Mini-project Report *on*

“Counselling Guidance System”

Submitted by

‘PANEL E

BATCH 3

NAME	PRN	ROLL
Sujal Dhanavade	1032201628	33
Harsch Jadhav	1032201695	35
Surojeet Ghosh	1032201731	41

Under the Guidance of

Prof. Shakti Kinger

At

MIT-WPU
॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

Dr. Vishwanath Karad
MIT WORLD PEACE UNIVERSITY | PUNE
TECHNOLOGY, RESEARCH, SOCIAL INNOVATION & PARTNERSHIPS

School of Computer Engineering and Technology

ABSTRACT

Counselling is a vital component of a person's life since it aids in the improvement of interpersonal relationships. Humans must cease ignoring this issue because it is essential for the development of mental wellness. The project "Counselling System," covers the gap in giving counselling in stressful situations. It answers the requirement to fill in the gaps in the traditional technique and make it more effective and immersive in this way.

TABLE OF CONTENTS

INTRODUCTION

Today, technology has been used to develop platforms that can digitally replicate real-life problems and improve the efficacy of dealing with them. Circumstances cannot be avoided; thus, it is critical to be prepared for any situation in which we can analyse and accomplish the task, regardless of how awful the situation is. As a result of today's events, a pandemic has struck many people's life, halting all sessions and physical contact. In this type of environment, the demand for counselling is growing since people are more likely to stay at home, causing them to make mistakes and experience other events that require counselling. However, because people are barred from making physical contact under the regulations that must be followed, it will be difficult for individuals to seek counselling. It is for this reason that the project "Counselling System" was implemented, which fills the gap in providing counselling in difficult situations.

PROBLEM DEFINITION

Problem Statement

- Counseling System Database Design

Overview

A well-designed database is critical for the efficient and accurate handling of information. A well-designed database will reduce the need for data entry and data correction, and it will aid in ensuring that data is easily accessible and usable by all parties. A well-designed database also makes data analysis and reporting easier to do. Data redundancy and inconsistency are common problems with poorly constructed databases, which make it difficult to manage and utilize the information stored within. An efficiently built database is more efficient and easier to use than an inefficiently designed database, in general.

