CHAPTER – 1

INTRODUCTION TO DATABASE

Database and database technology has a major impact on the growing use of computers. It is fair to say that databases play a critical role in almost all areas where computers are used, including business, electronic commerce, engineering, medicine, genetics, law, education, and library science.

The word database is so commonly used that we must begin by defining what the database is. Our initial definition is quite general. A database is a collection of related data. By data, we mean known facts that can be recorded and that have implicit meaning. For example, consider the names, telephone numbers, and addresses of the people you know. You may have recorded this data in an indexed address book or you may have stored it on a hard drive, using personal computers and software such as Microsoft excel. This collection of related data with an implicit meaning is a database.

The preceding definition of a database is quite general, for example, we may consider the collection of words that make up this page of text to be related data and hence to constitute a database. However, the common use of the term database is usually more restricted. A database has the following properties:

A database represents some aspect of the real world, sometimes called the mini world or the universe of discourse. The changes to the mini world are reflected in the database.

A database is a logically coherent collection of data with some inherent meaning. A random assortment of data cannot correctly be referred to as a database.

A database is designed, built and populated with data for a specific purpose. It has an intended group of users and some preconceived applications in which these users are interested.

In other words, a database has some source from which data is derived, some degree of interaction with events in the real world, and an audience that is actively interested in its contents. The end-users of the database may perform business transactions (for example a customer buys a camera) or events may happen that may cause the information in the

database to change. In order for a database to be accurate and reliable at all times, it must be a true reflection of the mini world that it represents; therefore changes must be reflected in the database as soon as possible

.A database can be of any size and complexity. A database may be generated and maintained manually or computerized. For example, a library card catalog is a database that may be created and maintained manually. A computerized database may be created and maintained either by a group of application programs written specifically for that task or by a database management system.
A database is a collection of data, typically describing the activities of one or more related organizations. For example, a university database might contain information about the following

Entities such as students, faculty, courses, and classrooms.
Relationships between entities, such as student's enrolment in courses, faculty teaching courses, and the use of rooms for courses.

File system versus DBMS

To understand the need for a DBMS, let us consider a motivating scenario: a company has a large collection (say 500 GB) of data on employees, departments, products, sales, and so on. This data is accessed concurrently by several employees. Questions about the data must be answered quickly, changes made to the data by different users must be applied consistently and access to certain parts of the data must be restricted. We can try to manage the data by storing it in operating system files. This approach has many drawbacks, including the following

● We probably do not have 500GB of main memory to hold all the data. We must, therefore, store data in a storage device such as a disk or tape and bring relevant parts into the main memory for processing as needed.
● Even if we have 500 GB of main memory, on computer systems with 32 bit addressing, we cannot refer directly to more than about 4 GB of data. We have to program some method of identifying all data items.
● We have to write special programs to answer each question a user may want to ask about the data. These programs are likely to be complex because of the large

volume of data to be searched.

● We must protect the data from the inconsistent changes made by different users accessing the data concurrently. If applications must address the details of such concurrent access, this adds greatly to their complexity.

● We must ensure that the data is restored to a consistent state if the system crashes while changes are being made.

● Operating systems provide only a password mechanism for security. This is not sufficiently flexible to enforce security policies in which different users have permission to access different subsets of the data.

A DBMS is a piece of software designed to make the preceding tasks easier. By storing data in DBMS rather than as a collection of operating system files, we can use the DBMS's features to manage the data in a robust and efficient manner. As the volume of data and the number of users grow hundreds of gigabytes of data and thousands of users are common in current corporate database DBMS support becomes indispensable.

1.1 DATABASE ENVIRONMENT SYSTEM

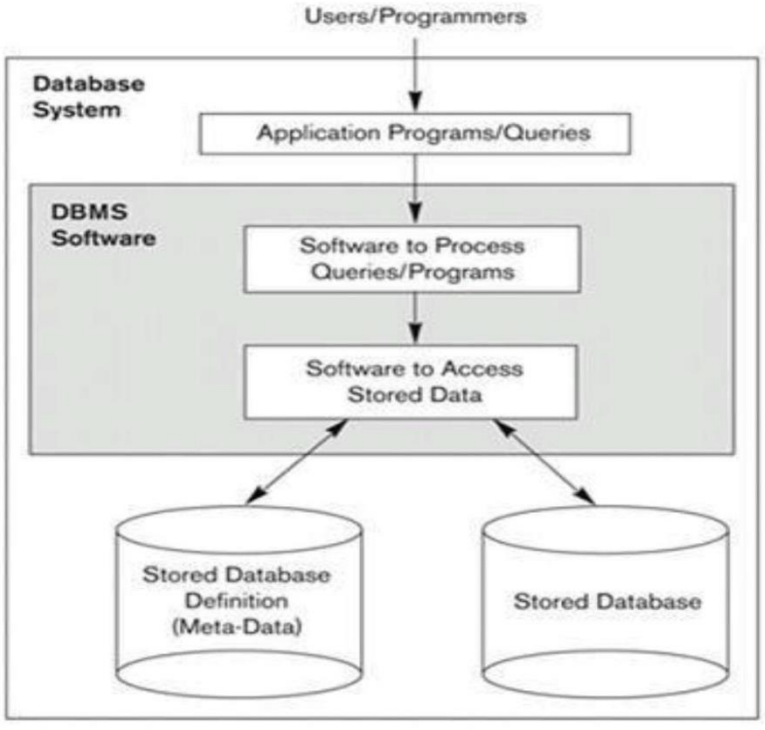

Fig 1.1: Simplified database environment system

A database management system (DBMS) is a collection of programs that enables users to create and maintain a database. The DBMS is a general-purpose software system that facilities the processes of defining, constructing, manipulating and sharing databases among

various users and applications. Defining a database involves specifying the data types, structures and constraints of the data to be stored in the database.

