# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

**BELAGAVI-590018**

# 

**A PROJECT REPORT**

**ON**

**NURSERY MANAGEMENT SYSTEM**

**BY**

**ASHWIN SHETTY RETHESH HV SHREYAS**

4SF17CS035 4SF19CS412 4SF19CS414

In the partial fulfillment of the requirement for V Sem. B. E. (CSE)

**DBMS LABORATORY WITH MINI PROJECT**

Under the guidance of

**POOJA N.S**

Asst. Professor, Dept. of CSE

****

**Department of Computer Science & Engineering**

**SAHYADRI**

**COLLEGE OF ENGINEERING & MANAGEMENT**

**Adyar, Mangaluru-575007**

**2020-21**

**SAHYADRI**

**COLLEGE OF ENGINEERING & MANAGEMENT**

# (Affiliated to Visvesvaraya Technological University, BELAGAVI)

**Adyar, Mangaluru – 07**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

*CERTIFICATE*



This is to certify that the project entitled “**ACTING SCHOOL MANAGEMENT SYSTEM”** issubmitted in partial fulfillment for the requirement of V sem. B. E. (Computer Science & Engineering), “DBMS LABORATORY WITH MINI PROJECT” during the year 2020 – 21 is a result of bonafide work carried out bythth

**ASHWIN SHETTY 4SF18CS028**

**RETHESH HV 4SF19CS412**

**SHREYAS****4SF19CS414**

**………………………… …………………………**

Mrs. Pooja N S Dr. Pushpalatha K

Asst. Prof. Dept. of CS&E HOD,Dept. of CS&E

SCEM, Mangaluru SCEM, Mangaluru

**Signature of the Examiners**

**1. ………………………….**

**2. ………………………….**

**ABSTRACT**

This Acting school Management System realized transmission and control of Acting school system, so as to

Facilitate the management and decision of sales, and reduce a bid burden for nursery business and for their managers.it also helps in improving the efficiency of nursery. Its requirement are to provide the basic information maintaince function of employers, managers and customers so that managers can use the functions add update and delete the basic information of products and their employees details. Acting school management system is very convenient input output and searching for available products so as to make work of this market so efficient. In aspect of software, the nursery management system uses java language and MySQL as the background database. Various configuration in computer including input and output internal memory and external memory capacities are meet the requirement of nursery managers.

**ACKNOWLEDGEMENT**

It is with the great satisfaction and emporia that we are submitting the Mini Project report on **“ACTING SCHOOL MANAGEMENT SYSTEM”**. We have completed the part of 5th semester **DBMS Laboratory with Project** of Bachelor of Engineering in Computer Science and Engineering of Visvesvaraya Technological University, Belagavi.

We take this opportunity to express our sincere thanks and indebtedness to our Project guide and mentor, **Mrs. Pooja N S Professor, Department of CSE**, for her support and guidance. Her vision and suggestions throughout the project period have been fundamental in the completion of project.

We extend our warm gratitude to Dr. Pushpalatha K, Head of the Department, Department of

CSE, for her constant support and advice that helped us to complete this project successfully.

We are extremely grateful to our beloved Principal, Dr. Rajesh S, Principal, Sahyadri College of engineering and management for encouraging us to come up with new ideas and to express them in a systematic manner.

We also like to thank all Teaching &Non-teaching staff of Sahyadri College of Engineering and

Management, Mangaluru for their kind co-operation during the course of my work.

Finally, we are thankful to our family and friends who helped us in our work and made the

project a successful one.

-----------------------------

1. **Ashwin Shetty**
2. **(4SF17CS035)**

-----------------------------

**Rethesh Hv**

1. **(4SF19CS412)**

-----------------------------

**Shreyas**

1. **(4SF19CS414)**

**PAGE INDEX**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Topic** | **Page No.** |
|  | Introduction | 6-7 |
|  | Design | 7 |
| 2.1. | E-R Diagram | 7-8 |
| 2.2. | Relational Schema | 9-10 |
| 2.3. | Schema Diagram | 11 |
| 3. | Normalization | 12-13 |
| 4. | Implementation | 14-17 |
| 5. | Result | 18-21 |
| 6. | Conclusion | 22 |
| 7. | References | 23 |

**CHAPTER : 1**

**INTRODUCTION**

Database is a collection of data and Management System is a set of programs to store and retrieve those data. Based on this we can define DBMS is a collection of inter-related data and set of programs to store & access those data in an easy and effective manner.

