A Project report on

Appointment Booking System

A Dissertation submitted to JNTU Hyderabad in partial fulfilment of academic requirements for the award of the degree

Bachelor of Technology

in

Computer Science and Engineering

Submitted by

PUNNA ABHISHEK

(23H55A0527)

SHEELAM ANVITHA

(23H55A0528)

SURAJ KUMAR SINGH

(23H55A0529)

Under the esteemed guidance of

Ms. Dipika Rath

(Asst Professor)

Department of Computer Science and Engineering

CMR COLLEGE OF ENGINEERING &TECHNOLOGY

(UGCAutonomous)

*Approved by AICTE *Affiliated to JNTUH *NAAC Accredited with A+ Grade

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD - 501401.

2022- 2026

CMR COLLEGE OF ENGINEERING &TECHNOLOGY

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD – 501401

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Mini Project I report entitled **"Appointment Booking System"** being submitted by P. Abhishek (23H55A0527), S. Anvitha (23H55A0528), Suraj Kumar Singh (23H55A0529) in partial fulfillment for the award **of Bachelor of Technology in** Computer Science and Engineering is a record of bonafide work Carried out his/her under my guidance and supervision.

The results embodies in this project report have not been submitted to any other University or Institute for the award of any Degree.

Ms. Dipika Rath   Dr . Siva Skandha Sanagala

Assistant Professor   **Associate Professor and HOD**
Dept. of CSE   **Dept. of CSE**

ACKNOWLEDGEMENT

P. Abhishek	(23H55AO527)
S. Anvitha	(23H55A0528)
Suraj Kumar Singh	(23H55A0529)

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

The Doctor Appointment System in Database Management System (DBMS) represents a cutting-edge solution designed to optimize and modernize the process of scheduling and managing appointments within healthcare facilities. This system addresses the challenges associated with traditional appointment systems, aiming to enhance patient experiences, streamline doctor scheduling, and integrate seamlessly with electronic health records (EHR).

The primary objectives of the proposed system include improving the overall efficiency of healthcare facilities by providing a user-friendly platform for patients and healthcare providers. Patients can conveniently schedule appointments with preferred doctors, receive automated reminders, and experience a seamless interface. For healthcare providers, the system facilitates better time management through efficient scheduling tools and immediate access to patient records during appointments.

CHAPTER 1
INTRODUCTION

CHAPTER 1
INTRODUCTION

1.1 PURPOSE

The purpose of this documentation is to provide a detailed overview of the Doctor Appointment Booking System implemented using Database Management System (DBMS). This system aims to streamline the process of scheduling and managing doctor appointments, enhancing the overall efficiency and patient experience within healthcare facilities.

The proposed Doctor Appointment Booking System in DBMS aims to revolutionize the way healthcare facilities manage appointments and improve patient access to medical services. By leveraging a robust database management system, the system aims to provide an efficient, secure, and user-friendly platform for patients and healthcare providers.

1.2 SCOPE

Doctor Appointment Booking System covers functionalities such as patient registration, doctor availability management, appointment scheduling, electronic health record integration, and administrative tools for healthcare staff. This system is designed to operate within a secure and compliant environment, ensuring the confidentiality and integrity of patient data.

CHAPTER 2
BACKGROUND WORK

CHAPTER 2
BACKGROUND WORK

2.1 ENTITIES

Doctor:

DoctorID (Primary Key)

Name

Specialization

Contact Information

Working Hours

Patient:

PatientID (Primary Key)

Name

Contact Information

Appointment:

AppointmentID (Primary Key)

DoctorID (Foreign Key referencing Doctor table)

PatientID (Foreign Key referencing Patient table)

AppointmentDateTime

Status (e.g., confirmed, canceled, completed)

Relationships:

A Doctor can have multiple Appointments.

A Patient can have multiple Appointments

CHAPTER 3
PROPOSED SYSTEM

CHAPTER 3
PROPOSED SYSTEM

3.1 OBJECTIVE OF PROPOSED MODEL

- **Improve Patient Experience:** Enhance the overall experience of patients by providing a seamless and user-friendly appointment scheduling process.

- **Optimize Doctor Scheduling:** Streamline the scheduling of doctor appointments to maximize the efficient use of healthcare professionals' time.

