equals(a1))

{

flag=0;

}

}

}catch(Exception e)

{

}

return flag;

}

}

class Log extends variables

{

public int register() {

Login obj1=new Login();

Scanner sc2=new Scanner(**System**.in);

System.out.println("Enter Your User id");

obj1.uid=sc2.nextLine();

System.out.println("Enter Your password");

obj1.passd=sc2.nextLine();

Registercheck r1=new Registercheck();

int c=r1.rch(obj1.uid,obj1.passd);

if(c==1)

{

try {

Class.forName("com.mysql.cj.jdbc.Driver");

Connection con=DriverManager.getConnection(url,username,passwd);

PreparedStatement pt=con.prepareStatement(query1);

pt.setString(1, obj1.uid);

pt.setString(2, obj1.passd);

int k=pt.executeUpdate();

System.out.println(k+" row/s changed");

}catch(Exception e)

{

System.out.println(e);

}

}else {

System.out.println("User id has taken already try another one ");

}

return c;

}

public int login()

{

int flag=0;

Login obj2=new Login();

Scanner sc1=new Scanner(System.in);

System.out.println("Enter Your User id");

obj2.uid=sc1.nextLine();

System .out.println("Enter Your Password");

obj2.passd=sc1.nextLine();

try

{

Class.forName("com.mysql.cj.jdbc.Driver");

Connection con=DriverManager.getConnection(url,username,passwd);

Statement at=con.createStatement();

ResultSet rs=at.executeQuery(query2);

while(rs.next())

{

String s1=rs.getString(1);

String s2=rs.getString(2);

if(obj2.uid.equals(s1) && obj2.passd.equals(s2))

{

flag=1;

System.out.println("Login done");

break;

}

}

if(flag==0)

{

System.out.println("Login Unsucess full");

}

}catch(Exception e)

{

System.out.println(e);

}

return flag;

}

}

**5.2 MODULE 2 FUNCTIONALITY**

**5.2.1** From here user choices will be taken into consideration if chooses he wants to build a mobile industry then the following operations will take place

class Mobilevar

{

String url="jdbc:mysql://localhost:3306/mobilephones";

String uname="root";

String passwd="msjashu456";

String query="select \* from places1";

String query1="select \* from resourses1";

public int p,e,c;

}

class Dataconnection extends Mobilevar

{

void query(String q)

{

try{

Class.forName("com.mysql.cj.jdbc.Driver");

Connection con=DriverManager.getConnection(url,uname,passwd);

Statement st=con.createStatement();

ResultSet rs=st.executeQuery(q);

while(rs.next())

{

System.out.println(rs.getInt(1)+" "+rs.getString(2));

}

st.close();

con.close();

}catch(Exception e)

{

System.out.println(e.getMessage());

}

}

}

class MobilePhones extends Mobilevar{

public void fun()

{

Dataconnection obj=new Dataconnection();

obj.query(query);

}

public void fun1()

{

Enter obj1 = new Enter();

obj1.f1();

Dataconnection obj=new Dataconnection();

System.out.println("Resourses Required for Manufacturing");

System.out.println();

obj.query(query1);

Scanner sc1= new Scanner(System.in);

System.out.println();

System.out.println("Enter Your cost of production");

p=sc1.nextInt();

Product objw=new Product();

System.out.println(objw.employeeSelector(p)+" employees should be hiered");

System.out.println(objw.pices()+" pices are manufactured with respect your budget");

System.out.println(objw.boxes()+" boxes are produced");

Enter obje=new Enter();

System.out.println("To continue with Financial management");

obje.f1();

System.out.println("Amout of salary to be paid for all "+objw.x+"employess");

System.out.println( objw.salary());

System.out.println("Gst of 18% is for your product"+objw.gst(p));

int c=p+objw.salary()+objw.gst(p);

System.out.println("Total cost of your manufacturing unit"+c);

System.out.println("cost per unit will be"+objw.cpu(c));

objw.comp();

System.out.println("Enter the price at which you want to sell");

float s1;

s1=sc1.nextFloat();

objw.prolo(s1,c);

System.out.println("Enter 1 to generate bill 2 to exit");

int i1=sc1.nextInt();

if(i1==1)

{

objw.gbill(p);

System.exit(0);

}else {

System.exit(0);

}

**}**

**5.3 MODULE 3 FUNCTIONALITY**

Functions from other class which has to be imported to the main class And used for processing the particular product data

package Product;

import java.sql.\*;

import java.util.Scanner;

public class Product {

public int x,y,z,q,pro;

float fa,transport=0;

public int employeeSelector(int p)

{

pro=p;

x=(p/10000)\*1;

return(x);

}

public int pices() {

y=(pro/5000);

return y;

}

public int boxes()

{

z=(y\*1)/10;

return z;

}

public int salary()

{

q=x\*1650;

return q;

}

public int gst(int p)

{

int x;

x=(18\*p)/100;

return(x);

}

public float cpu(float c)

{

float x=(c)/y;

return x;

}

public void comp()

{

System.out.println("List of other brands and their selling price");

try {

String url="jdbc:mysql://localhost:3306/mobilephones";

String uname="root";

String passwd="msjashu456";

String query="select \* from comp";

Class.forName("com.mysql.cj.jdbc.Driver");

Connection con=DriverManager.getConnection(url,uname,passwd);

Statement st=con.createStatement();