Database systems are basically developed for large amount of data. When dealing with huge amount of data, there are two things that require optimization: Storage of data and retrieval of data. A DBMS makes it possible for end users to create, read, update and delete data in a database. The DBMS essentially serves as an interface between the database and end users or application programs, ensuring that data is consistently organized and remains easily accessible.

The DBMS can offer both logical and physical data independence. That means it can protect users and applications from needing to know where data is stored or having to be concerned about changes to the physical structure of data.

* 1. **INTRODUCTION TO MYSQL**

MySQL is an open-source, fast reliable, and flexible relational database management system, typically used with PHP. It is a database system used for developing web-based software applications.

It was developed by Michael Widenius and David Axmark in 1994. It is presently developed, distributed, and supported by Oracle Corporation. It was written in C, C++.It is fully multithreaded by using kernel threads. It can handle multiple CPUs if they are available. It provides transactional and non-transactional storage engines.

* 1. **JAVA**

Java is a popular programming language. It is owned by Oracle, and more than 3 billion devices run Java. Java works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc.).It is open-source secure, fast, free and powerful. As Java is close to C++ and C#, it makes it easy for programmers to switch to Java.

* 1. **INTRODUCTION TO NET BEANS**

NetBeans IDE is a free, open source, integrated development environment (IDE) that enables you to develop desktop, mobile and web applications. The IDE supports application development in various languages, including Java, HTML5, PHP and C++.

The IDE provides integrated support for the complete development cycle, from project

creation through debugging, profiling and deployment. The IDE runs on Windows,

Linux, Mac OS X, and other UNIX-based systems.

* 1. **NECESSITY OF PROJECT**

This project **“NURSERY MANAGEMENT SYSTEM”** will help employees to quickly view what items are available in their industry, and they can add the sold details to the data base, and these records can track by managers.

**CHAPTER :2**

**DESIGN**

* 1. **ER DIAGRAM**

An entity relationship model describes interrelated things of interest in a specific domain of knowledge. The ER Diagram of our project is shown in the figure given below.

ER Diagram:

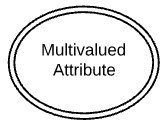


Fig 2.1

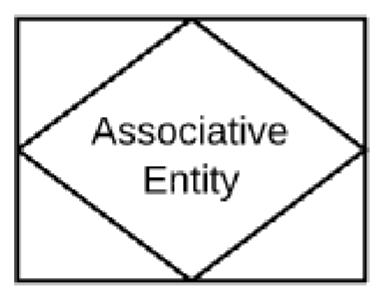


Fig 2.2

Relationship

Fig 2.3



Fig 2.4

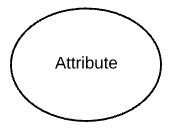


Fig 2.5

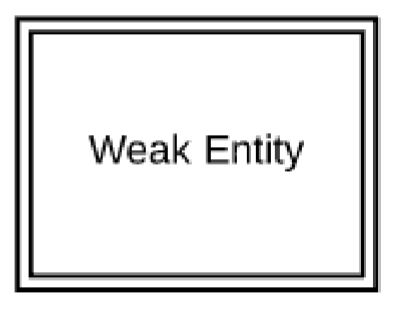
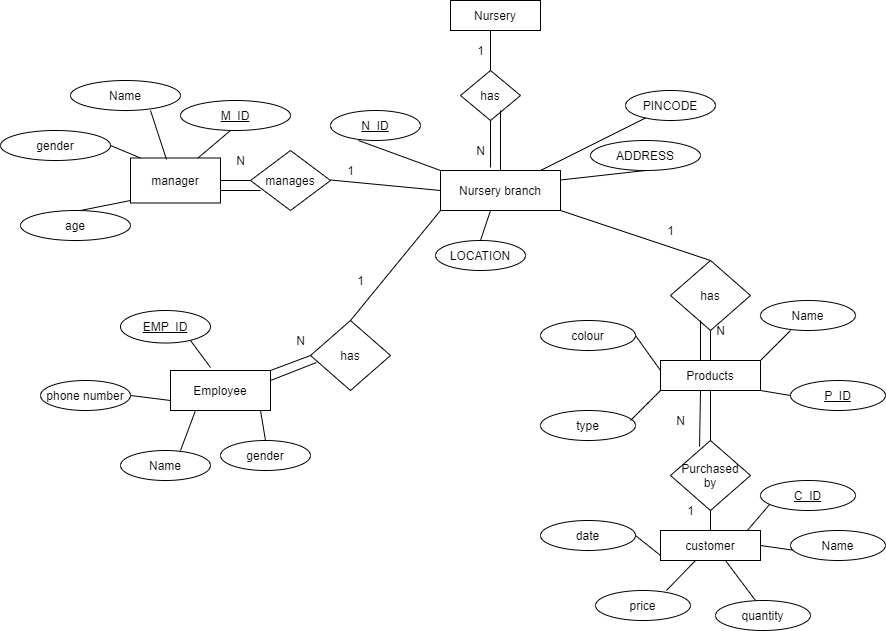


Fig 2.6



* 1. **Relational Schema**

Step 1: Mapping of strong entity

Nursery\_branch

|  |  |  |  |
| --- | --- | --- | --- |
| N\_ID | Loc | Pin\_code | Address |

Manager

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| M\_ID | Name | Gender | Age | N\_ID |

Employee

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| E\_name | Emp\_ID | Gender | Phone\_number | N\_ID |

Products

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| P\_name | P\_ID | type | color | N\_ID |

Customer

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| C\_name | C\_ID | P\_ID | quantity | price | date | Emp\_ID |

Strong entity is a relation

Entity is said to be a strong entity when it contains primary key.

Primary key has an attribute with unique values.

1. In the first relation N\_ID is a primary key.
2. In the second relation M\_IDis a primary key.
3. In the third relation Emp\_ID is a primary key.
4. In the fourth relation P\_ID is a primary key.
5. In the fifth relation C\_ID is a primary key.

STEP 2: Mapping of weak entity Types

There is no weak entity hence this step is ignored

STEP 3: Mapping of 1:1 relation type

There is no 1.1 relation hence this step is ignored

STEP 4: Mapping of 1: N relation types

For every 1: N relation types identify the entity the N side. Make primary key entity

Which is participation in 1 side as foreign key of entity which is N side. If there are any attributes for relation ships add to n side.

Manager

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| M\_ID | Name | Gender | Age | \_ \_\_N\_ID |

Employee

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| E\_name | Emp\_ID | Gender | Phone\_number | N\_ID |

Products

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| P\_name | P\_ID | type | color | N\_ID |

STEP 5**:** Mapping of M: N relation types

There is no M: N relation hence this step is ignored

STEP 6: Mapping of multivalued attributes

There is multivalued attributes relation hence this step is ignored

* 1. **SCHEMA DIAGRAM**

Nursery\_branch

|  |  |  |  |
| --- | --- | --- | --- |
| N\_ID | Loc | Pin\_code | Address |

Manager

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| M\_ID | Name | Gender | Age | N\_ID |

Employee

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| E\_name | Emp\_ID | Gender | Phonee\_number | N\_ID |

Products

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| P\_name | P\_ID | type | color | N\_ID |

Customer

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| C\_name | C\_ID | P\_ID | quantity | price | date | Emp\_ID |

**CHAPTER 3**

**NORMALIZATION**

Nursery\_branch

|  |  |  |  |
| --- | --- | --- | --- |
| N\_ID | Loc | Pin\_code | Address |

* The above relation is in 1NF because there are no multivalued attributes in the relational

Schema.

Manager

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| M\_ID | Name | Gender | Age | N\_ID |

* The above relation is in 1NF because there are no multivalued attributes in the relational

Schema.