TOOLS AND TECHNOLOGIES USED

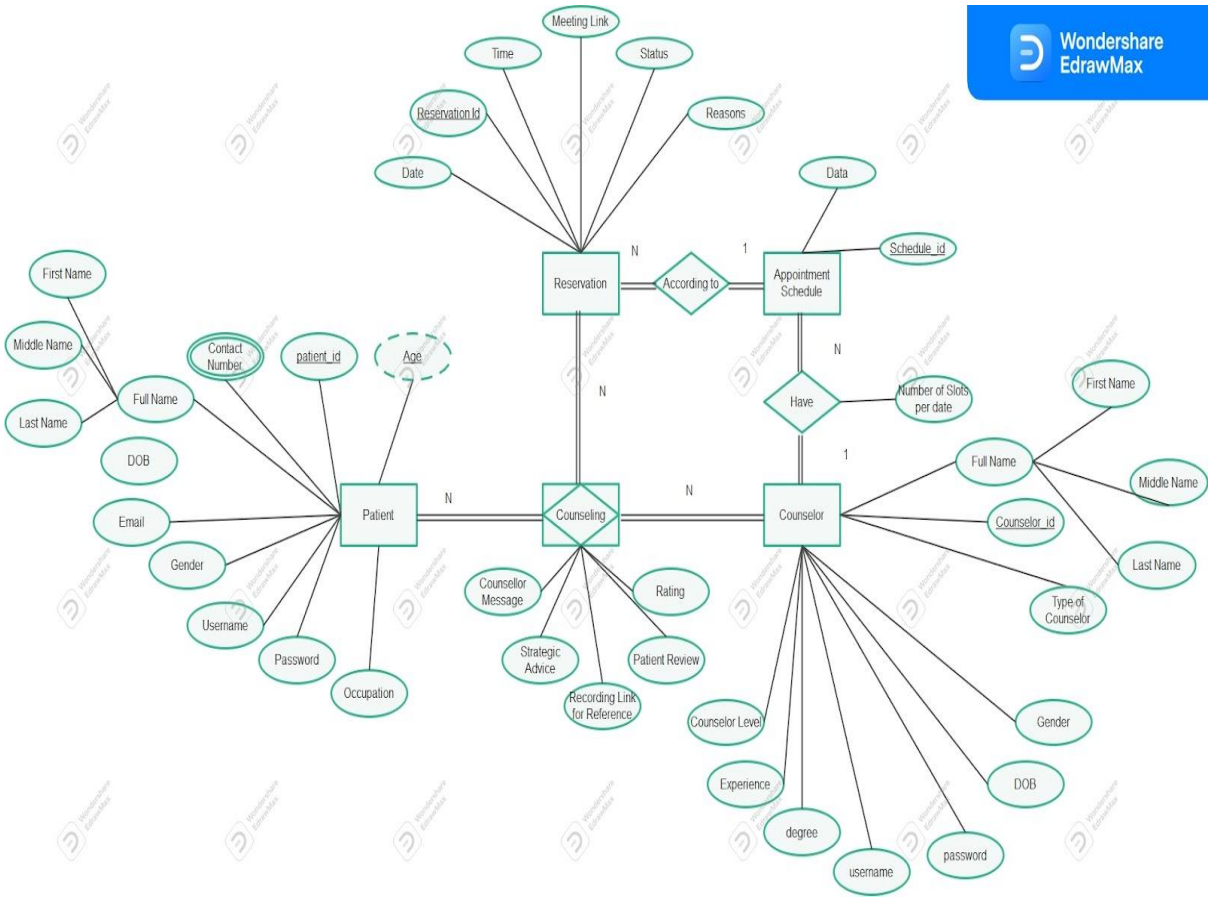
FRONTEND

- EJS
- BootStrap 5
- CSS
- JavaScript

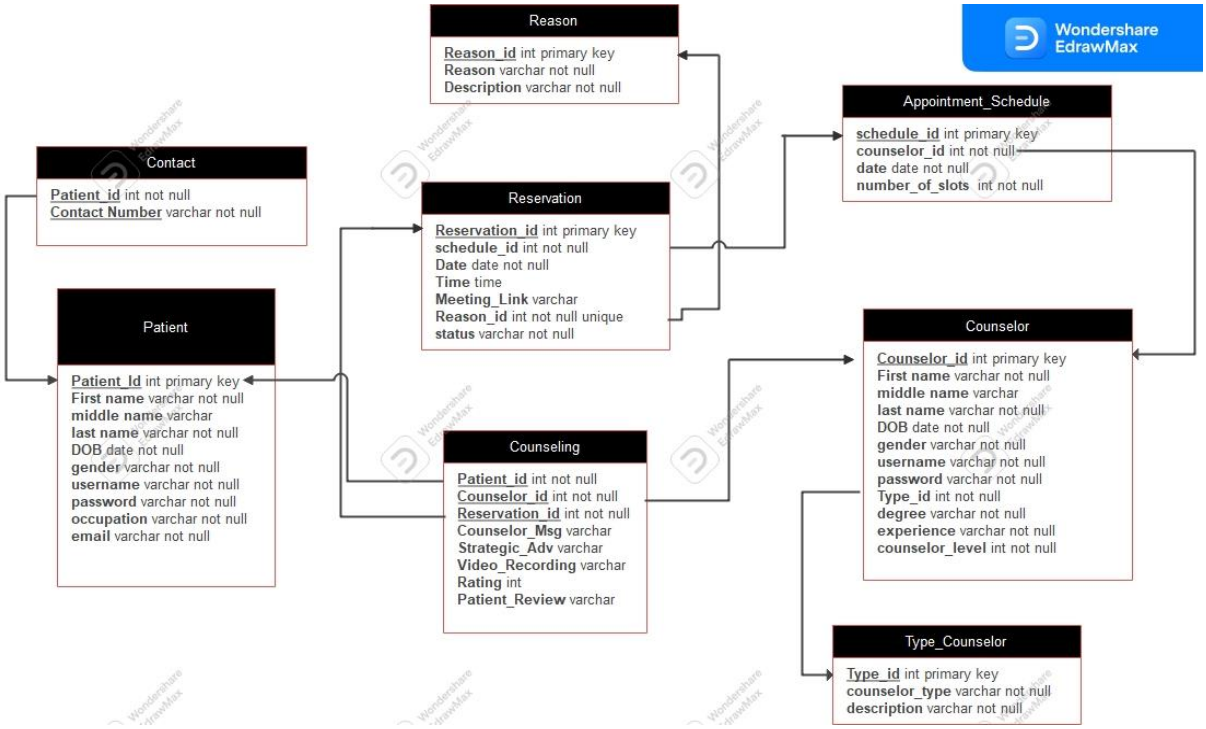
BACKEND

- NODEJS
- Express
- MySQL

ER DIAGRAM


Meeting Link
Time
Status
Reservation Id
Reasons
Date
Data
Schedule_id
Reservation
According to
Appointment Schedule
N
1
N
N
Have
Number of Slots per date
1
First Name
Middle Name
Last Name
Full Name
Contact Number
patient_id
Age
DOB
Email
Gender
Username
Password
Occupation
Patient
N
Counseling
N
Counselor
Counsellor Message
Rating
Strategic Advice
Patient Review
Recording Link for Reference
Full Name
First Name
Middle Name
Last Name
Counselor_id
Type of Counselor
Counselor Level
Experience
degree
username
password
Gender
DOB