- **Integrate Electronic Health Records (EHR):** Seamlessly integrate with existing EHR systems to provide doctors with immediate access to patient records during appointments**.**

3.2 TECHNOLOGIES USED IN PROPOSED MODEL

MySQL

MySQL is the world's most popular open source database. According to DB-Engines, MySQL ranks as the second-most-popular database, behind Oracle Database. MySQL powers many of the most accessed applications, including Facebook, Twitter, Netflix, Uber, Airbnb, Shopify, and Booking.com.

Since MySQL is open source, it includes numerous features developed in close cooperation with users over more than 25 years. So it's very likely that your favorite application or programming language is supported by MySQL Database.

PL/SQL

PL/SQL is a combination of SQL along with the procedural features of programming languages. It was developed by Oracle Corporation in the early 90's to enhance the capabilities of SQL. PL/SQL is one of three key programming languages embedded in the Oracle Database, along with SQL itself and Java. This tutorial will give you great understanding on PL/SQL to proceed with Oracle database and other advanced RDBMS concepts.

3.3 STEPWISE IMPLEMENTATION AND CODING

- Create a database named db
- Now create a table appointment
- Insert the attributes of appointment table
- Now create a table of customer and insert the records of customer details
- Create a table appstatus to know about appointment is conformed/pending/cancelled
- Now insert the required records in appstatus table
- Create a table docavalible to know whether doctor available or not at which day and at which time he is available
- Insert the records of docavailable
- Now queries to retrieve the records which are conformed and cancelled and pending
- And query to know the availability of doctor twice a week