* The above relation is 2NF because all the attributes in the relational are fully functionally dependent on primary key.
* The above relation is 3NF because there is no transitive dependency on primary key.

Employee

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Emp\_ID | E\_Name | Gender | Phone\_number | N\_ID |

* The above relation is in 1NF because there are no multivalued attributes in the relational

Schema.

* The above relation is 2NF because all the attributes in the relational are fully functionally dependent on primary key.
* The above relation is 3NF because there is no transitive dependency on primary key.

Products

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| P\_ID | P\_name | type | color | N\_ID |

* The above relation is in 1NF because there are no multivalued attributes in the relational

Schema.

* The above relation is 2NF because all the attributes in the relational are fully functionally dependent on primary key.
* The above relation is 3NF because there is no transitive dependency on primary key.

Customer

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| C\_ID | C\_name | P\_ID | quantity | price | date | Emp\_ID |

* The above relation is in 1NF because there are no multivalued attributes in the relational

Schema.

* The above relation is 2NF because all the attributes in the relational are fully functionally dependent on primary key.
* The above relation is 3NF because there is no transitive dependency on primary key.

**1st Normal Form (1NF)**

The relations are in 1NF since there is no multivalued attributes or nested relation.

**2nd Normal Form (2NF)**

The relation are in 2NF since every nonprime attributes are fully functionally dependent on part of the primary key.

**3rd Normal Form (3NF)**

The relations are in 3NF since no nonprime attributes are transitively dependent on primary key.

**CHAPTER 4**

**IMPLEMENTATION**

**4.1 Required Specification**

**Hardware Requirements**

**Software requirements**

* 1. **Table Structure**

**NURSERY\_BRANCH**

CREATE TABLE NURSERY

(

N\_ID VARCHAR2(45) PRIMARY KEY,

PIN\_CODE VARCHAR2(45),

LOC VARCHAR2(45));

|  |  |
| --- | --- |
| ATTRIBUTES | TYPE |
| N\_ID | VARCHAR2(45) |
| PIN\_CODE | VARCHAR2(45) |
| LOC | VARCHAR2(45) |

* 1. **MANAGER**

CREATE TABLE MANAGER

(

M\_ID VARCHAR2(45) PRIMARY KEY,

NAME VARCHAR2(45),

GENDER VARCHAR2(45),

AGE NUMBER (3),

N\_ID VARCHAR2(45) REFERENCES NURSERY(N\_ID) ON DELETE CASCADE);

|  |  |
| --- | --- |
| ATTRIBUTES | TYPE |
| M\_ID | VARCHAR2(45) |
| NAME | VARCHAR2(45) |
| GENDER | VARCHAR2(45) |
| AGE | VARCHAR2(45) |
| N\_ID | VARCHAR2(45) |

* 1. **EMPLOYEE**

CREATE TABLE EMPLOYEE

(

NAME VARCHAR2(45),

E\_ID VARCHAR2(45),

GENDER VARCHAR2(5),

PHONE\_NUMBER VARCHAR2(45),

N\_ID VARCHAR2(45)REFERENCES NURSERY(N\_ID) ON DELETE CASCADE );

|  |  |
| --- | --- |
| ATTRIBUTES | TYPE |
| NAME | VARCHAR2(45) |
| E\_ID | VARCHAR2(45) |
| GENDER | VARCHAR2(45) |
| N\_ID | VARCHAR2(45) |

* + 1. **PRODUCTS**

CREATE TABLE PRODUCT

(

P\_ID VARCHAR2(45) PRIMARY KEY,

P\_NAME VARCHAR2(45),

P\_COLOR VARCHAR2(45),

P\_TYPE VARCHAR2(45),

N\_ID VARCHAR2(45) REFERENCES NURSERY(N\_ID) ON DELETE CASCADE);

|  |  |
| --- | --- |
| ATTRIBUTES | TYPE |
| P\_ID | VARCHAR2(45) |
| P\_NAME | VARCHAR2(45) |
| P\_COLOUR | VARCHAR2(45), |
| P\_TYPE | VARCHAR2(45) |
| N\_ID | VARCHAR2(45) |

