

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**  
BELAGAVI, KARNATAKA- 590018



PROJECT REPORT  
ON  
**COLD STORAGE MANAGEMENT SYSTEM**

Submitted in partial fulfillment of the requirements as a part of DBMS Lab for the V Semester of degree of **Bachelor of Engineering in Computer Science and Engineering** of Visvesvaraya Technological University, Belagavi

Submitted by

**KHUSHI RANI**  
**1JS20CS079**

**M VARSHA**  
**1JS20CS086**

**RAJ HARSH**  
**1JS20CS124**

Under the Guidance of

**Dr. Rohitaksha K,**  
Assistant Professor, Dept of CSE  
JSSATE, Bengaluru

**Department of Computer Science and Engineering**



**JSS ACADEMY OF TECHNICAL EDUCATION**  
Dr. Vishnuvardhan Rd, Srinivaspura Post  
Bengaluru-560060

JSS MAHAVIDHYAPEETHA, MYSURU

JSS ACADEMY OF TECHNICAL EDUCATION, BENGALURU  
JSS CAMPUS ROAD, SRINIVASPURA, KENGERI - 560060



## DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

### CERTIFICATE

This is to certify that the project work entitled "COLD STORAGE MANAGEMENT SYSTEM" is a bonafide work carried out by Khushi Rani (1JS20CS079), M Varsha (1JS20CS086) and Raj Harsh (1JS20CS124) in partial fulfillment of the requirements for DBMS Laboratory with Mini Project (18CSL58) of 5th Semester Bachelor of Engineering in Computer Science and Engineering of the Visvesvaraya Technological University, Belgavi during the academic year 2022-2023. It is certified that all corrections and suggestions indicated for Internal Assessment have been incorporated in the report deposited in the department Library. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said degree.

**MR. ROHITHAKSHA K.**

Professor, Dept of CSE  
JSSATE, Bengaluru

**DR. P.B. MALLIKARJUNA**

Professor and Head, Dept of CSE  
JSSATE, Bengaluru

**Name of Examiner:**

1. ....

**Signature:**

.....

2. ....

.....

## **Abstract**

The World of data is constantly changing and evolving every second. This in turn has created a completely new dimension of growth and challenges for companies around the globe. By accurately recording data, storing, updating and tracking them on an efficient and regular basis, companies can address their challenges on one hand and make use of the immense potentials offered by this sector on the other hand. A database management system stores, organizes and manages a large amount of information within a single software application. The use of this system increases efficiency of business operations and reduce overall costs.

Cold storage stores are facilities that are specifically designed to maintain a low temperature in order to preserve perishable goods such as food, medicine, and flowers. These facilities play an important role in the food industry by extending the shelf life of products and reducing food waste. The use of cold storage technology allows products to be stored for longer periods of time while maintaining their freshness and quality. A typical cold storage facility includes a refrigeration system, which maintains the low temperature, and insulation to prevent heat gain.

The CSMS will also include a security module that will ensure that only authorized personnel have access to the system and the data stored within it. This will include the implementation of user authentication and access controls to ensure that only authorized personnel can access and modify the data. The system will also include an audit trail feature that will keep track of all the changes made to the data and who made them, in order to maintain transparency and accountability.

The CSMS will be designed to be scalable and adaptable, to cater to the needs of different types of cold storage facilities. The system can be easily customized to suit the specific requirements of the facility, including the type of products stored, the temperature and humidity range, and the capacity of the facility. This will enable the system to be used in various types of cold storage facilities such as refrigerated warehouses, frozen warehouses, and blast freezers. Additionally, the system can be integrated with other systems in the facility, such as automated inventory systems or temperature monitoring devices, to provide a comprehensive view of the facility's operations.

## **Acknowledgement**

We express our humble pranams to His Holiness Jagadguru Sri Sri Shivaratri Deshikendra Mahaswamiji for showering his blessings onus to receive good education and have a successful career.

The completion of any project involves the efforts of many people. We have been lucky enough to have received a lot of support from all ends during the course of this project. So. we take the opportunity to express our gratitude to all whose guidance and encouragement helped us emerge successful.

We are thankful for the resourceful guidance, timely assistance and graceful gesture of our guide Mr. Rohitaksha K., Assistant Professor and Mrs Snehalata N, Assistant Professor, Dept. of Computer Science and Engineering who has helped us in every aspect of our project work. We express our sincere thanks to our beloved principal Dr. Bhimasen Soragaon for having supported us in all academic endeavors,

Lastly, we would be immensely pleased to express our heartfelt thanks to all the teaching and non-teaching staff of the Department of Computer Science and Engineering and our friends for their timely help, support and guidance.

KHUSHI RANI  
M VARSHA  
RAJ HARSH

# **Contents**

## **Chapter 1: Preamble**

- 1.1. Introduction
  - 1.1.1. Python Programming
  - 1.1.2. VS Code
  - 1.1.3. MySQL
  - 1.1.4. Normalization
- 1.2. Objectives
- 1.3 Organization of Report
- 1.4 Summary

## **Chapter 2: Requirement Specification**

- 2.1 Hardware Specification
- 2.2 Software Specification
- 2.3 User Characteristics

## **Chapter 3: System design and Implementation**

- 3.1 Introduction
- 3.2 Relational Schema
- 3.3 ER Diagram
- 3.4 Queries
  - 3.4.1 Creating Table
  - 3.4.2 Inserting Values to table
  - 3.4.3 Deleting Values from table
  - 3.4.4 Stored Procedure
  - 3.4.5 Triggers
  - 3.4.6 Assertions
- 3.5 Pseudo Codes
  - 3.5.1 Algorithm for Login
  - 3.5.1 Algorithm for Table Display
  - 3.5.1 Algorithm for Insert
  - 3.5.1 Algorithm for Update
  - 3.5.1 Algorithm for Delete

## **Chapter 4: Results and Discussions**

- 4.1 Database
- 4.2 Python Programming using VS Code
- 4.3 Index page
- 4.4 About us
- 4.5 Booking Page
- 4.6 Storage Selection Page
- 4.7 Admin Login Page

- 4.8 Admin Section Page
- 4.9 Bookings
- 4.10 Triggers
  - 4.10.1 Triggers in the Database
  - 4.10.2 Database before deleting
  - 4.10.3 Deleting from bookings
  - 4.10.4 Trigger for updating user table
  - 4.10.5 User table after updating
  - 4.10.6 Changes in the database after deleting
- 4.11 Storage details
- 4.12 Stored procedures
  - 4.12.1 Storage procedure(i) execution
  - 4.12.2 Storage procedure(i) result
  - 4.12.3 Storage procedure(ii) execution
  - 4.12.4 Storage procedure(ii) result
- 4.13 Admin assertion
- 4.14 User Login page
- 4.15 User Section
- 4.15 Contact page
- 4.16 Reviews by the customers
- 4.17 Reviews by the customers stored in message list
- 4.18 Team members

## **Conclusion and Future Enhancements**

# List of Figures and Tables

## Chapter 3: System design and Implementation

### Figures:

- ER Diagram
- Relational Schema

### Tables:

- booking\_details
- booking\_list
- message\_list
- admin
- user
- storage\_list

## Chapter 4: Results and Discussions

### Figures:

- Database
- Python Programming using VS Code
- Index page
- Index page
- About us
- Booking Page
- Storage Selection Page
- Admin Login Page
- Admin Section Page
- Booking Page Initially
- Database after booking
- Triggers in the Database
- Database before deleting
- Deleting from bookings
- Trigger for updating user table
- User table after updating
- Changes in the database after deleting
- Storage details
- Storage procedures
- Stored procedure(i) execution
- Stored procedure(i) result
- Stored procedure(ii) execution
- Stored procedure(ii) result
- Admin assertion
- User Login page
- User Section
- Contact page
- Reviews by the customers
- Reviews by the customers stored in message list
- Team members

# Chapter 1: Preamble

## 1.1 Introduction

A database is an organized collection of data. A relational database, more restrictively, is a collection of schemas, tables, queries, reports, views, and other elements. A database management system (DBMS) is a computer-software application that interacts with end-users, other applications, and the database itself to capture and analyze data. A general-purpose DBMS allows the definition, creation, querying, update, and administration of databases. There is a need for an application to make it easy for industries and trading companies to maintain their records and have a track of goods. Supply chain management streamlines everything from product flow to unexpected natural disasters. With an effective COLD STORAGE MANAGEMENT, organizations can diagnose problems and disruptions correctly. COLD STORAGE MANAGEMENT plays an important role in moving items quickly and efficiently to destination.

## Database Management System (DBMS)

Following the technology progress in the areas of processors, computer memory, computer storage, and computer networks, the sizes, capabilities, and performance of databases and their respective DBMSs have grown in orders of magnitude. The development of database technology can be divided into three eras based on data model or structure: navigational, SQL/relational, and post-relational. The two main early navigational data models were the hierarchical model, epitomized by IBM's IMS system, and the CODASYL model (network model), implemented in a number of products such as IDMS. The relational model employs sets of ledger-style tables, each used for a different type of entity. Only in the mid-1980s did computing hardware become powerful enough to allow the wide deployment of relational systems (DBMSs plus applications). By the early 1990s, however, relational systems dominated in all large-scale data processing applications, and as of 2015 they remain dominant: IBM DB2, Oracle, MySQL, and Microsoft SQL Server are the top DBMS. The dominant database language, standardized SQL for the relational model, has influenced database languages for other data models.

### 1.1.1 Python Programming

Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library. Guido van Rossum began working on Python in the late 1980s as a successor to the ABC programming language and first released it in 1991 as Python 0.9.0. Python 2.0 was released in 2000 and introduced new features such as list comprehensions, cycle-detecting garbage collection, reference counting, and Unicode support. Python 3.0, released in 2008, was a major revision not completely backward-compatible with earlier versions. Python 2.7.18, released in 2020, was the last release of Python 2. Python consistently ranks as one of the most popular programming languages.

### 1.1.2 VS Code

**Visual Studio Code**, also commonly referred to as **VS Code**, is a source-code editor made by Microsoft with the Electron Framework, for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality. In the Stack Overflow 2021 Developer Survey, Visual Studio Code was ranked the most popular developer environment tool among 82,000 respondents, with 70% reporting that they use it.

Visual Studio Code is a source-code editor that can be used with a variety of programming languages, including C#, Java, JavaScript, Go, Node.js, Python, C++, C, Rust and Fortran. It is based on the Electron framework, which is used to develop Node.js web applications that run on the Blink layout engine. Visual Studio Code employs the same editor component (codenamed "Monaco") used in Azure DevOps. Visual

Studio Code can be extended via extensions, available through a central repository. This includes additions to the editor and language support.

### 1.1.3 Database - MySQL

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons. MySQL is released under an open-source license. So, we have nothing to pay to use it. MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages. MySQL uses a standard form of the well-known SQL data language.

MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc. It works very quickly and works well even with large data sets. MySQL is very friendly to PHP, the most appreciated language for web development. MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB) and is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

### 1.1.4 Normalization

Normalization is a process of organizing the data in database to avoid data redundancy, insertion anomaly, update anomaly & deletion anomaly. To overcome these anomalies, we need to normalize the data. There are 4 basic types of normalizations. They are:

- First normal form(1NF)
- Second normal form(2NF)
- Third normal form(3NF)
- Boyce & Codd normal form (BCNF)

First normal form (1NF) is defined as per rule as: an attribute (column) of a table cannot hold multiple values. It should hold only atomic values. This means that there shouldn't be repetition of data in the tables.

A table is said to be in 2NF if the two conditions stated are satisfied. The table is in First normal form and all the non-prime attribute are dependent on the proper subset of any candidate key of table. The attribute that is not part of any candidate key are known as non-prime attribute.

A table design is said to be in 3NF if the table is in 2NF and [Transitive functional dependency](#) of non-prime attribute on any super key are removed.

Boyce Codd normal form (BCNF) is the advance version of 3NF that's why it is also referred as 3.5NF. BCNF is stricter than 3NF. A table complies with BCNF if it is in 3NF and for every functional dependency  $X \rightarrow Y$ , X should be the super key of the table.

## 1.2 Objectives

Objectives of the **COLD STORAGE MANAGEMENT SYSTEM** are:

- Prolonged shelf-life of fruits and vegetables
- Temperature Control
- Cost-effective
- Transportation of products
- Frees up space
- Storage for Chemicals

## **1.3 Organization of Report**

**Chapter 1** provides the information about the basics of Java Swings and MySQL. In **Chapter 2**, we discuss the software and hardware requirements to run the above applications. **Chapter 3** gives the idea of the project and its actual implementation. **Chapter 4** discusses about the results and discussions of the program. **Chapter 5** concludes by giving the direction for future enhancement.

## **1.4 Summary**

The chapter discussed before is an overview about the **Python Programming, VS Code** and **MySQL Database DBMS**. The scope of study and objectives of the project are mentioned clearly. The organization of the report is been pictured to increase the readability. Further, coming up chapters depicts the use of various queries to implement various changes like insert, update, delete and also triggers to perform various functions.

