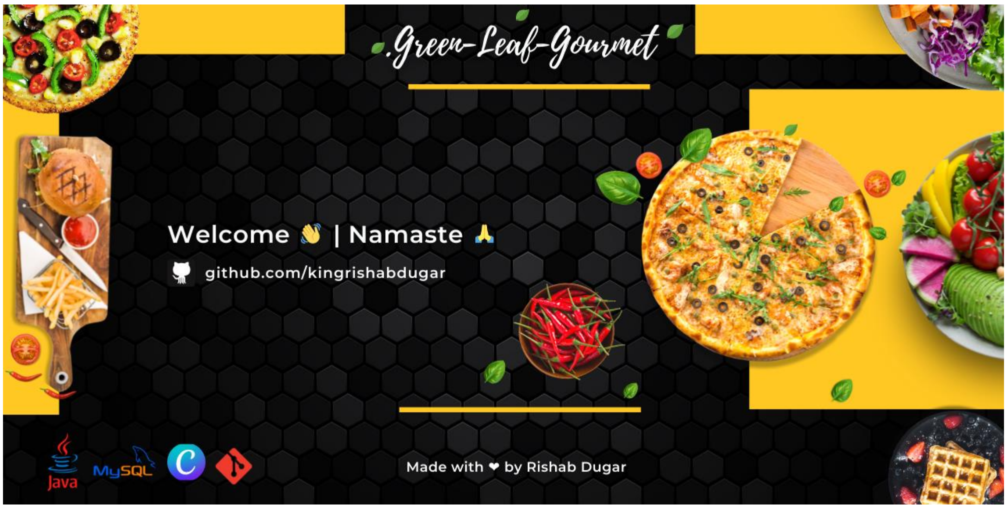
🍃Green-Leaf-Gourmet🍃

Foodies welcome here 😋😝 | Eat natural 🌱 – Eat organic 💭 | Directly from Mother Earth🌍

About 🍃Green-Leaf-Gourmet🍃



- *A full-stack Restaurant Management Desktop application with features to store and view information about users, add, remove & view items in a menu and efficiently manage orders. While ordering, users can select menu items with their quantity/number. Furthermore, They can calculate the total cost and generate a bill/receipt for the customer. With Administrator privileges one gets access to more features viz verifying and approving users, managing categories & products listed etc*
- *The Frontend (User Interface) of the application designed in Java Swing using the NetBeans GUI Builder.*
- *It combines all the benefits of traditional point-of-sale (POS) systems with tools that helps in managing user information, taking table reservations, streamline inventory management, manage bills, and build an online presence. It also integrates with the existing technology systems (such as accounting and employee management software)*
- *In today's fast-moving world, the necessity of management software is frequently neglected. It is looked down upon as little more than a mobile cashier. Management Software Systems, on the other hand, are capable of much more. They streamline operations, minimize wait times, and have a*

significant impact on customer happiness, which has a direct impact on the success of any venture.

Aim of the Project

The aim of the project is to provide a basic understanding of Full Stack Software Development and how it can be achieved in Java and MySQL, without the use of any additional framework.

During schooldays my ISC project was based on a Restaurant Management System , although it was a much simpler and basic project that ran on Blue J and returned the total bill for the user. Back in my 2nd year there was a competition known as 10daysofCode where we had to build an application from scratch within 10 days under the guidance of a mentor . Since I was much connected to Java then hence I opted for a mentor with specialization in Java projects and under his guidance I made a basic Java Application on NetBeans using JavaSwing and mySQL. Later on I kept on upgrading this project and its UI and added new features to it to make it more end- user friendly and attractive.

Why Java ?

Since it was done in the beginning of the 2nd year , I was not much proficient in C++ back then and felt more connected to Java due to previous experience in school. Also

Also JAVA, as we all know is one of the most popular OOP(Object Oriented Programming) Languages. This is one of the reasons why we chose JAVA for building this application as it is much easier to create database models and visualize them in the form of classes and objects.

