

National University Of Computer & Emerging Sciences CFD Campus

CL2006 - Database Systems - Lab (Spring 2024)

| Lab Instructor | Amna Waheed |
|----------------|--|
| Department | Computer Science |
| Group Members | 22F-3440 Tayyab Khan 22F-8766 Muhammad Azam |

Hospital Management System (PHASE # 3)

Phase 3
Database Systems
12/5/24

DETAILED SCHEMA

Users Table

- UserID (PK):
- Username: The user's chosen username.
- Password
- RoleID (FK): Links to the Roles table to specify if the user is an Admin, Employee, or Patient.

Profiles Table

- ProfileID (PK): Unique identifier for a profile.
- UserID (FK): Links to the Users table.
- FirstName
- LastName
- Email
- PhoneNumber
- Address
- DateOfBirth

Employees Table

- EmployeeID (PK)
- ProfileID (FK): Links to the Profiles table.
- DepartmentID (FK): Links to the Departments table.
- HireDate:
- Position: The role or job title within the hospital.
- Salary

Patients Table

- PatientID (PK): Unique identifier for a patient.
- ProfileID (FK): Links to the Profiles table.
- MedicalHistory: Textual description of the patient's medical history.
- CONSTRAINT NAMEFK PROFILES USERS

Departments Table

- DepartmentID (PK): Unique identifier for a department.
- DepartmentName: Name of the department.
- Description: A brief description of the department.

Appointments Table

- AppointmentID (PK): Unique identifier for an appointment.
- PatientID (FK): Links to the Patients table.
- EmployeeID (FK): Links to the Employees table.
- ScheduledTime: The start time of the appointment.
- EndTime: The end time of the appointment.
- Purpose: The reason for the appointment.
- Status: Status of the appointment (Scheduled, Completed, Cancelled).

Roles Table

- RoleID (PK): Unique identifier for a role.
- RoleName: The name of the role (e.g., Admin, Doctor, Nurse, Receptionist).

CustomerSupport Table

- SupportTicketID (PK): Unique identifier for a support ticket.
- SubmittedByUserID (FK): Links to the Users table (who submitted the ticket).
- IssueDescription: Detailed description of the issue or inquiry.
- DateSubmitted: The date the ticket was submitted.
- ResolutionStatus: Current status of the ticket (Open, In Progress, Resolved).

Prescriptions Table

- PrescriptionID (PK): Unique identifier for a prescription.
- PatientID (FK): Links to the Patients table.
- EmployeeID (FK): Links to the Employees table (prescribing doctor).
- Medication: Name and dosage of the medication prescribed.
- UsageInstructions: Instructions on how the patient should use the medication.
- DatePrescribed: The date when the medication was prescribed.

CustomerSupport Table

- SupportTicketID (PK): Unique identifier for a support ticket.
- SubmittedByUserID (FK): Links to the Users table (who submitted the ticket).
- IssueDescription: Detailed description of the issue or inquiry.
- DateSubmitted: The date the ticket was submitted.
- ResolutionStatus: Current status of the ticket (Open, In Progress, Resolved).

Tasks Table

- TaskID (PK): Unique identifier for a task.
- AssignedToEmployeeID (FK): Links to the Employees table.
- · Details: The description of the task.
- Deadline: Due date for the task completion.
- Status: Current status of the task (Assigned, In Progress, Completed).

Feedback Table

- FeedbackID (PK): Unique identifier for a feedback entry.
- GivenByUserID (FK): Links to the Users table.
- Subject: The subject or title of the feedback.
- Message: The detailed message of the feedback.
- Rating: Numerical rating given.
- DateGiven: The date the feedback was provided.

Sequences:

Sequences that I created in my database for the Primary key of the tables, sequences helped me not take care of the PK increments, whenever any insertion is made in any of the table, their primary keys are automatically generated based upon the sequences which is incremented by 1 and also starting from 1.