The database definition or description information is also stored by the DBMS in the form of a database catalog or dictionary, it is called Metadata. Constructing the database is the process of storing the data on some storage medium that is controlled by the DBMS. Manipulating a database includes functions such as querying the database to retrieve specific data, updating the database to reflect changes in the mini world and generating reports from the data. Sharing a database allows multiple users and programs to access the database simultaneously.

An application program accesses the database by sending queries or requests for data to DBMS. A query typically causes some data to be retrieved; a transaction may cause some data to be read and some data to be written into the database.

Other important functions provided by DBMS include protecting the database and maintaining it over a long period of time, protection includes system protectionagainst hardware or software malfunction and security protection against unauthorized or maliciousaccess. A typical large database may have a life cycle of many years, so the DBMS must be able to maintain the database system by allowing the system to evolve as requirements change over time.

It is not absolutely necessary to use general-purpose DBMS software to implement a computerized database. We could write our own set of programs to create and maintain the database, in effect creating our own special purpose DBMS software. In either case, whether we use a general-purpose DBMS or not we usually have deployed a considerable amount of complex software. In fact, most DBMSs are very complex software systems. Fig 1.1 shows a simplified database environment system

1.2 ADVANTAGES OF USING DBMS APPROACH

Using a DBMS to manage data has many advantages:

● **Data independence:** application program should not, ideally, be expected to details of data representation and storage, the DBMS provides an abstract view of the data

that hides such details.

● **Efficient Data Access:** A DBMS utilizes a variety of sophisticated techniques to store and retrieve data efficiently. This feature is especially important if the data is to be stored on an external device.

● **Data integrity and security:** if data is always accessed through DBMS, the DBMS can enforce integrity constraints. For example, before inserting salary information for an employee, the DBMS can check that the department budget is not exceeded. Also, it can enforce access controls that govern what data is visible to different classes of users.

● **Data Administration:** when several users share data, centralizing the administration of data can offer significant improvements. Experienced professionals who understand the nature of the data being managed, and how different groups of users use it, it can be responsible for organizing the data representation to minimize redundancy and for fine-tuning the storage of the data to make retrieval efficient.

● **Concurrent Access and Cash Recovery:** A DBMS schedules concurrent accesses to the data in such a manner that users can think of the data as being accessed by only one user at a time. Further, the DBMS protects users from the effects of system failures.

● **Reduced Application Development Time:** clearly, the DBMS supports important functions that are common to many applications accessing data in the DBMS. This, in conjunction with the high-level interface to data, facilities quick application development. DBMS applications are also likely to be more robust than a similar stand-alone application because many important tasks are handled by the DBMS.

1.3 ARCHITECTURE OF DATABASE

The Three-Schema Architecture

The goal of the three-schema architecture illustrated in the figure is to separate the user application from the physical database. In this architecture, schemas can be defined at the following three levels:

● The internal level has an internal schema, which describes the physical storage structure of the database. The internal schema uses a physical data model and describes the complete details of data storage and access paths for the database.

● The conceptual level has a conceptual schema, which describes the structure of the whole database for a community of users. The conceptual schema hides the details of physical storage structures and concentrates on describing entities, data types, relationships, user operations, and constraints. Usually, a representational data model is used to describe the conceptual schema when a database system is implemented. This implementation conceptual schema is often based on a conceptual schema design in a high-level data model.

● The external or view level includes a number of external schemas or user views. Each external schema describes the part of a database that a particular user group is interested in and hides the rest of the database from that user group. As in the previous level, each external schema is typically implemented using a representational data model, possibly based on external schema design in a high level data model.

Fig 1.2 shows the architecture of DBMS.

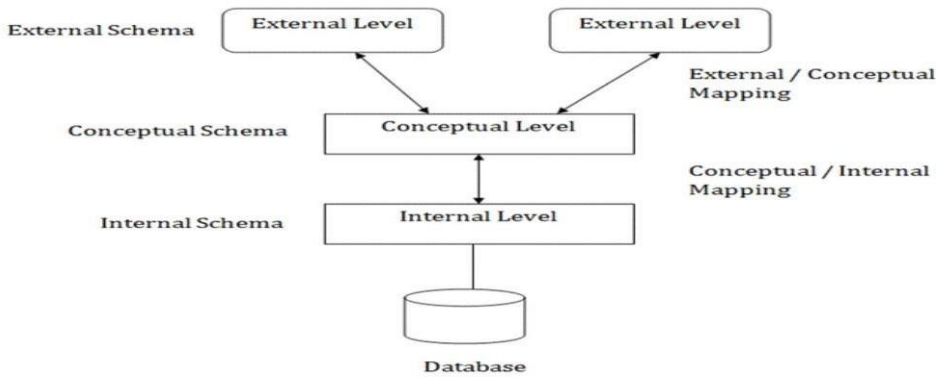

Fig 1.2: Architecture of DBMS

CHAPTER 2

INTRODUCTION TO PROJECT

RAILWAY RESERVATION MANAGEMENTSYSTEM

2.1 Brief description

● The mini-project entitled “Railway reservation management system“ is developed as apart of the fifth semester DBMS laboratory, for the partial fulfillment of the requirement for the BE(Information Science) course.