Swing is a set of program component s for Java programmers that **provide the ability to create graphical user interface (GUI) components, such as buttons and scroll bars, that are independent of the windowing system for specific operating system**

Full Stack Meaning ?

A Full Stack Project is simply one in which you create both the frontend and the backend, eventually connecting it to a database
Any application be it a website, or a desktop application or a mobile phone application has a certain user interface, something you see and work on. Now what you see and interact with is called the **Frontend** of that application, i.e the client-side application.
This invisible part of any application which governs the behaviour of the application is called the **Backend** (or server-side) of that application.
Storing data and accessing data is the most fundamental aspect of development and
we do that with the help of a **Database**.

Features

- *Attractive User Interface with separate sections for Ordering, viewing, editing & deleting categories & products, user details etc*
- *User-Friendly options for retrieving security details viz. "Forgot Password", "Change Security Question" etc*
- *Registration with Admin approval to manage all users with administrator access*
- *Get the total bill value with the option to generate a Bill receipt in PDF format*
- *View past Bill receipts generated with the option to filter by date and sort by time created*

💻 Tech Stacks used

Creating PDF with Java and iText

iText is a Java library which allows to create PDF, read PDF and manipulate them.

http://sourceforge.net/projects/itext/. The download contains one jar which is required if you want to use iText.

https://howtodoinjava.com/java/library/read-generate-pdf-java-itext/

New Features to be Implemented :

1. **Java Salted Password Hashing**

Hashing is great for protecting the passwords but has a minor flaw due to its deterministic nature.

To mitigate the above issue we could **Salt** the password, a **Salt** is a fixed-length **secure random** string which is added to the **password** before hashing and hence the hash will be different for the same password.

Hashing is a **cryptographic** function which converts any amount of data into a fixed length hash

MessageDigest md = MessageDigest.getInstance("SHA-256");

Salt should be generated using a **Cryptographically Secure Pseudo-Random Number Generator** (**CSPRNG**), for **Java** the **CSPRNG** is **java.security.SecureRandom**.

While Storing the password

- Generate a long random salt using **SecureRandom**
- Use the Hash function such as **SHA256** to hash both **Salt** and **Password** together
- Save both the **Salt** and the **Hash** separately in the database.

While Validating the password

- Retrieve the **Salt** and **Hash** from the database

- Use the same Hash function **(SHA256)** which is used while generating the hash
- Generate a **new Hash** with **new password** provided and the **Salt** retrieved from the database.
- Now compare the **new hash** with the **hash** from the database. If they match, then password provided is correct. Otherwise, the password is incorrect

2. **Repeat Previous Order option for a customer (when his name or email is entered)**
   It can be implemented using MySQL in a table maintaining previous orders of all customers
   Problems :
   Many entries for a particular bill id and customer … If there are many many items from different categories that he ordered previously, we will have to traverse linearly to get each item from the table
   Less organised way (No proper structure) and more costly to the system
   Also creating each table for each customer is not feasible

In general, SQL databases can scale vertically, meaning you can increase the load on a server by migrating to a larger server that adds more CPU, RAM or SSD capability. While vertical scalability is used most frequently, SQL databases can also scale horizontally through sharding or partitioning logic, although that's not well-supported.

Horizontal scaling means scaling by adding more machines to your pool of resources (also described as "scaling out"), whereas vertical scaling refers to scaling by adding more power (e.g. CPU, RAM) to an existing machine (also described as "scaling up")

NoSQL

NoSQL systems allow you to work with different data structures within a database

NoSQL databases are not relational, so they don’t solely store data in rows and tables. Instead, they generally fall into one of four types of structures:

- **Column-oriented,** where data is stored in cells grouped in a virtually unlimited number of columns rather than rows.
- **Key-value stores**, which use an associative array (also known as a dictionary or map) as their data model. This model represents data as a collection of key-value pairs.
- **Document stores**, which use documents to hold and encode data in standard formats, including XML, YAML, JSON (JavaScript Object Notation) and BSON. A benefit is that documents within a single database can have different data types.
- **Graph databases**, which represent data on a graph that shows how different sets of data relate to each other. Neo4j, RedisGraph (a graph module built into Redis) and OrientDB are examples of graph databases.