USER_ID_SEQ
SALARY_SEQ

PROFILE_ID_SEQ

PATIENT_SEQ

EMPLOYEE_ID_SEQ

APPOINTMENTID_SEQ

Triggers:

| USERS_TRIGGER |
|--------------------------------|
| PROFILES_TRIGGER |
| PATIENT_BIR |
| TRG_BEFORE_INSERT_APPOINTMENTS |

Details of the triggers are as follows:

Most of the triggers are fired for insertions to automatically assign the PrimaryKeys to each table.

What happens is whenever an insertion is done from the form, the new details are stored with each of them. For the handling of condition that when a new employee is added, his userID, profileID should also be made is also handled using the insertions query itself by using a procedure in the C# code where we implement the query.

| TRIGGER_NAME | TRIGGER_TYPE | TRIGGERING _EVENT | TABLE_NAME | TRIGGER_BODY |
|--|--------------------|----------------------|------------------|---|
| USERS_TRIGGE R | BEFORE EACH ROW | INSERT | USERS | BEGIN SELECT user_id_seq.NEXTVAL INTO :NEW.UserID FROM dual; END; |
| PROFILES_TRIG GER | BEFORE EACH ROW | INSERT | PROFILES | BEGIN SELECT profile_id_seq.NEXTVAL INTO :NEW.ProfileID FROM dual; END; |
| PATIENT_BIR | BEFORE EACH ROW | INSERT | PATIENTS | BEGIN SELECT patient_seq.NEXTVAL INTO :new.PATIENTID FROM dual; END; |
| TRG_BEFORE_IN SERT_APPOINTM ENTS | BEFORE EACH ROW | INSERT | APPOINTMENT S | BEGIN :NEW.AppointmentID := AppointmentID_SEQ.NEXTVAL; END; |

Handling Deletions:

FK_EMPLOYEEID_APPOINTMENTS R APPOINTMENTS

| FK_PATIENTID_APPOINTMENTS | R | APPOINTMENTS |
|----------------------------|---|--------------|
| FK_DEPARTMENT_ID_EMPLOYEES | R | EMPLOYEES |
| FK_PROFILE_ID | R | PATIENTS |
| FK_PROFILE_EMPLOYEE | R | EMPLOYEES |
| FK_PROFILES_USERS | R | PROFILES |
| FK_ROLE | R | USERS |

These are all the rules that are used for the deletions so far:

SELECT constraint_name, table_name, delete_rule
FROM user_constraints
WHERE constraint_type = 'R'
AND table_name IN ('APPOINTMENTS', 'EMPLOYEES', 'PATIENTS', 'PROFILES', 'USERS');

| CONSTRAINT_NAME | TABLE_NAME | DELETE_RULE |
|----------------------------|--------------|-------------|
| FK_PATIENTID_APPOINTMENTS | APPOINTMENTS | CASCADE |
| FK_EMPLOYEEID_APPOINTMENTS | APPOINTMENTS | SET NULL |
| FK_PROFILE_EMPLOYEE | EMPLOYEES | CASCADE |
| FK_DEPARTMENT_ID_EMPLOYEES | EMPLOYEES | SET NULL |
| FK_PROFILE_ID | PATIENTS | CASCADE |
| FK_PROFILES_USERS | PROFILES | CASCADE |
| FK_ROLE | USERS | NO ACTION |

Employee Deletion Procedure:

I made this procedure so whenever an employee is deleted from employees table, the profile and user is also deleted from their respective tables.

PROCEDURE DELETE_EMPLOYEE (p_EmployeeID IN EMPLOYEES.EMPLOYEEID%TYPE) IS

v_ProfileID PROFILES.PROFILEID%TYPE;

v UserID USERS.USERID%TYPE;

BEGIN

-- Find the associated ProfileID

SELECT ProfileID INTO v_ProfileID FROM EMPLOYEES WHERE EmployeeID = p_EmployeeID;

-- Delete the employee record

DELETE FROM EMPLOYEES WHERE EmployeeID = p_EmployeeID;

-- Delete the profile record

DELETE FROM PROFILES WHERE ProfileID = v_ProfileID;