## **Chapter 2 : Requirement Specifications**

### **2.1 SOFTWARE SPECIFICATION**

- **Operating System:** Windows 11 and Linux Mint
- **Front End:** HTML CSS
- **Rear End:** MySQL and Python Programming

### **2.2 HARDWARE SPECIFICATION**

- Processor: x86 compatible processor with 1.7 GHz Clock Speed
- RAM: 512 MB or greater
- Hard Disk: 20 GB or greater
- Monitor: VGA/SVGA
- Keyboard: 104 keys standard
- Mouse: 2/3 button. Optical/Mechanical.

### **2.3 USER CHARACTERISTICS**

Every user:

- Should be comfortable with basic working of the computer
- Must have basic knowledge of English
- Must carry a login ID and password used for authentication

# **Chapter 3: System Design and Implementation**

## **3.1 Introduction**

Systems design is the process or art of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. One could see it as the application of systems theory to product development. This Project is implemented using Python Programming and VS Code, which is proven to be a very efficient tool in the field of Java programming. It is done under Windows11 and Linux Mint platform. Library modules are used to create the objects and to translate them. Python programming language is used to implement the entire code. Interface to the program is provided with the help of MySQL Database.

## **3.2 Relational Schema**

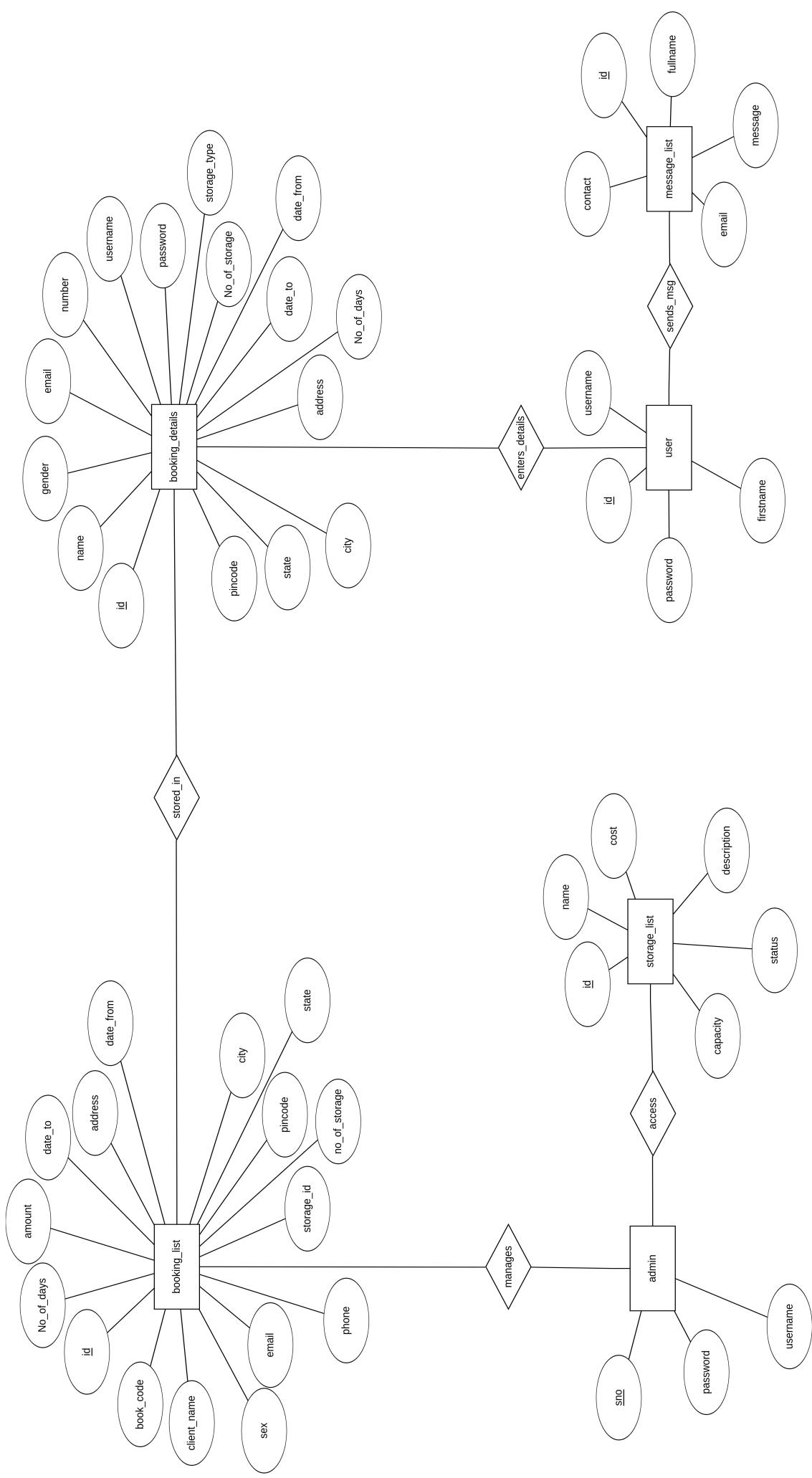
The schema diagram of a database system is its structure described in a formal language supported by the database management system (DBMS). The formal definition of a database schema is a set of formulas called integrity constraints imposed on a database. The term "schema" refers to the organization of data as a blueprint of how the database is constructed. These integrity constraints ensure compatibility between parts of the schema. All constraints are expressible in the same language. A database can be considered a structure in realization of the database language. The states of a created conceptual schema are transformed into an explicit mapping, the database schema. This describes how real-world entities are modeled in the database. A database schema specifies, based on the database administrator's knowledge of possible applications, the facts that can enter the database, or those of interest to the possible end-users.<sup>[2]</sup> The notion of a database schema plays the same role as the notion of theory in predicate calculus. A model of this "theory" closely corresponds to a database, which can be seen at any instant of time as a mathematical object. Thus a schema can contain formulas representing integrity constraints specifically for an application and the constraints specifically for a type of database, all expressed in the same database language.<sup>[1]</sup> In a relational database, the schema defines the tables, fields, relationships, views, indexes, packages, procedures, functions, queues, triggers, types, sequences, materialized views, synonyms, database links, directories, XML schemas, and other elements.

Figure 3.2 Schema is defined for an Inventory Management System. All the various table used are described in the following schema. The necessary Primary key's and the corresponding Foreign keys are also represented.

## **3.3 ER Diagram**

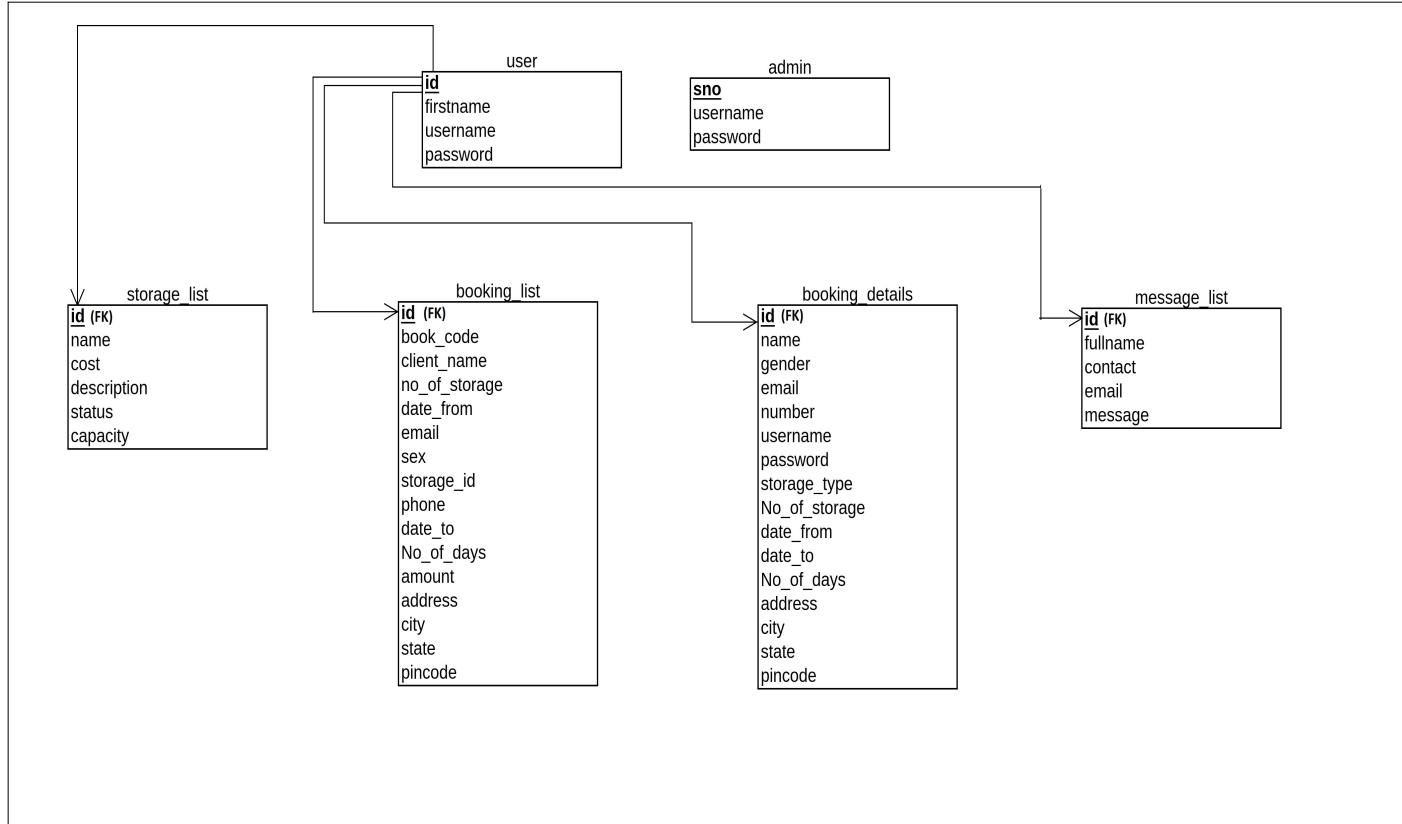
An entity–relationship model or the ER Diagram describes inter-related things of interest in a specific domain of knowledge. An ER model is composed of entity types and specifies relationships that can exist between instances of those entity types. In software engineering, an ER model is commonly formed to represent things a business needs to remember in order to perform business processes. Consequently, the ER model becomes an abstract data model, that defines a data or information structure which can be implemented in a database, typically a relational database. (**Figure 3.2** represents ER Diagram)

Entity–relationship modelling was developed for database and design by Peter Chen and published in a 1976 paper,<sup>[1]</sup> with variants of the idea existing previously, but today it is commonly used for teaching students the basics of data base structure.<sup>[2]</sup> Some ER models show super and subtype entities connected by generalization-specialization relationships,<sup>[3]</sup> and an ER model can be used also in the specification of domain-specific ontologies. An ER model is typically implemented as a database. In a simple relational database implementation, each row of a table represents one instance of an entity type, and each field in a table represents an attribute type.



**Fig: ER Diagram**

## **Relational Schema:**



**Figure 3.3 Schema Diagram**

## **3.4 Queries**

The below mentioned are all the queries used to perform various tasks in MySQL such as insert, delete, update. A short description of the query is also provided.

### **3.4.1 Creating Tables**

#### **1. booking\_details:**

```
CREATE TABLE `booking_details`  
(  
    `id` int(30) NOT NULL,  
    `name` varchar(50) NOT NULL,  
    `gender` text NOT NULL,  
    `email` varchar(50) NOT NULL,  
    `number` varchar(50) NOT NULL,  
    `username` varchar(50) NOT NULL,  
    `password` varchar(15) NOT NULL,  
    `storage_type` int(5) NOT NULL,  
    `No_of_storages` int(11) NOT NULL,  
    `date_from` date NOT NULL,  
    `date_to` date NOT NULL,
```

```

`No_of_days` int(5) NOT NULL,
`address` varchar(50) NOT NULL,
`city` text NOT NULL,
`state` text NOT NULL,
`pincode` int(10) NOT NULL
);

```

Field	Type	Null	Key	Default	Extra
id	int	not NULL	Primary Key	None	
name	varchar	not NULL		None	
gender	text	not NULL		None	
email	varchar	not NULL		None	
number	varchar	not NULL		None	
username	varchar	not NULL		None	
password	varchar	not NULL		None	
storage_type	text	not NULL		None	
No_of_storage	text	not NULL		None	
date_from	date	not NULL		None	
date_to	date	not NULL		None	
No_of_days	int	not NULL		None	
address	varchar	not NULL		None	
city	text	not NULL		None	
state	text	not NULL		None	
pincode	int	not NULL		None	

Table 3.4.1 booking\_details

## 2. booking\_list:

```

CREATE TABLE `booking_list`
(
    `id` int(30) NOT NULL,
    `book_code` varchar(100) NOT NULL,
    `client_name` varchar(50) NOT NULL,
    `gender` text NOT NULL,
    `email` varchar(50) NOT NULL,
    `number` varchar(50) NOT NULL,
    `username` varchar(50) NOT NULL,
    `password` varchar(15) NOT NULL,
    `storage_type` int(5) NOT NULL,
    `No_of_storages` int(11) NOT NULL,
    `date_from` date NOT NULL,
    `date_to` date NOT NULL,
    `No_of_days` int(5) NOT NULL,
    `address` varchar(50) NOT NULL,
    `city` text NOT NULL,
)