CODE

```
show databases;

create database db;

use db;

show databases;

show tables;

create table  app(id int(40) primary key, name varchar(20), age int(30));

insert into app(id,name,age)values(1,'aa',40);
```

```sql
select * from app;

create table Appointment(CustomerID int(20),  AppointmentTypeID int(20),
AppointmentDateTime datetime(6), Duration time(6),AppointmentStatusID
int(20));


INSERT INTO Appointment (CustomerID, AppointmentTypeID,
AppointmentDateTime, Duration, AppointmentStatusID)

VALUES (1, 2, '2022-01-01 14:00:00', 200, 1);

INSERT INTO Appointment (CustomerID, AppointmentTypeID,
AppointmentDateTime, Duration, AppointmentStatusID)

VALUES (2, 3, '2022-01-01 13:00:00', 100, 5);

INSERT INTO Appointment (CustomerID, AppointmentTypeID,
AppointmentDateTime, Duration, AppointmentStatusID)

VALUES (3, 4, '2022-01-01 16:00:00', 202, 12);

INSERT INTO Appointment (CustomerID, AppointmentTypeID,
AppointmentDateTime, Duration, AppointmentStatusID)

VALUES (4 ,5, '2022-03-01 09:00:00', 600, 1);

INSERT INTO Appointment (CustomerID, AppointmentTypeID,
AppointmentDateTime, Duration, AppointmentStatusID)

VALUES (5, 6, '2022-01-08 09:00:00', 200, 6);

INSERT INTO Appointment (CustomerID, AppointmentTypeID,
AppointmentDateTime, Duration, AppointmentStatusID)

VALUES (12, 7, '2022-01-17 14:00:00', 100, 7);

INSERT INTO Appointment (CustomerID, AppointmentTypeID,
AppointmentDateTime, Duration, AppointmentStatusID)

VALUES (11, 3, '2022-01-01 14:00:00', 200, 8);

INSERT INTO Appointment (CustomerID, AppointmentTypeID,
AppointmentDateTime, Duration, AppointmentStatusID)

VALUES (18, 2, '2022-01-01 14:00:00', 200, 11);

select * from Appointment;

create table customers(c_id int(10),Firstname varchar(12),Lastname