ResultSet rs=st.executeQuery(query);

while(rs.next())

{

System.out.println(rs.getInt(1)+" "+rs.getString(2)+" "+rs.getFloat(3));

}

st.close();

con.close();

}catch(Exception e)

{

System.out.println(e.getMessage());

}

}

public void prolo(float f,int c)

{

float M,N;

fa=f;

M=y\*f;

N=M-c;

if(M>c)

{

System.out.println("Profit has been generate which is"+N);

}else {

System.out.println("You have to change your price you are in loss");

}

}

public void transpo()

{

System.out.println("Enter the place distance in Kilometers");

Scanner obj1=new Scanner(System.in);

float distance=obj1.nextFloat();

System.out.println("Enter the Price per Kilometer ");

float price=obj1.nextFloat();

float totalprice=distance\*price;

transport=totalprice;

}

public void gbill(int p)

{

String d="MobilePhones";

Scanner sc=new Scanner(System.in);

System.out.println("Enter Number of pieces you want to sale");

int n=sc.nextInt();

if(n<=y)

{

float t=n\*fa;

System.out.println(" \*\*\*\*\* BILL FROM THE COMPANY \*\*\*\*\* ");

System.out.println("Description |"+" GST |"+"Number of pieces|"+" price per unit |"+" Amount|");

System.out.println(d+" | "+" 18%|"+n+" |"+fa+" |"+t+"|");

System.out.println("Tranportation Fee |"+" |"+transport);

System.out.println("Total Roundoff |"+" "+(t+transport)+ "|");

}else {

System.out.println("No stock available");

}

}

}

**5.4 MODULE 4 FUNCTIONALITY**

Connection to data base to fetch the information bringing the the

Data to front end part to process that on instructions

package Product;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.Statement;

//import Projectpack.Mobilevar;

public class product01

{

public String url="jdbc:mysql://localhost:3306/mobilephones";

public String uname="root";

public String passwd="msjashu456";

public String query="select \* from places1";

public String query1="select \* from resourses1";

public int p,e,c;

public void query(String q)

{

try{

Class.forName("com.mysql.cj.jdbc.Driver");

Connection con=DriverManager.getConnection(url,uname,passwd);

Statement st=con.createStatement();

ResultSet rs=st.executeQuery(q);

while(rs.next())

{

System.out.println(rs.getInt(1)+" "+rs.getString(2));

}

st.close();

con.close();

}catch(Exception e)

{

System.out.println(e.getMessage());