```

```

    `state` text NOT NULL,
    `pincode` int(10) NOT NULL
);

```

Field	Type	Null	Key	Default	Extra
id	int	not NULL	Primary Key	None	
book_code	varchar	not NULL		None	
client_name	varchar	not NULL		None	
gender	text	not NULL		None	
email	varchar	not NULL		None	
number	varchar	not NULL		None	
username	varchar	not NULL		None	
password	varchar	not NULL		None	
storage_type	text	not NULL		None	
No_of_storage	text	not NULL		None	
date_from	date	not NULL		None	
date_to	date	not NULL		None	
No_of_days	int	not NULL		None	
address	varchar	not NULL		None	
city	text	not NULL		None	
state	text	not NULL		None	
pincode	int	not NULL		None	

**Table 3.4.2 booking\_list**

### 3. message\_list:

```

CREATE TABLE `message_list`
(
    `id` int(5) NOT NULL,
    `fullname` text NOT NULL,
    `contact` text NOT NULL,
    `email` text NOT NULL,
    `message` text NOT NULL
);

```

Field	Type	Null	Key	Default	Extra
id	int	not NULL	Primary Key	None	
fullname	text	not NULL		None	
contact	text	not NULL		None	
email	text	not NULL		None	
message	text	not NULL		None	

**Table 3.4.3 message\_list**

### 4. admin:

```

CREATE TABLE `admin`
(
    `sno` int(1) NOT NULL,
    `username` varchar(20) NOT NULL,

```

```

`password` varchar(10) NOT NULL
);

```

Field	Type	Null	Key	Default	Extra
sno	int	not NULL	Primary Key	None	
username	varchar	not NULL		None	
password	varchar	not NULL		None	

**Table 3.4.4 admin**

#### 5. users:

```

CREATE TABLE `users`
(
    `id` int(50) NOT NULL,
    `firstname` varchar(250) NOT NULL,
    `username` text NOT NULL,
    `password` text NOT NULL
);

```

Field	Type	Null	Key	Default	Extra
id	int	not NULL	Primary Key	None	
firstname	varchar	not NULL		None	
username	text	not NULL		None	
password	text	not NULL		None	

**Table 3.4.5 users**

#### 6. storage\_list:

```

CREATE TABLE `storage_list`
(
    `id` int(30) NOT NULL,
    `name` text NOT NULL,
    `cost` float NOT NULL DEFAULT 0,
    `description` text NOT NULL,
    `Capacity` int(5) NOT NULL,
    `status` text NOT NULL DEFAULT 'Available'
);

```

Field	Type	Null	Key	Default	Extra
id	int	not NULL	Primary Key	None	
name	text	not NULL		None	
cost	float	not NULL		None	
description	text	not NULL		None	
capacity	int	not NULL		None	
status	text	not NULL		None	

**Table 3.4.5 storage\_lists**

### **3.4.2 Inserting values into Tables**

#### **Actual data stored in the database**

1. INSERT INTO `admin` (`sno`, `username`, `password`) VALUES(1, 'Varsha\_7', 'varsha7007');
2. INSERT INTO `booking\_details` (`id`, `name`, `gender`, `email`, `number`, `username`, `password`, `storage\_type`, `No\_of\_storages`, `date\_from`, `date\_to`, `No\_of\_days`, `address`, `city`, `state`, `pincode`) VALUES(2, 'Varsha', 'Female', 'varshacs086@gmail.com', '6299188694', 'varsha\_8', '1js20cs086', 1, 2, '2023-01-18', '2023-01-24', 0, 'JSSATE Girls Hostel', 'Bangalore', 'Karnataka', 560060);
3. INSERT INTO `booking\_list` (`id`, `book\_code`, `client\_name`, `sex`, `email`, `Phone`, `storage\_id`, `No\_of\_storages`, `date\_from`, `date\_to`, `No\_of\_days`, `amount`, `address`, `city`, `state`, `pincode`) VALUES(1, 'KVR-0000000001', 'M Kartik Kumar', 'Male', 'kartiksachu@gmail.com', '9876546752', 5, 2, '2021-12-17', '2021-12-31', 14, 14000, 'Jayanagar', 'Bangalore', 'Karnataka', 560090);
4. INSERT INTO `message\_list` (`id`, `fullname`, `contact`, `email`, `message`) VALUES(1, 'Khushi Rani', '09123456789', 'khushi@gmail.com', '--');
5. INSERT INTO `storage\_list` (`id`, `name`, `cost`, `description`, `Capacity`, `status`) VALUES(1, 'Storage 1', 250, 'KVR refrigerated containers come in a variety of sizes from 10ft to 45ft and have a temperature range of -40°C to +10°C. Most cold store units are portable and can be used to transport goods safely. Mega Cold Stores are also a popular option for larger businesses. These units are combined with multiple cold storage units to make a mega refrigerated solution.\r\nThe air conditioning cold room is used for long-term storage of fresh fruits, vegetables, etc. In addition to the control of temperature and moisture in the cold room, we need to consider the respiration of plants in the cold room at the same time, but also regulate and control O<sub>2</sub>, CO, N<sub>2</sub>, and ethylene, inhibits respiration and metabolism of fruits and vegetables in a dormant state, in order to achieve the purpose of long-term storage.\r\n\r\n', 5, 'Available');
6. INSERT INTO `users` (`id`, `firstname`, `username`, `password`) VALUES(1, 'Khushi', 'khushi\_rani', '4744ddea876b11dcb1d169fadf494418');

### **3.4.3 Query to demonstrate Deleting the value in the table:**

```
@app.route("/delete/<string:id>", methods=['GET','POST'])
def delete(id):
    booking_list=Booking_list.query.filter_by(id=id).first()
    db.session.delete(booking_list)
    db.session.commit()
    return "Successfully Deleted"
```

**Description:** This query is used to delete a row in booking\_details table when the name entered in text field matches in the booking\_details table of database.

### 3.4.4 Stored Procedures

A stored procedure is nothing more than prepared SQL code that the developer saves so the application can reuse the code over and over again. So, if the developer thinks about a query that you write over and over again, instead of having to write that query each time you would save it as a stored procedure and then just call the stored procedure to execute the SQL code that you saved as part of the stored procedure.

#### Procedure:

##### 1. getstorage:

```
DROP PROCEDURE `getstorage`; CREATE DEFINER='root'@'localhost'  
PROCEDURE `getstorage`(IN `sid` INT) NOT DETERMINISTIC CONTAINS SQL  
SQL SECURITY DEFINER select * from storage_list WHERE id=sid
```

##### 2. getuserdetails:

```
DROP PROCEDURE `getuserdetails`; CREATE DEFINER='root'@'localhost'  
PROCEDURE `getuserdetails`(IN `bcode` VARCHAR(100)) NOT  
DETERMINISTIC CONTAINS SQL SQL SECURITY DEFINER SELECT * from  
booking_list WHERE book_code=bcode
```

**Description:** Here whenever the Stored Procedure is called the storage details and user details table will be displayed accordingly

### 3.4.5 Triggers

Triggers are stored programs, which are automatically executed or fired when some events occur. Triggers are stored into database and invoked repeatedly, when specific condition match.

#### Trigger:

##### 1. get\_bookcode :

```
DROP TRIGGER IF EXISTS `get_bookcode`;  
CREATE DEFINER='root'@'localhost` TRIGGER `get_bookcode` BEFORE  
INSERT ON `booking_list` FOR EACH ROW BEGIN SET  
NEW.book_code=CONCAT ("KVR-",LPAD(LAST_INSERT_ID(),10 , "0"));  
END
```

##### 2. insert\_user :

```
DROP TRIGGER IF EXISTS `insert_user`;  
CREATE DEFINER='root'@'localhost` TRIGGER `insert_user` AFTER INSERT  
ON `booking_details` FOR EACH ROW INSERT INTO users VALUES (null,NEW  
.name,NEW.username,NEW.password)
```

##### 3. booking\_list :

```
DROP TRIGGER IF EXISTS `insert_booking_list`;  
CREATE DEFINER='root'@'localhost` TRIGGER `insert_booking_list` AFTER  
INSERT ON `booking_details` FOR EACH ROW INSERT INTO booking_list  
VALUES(null,null,NEW.name,NEW.gender,NEW.email,NEW.number,  
NEW.storage_type,NEW.No_of_storages,NEW.date_from,NEW.date_to,NEW.  
No_of_days,NEW.No_of_storages*NEW.No_of_days*500,NEW.address,  
NEW.city,NEW.state,NEW.pincode)
```

**Description:** This trigger is triggered when the insertion or deletion is done on the booking\_details table and the rows in booking\_lists are modified.

### 3.5 Pseudo Code

Pseudocode is a simple, informal language used to describe the logic of a computer program or algorithm. It is not a formal programming language, but rather a way to express algorithms and ideas in a readable and understandable form. The syntax of pseudocode is designed to be similar to that of a programming language, but it is not bound by the same strict rules and conventions.

Pseudocode is typically used to express the logic of a program or algorithm before it is implemented in a specific programming language. It is used to plan and design the logic of a program, and to communicate the logic to other programmers. Pseudocode is also used to teach programming concepts, as it is easier to understand than a specific programming language.

#### 3.5.1 Algorithm for Login

**Step 1:** BEGIN  
**Step 2:** Enter username and password  
**Step 3:** Verify the credentials entered with that in the login table  
**Step 4:** If Credentials match, then proceed to the Admin page  
    Else show login failed  
**Step 5:** End if  
**Step 6:** END

#### 3.5.2 Algorithm for Table Display

**Step 1:** BEGIN  
**Step 2:** Establish connection with the database using the username and password of the database.  
**Step 3:** Define the select query to retrieve all the values from the DBMS  
**Step 4:** Define DefaultTableModel for the table and fetch the details from the database.  
**Step 5:** END

#### 3.5.3 Algorithm for Insert

**Step 1:** BEGIN  
**Step 2:** Get all the necessary values required for insertion into variable defined in the method.  
**Step 3:** Define the query for insertion as stated above.  
**Step 4:** Execute the Query using the (Select \* from) the required table.  
**Step 5:** END

#### 3.5.4 Algorithm for Update

**Step 1:** BEGIN  
**Step 2:** Get all the necessary values required for updating the values into the variable defined in the method.  
**Step 3:** UPDATE table name  
    SET column1 = value1, column2 = value2, ...  
    WHERE condition;  
**Step 4:** Define the Query for Updating as stated above.  
**Step 5:** Execute the Query using the executeUpdate() method defined.  
**Step 6:** END

### **3.5.5 Algorithm for Delete**

- Step 1:** BEGIN
- Step 2:** Get the model number of the instrument which is to be deleted into variable defined in the method.
- Step 3:** Delete from table\_name where condition;
- Step 4:** Define the Query for deleting as stated above.
- Step 5:** Execute the Query using the executeUpdate() method defined .
- Step 6:** END

# Chapter 4: Results and Discussions

The project is compiled and executed using Python programming and MySQL. We have put in few screen shots here to show the working of our Application.

## phpMyAdmin (XAMPP)

The screenshot shows the phpMyAdmin interface for the 'CSM' database. The top navigation bar includes tabs for Structure, SQL, Search, Query, Export, Import, Operations, Privileges, Routines, Events, Triggers, Tracking, Designer, and Central columns. A 'Filters' section contains a search input field. The main area displays a table of tables with the following data:

Table	Action	Rows	Type	Collation	Size	Overhead
admin	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	32.0 Kib	-
booking_details	Browse Structure Search Insert Empty Drop	9	InnoDB	utf8mb4_general_ci	32.0 Kib	-
booking_list	Browse Structure Search Insert Empty Drop	8	InnoDB	utf8mb4_general_ci	48.0 Kib	-
message_list	Browse Structure Search Insert Empty Drop	7	InnoDB	utf8mb4_general_ci	32.0 Kib	-
storage_list	Browse Structure Search Insert Empty Drop	6	InnoDB	utf8mb4_general_ci	16.0 Kib	-
users	Browse Structure Search Insert Empty Drop	8	InnoDB	utf8mb4_general_ci	32.0 Kib	-
<b>6 tables</b>	<b>Sum</b>	<b>39</b>	<b>InnoDB</b>	<b>utf8mb4_general_ci</b>	<b>192.0 Kib</b>	<b>0 B</b>

Below the table, there are buttons for 'Check all' and 'With selected:'. Further down, there are links for 'Print' and 'Data dictionary', and a 'Create new table' dialog box. The dialog box has fields for 'Table name' (empty) and 'Number of columns' (set to 4), with a 'Create' button.