varchar(12));

INSERT INTO customers (C_id, Firstname, Lastname)

VALUES (1,'ram','k');
```

```sql
INSERT INTO customers (C_id, Firstname, Lastname)

VALUES (11,'ragu','l');

INSERT INTO customers (C_id, Firstname, Lastname)

VALUES (12,'abhi','s');

INSERT INTO customers (C_id, Firstname, Lastname)

VALUES (18,'sai','n');

INSERT INTO customers (C_id, Firstname, Lastname)

VALUES (2,'aalu','a');

INSERT INTO customers (C_id, Firstname, Lastname)

VALUES (2,'raju','c');

INSERT INTO customers (C_id, Firstname, Lastname)

VALUES (3,'varsha','d');

select * from customers;

create table doctor(id int(8), name varchar(15), phno int(15), email varchar(20),specialization varchar(15));

insert into doctor(id,name,phno,email,specialization)values(1,'dr-abc',899456,'ab@gmailcom', 2);

insert into doctor(id,name,phno,email,specialization)values(2,'dr-def',786450,'de@gmailcom', 1);

insert into doctor(id,name,phno,email,specialization)values(3,'dr-mno',556678,'mn@gmailcom', 6);

insert into doctor(id,name,phno,email,specialization)values(4,'dr-xyz',789960,'xx@gmailcom', 3);

insert into doctor(id,name,phno,email,specialization)values(5,'dr-klm',899456,'lm@gmailcom', 5);

insert into doctor(id,name,phno,email,specialization)values(6,'dr-hjk',896544,'hj@gmailcom', 4);

insert into doctor(id,name,phno,email,specialization)values(7,'dr-abc',899456,'ab@gmailcom', 1);

insert into doctor(id,name,phno,email,specialization)values(3,'dr-mno',556678,'mn@gmailcom', 6);

insert into doctor(id,name,phno,email,specialization)values(4,'dr-xyz',789960,'xx@gmailcom', 2);
```

```sql
select *from doctor;

create table speci(specialization int(15), name varchar(15));

insert into speci(specialization,name)values(1,'cardialagist');