● Our website has various kinds of information that helps regarding booking of tickets via railways .

● Users will be able to search the train availability ,the exact fare ,the arrival and departuretime of the train and they can also book the ticket by using the debit ,credit or master card and after booking the ticket if the user want to cancel it then they can easily do it also.

Objectives

The objective of the online railway reservation management system
Project is to design software to fully automate the process of issuing a railway ticket.
That is:-

1. To create a database of the trains
2. To search the trains it’s arrival and departure time,distance between source and destination.
3. To check the availability of the ticket.
4. To calculate fare.
5. To book the ticket.
6. To cancel the ticket if necessary

2.2 Scope

● Railway passengers frequently need to know about their ticket reservation status, ticket availability on a particular train or for a place, train arrival or departure details, special trains etc.. Customer information centers at the railway stations are unable to serve such queries at peak periods.

● The number of the reservation counters available to the passengers and customers are very less.

● On most of the reservation systems there are long queues, so it takes a long time for any individual to book the ticket.As now there are no call centers facilities available to solve the queries of the passengers.

● The online railway ticket reservation system aims to develop a web application which aims at providing trains details, trains availability, as well as the facility to book ticket in online for customers.

● So, we thought of developing a web based application which would provide the users all these facilities from his terminal only as well as help them in booking their tickets. The Application was to be divided into two parts namely the user part , and the administrator part. And each of these has their corresponding features.We decided to give the name of the website “RAILWAY RESERVATION MANAGEMENT SYSTEM”.The online railway ticket reservation system is developed using ASP.NET with C# as the backend in the .NET Framework.

Analysis

Railway reservation management system is a online ticket booking website, which is capable of booking ticket and serach the train availavility . This website is mainly created to fulfil the following requirements, it comprises of the following properties:-

● A central database that will store all information.

● An online website that will provide real- time information about the availability of tickets their prices .

● Every registered user is able to view his booking id that has been made in his/her name.

● Every registered user can change his password any time he wants to change.

2.3 Table description

2.3.1 USERINFO

USERINFO table has a attributes f_name ,I_name ,email,password,gender, marital,dob,mobile,ques,ans is used shown in table 2.1

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	f_name	varchar(50)	latin1_swedish_ci		No	None		
2	l_name	varchar(50)	latin1_swedish_ci		No	None		
3	email	varchar(50)	latin1_swedish_ci		No	None		
4	password	varchar(20)	latin1_swedish_ci		No	None		
5	gender	varchar(10)	latin1_swedish_ci		No	None		
6	marital	varchar(10)	latin1_swedish_ci		No	None		
7	dob	varchar(20)	latin1_swedish_ci		No	None		
8	mobile	bigint(10)			No	None		
9	ques	varchar(100)	latin1_swedish_ci		No	None		
10	ans	varchar(100)	latin1_swedish_ci		No	None		

TABLE 2.1 STRUCTURE OF USER INFO

2.3.2 TRAIN LIST

TRAIN LIST table has a atrributes number,name,origin,destination,arrival,departure, mon,tue,wed,thu,fri,sat,sun,1A,2A,3A,SL,General and number is used as a primary key as shown in table 2.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extr
1	Number	int(6)			No	None		
2	Name	varchar(20)	latin1_swedish_ci		No	None		
3	Origin	varchar(20)	latin1_swedish_ci		No	None		
4	Destination	varchar(20)	latin1_swedish_ci		No	None		
5	Arrival	varchar(10)	latin1_swedish_ci		No	None		
6	Departure	varchar(10)	latin1_swedish_ci		No	None		
7	Mon	varchar(2)	latin1_swedish_ci		No	None		
8	Tue	varchar(2)	latin1_swedish_ci		No	None		
9	Wed	varchar(2)	latin1_swedish_ci		No	None		
10	Thu	varchar(2)	latin1_swedish_ci		No	None		
11	Fri	varchar(2)	latin1_swedish_ci		No	None		
12	Sat	varchar(2)	latin1_swedish_ci		No	None		
13	Sun	varchar(2)	latin1_swedish_ci		No	None		
14	1A	int(11)			No	None		
15	2A	int(11)			No	None		
16	3A	int(11)			No	None		
17	SL	int(11)			No	None		
18	General	int(11)			No	None		

TABLE 2.2 STRUCTURE OF TRAIN LIST

2.3.3 SEATS

SEATS table has the attribute train_no,train_name,doj,1A,2A,3A,AC,CC,SL is used shown in the table 2.3

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Train_No	int(11)			No	None		
2	Train_Name	varchar(20)	latin1_swedish_ci		No	None		
3	doj	date			No	None		
4	1A	int(11)			No	None		
5	2A	int(11)			No	None		
6	3A	int(11)			No	None		
7	AC	int(11)			No	None		
8	CC	int(11)			No	None		
9	SL	int(11)			No	None		