Figure: Database

## VS Code

The screenshot shows the Visual Studio Code interface with the 'main.py' file open in the editor. The code is a Flask application with SQLAlchemy integration. It defines several database models: Message\_list, Storage\_list, Admin, User, and Booking\_details. The code uses annotations like db.Column and db.Integer to define the database schema. The 'PROBLEMS' and 'TERMINAL' tabs are visible at the bottom, showing command-line output related to the application's startup.

```
from contextlib import RedirectStream
from flask import Flask, render_template, request
from flask_sqlalchemy import SQLAlchemy

app = Flask(__name__)
app.config["SQLALCHEMY_DATABASE_URI"] = 'mysql+pymysql://root:@localhost/CSM'
db = SQLAlchemy(app)

class Message_list(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    fullname = db.Column(db.String(50), nullable=False)
    email = db.Column(db.String(20), nullable=False)
    contact=db.Column(db.String(12),nullable=False)
    message=db.Column(db.String,nullable=False)
    status=db.Column(db.Integer)
    date_created=db.Column(db.String,nullable=True)

class Storage_list(db.Model):
    id=db.Column(db.Integer, primary_key=True)
    name=db.Column(db.String(50), nullable=False)
    cost=db.Column(db.Float)
    description=db.Column(db.String)
    Capacity=db.Column(db.Integer)
    status=db.Column(db.String)

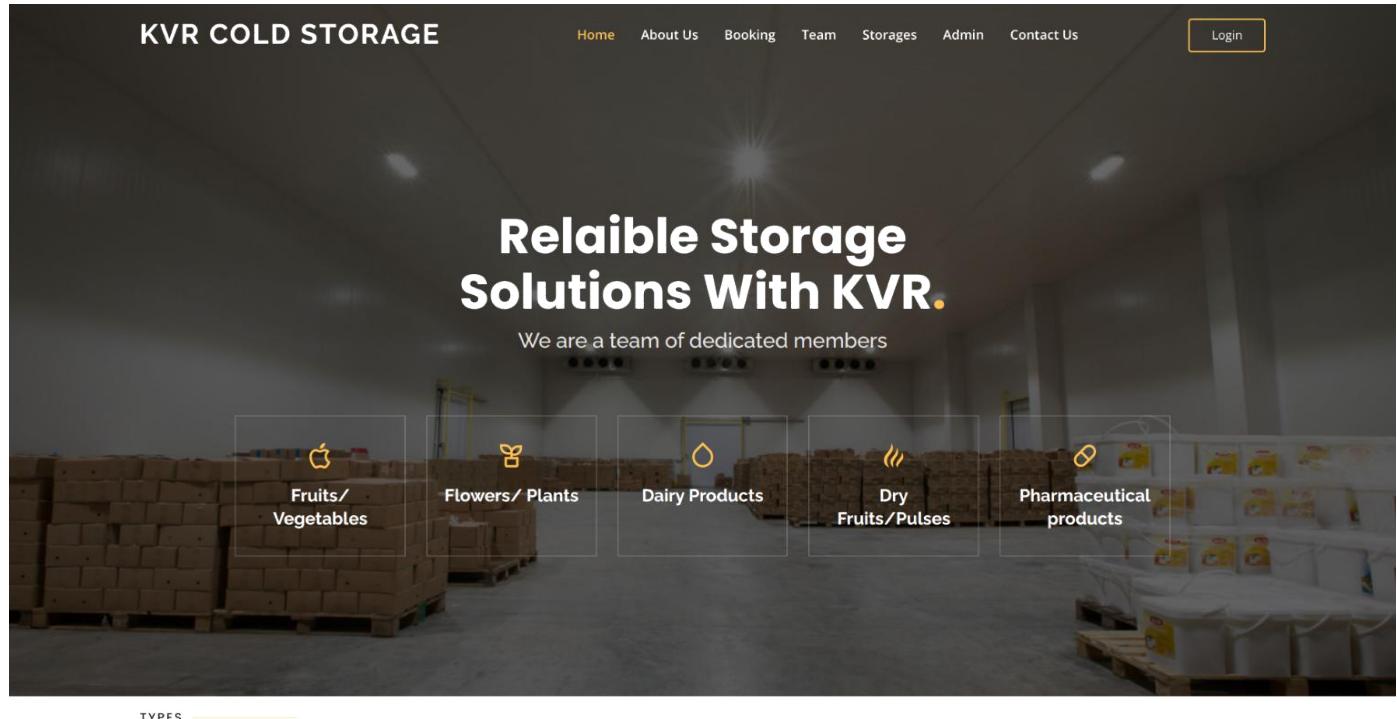
class Admin(db.Model):
    sno = db.Column(db.Integer, primary_key=True)
    username=db.Column(db.String(50), nullable=False)
    password=db.Column(db.String(50), nullable=False)

class Users(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    username=db.Column(db.String(50), nullable=False)
    password=db.Column(db.String(50), nullable=False)

class Booking_details(db.Model):
    PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
khushi@khushi-HP-Laptop-15s-grxxx:/media/khushi/New Volume/5th Sem/Mini Project/CSM$ /bin/python3 "/media/khushi/New Volume/5th Sem/Mini Project/CSM/main.py"
* Serving Flask app `main'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 113-313-146
Ln1 Col1 Spaces:4 UTF-8 LF Python 3.10.6 64-bit 78% 13:52
```

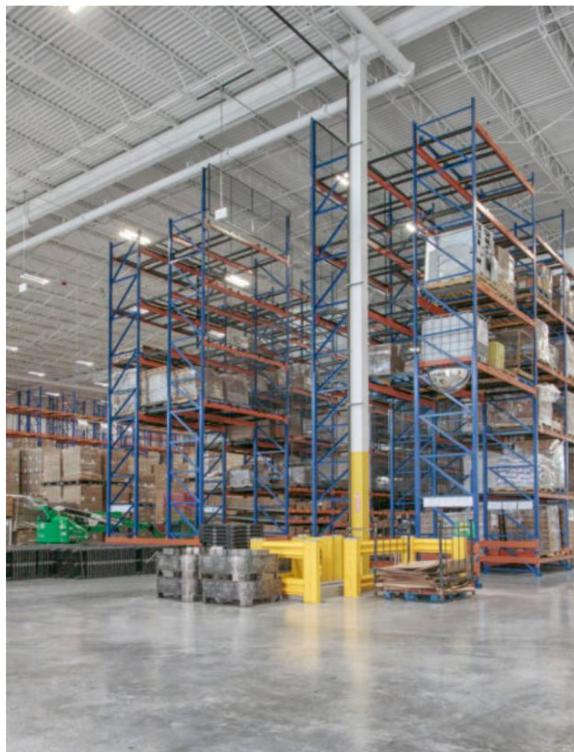
Figure: Python Programming using VS Code

# Index Page



TYPES

## TYPES OF STORAGES



### Refrigerated containers

It is the primary and efficient type of cold storage facility. They can be used for a small number of items that need optimum temperature storage. Further, workers can easily move them from one place to another.

### Blast Freezers and Chillers

These meet large storage requirements that need the items to be chilled when they reach the customer. High-end restaurants and similar companies use these.

### Cold rooms

These can be both refrigerants and blast freezers based on the requirement, except that they are much larger and in the form of entire rooms.

### Pharmaceutical grade cold storage warehouse

These are warehouses that have the cold storage technologies built-in with them. Used in hospitals and medical facilities, they are specially used to store blood, certain vaccines and biopharmaceuticals.

### Plant attached cold storage warehouse

When producers wish to keep the cold rooms within their factories, such facilities are built. Conveyor belts are used to send intermediary goods or finished products to cold storage. This makes it an on-site service.

### Dedicated custom cold storage warehouse facilities

It includes companies that have specific cold storage needs and need customized cold rooms. Instead of building entire cold rooms, they can get made one personally by 3PL logistic services or warehousing solution companies.

## Uses of Cold Storage

Cold storage is essentially meant for storing items that need specific low-temperature surroundings. There is a misconception that cold rooms can store only food items this way. However, cold rooms can store a variety of items

Tap to view more

**Figure:** Index page



## The Most Trusted Storage. Safe Storage Is What We Do.

We provide hassle-free cold storage solutions.



-->

### KVR COLD STORAGE.

Srinivaspura Road.JSSATE, Kengeri  
Bangalore 560060, India

Phone: +91 5589 5548 85  
Email: kvrstorage@gmail.com



#### Useful Links

- > Home
- > About us
- > Booking
- > Terms of service
- > Privacy policy
- > Storage for Fruits/Vegetables
- > Storage for Flowers/Plants
- > Storage for Dairy Products
- > Storage for Dry fruits/Pulses
- > Storage for Pharmaceutical products
- > Many more

#### Our Services

#### Our Newsletter

A Weekly eNewsletter showcases the latest news, trends, technologies etc impacting the stored products.

[Subscribe](#)

© Copyright KVR Cold Storage. All Rights Reserved



**Figure:** Index page

# About us

ABOUT

## ABOUT US

Welcome to KVR Cold Storage.



### KVR COLD STORAGE

*Cold storage is a facility that primarily stores items that are short-lived and highly likely to get spoilt under normal conditions. These may include fruits, vegetables, fish, meat etc. These food items are stored under optimum temperature (primarily low) and humid environment as required for individual items. Almost all cold storage rooms are designed such that these properties are pre-configured based on what is being stored. Some cold rooms are made such that these properties are adjustable.*

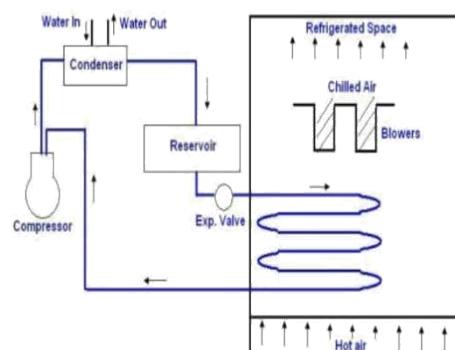


### Construction and Working of a Cold Storage Plant

A cold-storage facility works on a refrigeration system which helps maintain an adequate temperature and environment as per the specifications of each item being stored. These are the main components of a cold storage room:



- ❖ **Compressor** : It is the main component that runs the cold room. It is the only device that needs the energy to run. The compressor consumes almost all the power in a cold room. It is used to raise the temperature and pressure of the refrigerant vapour coming out of the Evaporator. As the pressure is increased, the boiling point increases and the compressor can condense the refrigerant (for example, ammonia) at the temperature of the condenser.
- ❖ **Condenser** : It is required to remove the heat from the refrigerant and the circulating water. It carries out phase change of the condenser from gas to liquid at high temperature and pressure. The condenser acts as a heat sink, and its heat exchange efficiency determines the efficiency of the cold storage plant.
- ❖ **Receiver** : The high-pressure liquid condensate is stored here. It is here that the refrigerant comes after phase change from the condenser. After it has reached the receiver component, the liquid refrigerant goes to the expansion valve to decrease the temperature and pressure.
- ❖ **Expansion Valve** : It reduces the temperature and pressure of the refrigerant using a throttling device. The throttling process occurs through friction and there is a change in the temperature and pressure of the refrigerant. Its pressure changes from that in the Receiver to that in the Evaporator.
- ❖ **Evaporator** : The cyclic process that decreases the temperature of the items stored takes place here. It takes heat from the storage compartment or atmosphere that is supposed to be cooled. This heat is then used to vaporize the liquid refrigerant. This way, the food items are cooled and preserved.
- ❖ **Blowers** : The cooled air is spread across the room through the convection process, thus, achieving the desired temperature of the room.



In short, the refrigerant's boiling point is reduced by increasing its temperature and pressure by the compressor. Heat is removed from the refrigerant as it is changed from gaseous to the liquid state by the condenser. Now the refrigerant is transferred to the reservoir for storage. Further, the refrigerant is moved to an expansion valve to reduce the temperature and pressure in the liquid state. The last step takes place in the Evaporator, where the heat from the surroundings is used to change the refrigerant again to a gaseous state, thus causing the cooling effect. The blower circulates this cool air.

**Figure: About us**

## Booking

- Bookings can be made by filling the given form in the booking section.
- The compartment will be allocated according to the booking requirements filled by the user in the booking form.