DATABASE SCHEMA



DDL/ DML/ DCL COMMANDS

DDL

```
create database counselingSystem;
use counselingSystem;

mysql> create table patient(
    patient_id int auto_increment,
    first_name varchar(50) not null,
    middle_name varchar(50) default "-" not null,
    last_name varchar(50) not null,
    dob date not null,
    gender char(2) default "NA" not null,
    username varchar(50) not null unique,
    password varchar(50) not null,
    occupation varchar(50) not null,
    email varchar(100) not null unique,
    primary key(patient_id));
Query OK, 0 rows affected (0.01 sec)



mysql> create table contact(
     patient_id int not null,
     contact varchar(12) not null unique,
     foreign key(patient_id) references patient(patient_id),
     primary key(patient_id, contact));
Query OK, 0 rows affected (0.02 sec)


mysql> create table type_counselor(
     type_id int auto_increment primary key,
     counselor_type varchar(50) not null,
     description_type varchar(500) not null);
Query OK, 0 rows affected (0.01 sec)


mysql> create table counselor(
     counselor_id int auto_increment primary key,
     first_name varchar(50) not null,
     middle_name varchar(50) default "-" not null,
     last_name varchar(50) not null,
     dob date not null,
     gender char(2) default "NA" not null,
```

```
     username varchar(50) not null unique,
     password varchar(50) not null,
     type_id int not null,
     degree varchar(50) not null,
      email varchar(100) not null unique,
     experience_years int not null,
     counselor_level int not null,
     foreign key(type_id) references type_counselor(type_id));
Query OK, 0 rows affected (0.02 sec)


mysql> create table appointment_Sched(
     schedule_id int auto_increment primary key,
     counselor_id int not null,
     apt_date date not null,
     numberOfSlots int not null default 10 check(numberOfSlots >=
0),
     foreign key(counselor_id) references counselor(counselor_id));
Query OK, 0 rows affected (0.03 sec)



mysql> create table reason(
     reason_id int auto_increment primary key,
     reason varchar(50) not null,
     descript varchar(500) not null);
Query OK, 0 rows affected (0.01 sec)


mysql> create table reservation(
     reservation_id int auto_increment primary key,
     schedule_id int not null,
     date_reserve date not null,
     time_reserve time,
     meeting_link varchar(200),
     reason_id int not null unique,
     status varchar(20) not null,
     foreign key(schedule_id) references
appointment_Sched(schedule_id),
     foreign key(reason_id) references reason(reason_id));
Query OK, 0 rows affected (0.03 sec)


mysql> create table counseling(
     patient_id int,
     counselor_id int,
     reservation_id int unique,
     counselor_msg varchar(200),
     strategic_Adv varchar(200),
     video_record varchar(200),
     rating int check(rating >= 0 and rating <= 5),
     patient_review varchar(200),
     foreign key(counselor_id) references counselor(counselor_id),
     foreign key(patient_id) references patient(patient_id),
```

```
     foreign key(reservation_id) references
reservation(reservation_id),
     primary key(patient_id, counselor_id, reservation_id));
Query OK, 0 rows affected (0.02 sec)

mysql> alter table patient auto_increment = 10000;
Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> alter table type_counselor auto_increment = 10000;
Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> alter table counselor auto_increment = 10000;
Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> alter table appointment_Sched auto_increment = 10000;
Query OK, 0 rows affected (0.15 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> alter table reason auto_increment = 10000;
Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> alter table reservation auto_increment = 10000;
Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> alter table counseling auto_increment = 10000;
Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

DCL

```
create user 'admin'@'localhost' identified by 'mysql123';
create user 'observer'@'localhost' identified by 'mysql245';
create user 'register'@'localhost' identified by 'mysql567';
grant all on counselingsystem.* to 'admin'@'localhost';
grant select on counselingsystem.* to 'observer'@'localhost';
grant select, insert on counselingsystem.* to
'register'@'localhost';
```

TRIGGERS

```
create trigger after_Counseling_update after update on counseling for each
row
begin
if new.counselor_msg is not null and new.strategic_Adv is not null and
new.video_record is not null then
update reservation set status = "completed" where reservation_id =
new.reservation_id;
end if;
end$$
```

PLSQL PROCEDURE/ FUNCTION

FUNCTIONS

```
CREATE DEFINER=`root`@`localhost` FUNCTION `getCountUser`(given_username
varchar(50)) RETURNS int
    DETERMINISTIC
BEGIN
      declare count int;
    select count(*) into count from patient where username =
given_username;

RETURN count;
END


CREATE DEFINER=`root`@`localhost` FUNCTION `getCountemail`(given_Email
varchar(50)) RETURNS int
    DETERMINISTIC
BEGIN
      declare count int;
    select count(*) into count from patient where email = given_Email;
RETURN count;
END


CREATE DEFINER=`root`@`localhost` FUNCTION
`getCountCounselorEmail`(given_Email varchar(50)) RETURNS int
    DETERMINISTIC
BEGIN
      declare count int;
    select count(*) into count from counselor where email = given_Email;
RETURN count;
END


CREATE DEFINER=`root`@`localhost` FUNCTION
`getCountCounselorUser`(given_username varchar(50)) RETURNS int
    DETERMINISTIC
BEGIN
      declare count int;
    select count(*) into count from counselor where username =
given_username;
RETURN count;
```

```
END


CREATE DEFINER=`root`@`localhost` FUNCTION `registerUser`(fname
varchar(50), mname varchar(50), lname varchar(50), birthDate date, gender
char(2), uname varchar(50), pass varchar(50), occ varchar(50), em
varchar(100)) RETURNS int
    DETERMINISTIC
BEGIN
      declare isUserAvailable, isEmailAvailable int;
    select getCountUser(uname) into isUserAvailable;
    select getCountemail(em) into isEmailAvailable;
    if isUserAvailable = 0 && isEmailAvailable = 0 then
            begin
                  insert into patient values(null, fname, mname, lname,
birthDate, gender, uname, pass, occ, em);
            return 1;
            end;
      else
            return 0;
      end if;
END