TABLE 2.3 STRUCTURE OF SEATS

2.3.4 INTERLIST

INTERLIST table has the attribute number,st1,st1arri,st2,st2arri,st3,st3arri,st4,st4arri, st5,st5arri,ori,oriarri,dest,desarri,name,mon,tue,wed,thu,fri,sat,sun as shown in Table 2.4

#	Name	Type	Collation	Attributes	Null	Default	Comments
1	Number	int(6)			Yes	NULL	
2	st1	varchar(10)	latin1_swedish_ci		Yes	NULL	
3	st1arri	varchar(10)	latin1_swedish_ci		Yes	NULL	
4	st2	varchar(10)	latin1_swedish_ci		Yes	NULL	
5	st2arri	varchar(10)	latin1_swedish_ci		Yes	NULL	
6	st3	varchar(10)	latin1_swedish_ci		Yes	NULL	
7	st3arri	varchar(10)	latin1_swedish_ci		Yes	NULL	
8	st4	varchar(10)	latin1_swedish_ci		Yes	NULL	
9	st4arri	varchar(10)	latin1_swedish_ci		Yes	NULL	
10	st5	varchar(10)	latin1_swedish_ci		Yes	NULL	
11	st5arri	varchar(10)	latin1_swedish_ci		Yes	NULL	
12	Ori	varchar(20)	latin1_swedish_ci		No	None	
13	Oriarri	varchar(10)	latin1_swedish_ci		No	None	
14	Dest	varchar(20)	latin1_swedish_ci		No	None	
15	Desarri	varchar(10)	latin1_swedish_ci		No	None	
16	Name	varchar(20)	latin1_swedish_ci		No	None	

17	Mon	varchar(2)	latin1_swedish_ci	No	None
18	Tue	varchar(2)	latin1_swedish_ci	No	None
19	Wed	varchar(2)	latin1_swedish_ci	No	None
20	Thu	varchar(2)	latin1_swedish_ci	No	None
21	Fri	varchar(2)	latin1_swedish_ci	No	None
22	Sat	varchar(2)	latin1_swedish_ci	No	None
23	Sun	varchar(2)	latin1_swedish_ci	No	None

TABLE 2.4 STRUCTURE OF INTERLIST

2.3.5 BOOKING

BOOKING table has theattributeuname,Tnumber,class,doj,DOB,fromstn,tostn, name,age,sex,status as shown in table 2.5

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	uname	varchar(15)	latin1_swedish_ci		No	None		
2	Tnumber	int(11)			No	None		
3	class	varchar(2)	latin1_swedish_ci		No	None		
4	doj	date			No	None		
5	DOB	date			No	None		
6	fromstn	varchar(15)	latin1_swedish_ci		No	None		
7	tostn	varchar(15)	latin1_swedish_ci		No	None		
8	Name	varchar(15)	latin1_swedish_ci		No	None		
9	Age	int(11)			No	None		
10	sex	varchar(10)	latin1_swedish_ci		No	None		
11	Status	varchar(20)	latin1_swedish_ci		No	None		

TABLE 2.5 STRUCTURE OF BOOKING

2.4 TRIGGERS

A trigger is a special type of stored procedure that automatically executes when a event occurs in the database server.

TABLE 2.6 Trigger in Train list

Chapter 3

DESIGN

3.1. ER Diagram

An entity–relationship model 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.