-- Delete the user record

DELETE FROM USERS WHERE UserID = (SELECT UserID FROM PROFILES WHERE ProfileID);

-- Commit the transaction

COMMIT;

END DELETE_EMPLOYEE;

FORMS IN C#

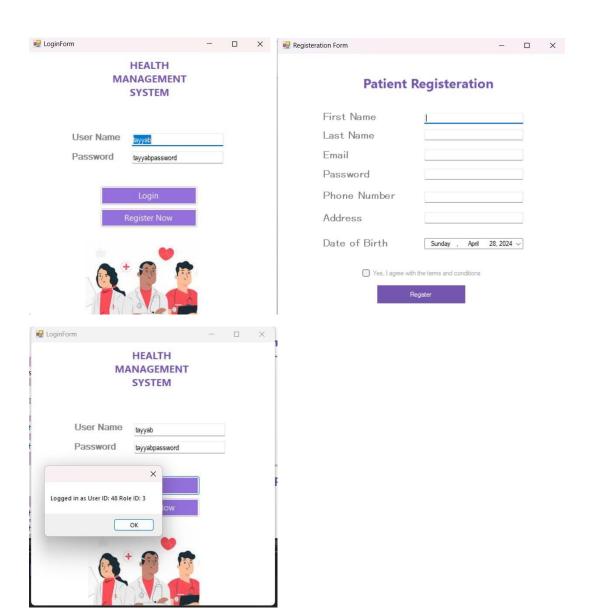
Form1.cs (LOGIN FORM)

Textboxes:

- Username
- Password

Buttons:

- LoginButton (proceeds to specific form of patient, employee or admin based upon role automatically)
- PatientRegister (proceeds to specific form for patient registration)



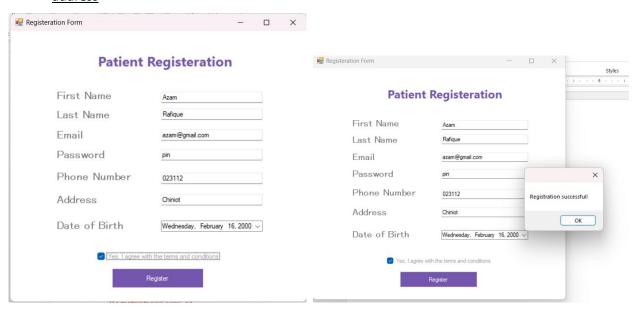
RegistrationForm.cs

Buttons:

- RegisterNow (inserts the data into Oracle database table respectively)

Textboxes:

- dateofbirth (DateTimePicker)
- <u>firstName</u>
- <u>lastName</u>
- <u>email</u>
- phoneNumber
- address



Before and After Insertion:



Corresponding Profile in the "Profiles Table":

| PROFILEID | USERID | FIRSTNAME | LASTNAME | EMAIL | PHONENUMBER | ADDRESS | DATEOFBIRTH |
|-----------|--------|-----------|----------|--------------------|-------------|--------------|-------------|
| 48 | 43 | Daniyal | Ahmed | dani@gmail.com | 342424 | Lahore | 04/28/2024 |
| 50 | 45 | Imran | Sadique | Imran@gmail.com | 1231 | Islamabad | 04/04/2024 |
| 54 | 51 | Asif | Khan | asif@gmail.com | 032124444 | Lahore | 02/09/2000 |
| 51 | 48 | Tayyab | Arshad | tayyab@example.com | 1234567890 | 123 Admin St | 12/20/2003 |
| 55 | 52 | Azam | Rafique | azam@gmail.com | 023112 | Chiniot | 02/16/2000 |

5 rows returned in 0.01 seconds <u>Download</u>

Workspace: F223440 User: F223440

Corresponding User in the "Users Table":

Results Explain Describe Saved SQL History

| USERID | USERNAME | PASSWORD | ROLEID |
|--------|----------------|----------------|--------|
| 43 | daniyal | daniyal123 | 2 |
| 45 | imran | imran123 | 2 |
| 48 | tayyab | tayyabpassword | 3 |
| 51 | asif@gmail.com | abc | 1 |
| 52 | azam@gmail.com | pin | 1 |