CREATE DEFINER=`root`@`localhost` FUNCTION `registerCounselor`(fname
varchar(50), mname varchar(50), lname varchar(50), birthDate date, gender
char(2),typeC varchar(50), uname varchar(50), pass varchar(50), degree
varchar(50), em varchar(100), exp int) RETURNS int
    DETERMINISTIC
BEGIN
      declare isUserAvailable, isEmailAvailable int;
    declare typeId int;
    select type_id into typeId from  type_counselor where counselor_type =
typeC;
    select getCountCounselorUser(uname) into isUserAvailable;
    select getCountCounselorEmail(em) into isEmailAvailable;
    if isUserAvailable = 0 && isEmailAvailable = 0 then
            begin
                  insert into counselor values(null, fname, mname, lname,
birthDate, gender, uname, pass, typeId,  degree, em, exp, 0);
            return 1;
            end;
      else
            return 0;
      end if;
END



CREATE DEFINER=`root`@`localhost` FUNCTION `loginUser`(uname varchar(50),
pass varchar(50)) RETURNS int
    DETERMINISTIC
BEGIN
      declare isUserAvailable int;
```

```
    select count(*) into isUserAvailable from patient where username =
uname and password = pass;
    if isUserAvailable = 1 then
            return 1;
      else
            return 0;
      end if;
END


CREATE DEFINER=`root`@`localhost` FUNCTION `loginCounselor`(uname
varchar(50), pass varchar(50)) RETURNS int
    DETERMINISTIC
BEGIN
      declare isUserAvailable int;
    select count(*) into isUserAvailable from counselor where username =
uname and password = pass;
    if isUserAvailable = 1 then
            return 1;
      else
            return 0;
      end if;
END



CREATE DEFINER=`root`@`localhost` FUNCTION `checkRating`(counselor int)
RETURNS float
    DETERMINISTIC
BEGIN
      declare avgRating float;
    select avg(rating) into avgRating from counseling where counselor_id =
counselor;
    if avgRating is null then
            return 0;
      end if;
      RETURN avgRating;
END



CREATE DEFINER=`root`@`localhost` FUNCTION `addContact`(id int, contact
varchar(12)) RETURNS int
    DETERMINISTIC
BEGIN
      declare count int;
    select count(*) into count from patient where patient_id = id;
    if count = 1 then
            insert into contact values(id, contact);
        return 1;
      else
            return 0;
    end if;
```

```
END


CREATE DEFINER=`root`@`localhost` FUNCTION `getSlots`(id int,
date_appointment date) RETURNS int
    DETERMINISTIC
BEGIN
      declare slots int;
    declare sched_Id int;
    declare finished int default 0;
    declare continue handler for not found set finished = 1;
    select numberOfSlots, schedule_id into slots, sched_Id from
appointment_Sched where apt_date = date_appointment and counselor_id = id;
    if finished = 1 then
            return 0;
      elseif slots >= 1 then
            return sched_Id;
      else
            return -1;
      end if;
END


CREATE DEFINER=`root`@`localhost` FUNCTION `setSlot`(id int, appoint_date
date) RETURNS int
    DETERMINISTIC
BEGIN
    declare count int;
    declare sched_id int;
      select count(*) into count from appointment_Sched where counselor_id
= id and apt_date = appoint_date;
    if count = 0 then
            insert into appointment_Sched values(null, id, appoint_date,
10);
        select schedule_id into sched_id from appointment_Sched where
counselor_id = id and apt_date = appoint_date;
        return sched_id;
      else
            return 0;
      end if;
END


CREATE DEFINER=`root`@`localhost` FUNCTION `booking`(patient_id int,
counselor_id int, dateReserve date, reason varchar(50), des varchar(500))
RETURNS int
    DETERMINISTIC
BEGIN
      declare slots int;
    declare sched_Id int;
    declare reasonId int;
    declare reservationId int;
    select getSlots(counselor_id, dateReserve) into slots;
    if slots = -1 then
```