Index	SQL	NoSQL
1)	Databases are categorized as Relational Database Management System (RDBMS).	NoSQL databases are categorized as Non-relational or distributed database system.
2)	SQL databases have fixed or static or predefined schema.	NoSQL databases have dynamic schema.
3)	SQL databases display data in form of tables so it is known as table-based database.	NoSQL databases display data as collection of key-value pair, documents, graph databases or wide-column stores.

4)	SQL databases are vertically scalable.	NoSQL databases are horizontally scalable.
5)	SQL databases use a powerful language "Structured Query Language" to define and manipulate the data.	In NoSQL databases, collection of documents are used to query the data. It is also called unstructured query language. It varies from database to database.
6)	SQL databases are best suited for complex queries.	NoSQL databases are not so good for complex queries because these are not as powerful as SQL queries.
7)	SQL databases are not best suited for hierarchical data storage.	NoSQL databases are best suited for hierarchical data storage.
8)	MySQL, Oracle, Sqlite, PostgreSQL and MS-SQL etc. are the example of SQL database.	MongoDB, BigTable, Redis, RavenDB, Cassandra, Hbase, Neo4j, CouchDB etc. are the example of nosql database

Challenges Faced ?

The first and foremost challenge was to choose a DBMS. When I searched on the web I found varying answers where some preferring the good old sql while some siding with the more modern and scalable nosql. As I had to complete the said project within a short span and I was advised by my Mentor that MySQL was relatively easy to learn , I went for it… I went through some docs and stackoverflow Q&A to learn how to connect the database with my project . I learnt some basic sql , went through some data structures in java like Array List (akin to vector in c++) etc to implement features like getAllRecords (Bill/Category/Product etc) order in BillDao

The next challenge I faced was to get the Bill Receipts in pdf format, because getting the bill receipts for the customer is a very important feature for any POS . I came to know about the open sourced iText library in Java that contains classes to generate PDF text in various fonts, create tables in PDF documents, add watermarks to pages, and so on

I learnt some basics of them to implement a simple Bill receipt though it has tons of more features that can be used to improve over the existing code like Password Protection / adding images & logos / changing fonts etc.

Also I wanted that as soon as the bill is generated it automatically opens in the default pdf reader for viewing instead of just being saved in a folder

Again stackoverflow came to the rescue and I got a simple code using File Protocol Handler that I implemented in Openpdf .java file in my project

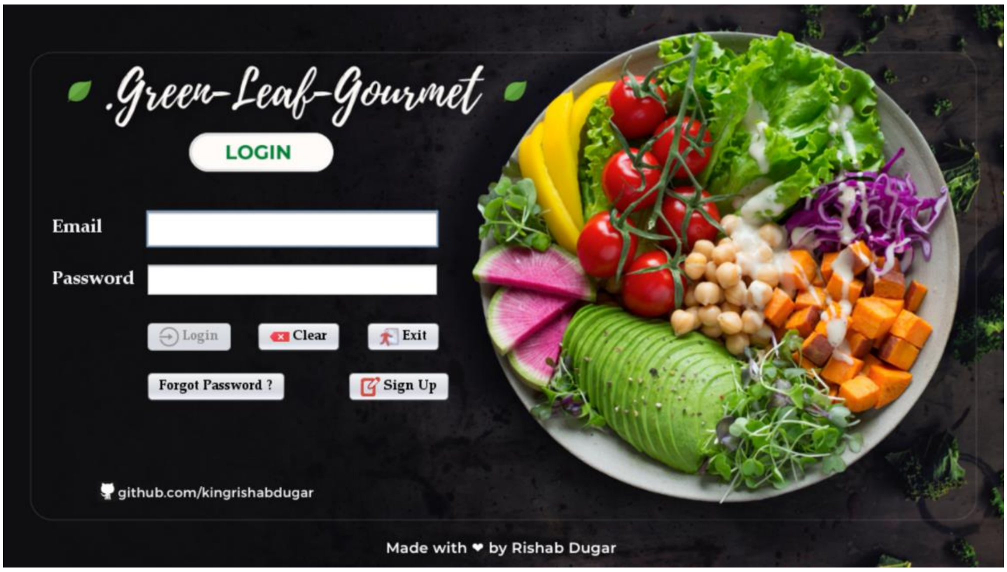
And then comes the small but irritating problems like while designing the JFrame the text boxes especially will automatically resize to different lengths as soon as I set the layout to absolute. However I found a hack, and as soon as I designed a JFrame I closed it while in the Free Layout , again opened it & set to absolute and that worked just fine. I don't know if this bug is of NetBeans or my system but this trick worked like a charm