5 rows returned in 0.01 seconds <u>Download</u>

Workspace: F223440 User: F223440

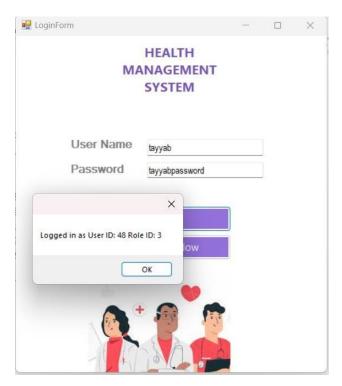
Admin.cs

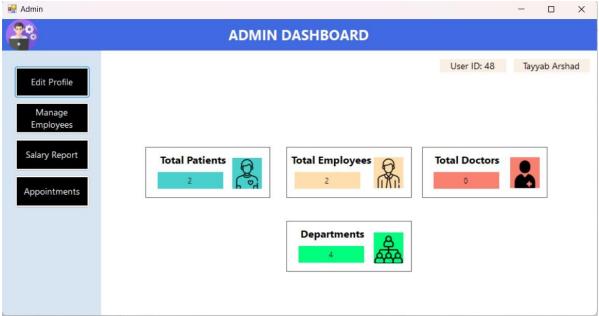
Buttons:

- Check Profile (open CheckProfile.cs form)
- ManageEmployeesButton (open ManageEmployee.cs form)

Labels:

- PatientCount (tells the number of patients that are in the database)
- EmployeeCount (tells the number of employees are in the database)
- DoctorCount (tells the number of doctors that are in the database)
- UserIdLabel (shows the current userId of the admin)
- nameLabel (shows the current name of the admin)



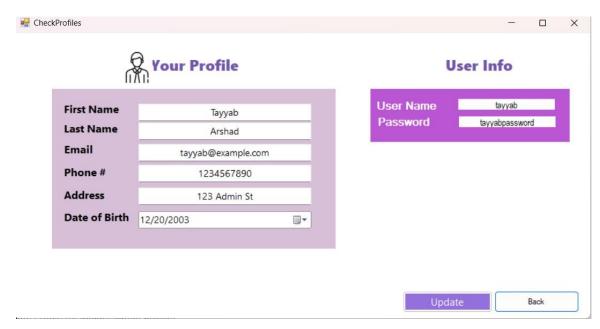


CheckProfile.cs

Buttons:

- Back (go back to Admin.cs)
- Update Profile (to update admin profile)

Edit Profile: Admin can "edit his profile":



If we change the last name to Khan instead of Arshad, the details will be updated in the database as well.

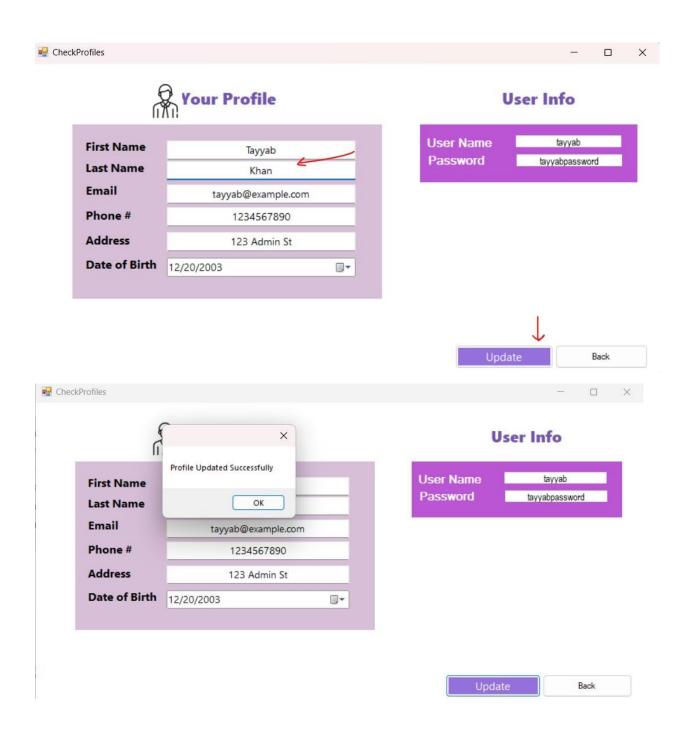
Before Updation:

| 00 | 02 | ALUITI | rangac | azam@gmail.com | 020112 | Official | 02/10/2000 |
|-----------|--------|-----------|----------|--------------------|-------------|--------------|------------|
| 55 | 52 | Azam | Rafigue | azam@gmail.com | 023112 | Chiniot | 02/16/2000 |
| 51 | 48 | Tayyab | Arshad | tayyab@example.com | 1234567890 | 123 Admin St | 12/20/2003 |
| 54 | 51 | Asif | Khan | asif@gmail.com | 032124444 | Lahore | 02/09/2000 |
| 50 | 45 | Imran | Sadique | Imran@gmail.com | 1231 | Islamabad | 04/04/2024 |
| 48 | 43 | Daniyal | Ahmed | dani@gmail.com | 342424 | Lahore | 04/28/2024 |
| PROFILEID | USERID | FIRSTNAME | LASTNAME | EMAIL | PHONENUMBER | ADDRESS | DATEOFBIRT |

5 rows returned in 0.01 seconds <u>Download</u>

Workspace: F223440 User: F223440

After Updation:



Results Explain Describe Saved SQL History

| PROFILEID | USERID | FIRSTNAME | LASTNAME | EMAIL | PHONENUMBER | ADDRESS | DATEOFBIRTH |
|-----------|--------|-----------|----------|--------------------|-------------|--------------|-------------|
| 48 | 43 | Daniyal | Ahmed | dani@gmail.com | 342424 | Lahore | 04/28/2024 |
| 50 | 45 | Imran | Sadique | Imran@gmail.com | 1231 | Islamabad | 04/04/2024 |
| 54 | 51 | Asif | Khan | asif@gmail.com | 032124444 | Lahore | 02/09/2000 |
| 51 | 48 | Tayyab | Khan | tayyab@example.com | 1234567890 | 123 Admin St | 12/20/2003 |
| 55 | 52 | Azam | Rafique | azam@gmail.com | 023112 | Chiniot | 02/16/2000 |

Workspace: F223440 User: F223440

ManageEmployee:

Buttons:

backButton (opens the Admin.cs form)

Grid:

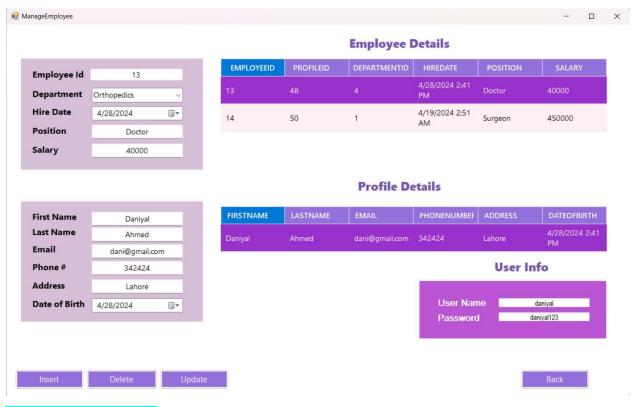
- EmployeesGrid (shows employees table from oracle database)

ComboBox:

- Departments (selection of departments from : DEPARTMENTID DEPARTMENTNAME: 1 for Cardiology, 2 for Oncology, 3 for Neurology, 4 for Orthopedics)

Labels/Textboxes:

- EmpId (label: displays the selected employee id)
- Other textboxes are used to edit, insert , or update the employees. For this purpose I have saved the current selected employee's EmployeeId, and that is run for SQL query when updating is required.