}

}

}

**CHAPTER 6**

**RESULTS**

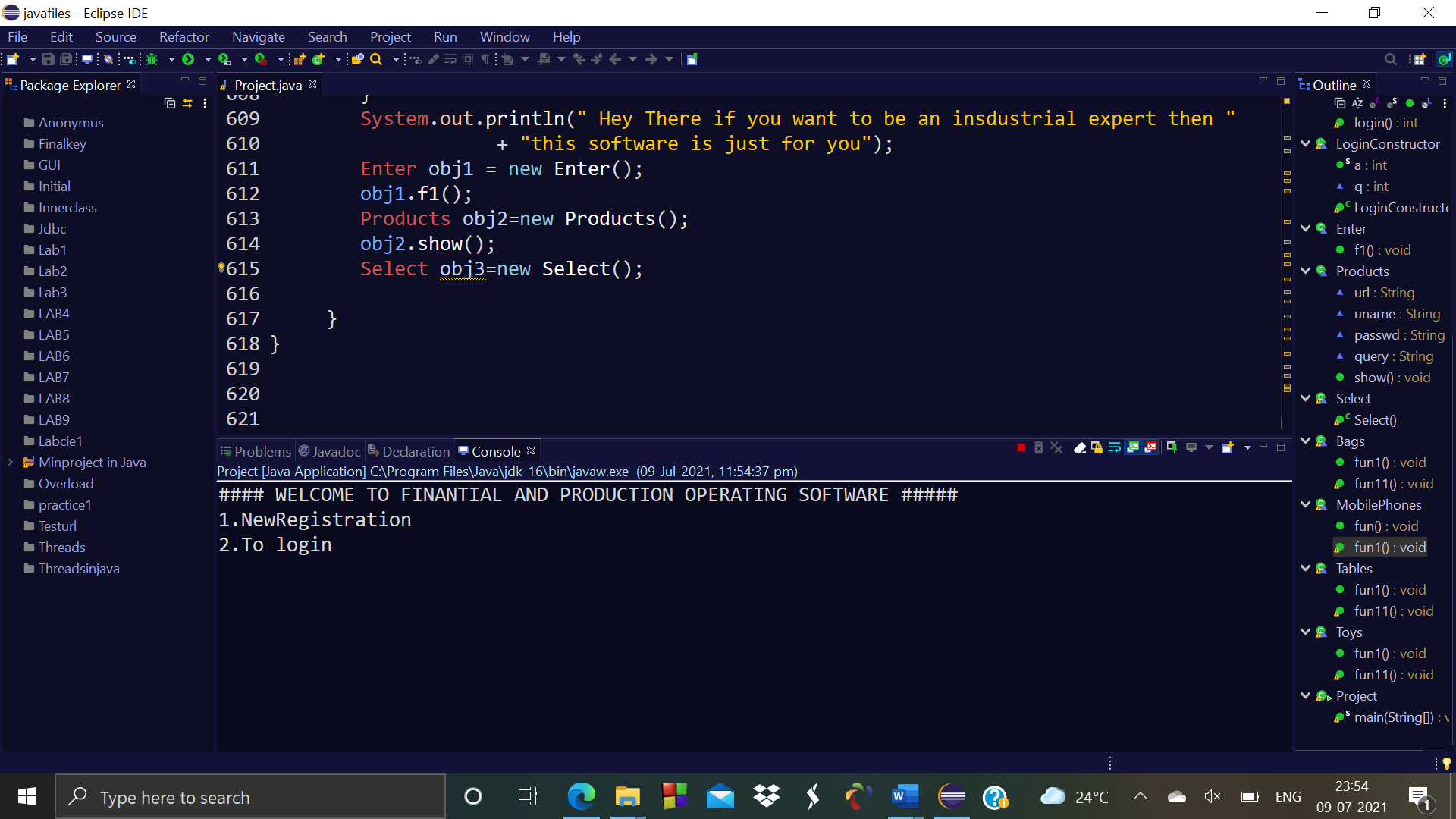


Figure 6.1 Login Page

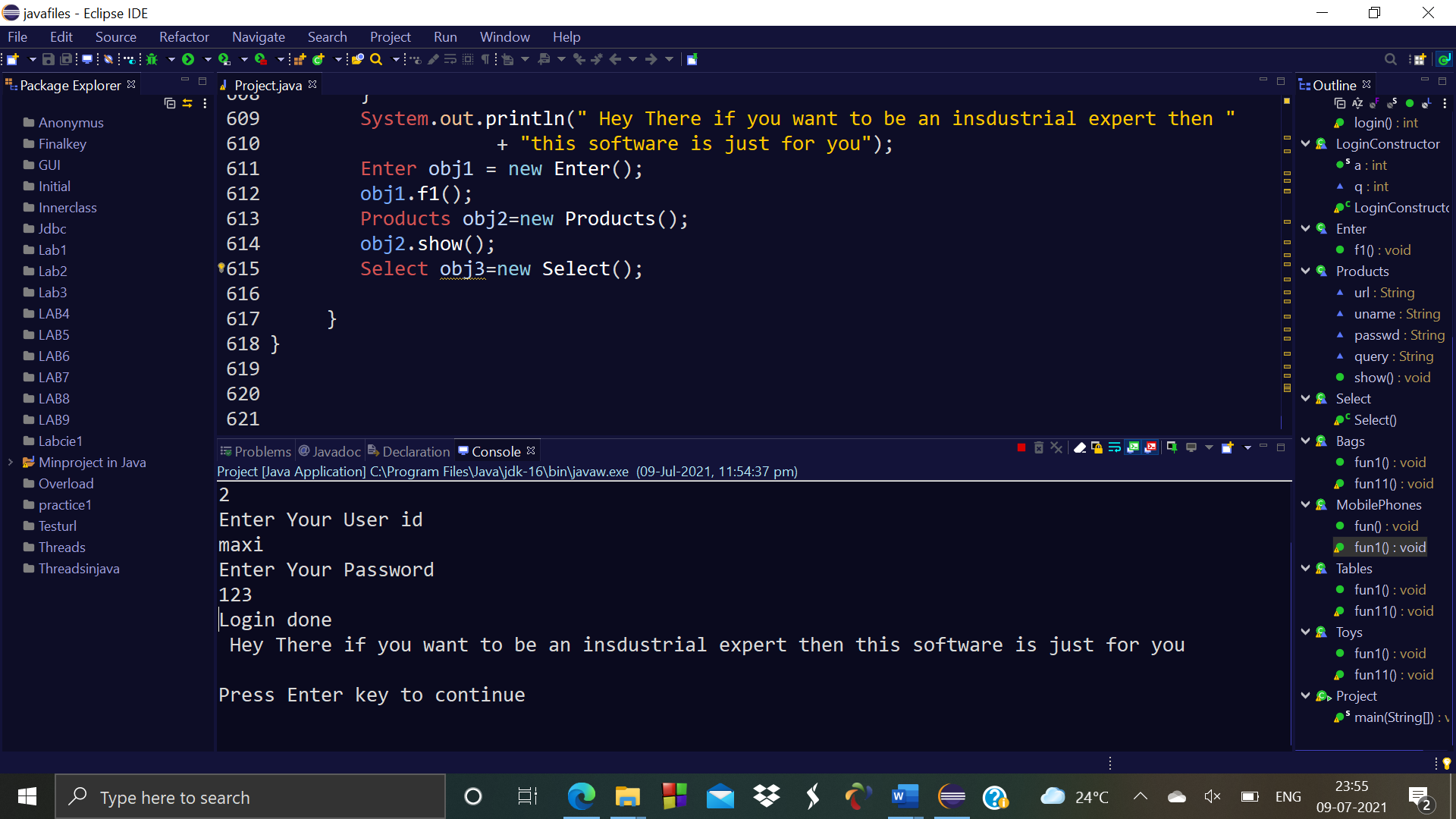


Figure 6.2 Authentication of details

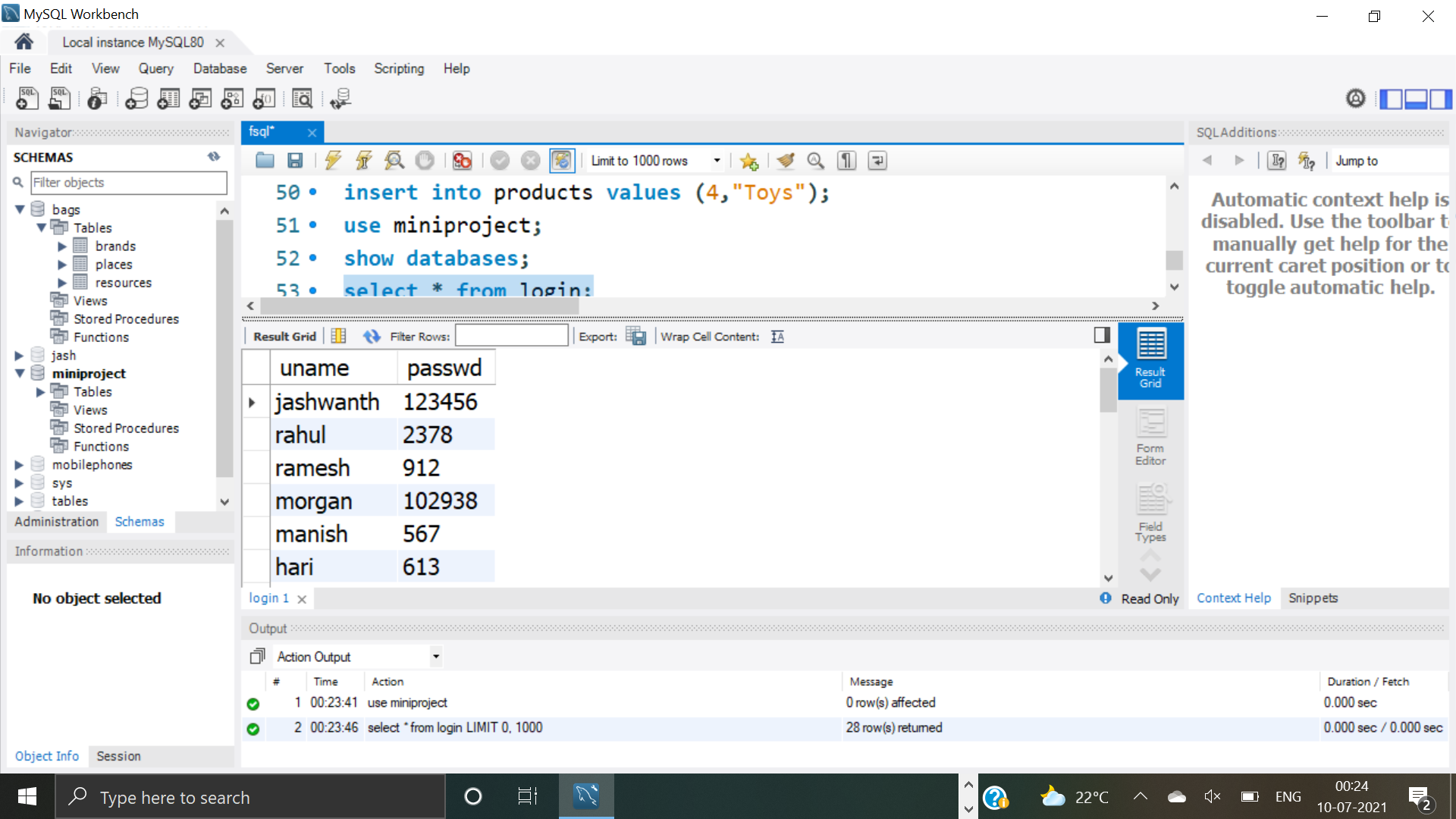


Figure 6.3 Backend where user details are stored in data base

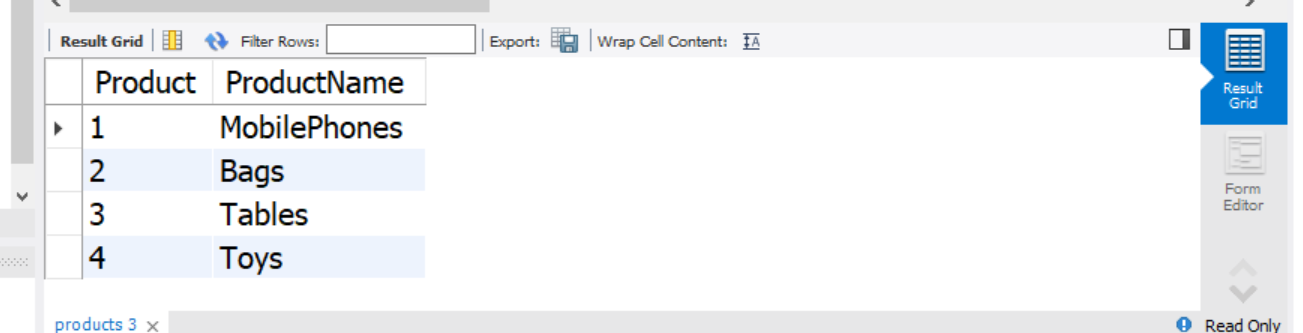