Overview ⚜

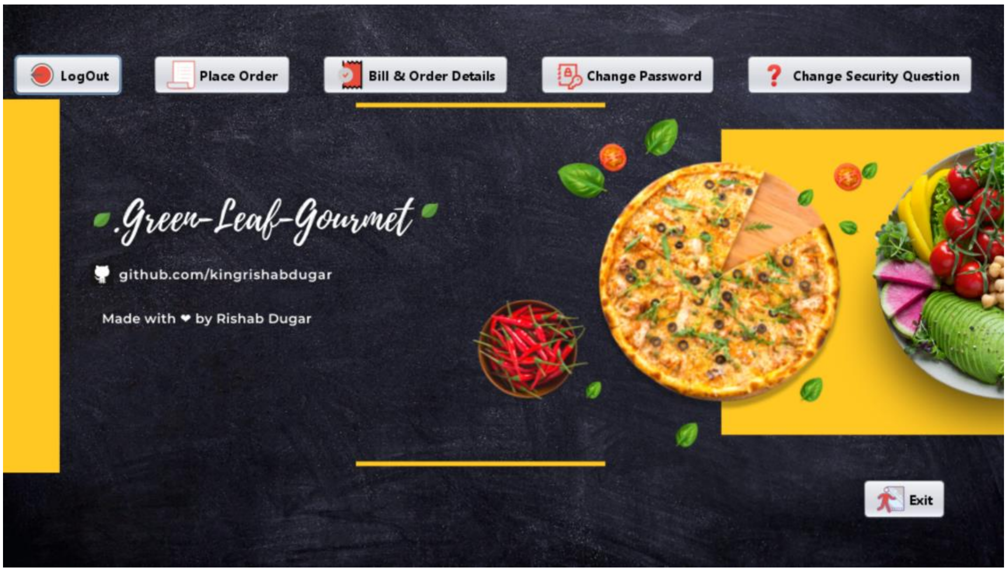
Demo | Working Example of the Application 📲

(P.S. If the GIF doesn't load the Demo / Working example can be accessed here : https://bit.ly/3PWPqra)