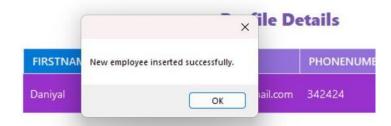
Inserting New Employee: Farhan

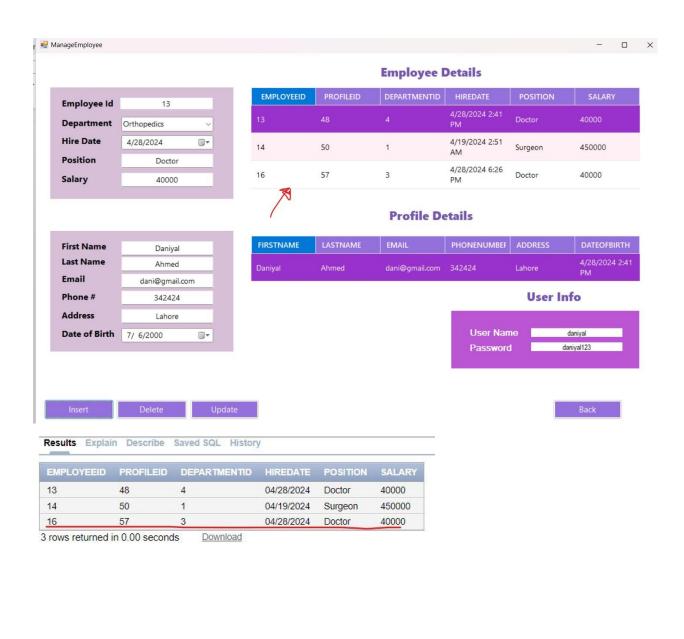


| Results Explain | n Describe | Saved SQL History | / | | |
|-----------------|------------|-------------------|------------|----------|--------|
| EMPLOYEEID | PROFILEID | DEPARTMENTID | HIREDATE | POSITION | SALARY |
| 13 | 48 | 4 | 04/28/2024 | Doctor | 40000 |
| 14 | 50 | 1 | 04/19/2024 | Surgeon | 450000 |

Workspace: F223440 User: F223440

After Insertion:





Insertion in Employees Table

Workspace: F223440 User: F223440

| PROFILEID | USERID | FIRSTNAME | LASTNAME | EMAIL | PHONENUMBER | ADDRESS | DATEOFBIRTH |
|-----------|--------|-----------|----------|--------------------|-------------|--------------|-------------|
| 48 | 43 | Daniyal | Ahmed | dani@gmail.com | 342424 | Lahore | 04/28/2024 |
| 50 | 45 | Imran | Sadique | Imran@gmail.com | 1231 | Islamabad | 04/04/2024 |
| 54 | 51 | Asif | Khan | asif@gmail.com | 032124444 | Lahore | 02/09/2000 |
| 51 | 48 | Tayyab | Khan | tayyab@example.com | 1234567890 | 123 Admin St | 12/20/2003 |
| 55 | 52 | Azam | Rafique | azam@gmail.com | 023112 | Chiniot | 02/16/2000 |
| 57 | 54 | Farhan | Ghafoor | farhan@gmail.com | 1001232 | Karachi | 07/06/2000 |

Workspace: F223440 User: F223440

Insertion in Profiles Table

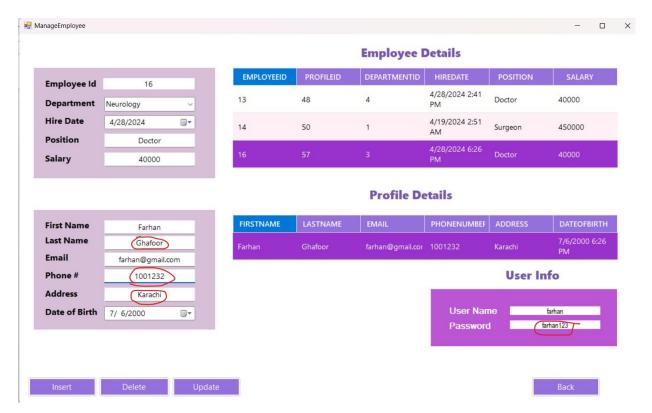
| USERID | USERNAME | PASSWORD | ROLEID |
|--------|----------------|----------------|--------|
| 43 | daniyal | daniyal123 | 2 |
| 45 | imran | imran123 | 2 |
| 48 | tayyab | tayyabpassword | 3 |
| 51 | asif@gmail.com | abc | 1 |
| 52 | azam@gmail.com | pin | 1 |
| 54 | farhan | farhan123 | 2 |