```
            return 0;
      elseif slots = 0 then
            select setSlot(counselor_id, dateReserve) into sched_Id;
        insert into reason values(null, reason, des);
        select last_insert_id() into reasonId;
        insert into reservation values(null, sched_Id, dateReserve, null,
null, reasonId, "No");
        select last_insert_id() into reservationId;
        insert into counseling values(patient_id, counselor_id,
reservationId, null, null, null, null, null);
        update appointment_Sched set numberOfSlots = numberOfSlots - 1
where schedule_id = sched_Id;
        return 1;
      else
            insert into reason values(null, reason, des);
        select last_insert_id() into reasonId;
        insert into reservation values(null, slots, dateReserve, null,
null, reasonId, "No");
        select last_insert_id() into reservationId;
        insert into counseling values(patient_id, counselor_id,
reservationId, null, null, null, null, null);
        update appointment_Sched set numberOfSlots = numberOfSlots - 1
where schedule_id = slots;
        return 1;
      end if;
END
```

PROCEDURE

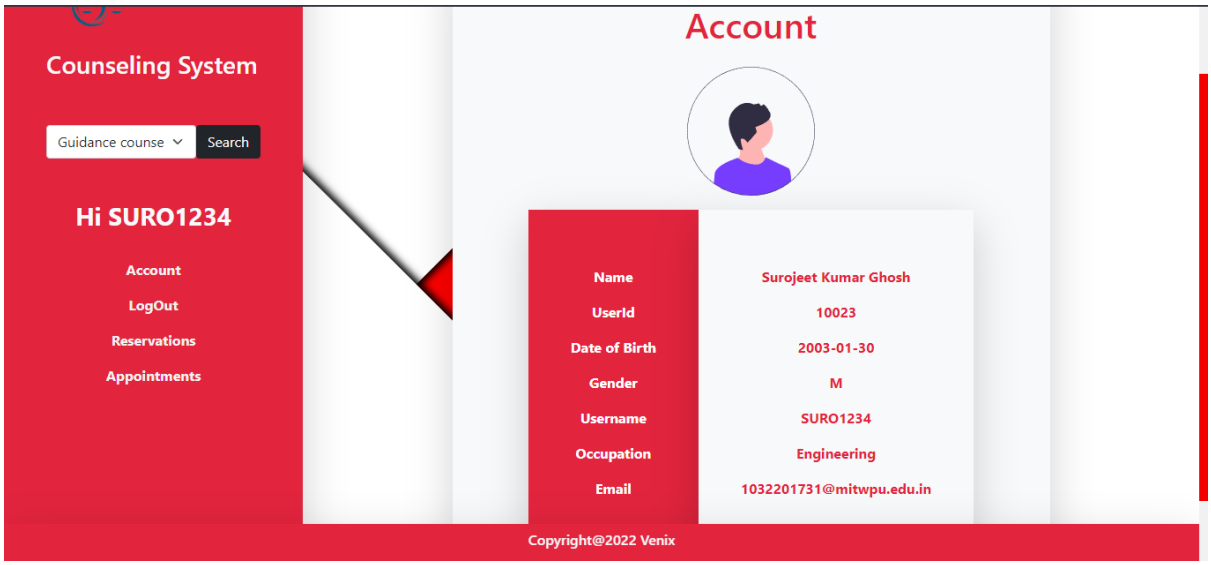
```
CREATE DEFINER=`root`@`localhost` PROCEDURE `giveReview`(in reserveId
int,  in Patientrating int, in review varchar(200))
BEGIN
      update counseling set patient_review = review, rating =