BOOKING

## BOOK YOUR STORAGE

Name:  
Your name

Gender:  
male/female/others

Email:  
Email

Phone Number:  
Phone number

Username:  
username name

Password:  
Password

Storage Type:  
Type:1,2,3,etc

Number of Storages:  
1,2,3,etc

Booking Dates  
From: dd/mm/yyyy To: dd/mm/yyyy

Number of days:  
1,2,3,etc

Address:  
Address

City:  
City

State:  
State

Pincode:  
pincode

Confirm Cancel

**Figure:** Booking Page

## Storages

- Storages available along with their fares for 1 day

STORAGES

### CHECK OUR STORAGES



Storage 1

[Book](#) [Details](#)

250 INR /day/storage



Storage 2

[Book](#) [Details](#)

350 INR /day/storage



Storage 3

[Book](#) [Details](#)

280 INR /day/storage



Storage 4

[Book](#) [Details](#)

450 INR /day/storage



Storage 5

[Book](#) [Details](#)

500 INR /day/storage



Storage 6

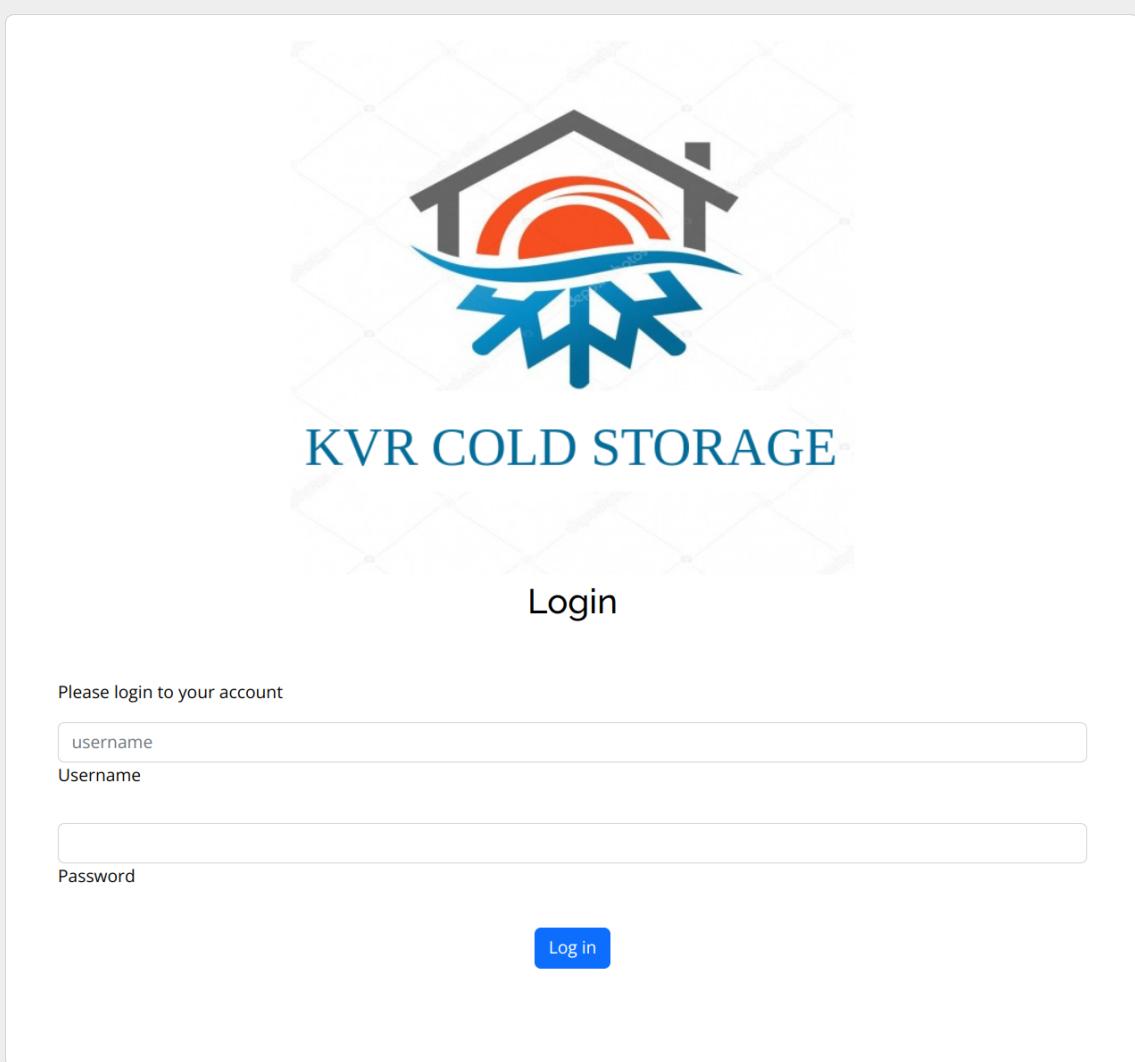
[Book](#) [Details](#)

550 INR /day/storage

**Figure:** Storage Selection Page

## Admin Login

LOGIN  
**LOGIN**



**Figure:** Admin Login Page

## Admin Section

There are two options under admin sections:

- Booking Details
- Reviews

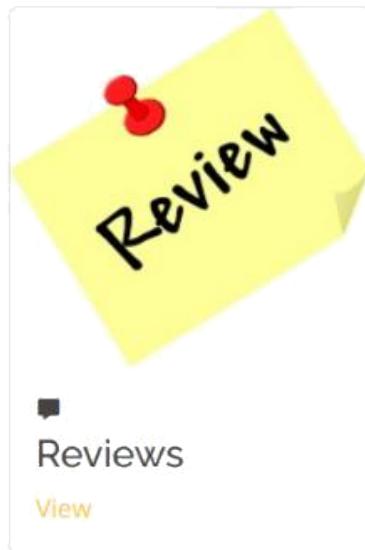
ADMIN

# ADMIN SECTION



 Booking Details

[View](#)



 Reviews

[View](#)

**Figure:** Admin Section Page

## Bookings

- Booking details available in the admin section of the KVR Cold Storages

DETAILS

### LIST OF ALL STORAGE BOOKINGS

id	Booking code	Client Name	Storage Id	Sex	Email	Phone	No of storages	Date From	Date To	No of Days	Amount	Address	City	State	Pincode
1	KVR-0000000001	M Kartik Kumar	5	Male	kartiksachu@gmail.com	9876546752	2	2021-12-17	2021-12-31	14	14000.0	Jayanagar	Bangalore	Karnataka	560090
2	KVR-0000000002	K Homi	6	Female	imoh@gmail.com	9765376521	3	2021-12-20	2022-01-01	11	18150.0	Block no.1, Shastrinagar	Jamshedpur	Jharkhand	830050
7	KVR-0000000023	Anushka Sinha	1	Female	anu@gmail.com	2345678912	2	2023-01-25	2023-01-30	5	2500.0	JSSATE Girls Hostel	Bangalore	Karnataka	560060
8	KVR-0000000024	Aditya Vaibhav	2	Male	adi23@gmail.com	6299188694	5	2023-01-20	2023-01-25	5	8750.0	Bayapanahalli	Bangalore	Karnataka	560080
12	KVR-0000000028	Sudeep Kumar	4	Male	sudeepdino@gmail.com	1234567891	2	2023-01-20	2023-01-27	7	6300.0	South end circle	Bangalore	Karnataka	560060
14	KVR-0000000030	Pallavi A	5	Female	pallavi@gmail.com	2345678912	7	2023-01-20	2023-01-22	2	7000.0	Indranagar	Bangalore	Karnataka	560080
16	KVR-0000000032	M Rachita	6	Female	rachita@gmail.com	8765432109	9	2023-01-20	2023-01-30	10	45000.0	block no 1,shastrinagar,kadma	Jamshedpur	Jharkhand	850010
26	KVR-0000000015	Vishnu Ojha	2	Male	vishnu@gmail.com	8765432109	1	2023-01-21	2023-01-31	10	10000.0	Bayapanahalli	Bangalore	Karnataka	560080

**Figure:** Booking Page Initially

## booking \_details Table

- Booking details available in the database of the KVR Cold Storages

The screenshot shows the MySQL Workbench interface with the 'booking\_details' table selected. The table contains 10 rows of data, each representing a booking entry. The columns include id, name, gender, email, number, username, password, storage\_type, No\_of\_storages, date\_from, date\_to, No\_of\_days, address, city, state, and pincode. The data includes various names like Varsha, Homi K, Kartik Kumar, Anushka Sinha, Aditya Vaibhav, Sudeep Kumar, Pallavi A, M Rachita, Vishnu Ojha, and Vaishnavi, along with their respective details such as email addresses, phone numbers, and booking dates.

	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	<b>id</b>	<b>name</b>	<b>gender</b>	<b>email</b>	<b>number</b>	<b>username</b>	<b>password</b>	<b>storage_type</b>	<b>No_of_storages</b>	<b>date_from</b>	<b>date_to</b>	<b>No_of_days</b>	<b>address</b>	<b>city</b>	<b>state</b>	<b>pincode</b>
<input type="checkbox"/>	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	2	Varsha	Female	varshacs086@gmail.com	6299188694	varsha_8	1js20cs086	1	2	2023-01-18	2023-01-24	0	JSSATE Girls Hostel	Bangalore	Karnataka	560060
<input type="checkbox"/>	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	14	Homi K	Female	imoh@gmail.com	1234567891	imoh_5	nitjsr	2	3	2023-01-20	2023-01-24	4	block no 1,shastrinagar,kadma	Jamshedpur	Jharkhand	850010
<input type="checkbox"/>	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	22	Kartik Kumar	Male	kartiksachu@gmail.com	6299188694	kartik_2	sirmvit	3	2	2023-01-20	2023-01-30	10	SMVIT Boys Hostel	Bangalore	Karnataka	560080
<input type="checkbox"/>	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	24	Anushka Sinha	Female	anu@gmail.com	2345678912	anu_sinha	1js20cs033	1	2	2023-01-25	2023-01-30	5	JSSATE Girls Hostel	Bangalore	Karnataka	560060
<input type="checkbox"/>	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	25	Aditya Vaibhav	Male	adi23@gmail.com	6299188694	adi_23	adityapandey	2	1	2023-01-20	2023-01-25	5	Bayapanahalli	Bangalore	Karnataka	560080
<input type="checkbox"/>	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	29	Sudeep Kumar	Male	sudeepdino@gmail.com	1234567891	dino@gmail.com	sudeppallavi	4	2	2023-01-20	2023-01-27	7	South end circle	Bangalore	Karnataka	560060
<input type="checkbox"/>	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	31	Pallavi A	Female	pallavi@gmail.com	2345678912	pallavi_7	pallavisudeep	5	1	2023-01-20	2023-01-22	2	Indranagar	Bangalore	Karnataka	560080
<input type="checkbox"/>	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	33	M Rachita	Female	rachita@gmail.com	8765432109	rachita_3	123456	6	2	2023-01-20	2023-01-30	10	block no 1,shastrinagar,kadma	Jamshedpur	Jharkhand	850010
<input type="checkbox"/>	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	45	Vishnu Ojha	Male	vishnu@gmail.com	8765432109	vishnu_8	sdfghjkjhg	2	1	2023-01-21	2023-01-31	10	Bayapanahalli	Bangalore	Karnataka	560080
<input type="checkbox"/>	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	48	Vaishnavi	Female	vaishnavi07@gmail.com	8765432109	vaish_07	sdfghjk45@dfg	3	2	2023-01-28	2023-01-31	4	Indranagar	Bangalore	Karnataka	560080

**Figure:** Database after booking

## Triggers

Insertion and deletions performed on booking\_details table is reflected back to the booking\_list managed by the admin of the database. This is achieved with the help of triggers.

### Trigger:

#### 1. get\_bookcode :

```
DROP TRIGGER IF EXISTS `get_bookcode`;
CREATE DEFINER='root'@'localhost` TRIGGER `get_bookcode` BEFORE
    INSERT ON `booking_list` FOR EACH ROW BEGIN SET
        NEW.book_code=CONCAT ("KVR-",LPAD(LAST_INSERT_ID(0,10 ),"0"));
    END
```

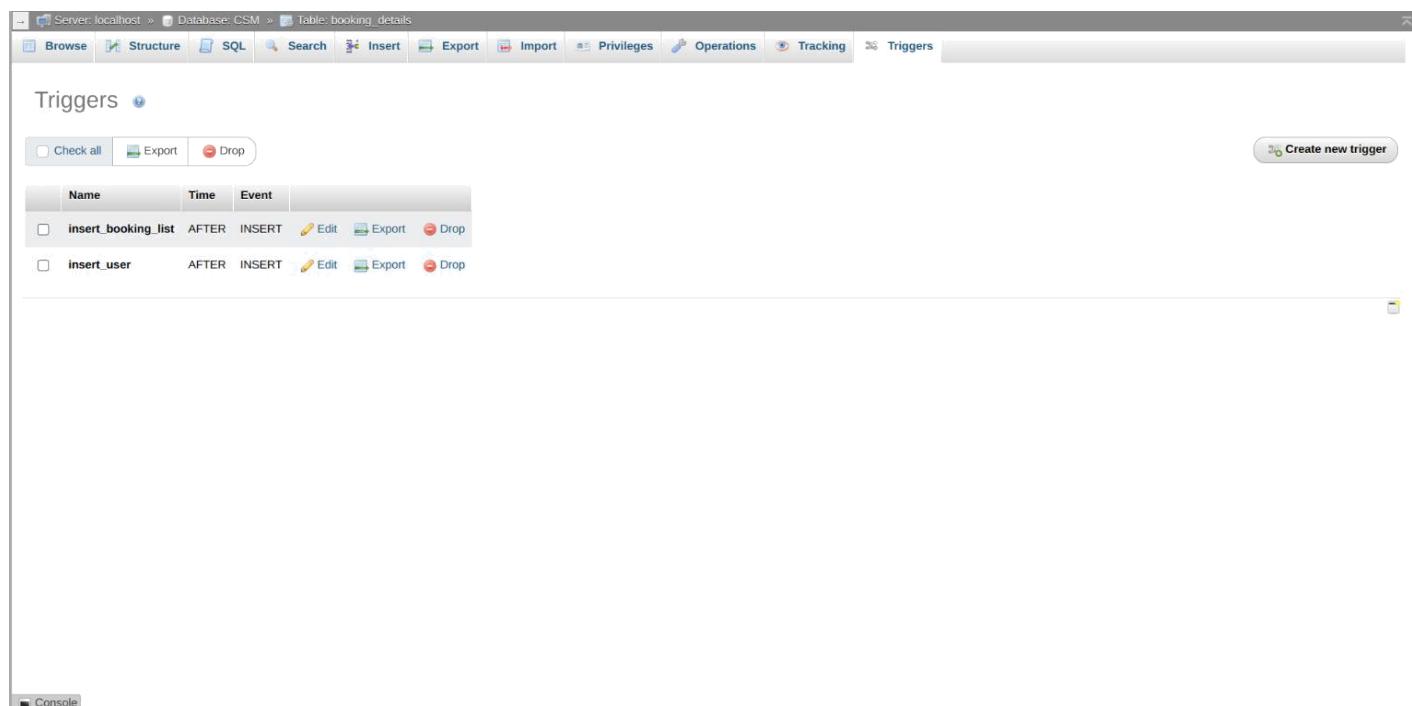
#### 2. insert\_user :

```
DROP TRIGGER IF EXISTS `insert_user`;
CREATE DEFINER='root'@'localhost` TRIGGER `insert_user` AFTER INSERT
ON `booking_details` FOR EACH ROW INSERT INTO users VALUES (null,NEW
.name,NEW.username,NEW.password)
```

#### 3. booking\_list :

```
DROP TRIGGER IF EXISTS `insert_booking_list`;
CREATE DEFINER='root'@'localhost` TRIGGER `insert_booking_list` AFTER
INSERT ON `booking_details` FOR EACH ROW INSERT INTO booking_list
VALUES(null,null,NEW.name,NEW.gender,NEW.email,NEW.number,
    NEW.storage_type,NEW.No_of_storages,NEW.date_from,NEW.date_to,NEW.
    No_of_days,NEW.No_of_storages*NEW.No_of_days*500,NEW.address,
    NEW.city,NEW.state,NEW.pincode)
```

**Description:** This trigger is triggered when the insertion or deletion is done on the booking\_details table and the rows in booking\_lists are modified.