Figure 6.4 Backend where Product details are stored

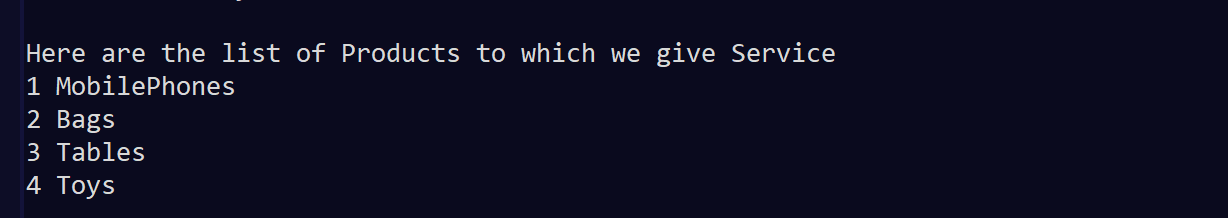


Figure 6.5 frontend where backend details are displayed

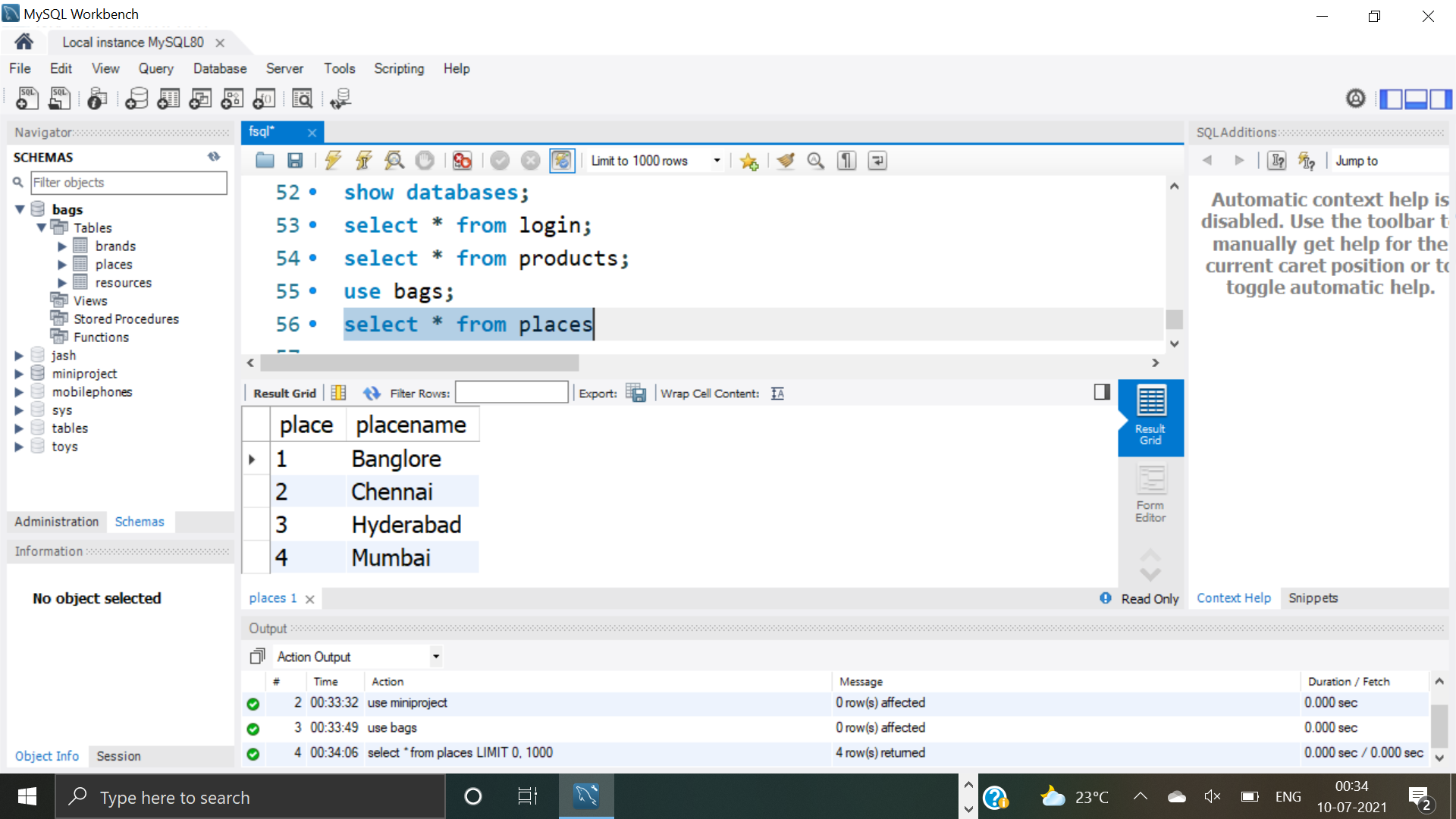


Figure 6.6 Back-end where place details with respect to product is stored

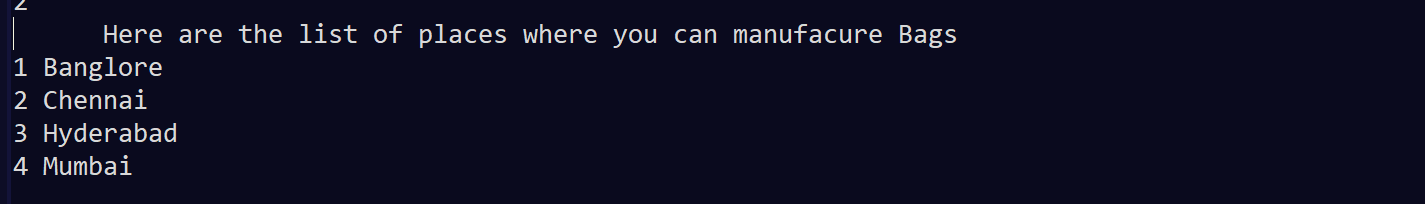