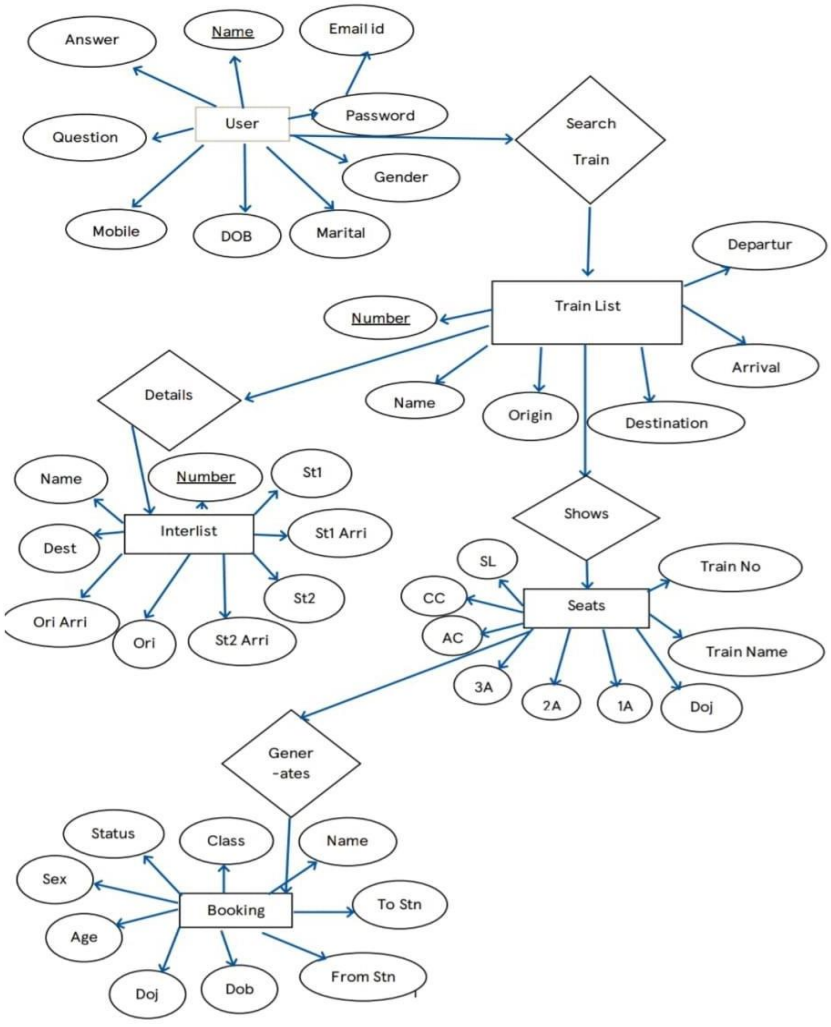

Figure 3.1 E R Diagram

This ER Diagram gives a brief idea about the relations existing between the tables and tells about the primary and the foreign keys being used in this Database.

3.2 NORMALIZATION

3.3.1 First Normal Form (1NF)

As the domain of all attributes of all relations in the database has atomic value and no tuples can have a set of these values, all relations are in 1NF.

3.3.2 Second Normal Form (2NF)

As there is no partial dependency in the database, i.e. all nonprime attributes of a relation are fully functionally dependent on the primary key of the relation schema, all relations are in 2NF.

3.3.3 Third Normal Form (3NF)

As all relations are in 2NF and no non-prime attribute of a relation schema is transitively dependent on the primary key, all relations are in 3NF.

3.3 SCHEMA DIAGRAM

A database schema can be represented in a visual diagram, which shows the database object and their relationship which represents the logical view of the database and how the relationships among them are represented.

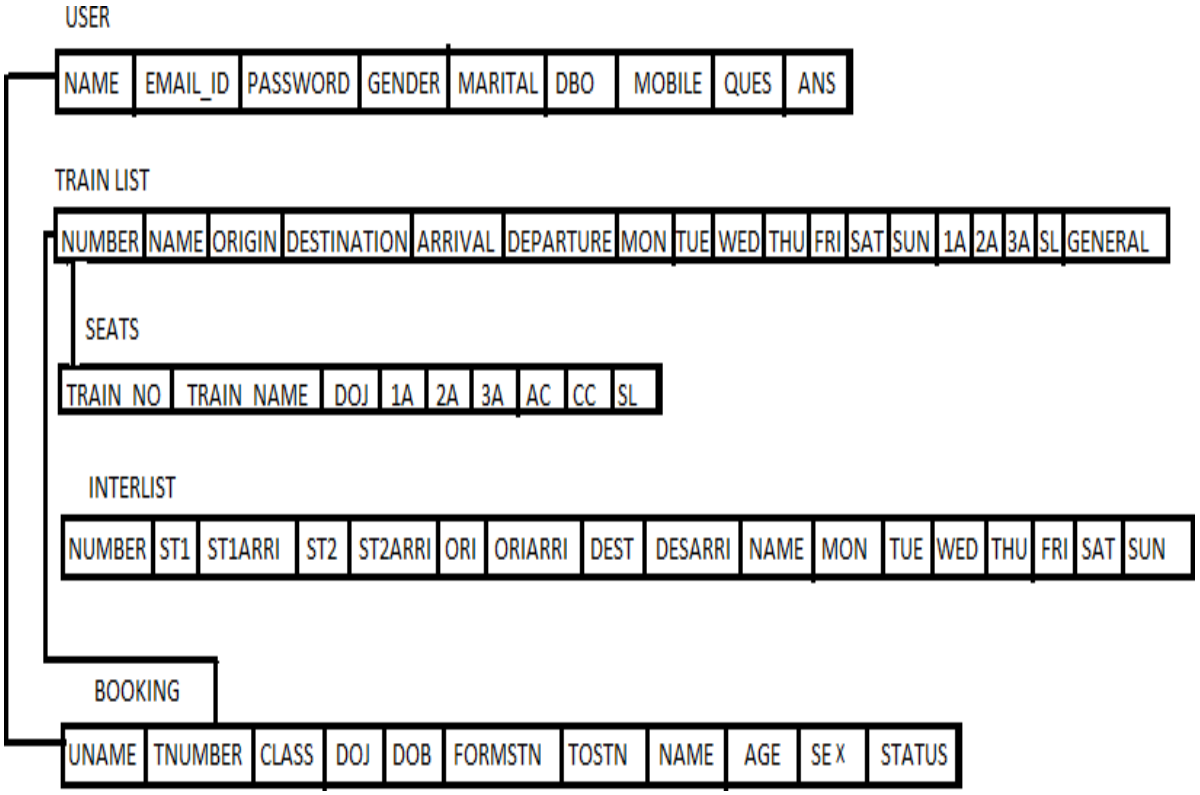

Figure 3.2 SCHEMA DIAGRAM

This Schema Diagram in Fig 3.2 represents different tables used

CHAPTER - 4

HARDWARE AND SOFTWARE REQUIREMENTS

4.1 FUNCTIONAL AND NON FUNCTIONAL REQUIREMENTS

4.1.1 Functional or specific Requirements

The required software is used for ordering food online. The system should satisfy the following requirements:

1. Logging into the system
2. Signup option
3. View Menu Details
4. Order Option
6. View User Orders
7. Logout option

4.1.2 Non functional Requirements

All of the application data is stored in an Oracle database, and therefore an Oracle Database must also be installed on the host computer. As with Apache2, this software is freely available and can be installed and run under most operating systems. The server hardware can be any computer capable of running both the web and database servers and handling the expected traffic. For a small scale restaurant that is not expecting to see much web traffic, an average personal computer may be appropriate. Once the site starts generating more hits, though, it will likely be necessary to upgrade to a dedicated host to ensure proper performance. The exact cut-offs will need to be determined through a more thorough stress testing of the system.