* + 1. **CUSTOMER**

CREATE TABLE CUSTOMER

(

C\_ID VARCHAR2(45) PRIMARY KEY,

NAME VARCHAR2(45),

PRICE NUMBER (45),

QUANTITY NUMBER (100),

P\_DATE DATE,

E\_ID VARCHAR2(45) REFERENCES EMPLOYEE (P\_ID) ON DELETE CASCADE

P\_ID VARCHAR2(45) REFERENCES PRODUCTS(P\_ID) ON DELETE CASCADE);

|  |  |
| --- | --- |
| ATTRIBUTES | TYPE |
| C\_ID | VARCHAR2(45) |
| NAME | VARCHAR2(45) |
| PRICE | NUMBER (45) |
| QUANTITY | NUMBER (45) |
| P\_DATE | DATE |
| E\_ID | VARCHAR2(45) |
| P\_ID | VARCHAR2(45) |

**4.2 FUNCTIONALTIES**

**4.2.1 Connecting to Database**

**4.2.2 Insert**

**4.2.3 Delete**

**4.2.4 Update**

**4.2.5 Tigger**

**CHAPTER 5**

**RESULT**

* 1. **Snapshots**
     1. **Login page**

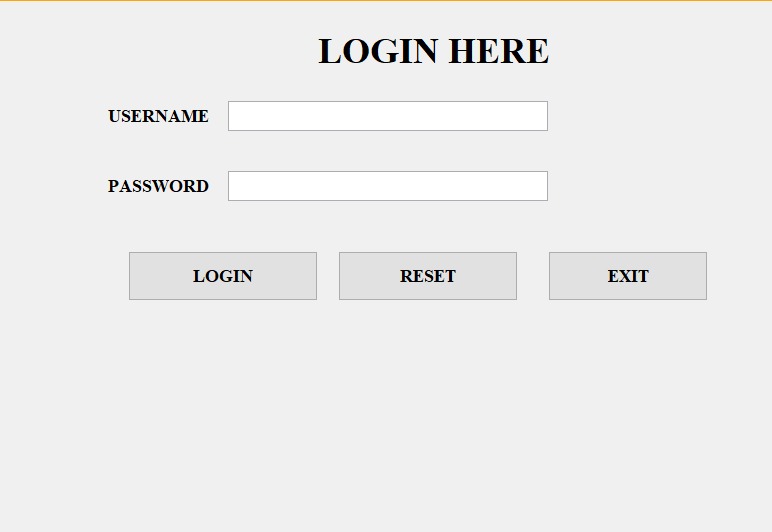
****

Fig 5.1

This page shows the login of the manager using his credentials after which he can manipulate the data beign stored in database.

* + 1. **Home page**

****

Fig 5.2

This page shows the over view of the employee, products, customer page. Manager can manipulate the details who buys the product in this nursery.

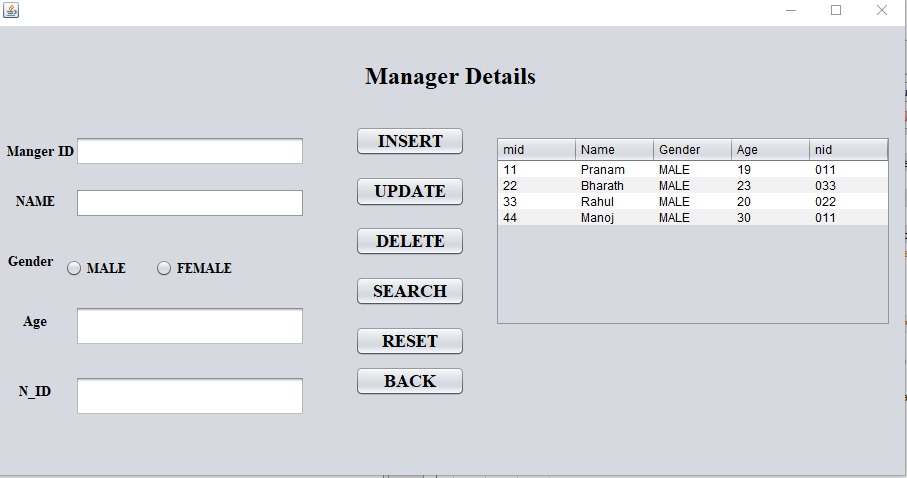
* + 1. **Manager**

Fig 5.3

This table shows the manager details, the admin can manipulate the manager details and can add new manager or update or delete the manager details.