Figure 6.7 Front-end part where data is displayed to the user

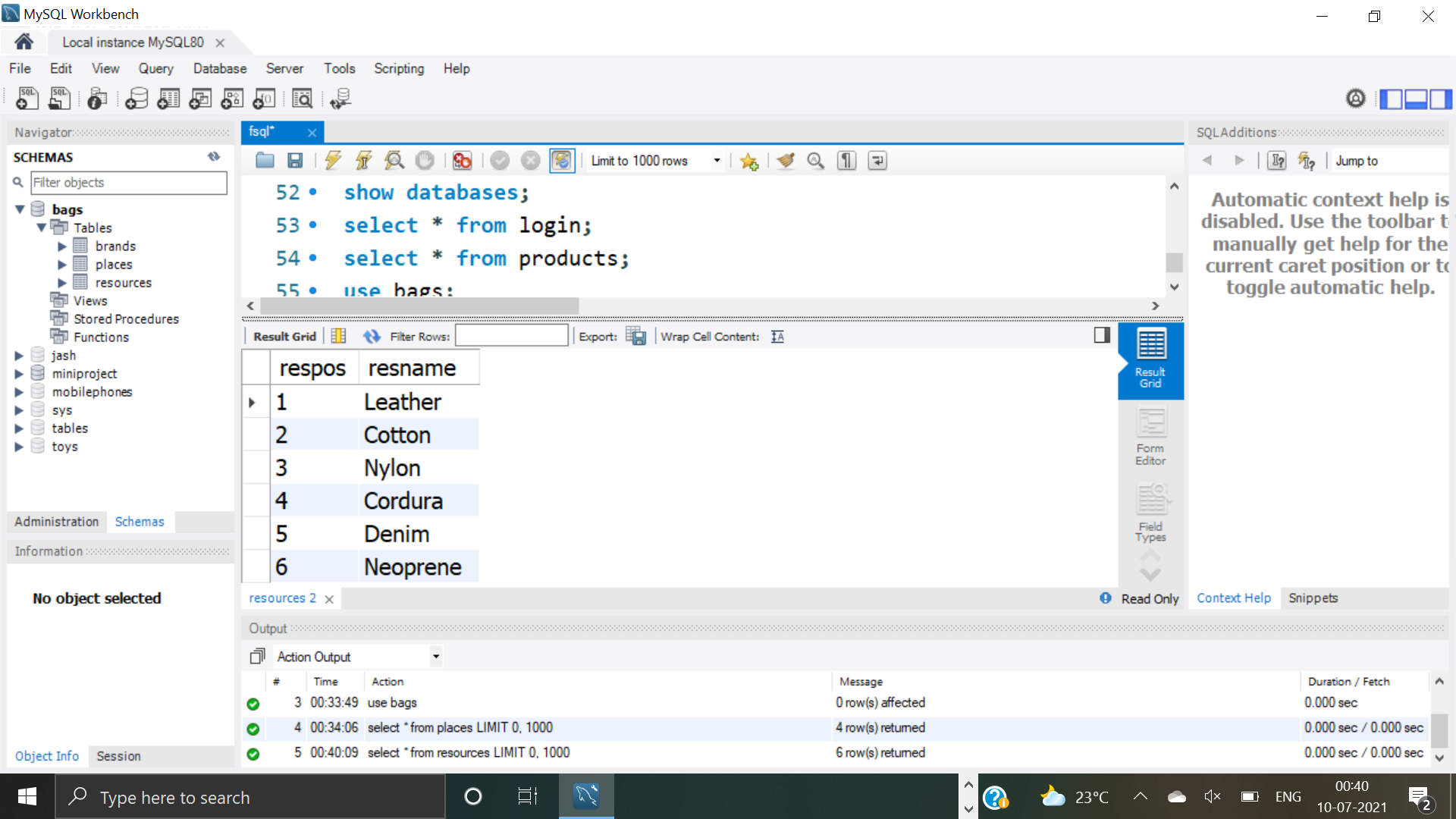


Figure 6.8 Backend where product resources data is stored

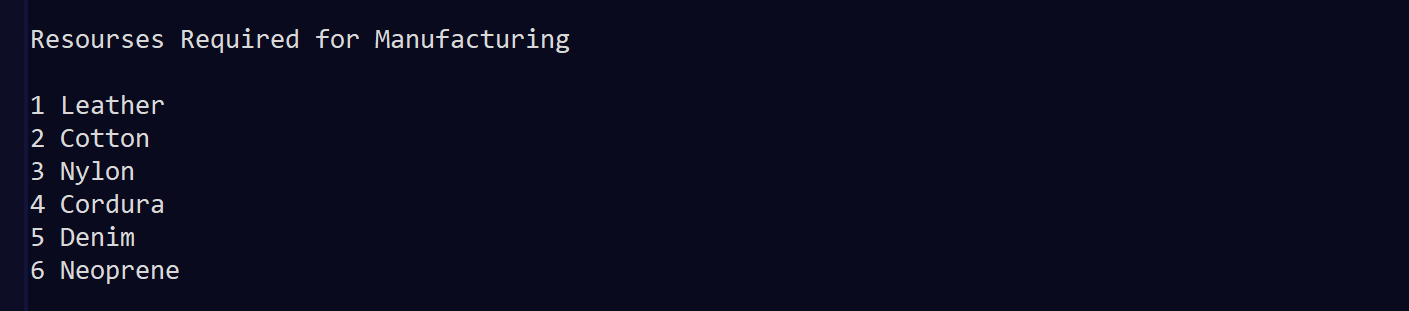


Figure 6.9 Front-end part where data is displayed to the user

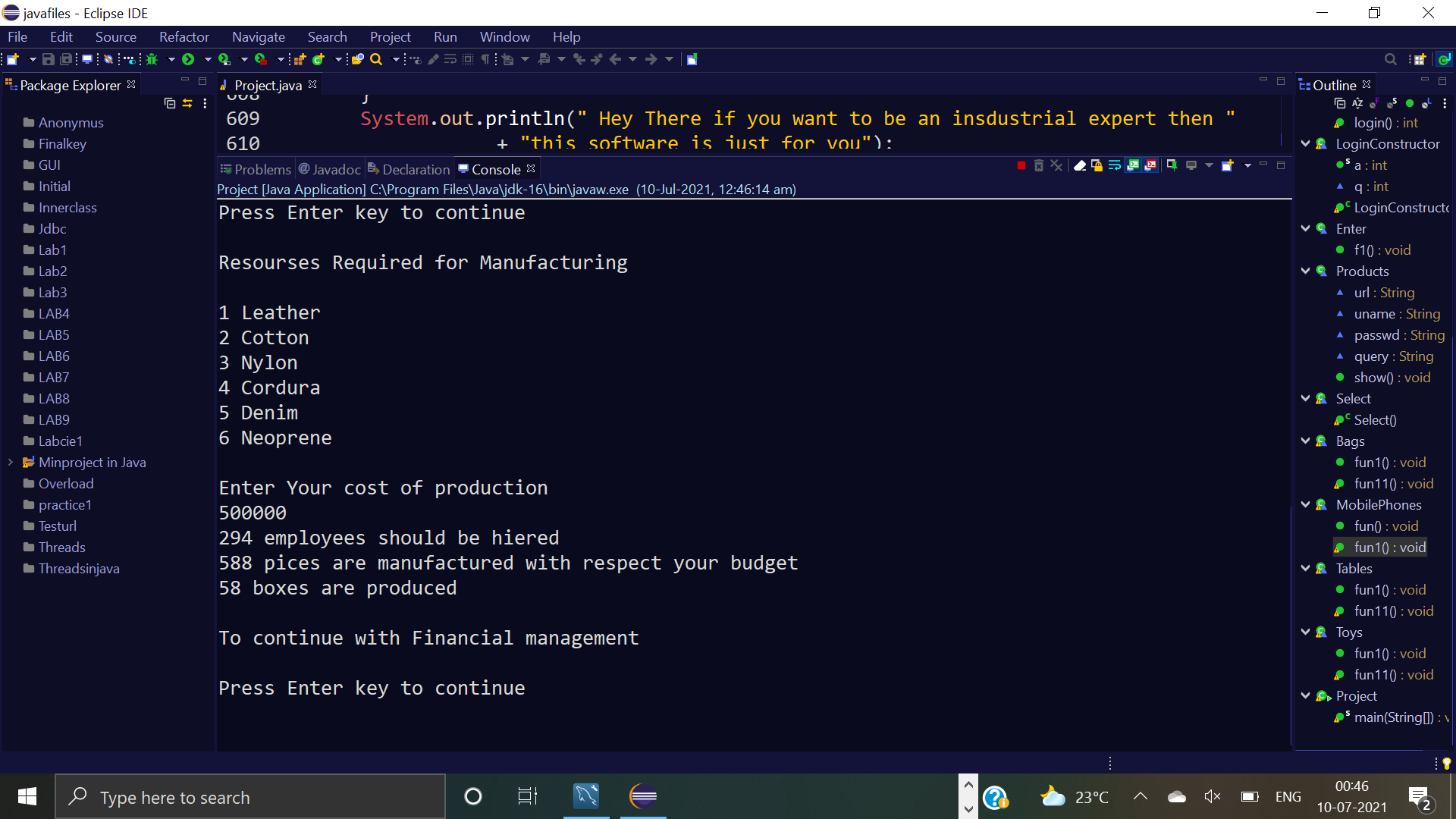