insert into speci(specialization,name)values(2,'neurosurgon');

insert into speci(specialization,name)values(3,'dentist');

insert into speci(specialization,name)values(4,'artho');

insert into speci(specialization,name)values(5,'physio');

insert into speci(specialization,name)values(6,'ent');

insert into speci(specialization,name)values(7,'skin doc');

select * from speci;

create table appstatus(AppointmentStatusID int(10), status
varchar(15),CustomerID int(9));

insert into
appstatus(AppointmentStatusID,status,CustomerID)values(1,'conformed',1);

insert into
appstatus(AppointmentStatusID,status,CustomerID)values(5,'pending',2);

insert into
appstatus(AppointmentStatusID,status,CustomerID)values(12,'conformed',3);

insert into
appstatus(AppointmentStatusID,status,CustomerID)values(1,'cancelled',4);

insert into
appstatus(AppointmentStatusID,status,CustomerID)values(6,'conformed',5);

insert into
appstatus(AppointmentStatusID,status,CustomerID)values(7,'pending',12);

insert into
appstatus(AppointmentStatusID,status,CustomerID)values(8,'conformed',11);

insert into
appstatus(AppointmentStatusID,status,CustomerID)values(11,'cancelled',18);

select * from appstatus;

create table  docavailable(idd int(6), id int(8), day varchar(9), start time(6),
end time(6), open varchar(6), mintperpatient time(6));

insert into docavailable(idd,id,day,start  ,end , open,mintperpatient)values(1,
2, 'monday', 09,22,'yes',30);
```

```sql
insert into docavailable(idd,id,day,start ,end , open,mintperpatient)values(2, 3, 'tuesday', 07,23,'yes',20);

insert into docavailable(idd,id,day,start ,end , open,mintperpatient)values(3, 1, 'wednesday', 09,20,'yes',25);

insert into docavailable(idd,id,day,start ,end , open,mintperpatient)values(4, 5, 'monday', 09,21,'yes',30);