6 rows returned in 0.00 seconds <u>Download</u>

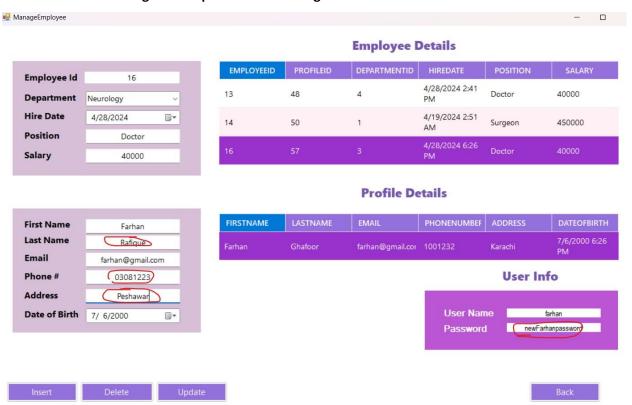
Workspace: F223440 User: F223440

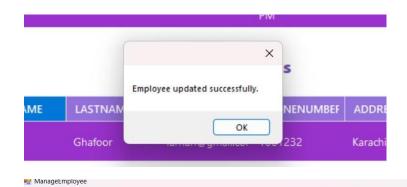
Insertion in USERS table as well

After Updation



These values were changed and updated to following:





Employee Details



| EMPLOYEEID | PROFILEID | DEPARTMENTID | HIREDATE | POSITION | SALARY |
|------------|-----------|--------------|----------------------|----------|--------|
| 13 | 48 | 4 | 4/28/2024 2:41 PM | Doctor | 40000 |
| 14 | 50 | 1 | 4/19/2024 2:51 AM | Surgeon | 450000 |
| 16 | 57 | 3 | 4/28/2024 6:26 PM | Doctor | 40000 |

Profile Details





User Info

- u ×



| Insert | Delete | Update | | |
|--------|----------------|-------------------|--------|--|
| USERID | USERNAME | PASSWORD | ROLEID | |
| 43 | daniyal | daniyal123 | 2 | |
| 45 | imran | imran123 | 2 | |
| 48 | tayyab | tayyabpassword | 3 | |
| 51 | asif@gmail.com | abc | 1 | |
| 52 | azam@gmail.com | pin | 1 | |
| 54 | farhan | newFarhanpassword | 2 | |

6 rows returned in 0.01 seconds <u>Download</u>

Workspace: F223440 User: F223440

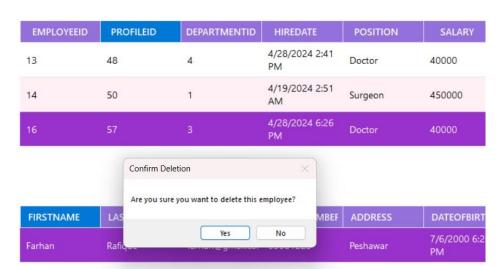
| PROFILEID | USERID | FIRSTNAME | LASTNAME | EMAIL | PHONENUMBER | ADDRESS | DATEOFBIRTH |
|-----------|--------|-----------|----------|--------------------|-------------|--------------|-------------|
| 48 | 43 | Daniyal | Ahmed | dani@gmail.com | 342424 | Lahore | 04/28/2024 |
| 50 | 45 | Imran | Sadique | Imran@gmail.com | 1231 | Islamabad | 04/04/2024 |
| 54 | 51 | Asif | Khan | asif@gmail.com | 032124444 | Lahore | 02/09/2000 |
| 51 | 48 | Tayyab | Khan | tayyab@example.com | 1234567890 | 123 Admin St | 12/20/2003 |
| 55 | 52 | Azam | Rafique | azam@gmail.com | 023112 | Chiniot | 02/16/2000 |
| 57 | 54 | Farhan | Rafique | farhan@gmail.com | 03081223 | Peshawar | 07/06/2000 |