Figure 6.10 Production Management with respect to product

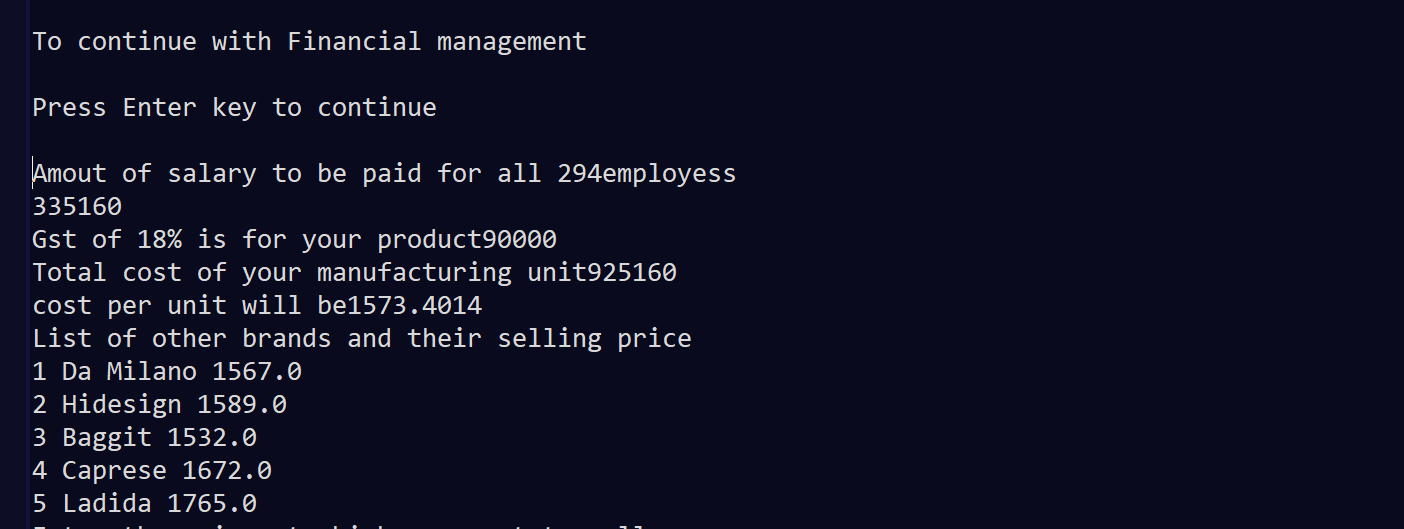


Figure 6.11 Financial Management with respect to product

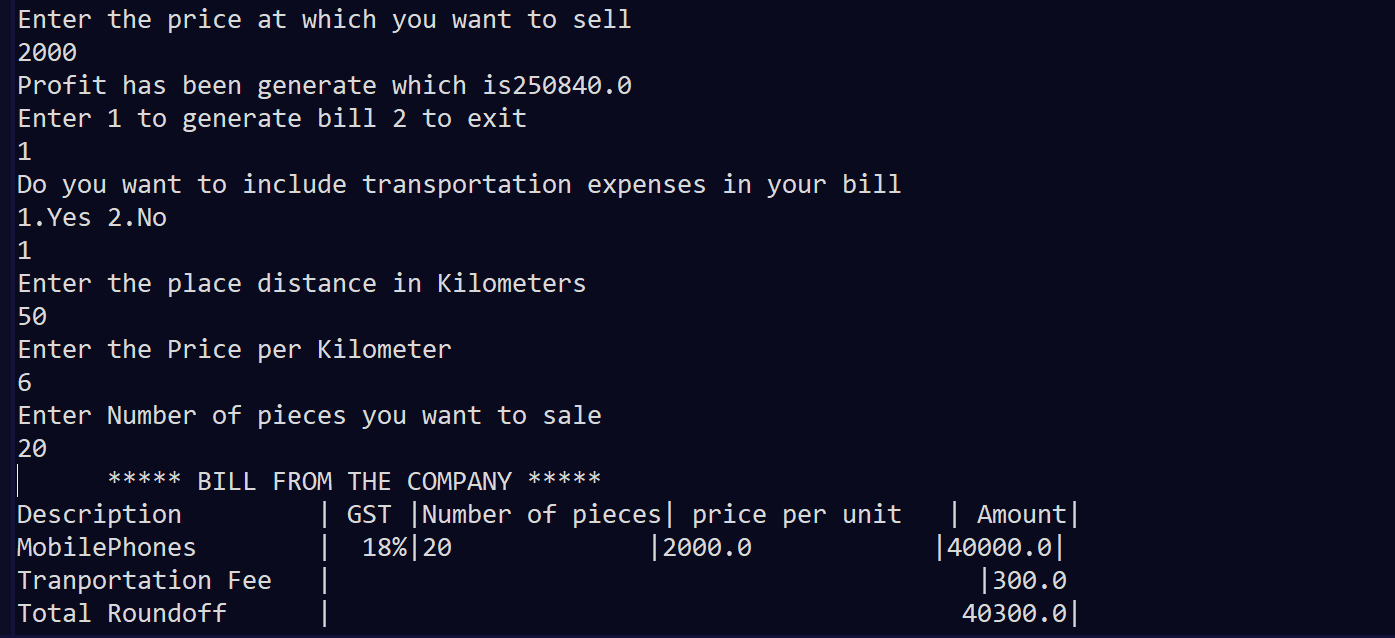


Figure 6.12 Checking probability of loss or profit and generating bill for the importer

**CHAPTER 7**

**CONCLUSION**

• This project has more scope in the future and can be integrated into a web or mobile app rather than a desktop application.

• This project is successfully implemented with all the features mentioned earlier.

• if this application is properly deployed then anyone with this application can easily find solutions for his industrial problems

• Therefore, we are successfully able to reach the goals and target of the project.

**REFERENCES**

[1] <https://www.programiz.com/>

[2] <https://www.geeksforgeeks.org/>

[3] <https://www.javatpoint.com/>

[4] <https://www.tutorialspoint.com/>

[5] <https://www.edureka.co/>

[6] <https://www.tutorialsteacher.com/>