insert into docavailable(idd,id,day,start ,end , open,mintperpatient)values(5, 4, 'friday', 10,22,'yes',30);

insert into docavailable(idd,id,day,start ,end , open,mintperpatient)values(6, 7, 'monday', 09,22,'yes',30);

select * from docavailable;

# to display list of patients who has conformed there booking

select

c.Firstname as c_id1,

a.AppointmentDateTime as Appointmentdate,

b.status as  AppointmentStatus,

d.name as Doctorname,

c.c_id as c_id

from

customers c,

Appointment a,

appstatus b,

doctor d

where a.AppointmentStatusID=b.AppointmentStatusID

and  a.CustomerID=b.CustomerID

and a.AppointmentTypeID=d.id

and b.AppointmentStatusID=c.c_id

and b.status='conformed';


# to display patients who had cancelled there appointmenta

select

c.Firstname as c_id1,
```

```sql
a.AppointmentDateTime as Appointmentdate,
b.status as  AppointmentStatus,
c.c_id as cid,
d.name as Doctorname
from
customers c,
Appointment a,
appstatus b,
doctor d
where a.AppointmentStatusID=b.AppointmentStatusID
and  a.CustomerID=b.CustomerID
and a.AppointmentTypeID=d.id
and b.AppointmentstatusID=c.c_id
and b.status='cancelled';
# determine list of canceled patients

# specify specialization of doctor
select
a.name as doctor_name,
a.phno ,
a.email,
b.specialization,
b.name
from
doctor a,
speci b
where
a.specialization=b.specialization;

#doctor avaliable twice a week
```
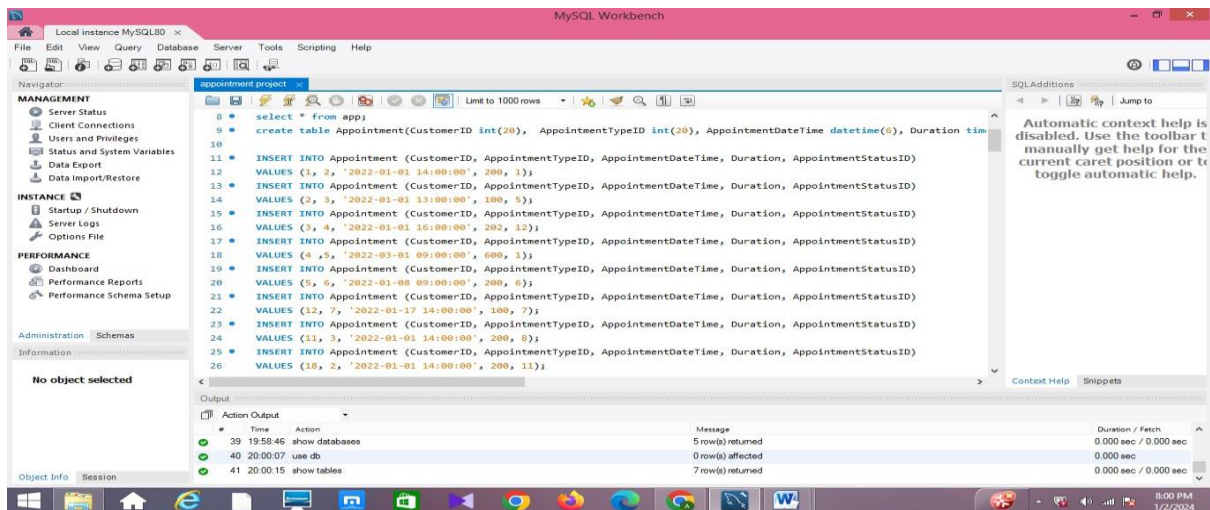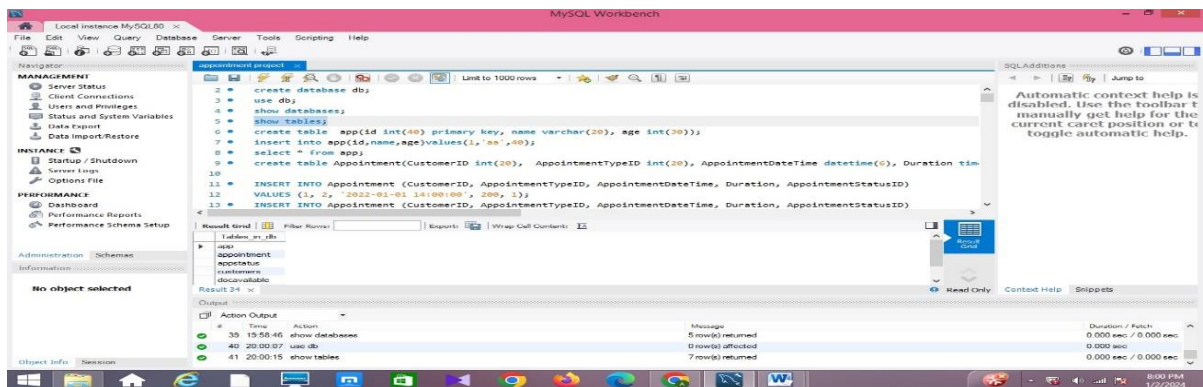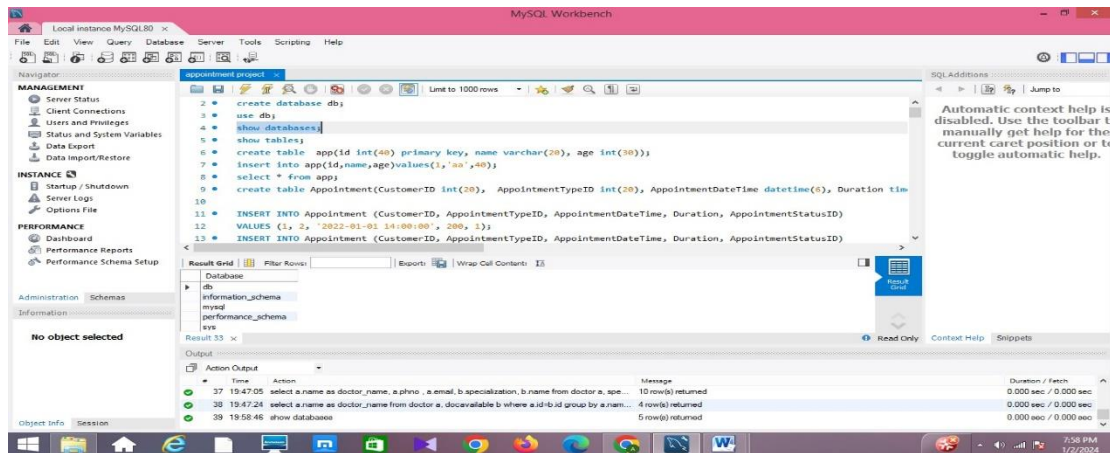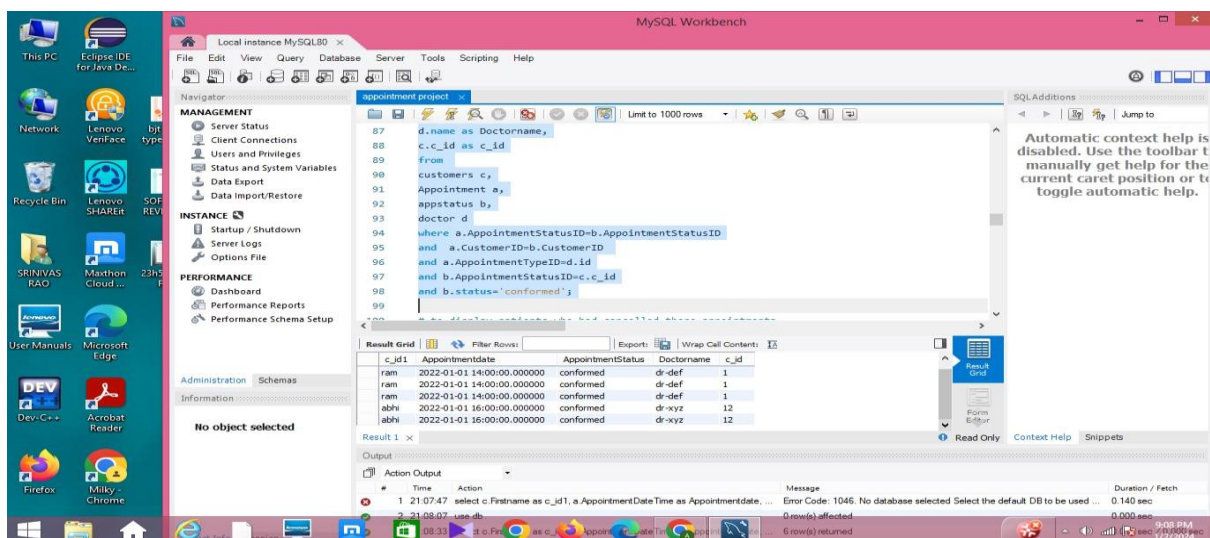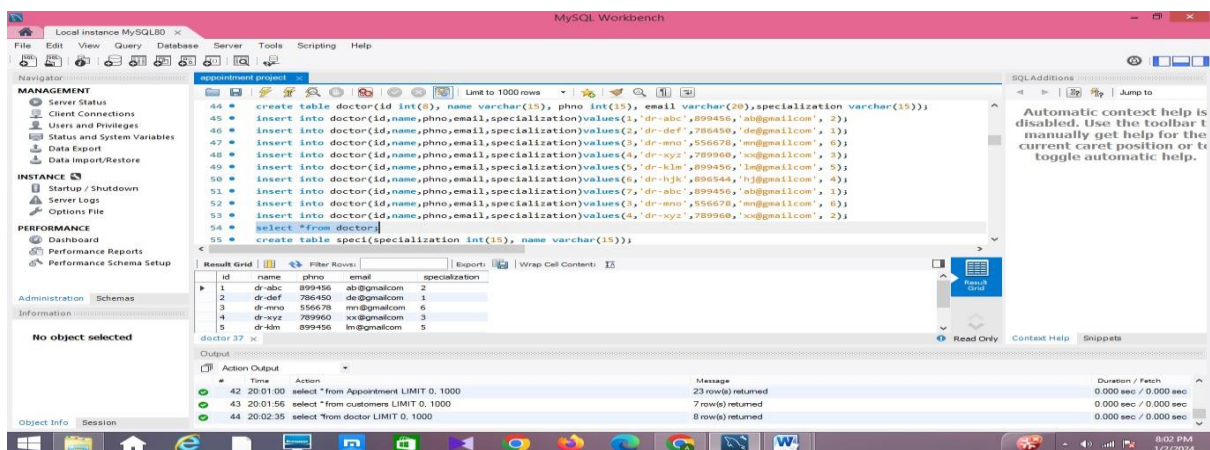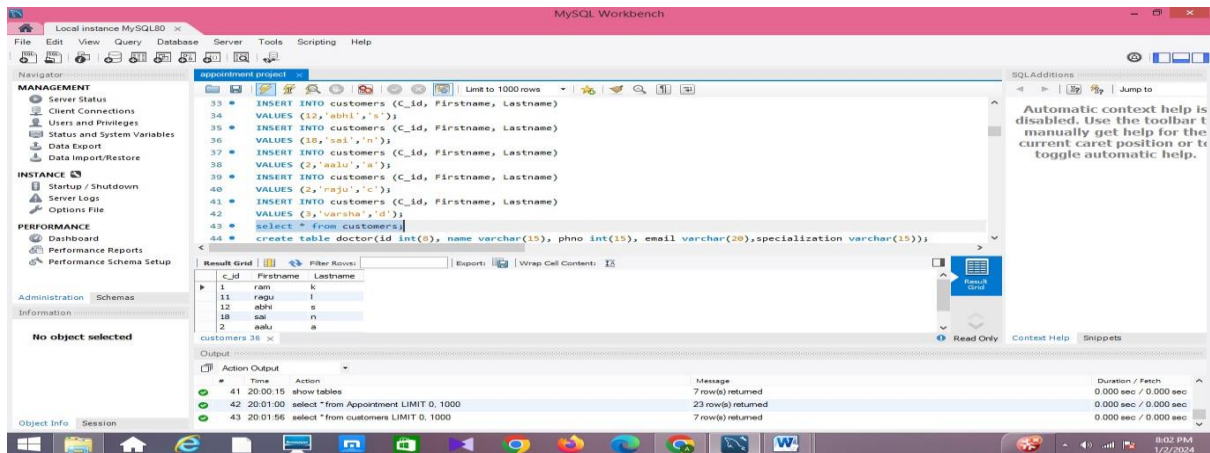
```sql
select
a.name as doctor_name
from
doctor a,
docavailable b
where a.id=b.id
group by a.name
having count(day)>1;
```