Patientrating where reservation_id = reserveId;
END
```

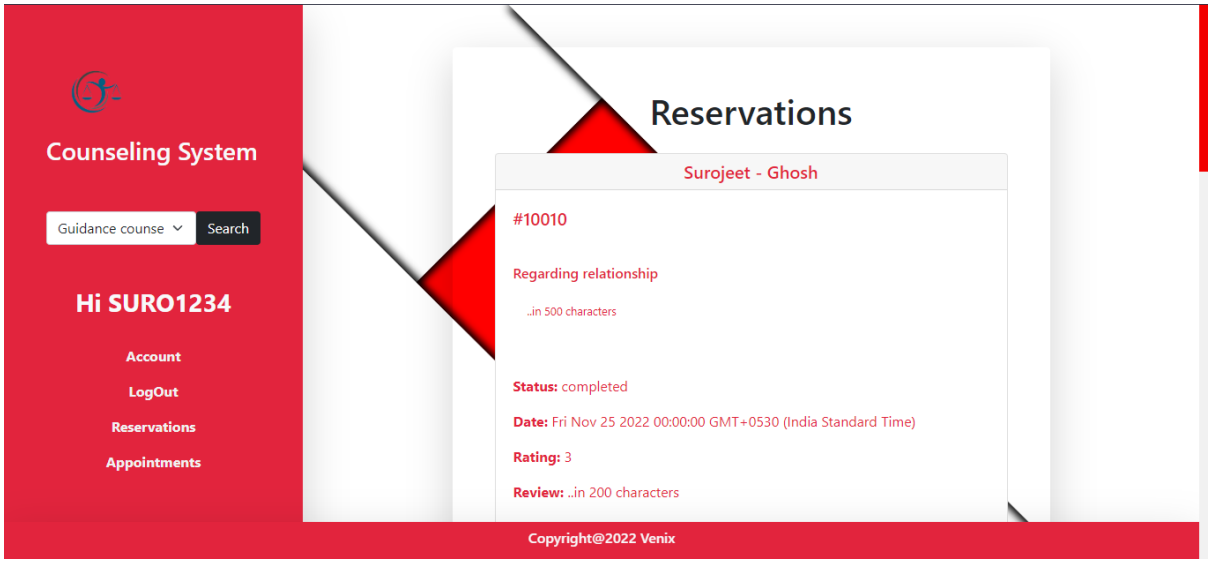
```
CREATE DEFINER=`root`@`localhost` PROCEDURE `setLink`(in id int,in link
varchar(200),in timeReserve time)
BEGIN
      declare timeR time;
    declare mlink varchar(200);
    select time_reserve, meeting_link into timeR, mlink from reservation
where reservation_id = id;
    if timeR is null and mlink is null then
            update reservation set time_reserve = timeReserve,
meeting_link = link where reservation_id = id;
      end if;
END
```

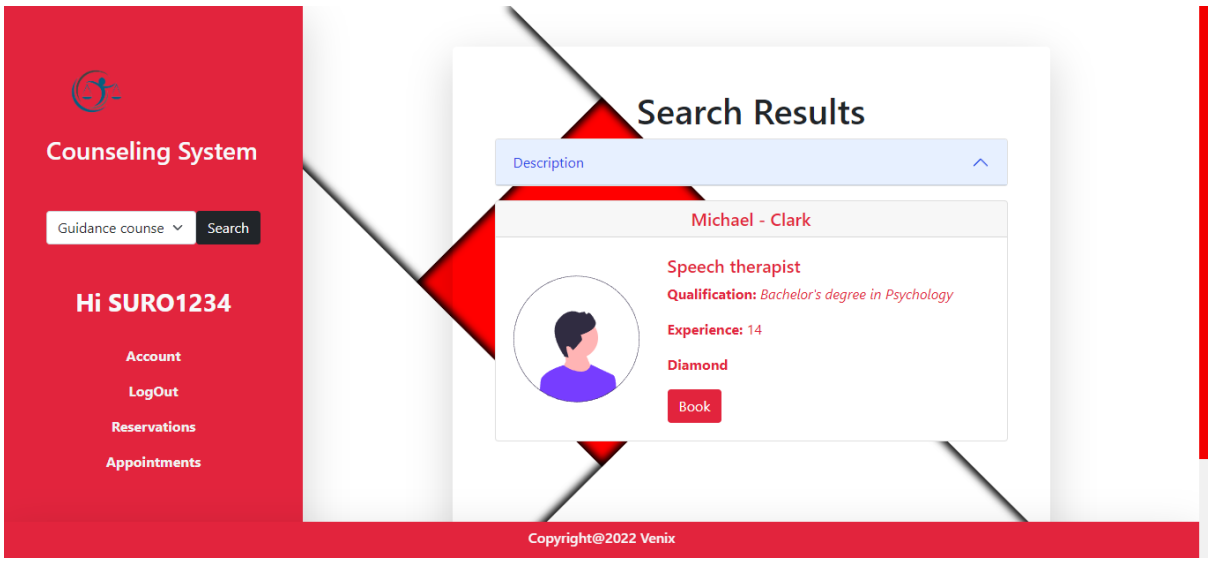
```
CREATE DEFINER=`root`@`localhost` PROCEDURE `setAdvice`(in patientId
int,in counselorId int,in reserveId int,in msg varchar(200),in adv
varchar(200),in record varchar(200))
BEGIN
      declare Cmsg varchar(200);
    declare Sadv varchar(200);
    declare rlink varchar(200);
    select counselor_msg, strategic_Adv,video_record  into Cmsg, Sadv,
rlink from counseling where patient_id = patientId and counselor_id =
counselorId and reservation_id = reserveId;
    if Cmsg is null and Sadv is null and rlink is null then
            update counseling set counselor_msg = msg, strategic_Adv =
adv, video_record = record where patient_id = patientId and counselor_id =
counselorId and reservation_id = reserveId;
      end if;