4.2 Hardware requirement

- A desktop or laptop with a proper internet connection
- 2 500GB or 60GB of the hard disk
- 3.4GB 2GB of the RAM
- 4 Windows 7 or 8 or 10 Operating system

4.3 Software requirements

4.3.1 Server side

1. Programming language: PHP 5.6.31
2. Web Server: Apache 2.4.27
3. Database: SQL 5.7.19

4.3.2 Client side

1. Programming language: JAVASCRIPT, HTML, CS
2. OS: windows7/8/10
3. MYSQL server

4.3.1.1 PHP

PHP is a server-side scripting language designed primarily for web development but also used as a general programming language PHP code may be embedded into HTML or HTML5 markup or it can be used in combination with various web template systems, web content management systems, and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server. The web server software combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated webpage.

4.3.1.2 WEB SERVER: APACHE

Apache is the most widely used web server software. Developed and maintained by Apache Software Foundation, Apache is open-source software available for free. It runs on 67% of all web servers in the world. It is fast, reliable, and secure. It can be highly customized to meet the needs of many different environments by using extensions and modules. Most WordPress hosting providers use Apache as their web server software. However, WordPress can run on other web server software as well.

4.3.1.3 HTML

HTML is an acronym that stands for HyperText Markup Language.
HyperText: HyperText simply means "Text within Text". A text has a link within it, is a hypertext. Every time you click on a word that brings you to a new webpage, you have clicked on a hypertext.
Markup language: A markup language is a programming language that is used to make text more interactive and dynamic. It can turn a text into images, tables, links, etc. An HTML document is made of many HTML tags and each HTML tag contains different content

.

4.3.1.4 JAVASCRIPT

Javascript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted

CONCLUSION

Our system can successfully give information on any train,find trains running between twostations ,book tickets and cancel tickets.This system could be used for official trainbooking.However several other features could be added like booking meals on trains etc.

Also payment gateways have to be implemented to make sure the transactions happen securely.

FUTURE ENHANCEMENT

- We can even further make it private and secured by implementing Log-in IDs and encrypting them with passwords.
- We can give away this software for more number of people and organizations to conduct a Beta Testing and based upon the results we can just make those changes and be assured of the application developed.
- We can make it more space and resource efficient so that this application consumes lesser RAM and ROM and battery power (if available).

REFERENCES

[1] https://github.com/samgakii123/

[2] http://php.net/

[3] https://www.http://en.wikipedia.org/wiki/PHP

[4] https://www.w3shools.com

APPENDIX 'A'- CODE SNIPPETS

A.1 DATABASE CONNECTION

The connect() / mysqli_connect() function opens a new connection to the MySQL server with the following syntax :

mysqli_connect(*host, username, password, dbname, port, socket*)

```
mysqli_select_db($conn,"$db_name") or die("cannot select db");
```

FIG A.1 DATABASE CONNECTION

A.2 INSERT QUERY

This query is used to insert a booking .

```
$sql="INSERT INTO $tbl_name(uname,Tnumber,class,doj,DOB,fromstn,tostn,Name,Age,sex,Status)
VALUES ('$uname','$num','$seat','$doj','$dob','$fromstn','$tostn','$name','$age','$sex','$status')";
$result=$conn->query($sql);
echo "$sql</br>";
if(!$result) die ($conn->error);
```

FIG A.2 INSERT QUERY

A.3 SELECT QUERY

In this query,all the details are fetched using SELECT* command

```
$uname=$_SESSION['name'];
$num=$_GET['tno'];
$seat= $_GET['selct'];
$name=$_GET['name1'];
$age=$_GET['age1'];
$sex=$_GET['sex1'];
$fromstn=$_GET[ 'fromstn'];
$tostn=$_GET['tostn'];
$doj=$_GET['doj'];
$dob=$_GET['dob'];
echo "..".$num."..".$name."..".$age."..".$sex."..".$seat."..";

$sql1="SELECT ".$seat." from seats_availability where Train_No='".$num."' and doj='".$doj."'";
$result1=$conn->query($sql1);
```

FIG A.3 SELECT QUERY

A.4 UPDATE QUERY

Here the update query is called to update the user of an already existing based on its name and category id,address etc respectively.

```
$sql="UPDATE $tbl_name SET l_name='$lname',email='$mail',gender='$gender',marital='$marital',dob='$dob',mobile='$mobile',ques='$ques',ans='$ans' WHERE f_name='$name'";
$result=mysqli_query($conn,$sql);


$_SESSION['error']==4;
```