CHAPTER 4
RESULTS AND DISCUSSION

4.1 RESULT

MySQL Workbench — Local instance MySQL80 — appointment project

```
33   INSERT INTO customers (C_id, Firstname, Lastname)
34     VALUES (12,'abhi','s');
35   INSERT INTO customers (C_id, Firstname, Lastname)
36     VALUES (18,'sai','n');
37   INSERT INTO customers (C_id, Firstname, Lastname)
38     VALUES (2,'aalu','a');
39   INSERT INTO customers (C_id, Firstname, Lastname)
40     VALUES (2,'raju','c');
41   INSERT INTO customers (C_id, Firstname, Lastname)
42     VALUES (3,'varsha','d');
43   select * from customers;
44   create table doctor(id int(8), name varchar(15), phno int(15), email varchar(20),specialization varchar(15));
```

Result Grid:

c_id	Firstname	Lastname
1	ram	k
11	ragu	l
12	abhi	s
18	sai	n
2	aalu	a

Action Output:

#	Time	Action	Message	Duration / Fetch
41	20:00:15	show tables	7 row(s) returned	0.000 sec
42	20:01:00	select * from Appointment LIMIT 0, 1000	23 row(s) returned	0.000 sec / 0.000 sec
43	20:01:56	select * from customers LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec

MySQL Workbench — Local instance MySQL80 — appointment project

```
44   create table doctor(id int(8), name varchar(15), phno int(15), email varchar(20),specialization varchar(15));
45   insert into doctor(id,name,phno,email,specialization)values(1,'dr-abc',899456,'ab@gmailcom', 2);
46   insert into doctor(id,name,phno,email,specialization)values(2,'dr-def',786450,'de@gmailcom', 1);
47   insert into doctor(id,name,phno,email,specialization)values(3,'dr-mno',556678,'mn@gmailcom', 6);
48   insert into doctor(id,name,phno,email,specialization)values(4,'dr-xyz',789960,'xx@gmailcom', 3);
49   insert into doctor(id,name,phno,email,specialization)values(5,'dr-klm',899456,'lm@gmailcom', 5);
50   insert into doctor(id,name,phno,email,specialization)values(6,'dr-hjk',896544,'hj@gmailcom', 4);
51   insert into doctor(id,name,phno,email,specialization)values(7,'dr-abc',899456,'ab@gmailcom', 1);
52   insert into doctor(id,name,phno,email,specialization)values(3,'dr-mno',556678,'mn@gmailcom', 6);
53   insert into doctor(id,name,phno,email,specialization)values(4,'dr-xyz',789960,'xx@gmailcom', 2);
54   select *from doctor;
55   create table speci(specialization int(15), name varchar(15));
```

Result Grid:

id	name	phno	email	specialization
1	dr-abc	899456	ab@gmailcom	2
2	dr-def	786450	de@gmailcom	1
3	dr-mno	556678	mn@gmailcom	6
4	dr-xyz	789960	xx@gmailcom	3
5	dr-klm	899456	lm@gmailcom	5

Action Output:

#	Time	Action	Message	Duration / Fetch
42	20:01:00	select * from Appointment LIMIT 0, 1000	23 row(s) returned	0.000 sec / 0.000 sec
43	20:01:56	select * from customers LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec
44	20:02:35	select *from doctor LIMIT 0, 1000	8 row(s) returned	0.000 sec / 0.000 sec

MySQL Workbench — Local instance MySQL80 — appointment project

```
87   d.name as Doctorname,
88   c.c_id as c_id
89   from
90   customers c,
91   Appointment a,
92   appstatus b,
93   doctor d
94   where a.AppointmentStatusID=b.AppointmentStatusID
95   and  a.CustomerID=b.CustomerID
96   and a.AppointmentTypeID=d.id
97   and b.AppointmentStatusID=c.c_id
98   and b.status='conformed';
99
```

Result Grid:

c_id1	Appointmentdate	AppointmentStatus	Doctorname	c_id
ram	2022-01-01 14:00:00.000000	conformed	dr-def	1
ram	2022-01-01 14:00:00.000000	conformed	dr-def	1
ram	2022-01-01 14:00:00.000000	conformed	dr-def	1
abhi	2022-01-01 16:00:00.000000	conformed	dr-xyz	12
abhi	2022-01-01 16:00:00.000000	conformed	dr-xyz	12

Action Output:

#	Time	Action	Message	Duration / Fetch
1	21:07:47	select c.Firstname as c_id1, a.AppointmentDateTime as Appointmentdate, ...	Error Code: 1046. No database selected Select the default DB to be used ...	0.140 sec
2	21:08:07	use db	0 row(s) affected	0.000 sec

Screenshot 1 — MySQL Workbench

```
122    c.c_id as cid,
123    d.name as Doctorname
124    from
125    customers c,
126    Appointment a,
127    appstatus b,
128    doctor d
129    where a.AppointmentStatusID=b.AppointmentStatusID
130    and   a.CustomerID=b.CustomerID
131    and a.AppointmentTypeID=d.id
132    and b.status='pending';
133
```