END



CREATE DEFINER=`root`@`localhost` PROCEDURE `updateLevel`()
BEGIN
      declare finished int default 0;
    declare avgRate float;
    declare id int;
    declare levelCounselor int;
    declare cursor_Counselor cursor for select counselor_id from
counselor;
    declare continue handler for not found set finished = 1;
    open cursor_Counselor;
    startLoop: loop
            begin
                  fetch cursor_Counselor into id;
            if finished = 1 then
                        leave startLoop;
                  end if;
            select checkRating(id) into avgRate;
            if avgRate >= 0 and avgRate < 1 then
                        set levelCounselor = 1;
                  elseif avgRate >= 1 and avgRate < 2 then
                        set levelCounselor = 2;
                  elseif avgRate >= 2 and avgRate < 3 then
                        set levelCounselor = 3;
                  elseif avgRate >= 3 and avgRate < 4 then
                        set levelCounselor = 4;
                  elseif avgRate >= 4 and avgRate <= 5 then
                        set levelCounselor = 5;
                  end if;
            update counselor set counselor_level = levelCounselor where
counselor_id = id;
        end;
      end loop startLoop;
    close cursor_Counselor;
END
```

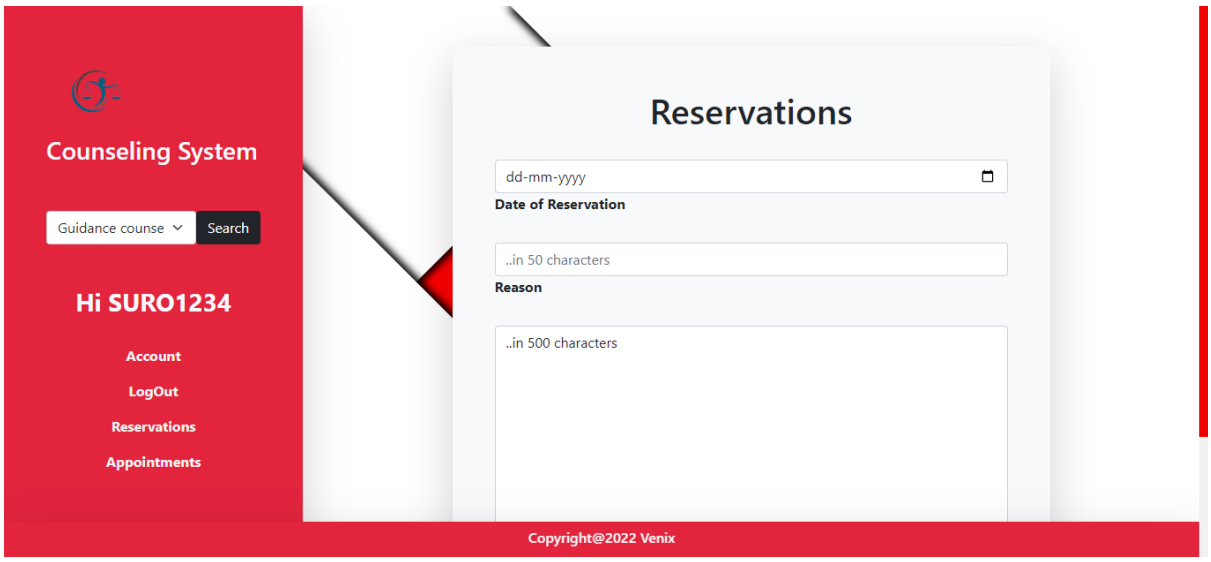
FRONTEND GUI SCREENSHOTS