**Figure:** Triggers in the Database

## 1. Database before delete

Showing rows 0 - 8 (9 total). Query took 0.0002 seconds.

SELECT \* FROM `booking\_list`

Profile [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

<input type="checkbox"/> Show all	Number of rows:	25	Filter rows:	Search this table	Sort by key:	None															
Extra options																					
		<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	<input type="checkbox"/> id	book_code	client_name	sex	email	Phone	storage_id	No_of_storages	date_from	date_to	No_of_days	amount	address	city	state	pincode
		<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	1	KVR-0000000001	M Kartik Kumar	Male	kartiksachu@gmail.com	9876546752	5	2	2021-12-17	2021-12-31	14	14000	Jayanagar	Bangalore	Karnataka	560090
		<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	2	KVR-0000000002	K Homi	Female	imoh@gmail.com	9765376521	6	3	2021-12-20	2022-01-01	11	18150	Block no.1, Shastrinagar	Jamshedpur	Jharkhand	830050
		<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	7	KVR-0000000023	Anushka Sinha	Female	anu@gmail.com	2345678912	1	2	2023-01-25	2023-01-30	5	2500	JSSATE Girls Hostel	Bangalore	Karnataka	560060
		<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	8	KVR-0000000024	Aditya Vaibhav	Male	adi23@gmail.com	6299188694	2	5	2023-01-20	2023-01-25	5	8750	Bayapanahalli	Bangalore	Karnataka	560080
		<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	12	KVR-0000000028	Sudeep Kumar	Male	sudeepdino@gmail.com	1234567891	4	2	2023-01-20	2023-01-27	7	6300	South end circle	Bangalore	Karnataka	560060
		<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	14	KVR-0000000030	Pallavi A	Female	pallavi@gmail.com	2345678912	5	7	2023-01-20	2023-01-22	2	7000	Indranagar	Bangalore	Karnataka	560080
		<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	16	KVR-0000000032	M Rachita	Female	rachita@gmail.com	8765432109	6	9	2023-01-20	2023-01-30	10	45000	block no 1,shastrinagar,kadma	Jamshedpur	Jharkhand	850010
		<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	26	KVR-0000000015	Vishnu Ojha	Male	vishnu@gmail.com	8765432109	2	1	2023-01-21	2023-01-31	10	10000	Bayapanahalli	Bangalore	Karnataka	560080
		<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	29	KVR-0000000018	Vaishnavi	Female	vaishnavi07@gmail.com	8765432109	3	2	2023-01-28	2023-01-31	4	4000	Indranagar	Bangalore	Karnataka	560080

Check all With selected:  Edit  Copy  Delete  Export

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

Query results operations

Print  Copy to clipboard  Export  Display chart  Create view

Bookmark this SQL query

Label:  Let every user access this bookmark

Bookmark this SQL query

Console

Figure: Database before deleting

## 2. Deleting an entry

DETAILS

**LIST OF ALL STORAGE BOOKINGS**

Booking id	Booking code	Client Name	Storage Id	Sex	Email	Phone	No of storages	Date From	Date To	No of Days			Address	City	State	Pincode
										Date	From	To	Days	Amount		
1	KVR-0000000001	M Kartik Kumar	5	Male	kartiksachu@gmail.com	9876546752	2	2021-12-17	2021-12-31	14	14000.0	Jayanagar	Bangalore	Karnataka	560090	<button>Delete</button>
2	KVR-0000000002	K Homi	6	Female	imoh@gmail.com	9765376521	3	2021-12-20	2022-01-01	11	18150.0	Block no.1, Shastrinagar	Jamshedpur	Jharkhand	830050	<button>Delete</button>
7	KVR-0000000023	Anushka Sinha	1	Female	anu@gmail.com	2345678912	2	2023-01-25	2023-01-30	5	2500.0	JSSATE Girls Hostel	Bangalore	Karnataka	560060	<button>Delete</button>
8	KVR-0000000024	Aditya Vaibhav	2	Male	adi23@gmail.com	6299188694	5	2023-01-20	2023-01-25	5	8750.0	Bayapanahalli	Bangalore	Karnataka	560080	<button>Delete</button>
12	KVR-0000000028	Sudeep Kumar	4	Male	sudeepdino@gmail.com	1234567891	2	2023-01-20	2023-01-27	7	6300.0	South end circle	Bangalore	Karnataka	560060	<button>Delete</button>
14	KVR-0000000030	Pallavi A	5	Female	pallavi@gmail.com	2345678912	7	2023-01-20	2023-01-22	2	7000.0	Indranagar	Bangalore	Karnataka	560080	<button>Delete</button>
16	KVR-0000000032	M Rachita	6	Female	rachita@gmail.com	8765432109	9	2023-01-20	2023-01-30	10	45000.0	block no 1,shastrinagar,kadma	Jamshedpur	Jharkhand	850010	<button>Delete</button>
26	KVR-0000000015	Vishnu Ojha	2	Male	vishnu@gmail.com	8765432109	1	2023-01-21	2023-01-31	10	10000.0	Bayapanahalli	Bangalore	Karnataka	560080	<button>Delete</button>
28	KVR-0000000017	Vaishnavi	3	Female	vaishnavi07@gmail.com	8765432109	2	2023-01-28	2023-01-31	4	4000.0	Indranagar	Bangalore	Karnataka	560080	<button>Delete</button>

Figure: Deleting from bookings

## Trigger for updating User table

The screenshot shows the 'Triggers' section of the phpMyAdmin interface for the 'booking' table. A single trigger, 'get\_bookcode', is listed. It is defined as a BEFORE INSERT trigger on the 'booking' table. The trigger is associated with the 'Edit', 'Export', and 'Drop' buttons.

**Figure:** Trigger for updating user table

## Database after updating user table

The screenshot shows the 'users' table in the phpMyAdmin interface. The table has 18 rows, each representing a user with columns: id, firstname, username, and password. The data includes various names like Khushi, M, Raj, Anushka, Nikhil, Pallavi, M Rachita, Vishnu, and Vaishnavi, along with their corresponding usernames and passwords.

	id	firstname	username	password
<input type="checkbox"/>	1	Khushi	khushi_rani	4744ddea876b11dc81d169fadf494418
<input type="checkbox"/>	2	M	m_varsha	0192023a7bbd73250516f069df18b500
<input type="checkbox"/>	3	Raj	raj_harsh	0192023a7bbd73250516f069df18b500
<input type="checkbox"/>	4	Anushka	anushka_sinha	4744ddea876b11dc81d169fadf494418
<input type="checkbox"/>	5	Nikhil	nikhil_rajput	0192023a7bbd73250516f069df18b500
<input type="checkbox"/>	6	Pallavi A	pallavi_7	pallavisudeep
<input type="checkbox"/>	7	M Rachita	rachita_3	123456
<input type="checkbox"/>	15	Vishnu Ojha	vishnu_8	sdfghjkjhg
<input type="checkbox"/>	18	Vaishnavi	vaish_07	sdfghjk45@dfg

**Figure:** User table after updating

### 3. Database after deleting an entry

After an entry has been deleted from the database, that particular row will be removed from the booking\_details table of the database and the corresponding changes will be reflected in the booking\_list table of the database. This was achieved using triggers.

The similar changes will be observed in the database whenever an entry has been made such as booking a particular storage from the Cold Storage.

The screenshot shows the MySQL Workbench interface with the following details:

- Server:** localhost
- Database:** CSM
- Table:** booking\_list

The table structure is as follows:

	<b>id</b>	<b>book_code</b>	<b>client_name</b>	<b>sex</b>	<b>email</b>	<b>Phone</b>	<b>storage_id</b>	<b>No_of_storages</b>	<b>date_from</b>	<b>date_to</b>	<b>No_of_days</b>	<b>amount</b>	<b>address</b>	<b>city</b>	<b>state</b>	<b>pincode</b>
<input type="checkbox"/>	1	KVR-0000000001	M Kartik Kumar	Male	kartiksachu@gmail.com	9876546752	5	2	2021-12-17	2021-12-31	14	14000	Jayanagar	Bangalore	Karnataka	560090
<input type="checkbox"/>	2	KVR-0000000002	K Homi	Female	imohi@gmail.com	9765376621	6	3	2021-12-20	2022-01-01	11	18150	Block no.1, Shastri Nagar	Jamshedpur	Jharkhand	830050
<input type="checkbox"/>	7	KVR-0000000023	Anushka Sinha	Female	anu@gmail.com	2345678912	1	2	2023-01-25	2023-01-30	5	2500	JSSATE Girls Hostel	Bangalore	Karnataka	560060
<input type="checkbox"/>	8	KVR-0000000024	Aditya Vaibhav	Male	adi23@gmail.com	6299188694	2	5	2023-01-20	2023-01-25	5	8750	Bayapanahalli	Bangalore	Karnataka	560080
<input type="checkbox"/>	12	KVR-0000000028	Sudeep Kumar	Male	sudeepdino@gmail.com	1234567891	4	2	2023-01-20	2023-01-27	7	6300	South end circle	Bangalore	Karnataka	560060
<input type="checkbox"/>	14	KVR-0000000030	Pallavi A	Female	pallavi@gmail.com	2345678912	5	7	2023-01-20	2023-01-22	2	7000	Indranagar	Bangalore	Karnataka	560080
<input type="checkbox"/>	16	KVR-0000000032	M Rachita	Female	rachita@gmail.com	8765432109	6	9	2023-01-20	2023-01-30	10	45000	block no 1.shastrinagar,kadma	Jamshedpur	Jharkhand	850010
<input type="checkbox"/>	26	KVR-0000000015	Vishnu Ojha	Male	vishnu@gmail.com	8765432109	2	1	2023-01-21	2023-01-31	10	10000	Bayapanahalli	Bangalore	Karnataka	560080

Below the table, there are several buttons for operations like Check all, Edit, Copy, Delete, and Export. There is also a "Query results operations" section with Print, Copy to clipboard, Export, Display chart, and Create view options. At the bottom, there is a "Bookmark this SQL query" section with a Label input field and a checkbox for Let every user access this bookmark, along with a "Console" button.