FIG A.4 UPDATE QUERY

APPENDIX 'B'- SCREENSHOTS

B.1 HOME PAGE

This is the first window when the application is executed as shown in Fig B.1

FIG B.1 HOME PAGE

B.2 SIGN UP PAGE

The page allows admin to sign up to database as shown in Fig B.2


Indian Railways
HOME
FIND TRAIN
RESERVATION
PROFILE
BOOKING HISTORY
Signup
First Name *
Enter the First name
Last Name *
Enter the Last name
Email ID *
Enter the valid email id
Password *
Enter the password
Confirm Password *
Re-type the password
Gender *
MALE

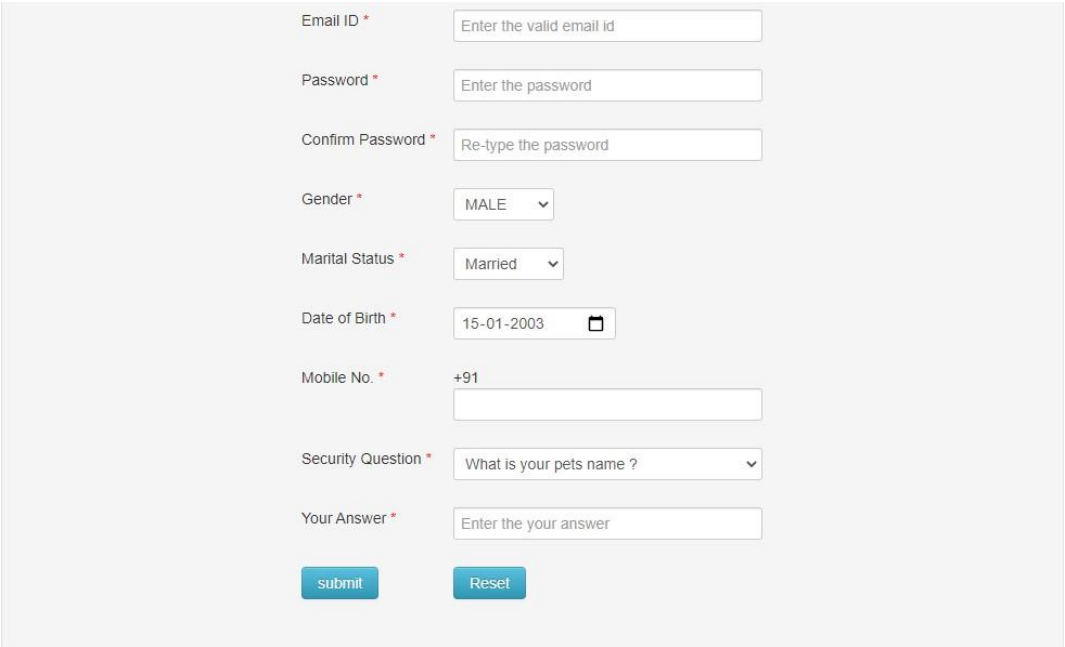

FIG B.2 SIGN UP PAGE

B.3 ADMIN LOGIN PAGE

This page allows admin to login and make changes to data base as shown in Fig B.3