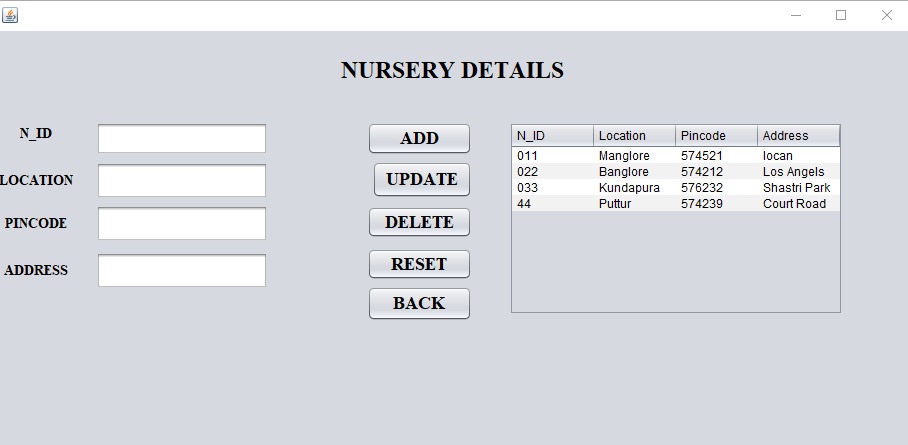
* + 1. **Nursery Branch**

Fig 5.4

This table shows the nursery details. admin can alter the nursery table, he can add new branch.

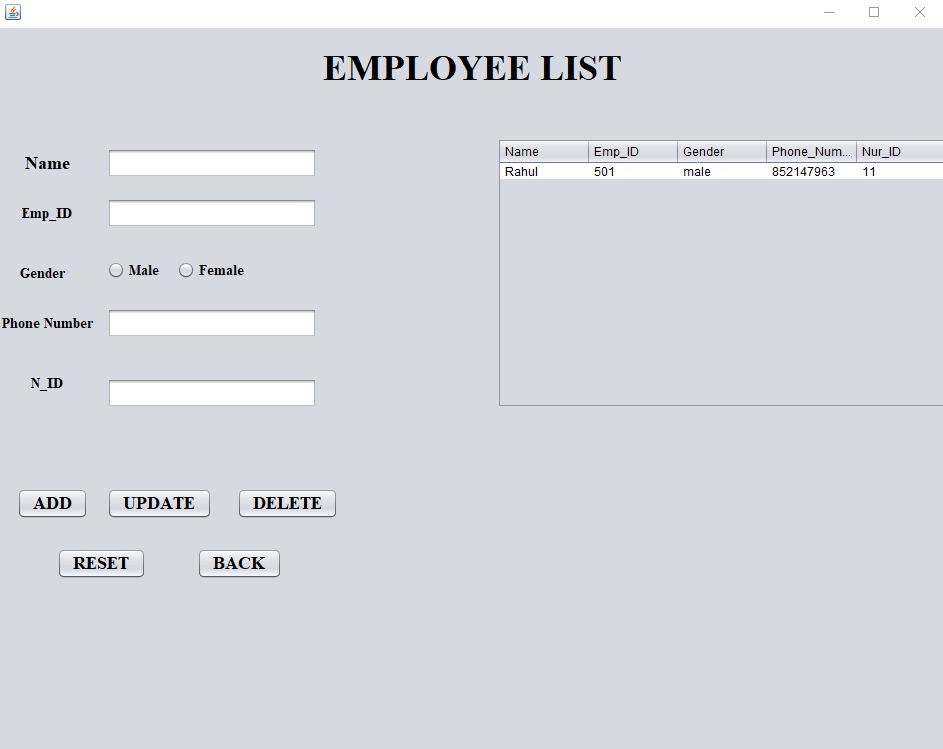
* + 1. **Employee**

Fig 5.4

This page shows the details of employees to the manager, manager can alter the employee details.