**Figure:** Changes in the database after deleting

# Storage Procedures

## 1. Storage details

DETAILS

### STORAGE DETAILS

<b>id</b>	<b>Name</b>	<b>Cost</b>	<b>Description</b>	<b>Capacity</b>	<b>Status</b>
1	Storage 1	250.0	KVR refrigerated containers come in a variety of sizes from 10ft to 45ft and have a temperature range of -40°C to +10°C. Most cold store units are portable and can be used to transport goods safely. Mega Cold Stores are also a popular option for larger businesses. These units are combined with multiple cold storage units to make a mega refrigerated solution. The air conditioning cold room is used for long-term storage of fresh fruits, vegetables, etc. In addition to the control of temperature and moisture in the cold room, we need to consider the respiration of plants in the cold room at the same time, but also regulate and control O2, CO, N2, and ethylene, inhibits respiration and metabolism of fruits and vegetables in a dormant state, in order to achieve the purpose of long-term storage.	5	Available
2	Storage 2	350.0	Flower storage cold room, 100–200 PU/PIR panel with good insulation, high efficiency condensing unit, power consumption saving cost. Suitable temperature range:- (1) For ordinary flowers, 1–3 °C. (2) For tropical flowers, 10–12°C, humidity: 60%-85%. (Fresh flowers are placed in such a cold storage environment with a shelf life of up to 1 month). The air conditioning cold room is used for long-term storage of plants, flowers, etc. In addition to the control of temperature and moisture in the cold room, we need to consider the respiration of plants in the cold room at the same time, but also regulate and control O2, CO, N2, and ethylene, inhibits respiration and metabolism of plants in a dormant state, in order to achieve the purpose of long-term storage.	5	Available
3	Storage 3	280.0	Cold storage for Dairy products, temperatures ranging from -40°C to +18°C, suitable for a variety of dairy products. Temperature range: 0–25°C, -5–25°C, -15–25°C, -25–25°C, -35–25°C, -50–25°C. Panels surface for choice SUS304: Best for anti-rust/corrosion. Salinization sheet: One more salinization layer than color plate, anti-rust/corrosion/scratch, feel like leather and plastic, nice. Color plate: Most popular and cost-effective. Galvanized sheet: Very hard and heavy. Aluminum: Nice and soft, we use 0.7/0.8 mm, hard enough.	5	Available
4	Storage 4	450.0	Automation Grade: Fully Automatic; Capacity: 4Ton; Temperature: -5 to +2 Degree Celsius; Dimensions: 8x8 Feet; Relative Humidity: 10-95%. This unit is properly ventilated and free from dirt, dust & rodents that assure complete safety of the stored products. Sub-divided into various categories for easy recognition and quick retrieval of products, this storage unit is installed with power back-up facility. Periodical checking of the stored lots is done for timely spray of insecticides for any visible insect activity.	5	Available
5	Storage 5	500.0	Room Temperature: Most of the Pharma products storage and activities are done at the room temperature at 20 to 25 degree Celsius. The controlled samples are also stored in this temperature. Cold Storage Conditions: 8 to 15 degree Celsius is popularly known as cool storage conditions. Some drugs degrade in room temperature. These drugs are stored in the cold storage conditions. FEATURES:- .The temperature range and amount/ quantity of the medicine is taken care effectively. .Temperature controls Fully air mapped and secure External temperature logging and data tracking .Cargo placement (evade areas where temperature variation). The volume of medicinal product is taken care of effectively. GMP-compliant to protect any kind of damage, contamination, mix-ups and degradation of the products.	10	Available
6	Storage 6	550.0	KVR offers a range of modular, cold room storage units that are ideal for use in many industries. The range features units of all shapes and sizes. However, they are all highly specified to give optimum control and monitoring of temperature. Complete control makes general storage and stability testing much easier, with temperature modification to +/-0.05°C. Our modular applications lend themselves well to pharmaceutical cold rooms. In many cases Pharmaceutical companies will need a quick solution, that can be delivered and installed on site. Our modular cold rooms are ideal for this scenario, and with temperature ranges from -40°C to +60°C they will meet most storage requirements. Offering complete temperature uniformity through forced air circulation these modular cold room cold storage units are ideal for use in the pharma industry. Boasting internal and emergency lighting as standard and alarm systems to cover numerous types of change with in the cold room. Should you require a modular cold room freezer, our units can generate temperatures of down to -40°C, more than cold enough to freeze most substances. For stability testing, the modular cold rooms offer control and monitoring to an accuracy of at least +/-0.5°C and are calibrated at a minimum of three points across their normal reading range.	10	Available

**Figure:** Storage details

## 2. Stored Procedures in the database

### Procedure:

#### 1. getstorage:

```
DROP PROCEDURE `getstorage` ; CREATE DEFINER=`root`@`localhost`  
PROCEDURE `getstorage`(`IN `sid` INT) NOT DETERMINISTIC CONTAINS SQL  
SQL SECURITY DEFINER select * from storage_list WHERE id=sid
```

#### 2. getuserdetails:

```
DROP PROCEDURE `getuserdetails` ; CREATE DEFINER=`root`@`localhost`  
PROCEDURE `getuserdetails`(`IN `bcode` VARCHAR(100)) NOT  
DETERMINISTIC CONTAINS SQL SQL SECURITY DEFINER SELECT * from  
booking_list WHERE book_code=bcode
```

**Description:** Here whenever the Stored Procedure is called the storage details and user details table will be displayed accordingly

The screenshot shows the MySQL Workbench interface with the 'Routines' tab selected. The toolbar at the top includes buttons for Structure, SQL, Search, Query, Export, Import, Operations, Privileges, Routines, Events, Triggers, Tracking, Designer, and Central columns. Below the toolbar, there are buttons for Check all, Export, Drop, and a search bar. A 'Create new routine' button is also present. The main area displays a table with two rows of stored procedures:

Name	Type	Returns	Edit	Execute	Export	Drop
getstorage	PROCEDURE					
getuserdetails	PROCEDURE					

**Figure:** Stored procedures

## Stored Procedure(i) execution

The screenshot shows the phpMyAdmin interface for a database named 'CSM'. On the left, the database structure is visible, including a 'ColdStorageManagement' schema with tables like 'admin', 'booking\_details', 'message\_list', 'storage\_list', and 'users'. In the center, under the 'Routines' tab, there is a modal window titled 'Execute routine `getstorage`'. This window contains a table for 'Routine parameters' with one entry: 'sid' of type INT with a value of '1'. Below the table are 'Go' and 'Close' buttons.

Figure: Stored procedure(i) execution

## Stored Procedure(i) result

The screenshot shows the phpMyAdmin interface after executing the stored procedure. A success message at the top states 'Your SQL query has been executed successfully' and '1 row affected by the last statement inside the procedure'. The query executed was 'SET @p0='1'; CALL `getstorage` (@p0);'. Below this, the 'Execution results of routine `getstorage`' section displays a table with one row of data. The table columns are 'id', 'name', 'cost', 'description', 'Capacity', and 'status'. The data row is: id=1, name='Storage', cost=250, description='KVR refrigerated containers come in a variety of sizes from 10ft to 45ft and have a temperature range of -40°C to +10°C. Most cold store units are portable and can be used to transport goods safely. Mega Cold Stores are also a popular option for larger businesses. These units are combined with multiple cold storage units to make a mega refrigerated solution. The air conditioning cold room is used for long-term storage of fresh fruits, vegetables, etc. In addition to the control of temperature and moisture in the cold room, we need to consider the respiration of plants in the cold room at the same time, but also regulate and control O<sub>2</sub>, CO, N<sub>2</sub>, and ethylene, inhibits respiration and metabolism of fruits and vegetables in a dormant state, in order to achieve the purpose of long-term storage.', Capacity=5, status=Available. At the bottom, the 'Routines' section lists two procedures: 'getstorage' and 'getuserdetails'.

Figure: Stored procedure(i) result

## Stored Procedures(ii) execution

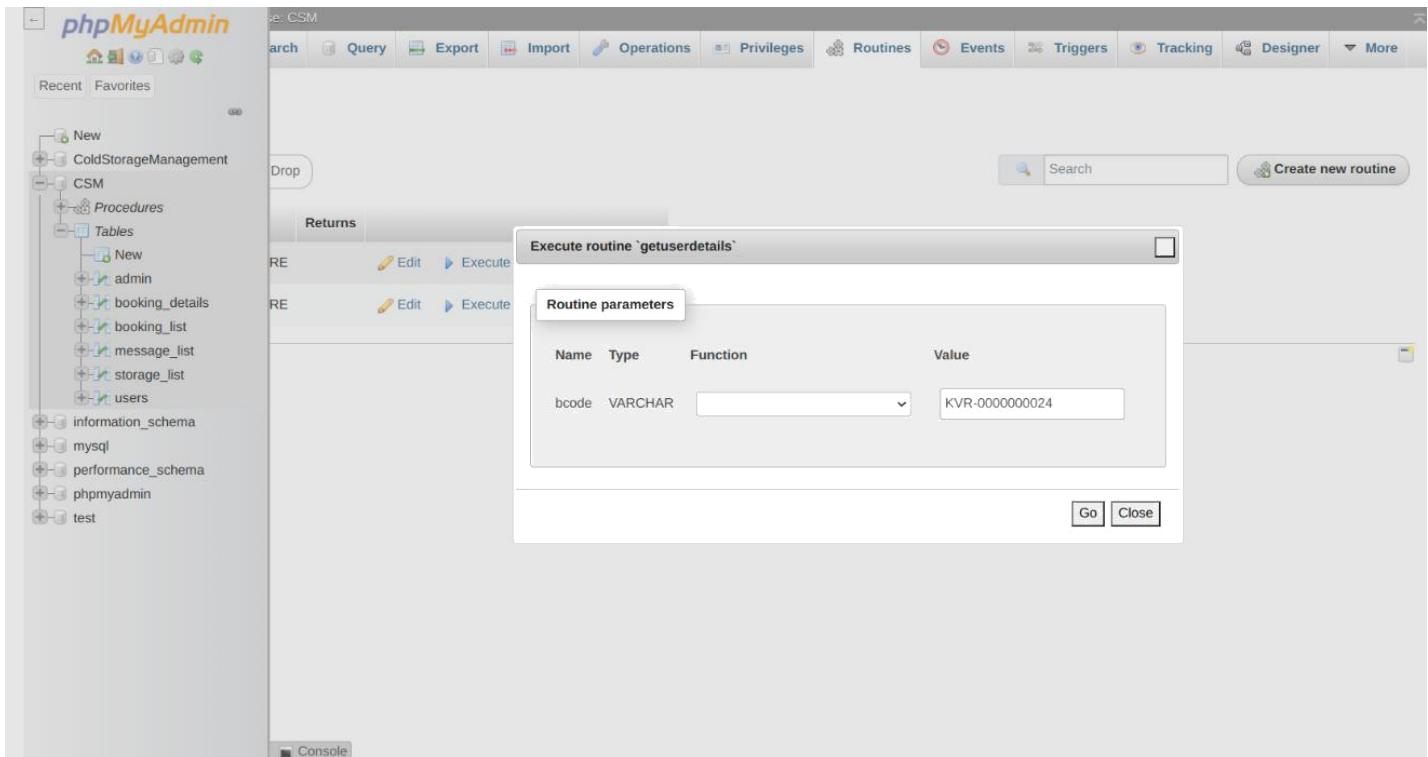


Figure: Stored procedure(ii) execution

## Stored Procedure(ii) result

The screenshot shows the phpMyAdmin interface for the 'CSM' database. At the top, a message says 'Your SQL query has been executed successfully. 1 row affected by the last statement inside the procedure.' Below this, the SQL query 'SET @p0='KVR-0000000024'; CALL `getuserdetails`(@p0);' is shown. Under 'Execution results of routine `getuserdetails`', there is a table with one row of data:

id	book_code	client_name	sex	email	Phone	storage_id	No_of_storages	date_from	date_to	No_of_days	amount	address	city	state	pincode
8	KVR-0000000024	Aditya Vaibhav	Male	adi23@gmail.com	6299188694	2	5	2023-01-20	2023-01-25	5	8750	Bayapanahalli	Bangalore	Karnataka	560080

Below the results, the 'Routines' section lists two procedures:

Name	Type	Returns
getstorage	PROCEDURE	
getuserdetails	PROCEDURE	

Figure: Stored procedure(ii) result

## Assertions

In a database management system (DBMS), an assertion is a statement that a certain condition must always hold true within the database. Assertions are used to define integrity constraints, which ensure that the data in the database is consistent and accurate. Examples of assertions include statements such as "every customer must have a unique ID," or "the total amount of an invoice must be greater than zero." DBMSs typically provide mechanisms for defining and enforcing assertions, such as the use of triggers or check constraints.

The Admin control is the assertion in this project as it cannot be modified or removed under any circumstances.