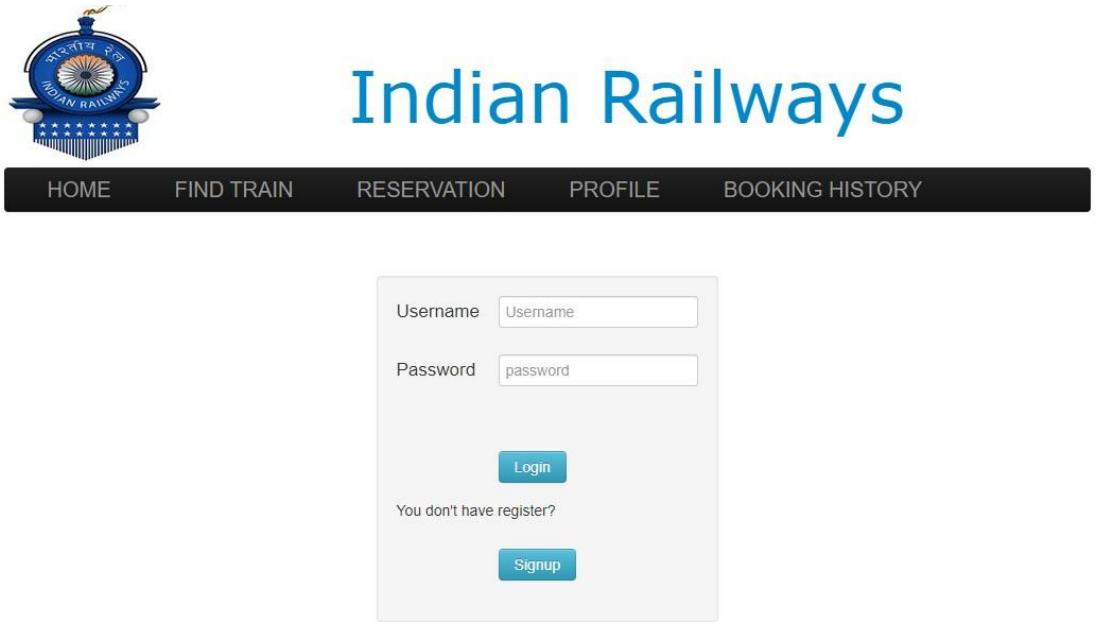

FIG B.3 ADMIN LOGIN PAGE

B.4 ADMIN FIND TRAIN PAGE

This page allows admin to find the train as shown in the FigB.4

FIG B.4 ADMIN FIND TRAIN PAGE

B.5 ADMIN BOOKING RESERVATION PAGE

This page allows admin to book the tickets as show in the Fig B.5

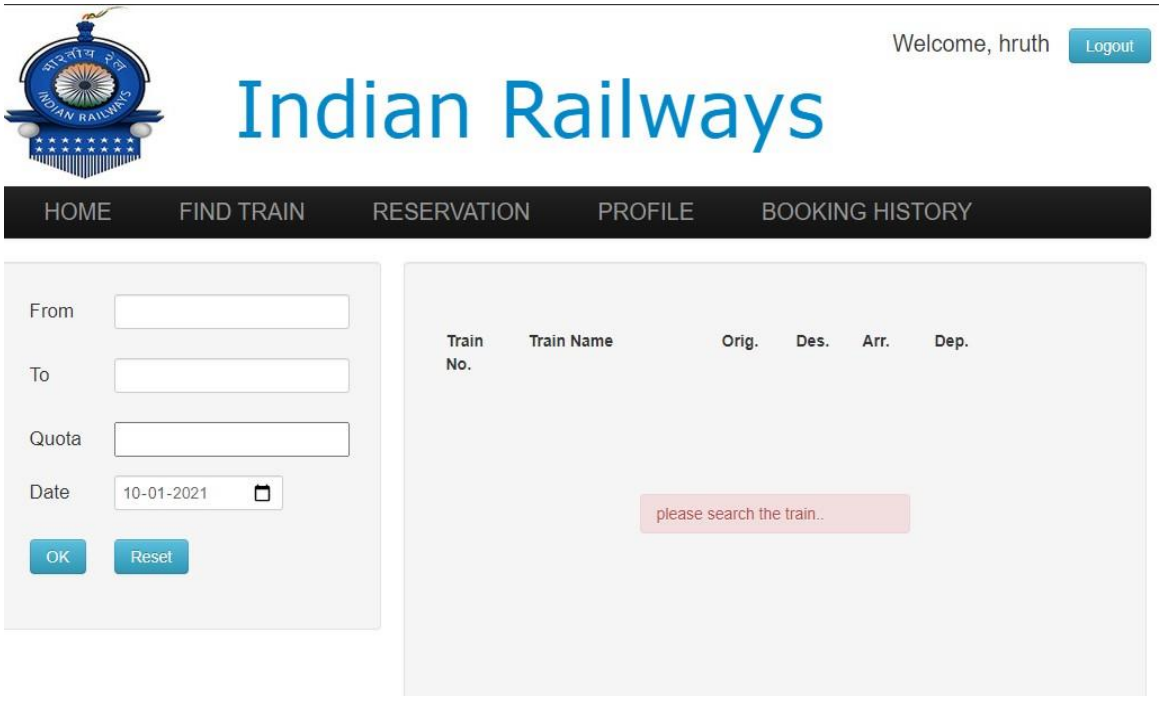

FIG B.5 ADMIN BOOKING RESERVATION PAGE

B.6 ADMIN BOOKING HISTORY

This page allows admin to see the Booking history as shown in the fig B.6

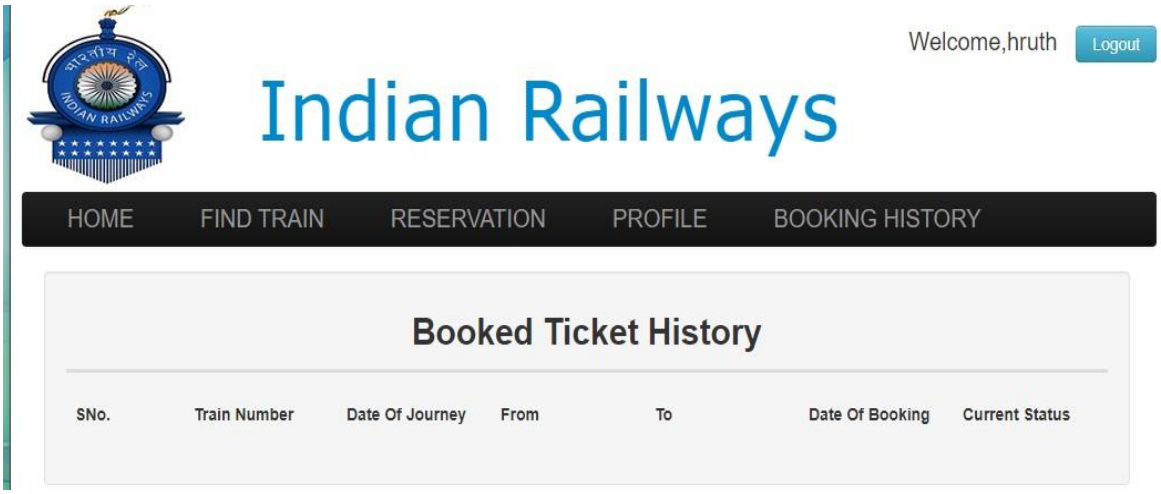

FIG B.6 ADMIN BOOKING HISTORY