* + 1. **Products**

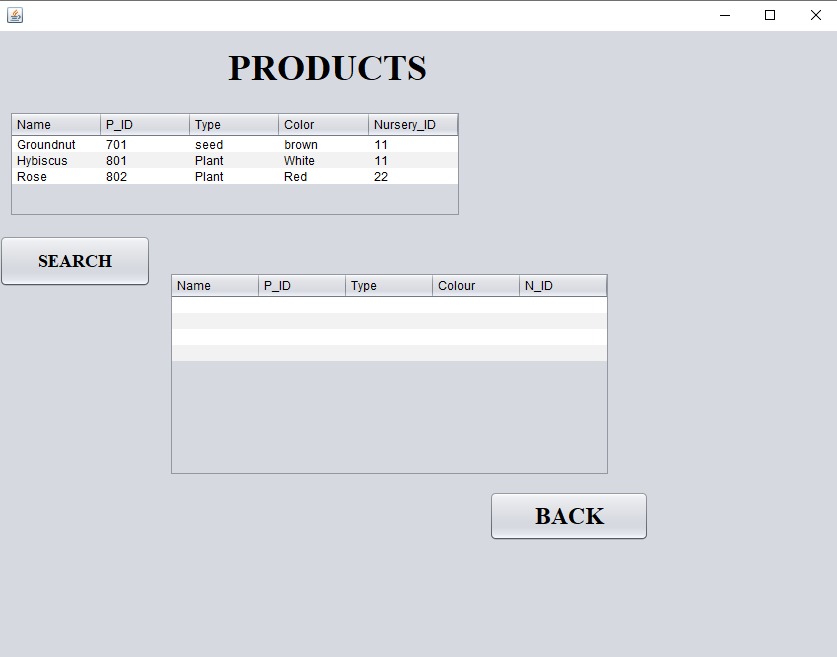


Fig 5.5

This page shows the product details to managers, manager can manipulate the products table.

* + 1. **Customer**

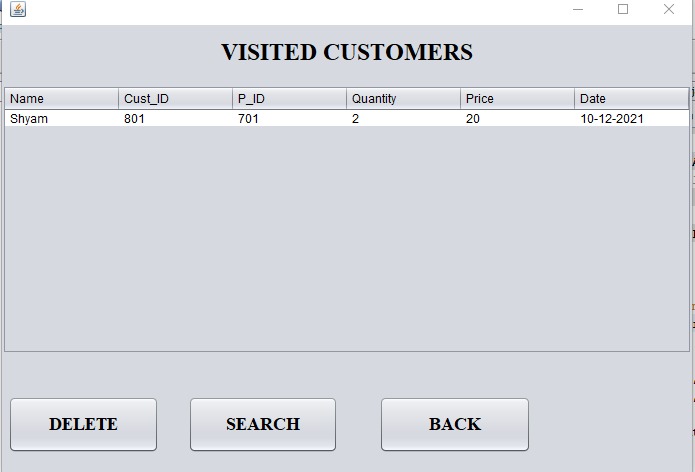
****

Fig 5.6

This table shows the details of customer who buys the product and manager can keep track of the customer.

**CONCLUSION**

Nursery management system is an application which can be implemented easily to store and retrieve the nursery products. The main objective of the Nursery management system is to manage the nursery products, employers and managers.The application is user-friendly and anyone can be trained in a short spam of time to master the usage of this application. Nursery Management System has been developed and automate the over-all processing of any nursery details. The project offers user to enter the data through simple and interactive manner. Data storage and retrieval will become easier and faster to maintain. This project is a very flexible software.

**REFERENCES**

* [www.youtube.com](http://www.youtube.com)
* [www.wikipedia.com](http://www.wikipedia.com)
* <https://www.mysqltutorial.org>
* Dbms textbook
* Sql study txtboookk