The screenshot shows the MySQL Workbench interface with the following details:

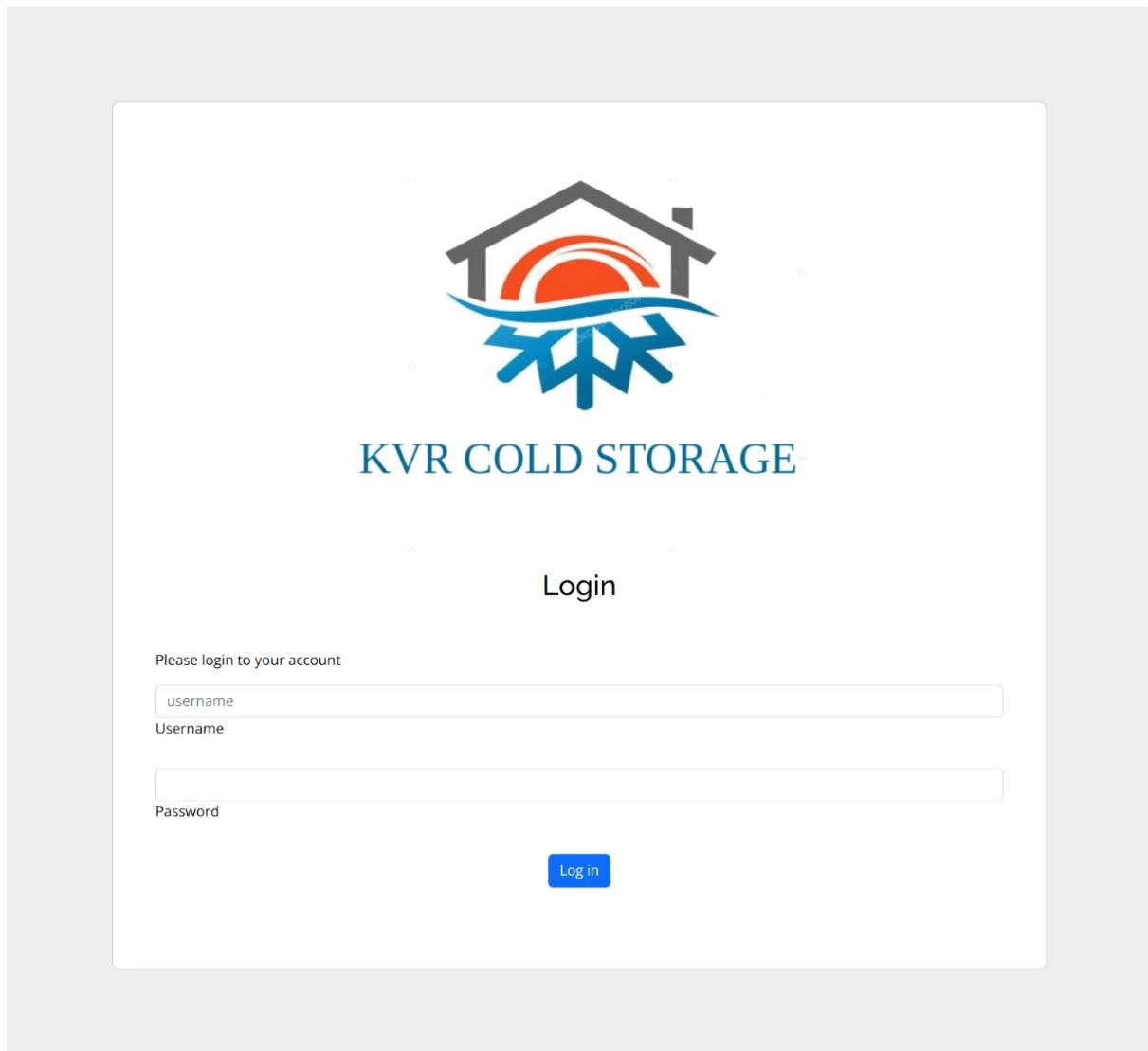
- Toolbar:** Browse, Structure, SQL, Search, Insert, Export, Import, Privileges, Operations, Tracking, Triggers.
- Status Bar:** Showing rows 0 - 2 (3 total, Query took 0.0001 seconds.)
- Query Editor:** SELECT \* FROM `admin`
- Data Grid:** Shows the 'admin' table with columns sno, username, password. Data rows:
  - 1 Varsha\_7 varsha7007
  - 2 Khushi\_rani happy22
  - 3 Raj\_Harsh raj\_shah7
- Action Buttons:** Edit, Copy, Delete for each row.
- Table Navigation:** Up, Down, Check all, With selected: Edit, Copy, Delete, Export.
- Filter and Sort:** Show all, Number of rows: 25, Filter rows: Search this table, Sort by key: None.
- Query Results Operations:** Print, Copy to clipboard, Export, Display chart, Create view.
- Bookmark Options:** Bookmark this SQL query, Label: (empty), Let every user access this bookmark.
- Console:** A small window at the bottom left.

**Figure:** Admin assertion

## User Login

LOGIN

**LOGIN**



**Figure:** User Login page

## User's Section

USER

**USER SECTION**



**Figure:** User Section

# Contact Us

CONTACT

## CONTACT US



**Location:**  
Srinivaspura Road, JSSATE, Kengeri, Bangalore 560060, India

**Email:**  
kvrstorage@gmail.com

**Call:**  
+91 5589 5548 85

Your Name

Your Email

Phone number

Message

**Figure:** Contact page

## Reviews

REVIEW DETAILS				
REVIEWS				
<b>id</b>	<b>Name</b>	<b>Contact</b>	<b>Email</b>	<b>Message</b>
1	Khushi Rani	09123456789	khushi@gmail.com	Excellent quality and value for money.
2	M Varsha	097894561423	mvarsha@gmail.com	Excellent service and amazing quality
3	Raj Harsh	09123456789	rajharsh@gmail.com	Various kinds of suitable and promising PCMs are provided by them. Satisfied with the service.
4	Anushka Sinha	097894561423	anushkasinha@gmail.com	Good service.
5	Nikhil Kumar Rajput	097894561423	nikhilrajput@gmail.com	I would recommend it to others.Used 2 storages to store the flowers and were perfectly fresh.
20	Aditya	098765234567	adi23@gmail.com	Amazing quality service and equiped with all modern day technologies.
24	Homi	8976543678	imoh@gmail.com	Amazing experience!

**Figure:** Reviews by the customers

## Reviews stored in database

Server: localhost > Database: CSM > Table: message_list																																																																													
<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">SQL</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Export</a> <a href="#">Import</a> <a href="#">Privileges</a> <a href="#">Operations</a> <a href="#">Tracking</a> <a href="#">Triggers</a>																																																																													
Showing rows 0 - 6 ( total, Query took 0.0001 seconds.)																																																																													
<pre>SELECT * FROM `message_list`</pre>																																																																													
<a href="#">Profiling</a> <a href="#">Edit inline</a> <a href="#">Edit</a> <a href="#">Explain SQL</a> <a href="#">Create PHP code</a> <a href="#">Refresh</a>																																																																													
<input type="checkbox"/> Show all   Number of rows: 25 <input type="button" value="Filter rows: Search this table"/> Sort by key: None																																																																													
<a href="#">Extra options</a>																																																																													
<table border="1"><thead><tr><th></th><th><a href="#">Edit</a></th><th><a href="#">Copy</a></th><th><a href="#">Delete</a></th><th><b>id</b></th><th><b>fullname</b></th><th><b>contact</b></th><th><b>email</b></th><th><b>message</b></th></tr></thead><tbody><tr><td><input type="checkbox"/></td><td><a href="#">Edit</a></td><td><a href="#">Copy</a></td><td><a href="#">Delete</a></td><td>1</td><td>Khushi Rani</td><td>09123456789</td><td>khushi@gmail.com</td><td>Excellent quality and value for money.</td></tr><tr><td><input type="checkbox"/></td><td><a href="#">Edit</a></td><td><a href="#">Copy</a></td><td><a href="#">Delete</a></td><td>2</td><td>M Varsha</td><td>097894561423</td><td>mvarsha@gmail.com</td><td>Excellent service and amazing quality</td></tr><tr><td><input type="checkbox"/></td><td><a href="#">Edit</a></td><td><a href="#">Copy</a></td><td><a href="#">Delete</a></td><td>3</td><td>Raj Harsh</td><td>09123456789</td><td>rajharsh@gmail.com</td><td>Various kinds of suitable and promising PCMs are p...</td></tr><tr><td><input type="checkbox"/></td><td><a href="#">Edit</a></td><td><a href="#">Copy</a></td><td><a href="#">Delete</a></td><td>4</td><td>Anushka Sinha</td><td>097894561423</td><td>anushkasinha@gmail.com</td><td>Good service.</td></tr><tr><td><input type="checkbox"/></td><td><a href="#">Edit</a></td><td><a href="#">Copy</a></td><td><a href="#">Delete</a></td><td>5</td><td>Nikhil Kumar Rajput</td><td>097894561423</td><td>nikhilrajput@gmail.com</td><td>I would recommend it to others.Used 2 storages to ...</td></tr><tr><td><input type="checkbox"/></td><td><a href="#">Edit</a></td><td><a href="#">Copy</a></td><td><a href="#">Delete</a></td><td>20</td><td>Aditya</td><td>098765234567</td><td>adi23@gmail.com</td><td>Amazing quality service and equiped with all moder...</td></tr><tr><td><input type="checkbox"/></td><td><a href="#">Edit</a></td><td><a href="#">Copy</a></td><td><a href="#">Delete</a></td><td>24</td><td>Homi</td><td>8976543678</td><td>imoh@gmail.com</td><td>Amazing experience!</td></tr></tbody></table>							<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	<b>id</b>	<b>fullname</b>	<b>contact</b>	<b>email</b>	<b>message</b>	<input type="checkbox"/>	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	1	Khushi Rani	09123456789	khushi@gmail.com	Excellent quality and value for money.	<input type="checkbox"/>	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	2	M Varsha	097894561423	mvarsha@gmail.com	Excellent service and amazing quality	<input type="checkbox"/>	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	3	Raj Harsh	09123456789	rajharsh@gmail.com	Various kinds of suitable and promising PCMs are p...	<input type="checkbox"/>	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	4	Anushka Sinha	097894561423	anushkasinha@gmail.com	Good service.	<input type="checkbox"/>	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	5	Nikhil Kumar Rajput	097894561423	nikhilrajput@gmail.com	I would recommend it to others.Used 2 storages to ...	<input type="checkbox"/>	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	20	Aditya	098765234567	adi23@gmail.com	Amazing quality service and equiped with all moder...	<input type="checkbox"/>	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	24	Homi	8976543678	imoh@gmail.com	Amazing experience!
	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	<b>id</b>	<b>fullname</b>	<b>contact</b>	<b>email</b>	<b>message</b>																																																																					
<input type="checkbox"/>	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	1	Khushi Rani	09123456789	khushi@gmail.com	Excellent quality and value for money.																																																																					
<input type="checkbox"/>	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	2	M Varsha	097894561423	mvarsha@gmail.com	Excellent service and amazing quality																																																																					
<input type="checkbox"/>	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	3	Raj Harsh	09123456789	rajharsh@gmail.com	Various kinds of suitable and promising PCMs are p...																																																																					
<input type="checkbox"/>	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	4	Anushka Sinha	097894561423	anushkasinha@gmail.com	Good service.																																																																					
<input type="checkbox"/>	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	5	Nikhil Kumar Rajput	097894561423	nikhilrajput@gmail.com	I would recommend it to others.Used 2 storages to ...																																																																					
<input type="checkbox"/>	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	20	Aditya	098765234567	adi23@gmail.com	Amazing quality service and equiped with all moder...																																																																					
<input type="checkbox"/>	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	24	Homi	8976543678	imoh@gmail.com	Amazing experience!																																																																					
<a href="#">Check all</a> With selected: <a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a> <a href="#">Export</a>																																																																													
<input type="checkbox"/> Show all   Number of rows: 25 <input type="button" value="Filter rows: Search this table"/> Sort by key: None																																																																													
<a href="#">Query results operations</a>																																																																													
<a href="#">Print</a> <a href="#">Copy to clipboard</a> <a href="#">Export</a> <a href="#">Display chart</a> <a href="#">Create view</a>																																																																													
<a href="#">Bookmark this SQL query</a>																																																																													
Label: <input type="text"/> <input type="checkbox"/> Let every user access this bookmark																																																																													
<a href="#">Bookmark this SQL query</a>																																																																													
<a href="#">Console</a>																																																																													

**Figure:** Reviews by the customers stored in message list

## Team

TEAM

### CHECK OUR TEAM



**M Varsha**

Marketing Officer



**Khushi Rani**

Warehouse Supervisor



**Raj Harsh**

Sales Manager

**Figure:** Team Members

## Conclusion and Future Enhancements

### Conclusion:

A Cold Storage Management System (CSMS) is a database-driven application that helps in managing the inventory and operations of a cold storage facility. The CSMS allows users to track the storage of perishable goods, such as fruits and vegetables, by monitoring the temperature and humidity levels, as well as the inventory of products stored.

The CSMS implemented in this project KVR Cold Storage Management System includes features such as:

- Real-time monitoring of temperature and humidity levels in the cold storage units
- Inventory management of products stored in the facility
- Recording of product information such as expiry dates, storage conditions, and quantity
- Automated alerts for temperature and humidity levels that fall outside of the acceptable range
- Generation of reports for inventory and temperature/humidity data

The implementation of a CSMS ensures that the stored products are always in good condition and reduces the risk of spoilage. It also helps in improving the efficiency of the cold storage facility by providing real-time monitoring of the inventory and temperature/humidity levels. Overall, the Cold Storage Management System is an important tool for managing the operations of a cold storage facility, and it can help in reducing costs and increasing profits for the facility owners.

### Future Enhancements:

There are several potential enhancements that could be made to the Cold Storage Management System (CSMS) in the future:

1. Integration with other systems: The CSMS could be integrated with other systems such as a transportation management system, to track the movement of goods in and out of the cold storage facility.
2. IoT integration: The CSMS could be integrated with Internet of Things (IoT) devices, such as sensors, to collect real-time data on temperature, humidity, and other environmental factors in the cold storage units.
3. Predictive analytics: The CSMS could use predictive analytics to forecast the demand for products and optimize inventory levels, reducing the risk of stockouts and overstocking.
4. Mobile app: A mobile app for the CSMS could be developed, allowing users to access the system and view inventory data, temperature and humidity levels, and other important information from anywhere, at any time.
5. Automated ordering: The system could have an automated ordering system to order the product when the stock is low.
6. Data visualization: The CSMS could include data visualization tools to help users understand the data and trends in the system more easily.
7. Compliance and Audit tracking: The CSMS could be enhanced to include compliance and audit tracking features that allow users to track compliance with industry regulations and standards.

All of these enhancements can help in increasing the efficiency, accuracy, and scalability of the CSMS, making it an even more valuable tool for managing the operations of a cold storage facility.