Sign Up & Login



Home Page | Admin & User Interface View



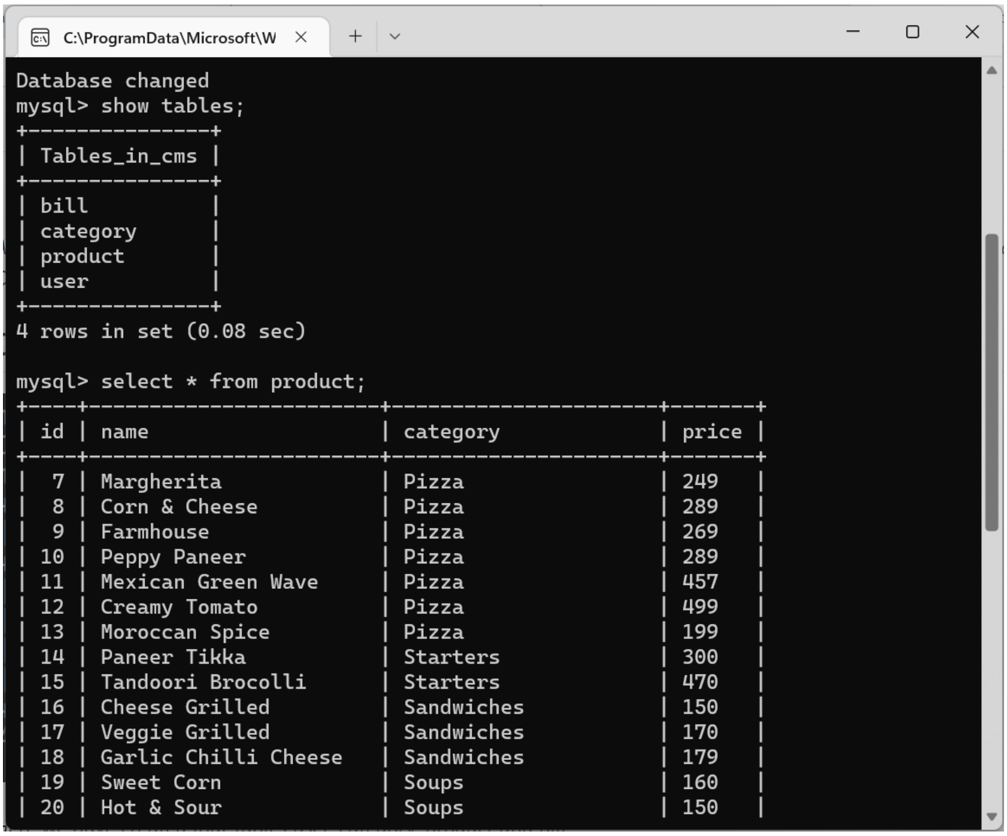
A user can Signup / Login (after Admin approval) and has access to the following features :

- Place an Order
- Generate Bill and receipt in PDF
- Change His/Her Password & Security Question
- Access the past bill receipts generated by all users with the feature to filter by date, time & email of the staff on duty

An admin can access the following features in addition to the features enjoyed by general users:

- Verify & Approve New Users
- Manage & Add/Delete Categories ,
- Add New Product ,
- View, Edit and Delete any Product

Database



C:\ProgramData\Microsoft\W

```
Database changed
mysql> show tables;
+---------------+
| Tables_in_cms |
+---------------+
| bill          |
| category      |
| product       |
| user          |
+---------------+
4 rows in set (0.08 sec)

mysql> select * from product;
+----+----------------------+----------------------+-------+
| id | name                 | category             | price |
+----+----------------------+----------------------+-------+
|  7 | Margherita           | Pizza                | 249   |
|  8 | Corn & Cheese        | Pizza                | 289   |
|  9 | Farmhouse            | Pizza                | 269   |
| 10 | Peppy Paneer         | Pizza                | 289   |
| 11 | Mexican Green Wave   | Pizza                | 457   |
| 12 | Creamy Tomato        | Pizza                | 499   |
| 13 | Moroccan Spice       | Pizza                | 199   |
| 14 | Paneer Tikka         | Starters             | 300   |
| 15 | Tandoori Brocolli    | Starters             | 470   |
| 16 | Cheese Grilled       | Sandwiches           | 150   |
| 17 | Veggie Grilled       | Sandwiches           | 170   |
| 18 | Garlic Chilli Cheese | Sandwiches           | 179   |
| 19 | Sweet Corn           | Soups                | 160   |
| 20 | Hot & Sour           | Soups                | 150   |
```

To create a new MySQL database via MySQL Command Line Client:

- Check if MySQL is Running . On Windows : Win + R -> services.msc -> MySQL -> Right-Click -> Start Service
- Run the client
- Enter your password
- Execute the create database command.

- Remember to update your Password On Line no. 19 in dao -> ConnectionProvider.java besides root for the application to RUN and connect with your database locally.