Result Grid:

c_id1	Appointmentdate	AppointmentStatus	cid	Doctorname
raju	2022-01-17 14:00:00.000000	pending	2	dr-abc
raju	2022-01-01 13:00:00.000000	pending	2	dr-mno
raju	2022-01-01 13:00:00.000000	pending	2	dr-mno
raju	2022-01-01 13:00:00.000000	pending	2	dr-mno
varsha	2022-01-01 13:00:00.000000	pending	3	dr-mno

Action Output:

#	Time	Action	Message	Duration / Fetch
45	20:03:07	select * from speci LIMIT 0, 1000	8 row(s) returned	0.016 sec / 0.000 sec
46	20:05:14	select * from docavailable LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec
47	20:07:06	select c.Firstname as c_id1, a.AppointmentDateTime as Appointmentdate, b.status as Appoi...	49 row(s) returned	0.000 sec / 0.000 sec

---

Screenshot 2 — MySQL Workbench

```
101 •  select
102    c.Firstname as c_id1,
103    a.AppointmentDateTime as Appointmentdate,
104    b.status as  AppointmentStatus,
105    c.c_id as cid,
106    d.name as Doctorname
107    from
108    customers c,
109    Appointment a,
110    appstatus b,
111    doctor d
112    where a.AppointmentStatusID=b.AppointmentStatusID
113    and   a.CustomerID=b.CustomerID
114    and a.AppointmentTypeID=d.id
115    and b.AppointmentstatusID=c.c_id
116    and b.status='cancelled';
117    # determine list of canceled patients
118
119    # specify specialization of doctor
```

Action Output:

#	Time	Action	Message	Duration / Fetch
2	21:08:07	use db	0 row(s) affected	0.000 sec
3	21:08:33	select c.Firstname as c_id1, a.AppointmentDateTime as Appointmentdate, b.status as Appoi...	6 row(s) returned	0.016 sec / 0.000 sec
4	21:09:04	select c.Firstname as c_id1, a.AppointmentDateTime as Appointmentdate, b.status as Appoi...	2 row(s) returned	0.000 sec / 0.000 sec

---

Screenshot 3 — MySQL Workbench

```
132    #doctor avaliable twice a week
133    select
134    a.name as doctor_name
135    from
136    doctor a,
137    docavailable b
138    where a.id=b.id
139    group by a.name
140    having count(day)>1;
141
142
143
```

Result Grid:

doctor_name
dr-abc
dr-def
dr-mno
dr-xyz

Action Output:

#	Time	Action	Message	Duration / Fetch
47	20:07:06	select c.Firstname as c_id1, a.AppointmentDateTime as Appointmentdate, b.status as Appoi...	49 row(s) returned	0.000 sec / 0.000 sec
48	20:10:17	select a.name as doctor_name, a.phno , a.email, b.specialization, b.name from doctor a, spe...	10 row(s) returned	0.000 sec / 0.000 sec
49	20:11:45	select a.name as doctor_name from doctor a, docavailable b where a.id=b.id group by a.nam...	4 row(s) returned	0.000 sec / 0.000 sec

CHAPTER 5
CONCLUSION

CHAPTER 5
CONCLUSION

In conclusion, the implementation of a Doctor Appointment Booking System in a Database Management System (DBMS) offers numerous advantages for both healthcare providers and patients. The system streamlines the appointment scheduling process, enhances operational efficiency, and improves overall patient experience.

A well-designed Doctor Appointment Booking System in a DBMS not only streamlines administrative processes but also significantly improves the overall healthcare experience for both providers and patients. It stands as a testament to the integration of technology to enhance healthcare delivery and efficiency.

REFERENCES

REFERENCES

1. **Books on Database Management Systems:**

Look for books covering database design, implementation, and management, with a focus on healthcare systems. Some recommendations include "Database System Concepts" by Abraham Silberschatz, Henry F. Korth, and S. Sudarshan.

2. **Chatgpt**

Collected information from chatgpt and implemented our project

3. **Healthcare Information Systems:**

A Practical Approach for Healthcare Management" by Karen A. Wager, Frances W. Lee, and John P. Glaser.