User's POV

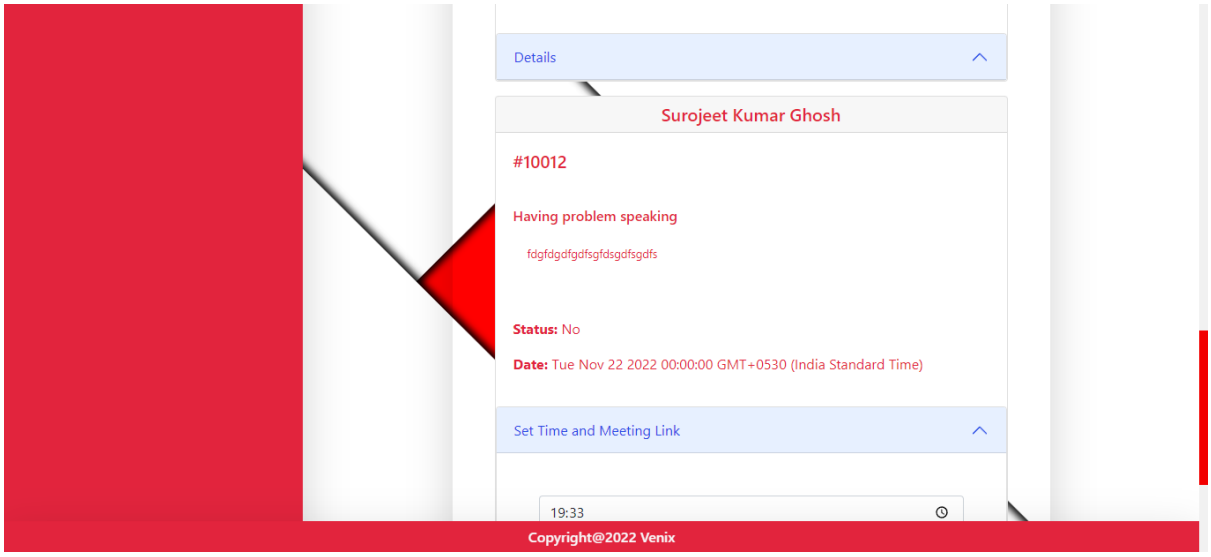


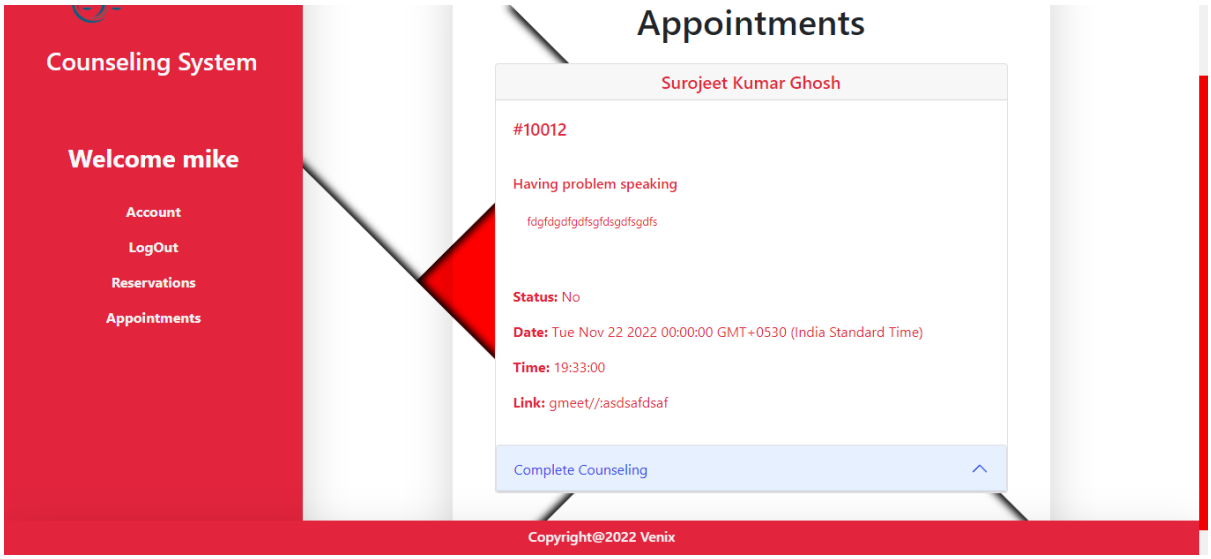

Counseling System
Guidance counse
Search
Hi SURO1234
Account
LogOut
Reservations
Appointments
Reservations
Surojeet - Ghosh
#10010
Regarding relationship
..in 500 characters
Status: completed
Date: Fri Nov 25 2022 00:00:00 GMT+0530 (India Standard Time)
Rating: 3
Review: ..in 200 characters
Copyright@2022 Venix

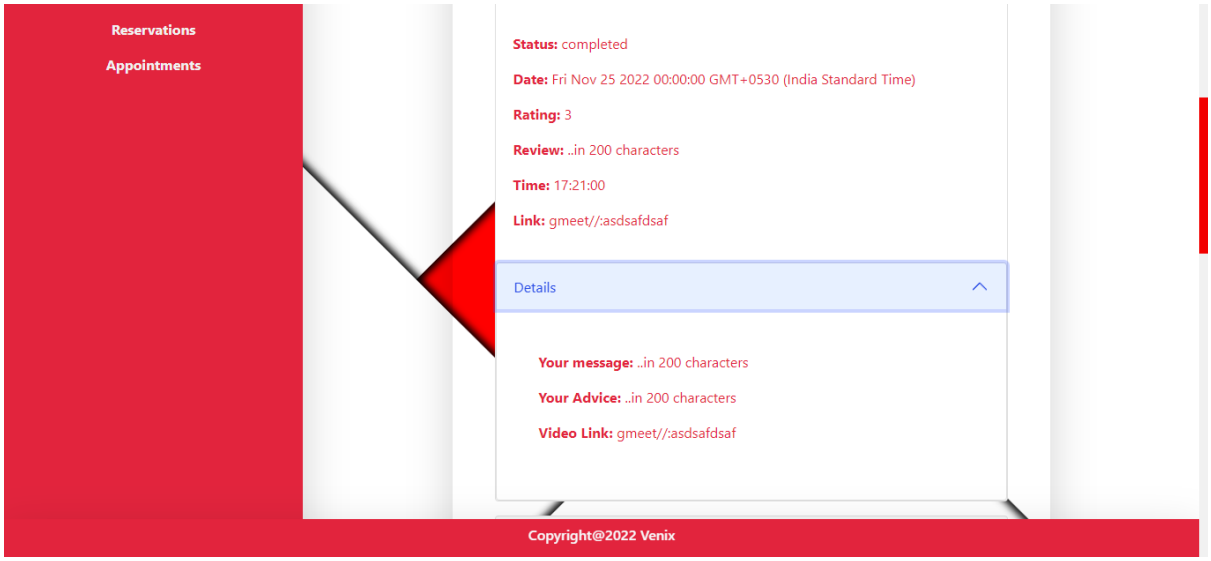

Counseling System
Guidance counse
Search
Hi SURO1234
Account
LogOut
Reservations
Appointments
Search Results
Description
Michael - Clark
Speech therapist
Qualification: Bachelor's degree in Psychology
Experience: 14
Diamond
Book
Copyright@2022 Venix


Counseling System
Guidance counse
Search
Hi SURO1234
Account
LogOut
Reservations
Appointments
Reservations
dd-mm-yyyy
Date of Reservation
..in 50 characters
Reason
..in 500 characters
Copyright@2022 Venix

Counsellor’s POV






Reservations
Appointments
Status: completed
Date: Fri Nov 25 2022 00:00:00 GMT+0530 (India Standard Time)
Rating: 3
Review: ..in 200 characters
Time: 17:21:00
Link: gmeet//:asdsafdsaf
Details
Your message: ..in 200 characters
Your Advice: ..in 200 characters
Video Link: gmeet//:asdsafdsaf
Copyright@2022 Venix

COUNCLUSION

Hence, we developed a Web Application for a counselling system using MySQL and Web Technologies Stack.