6 rows returned in 0.00 seconds Download

Workspace: F223440 User: F223440

DELETING FARHAN

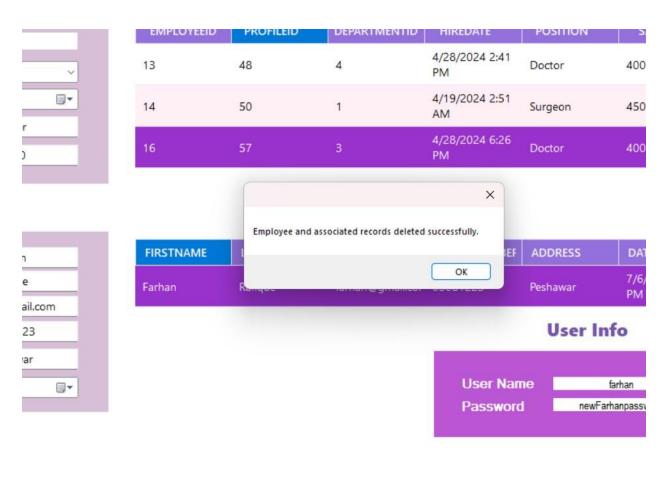


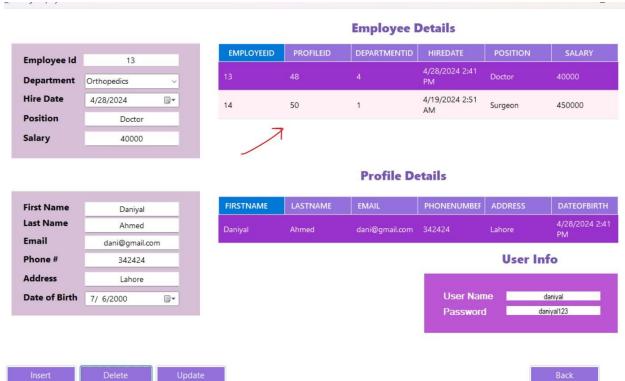






User Info





| USERID | USERNAME | PASSWORD | ROLEID |
|--------|----------------|----------------|--------|
| 43 | daniyal | daniyal123 | 2 |
| 45 | imran | imran123 | 2 |
| 48 | tayyab | tayyabpassword | 3 |
| 51 | asif@gmail.com | abc | 1 |
| 52 | azam@gmail.com | pin | 1 |

5 rows returned in 0.00 seconds <u>Download</u>

| Workspace: | F223440 | User: | F223440 | |
|------------|---------|-------|---------|--|
| | | | | |

| PROFILEID | USERID | FIRSTNAME | LASTNAME | EMAIL | PHONENUMBER | ADDRESS | DATEOFBIRTH |
|-----------|--------|-----------|----------|--------------------|-------------|--------------|-------------|
| 48 | 43 | Daniyal | Ahmed | dani@gmail.com | 342424 | Lahore | 04/28/2024 |
| 50 | 45 | Imran | Sadique | Imran@gmail.com | 1231 | Islamabad | 04/04/2024 |
| 54 | 51 | Asif | Khan | asif@gmail.com | 032124444 | Lahore | 02/09/2000 |
| 51 | 48 | Tayyab | Khan | tayyab@example.com | 1234567890 | 123 Admin St | 12/20/2003 |
| 55 | 52 | Azam | Rafique | azam@gmail.com | 023112 | Chiniot | 02/16/2000 |

5 rows returned in 0.00 seconds <u>Download</u>



Workspace: F223440 User: F223440

| EMPLOYEEID | PROFILEID | DEPARTMENTID | HIREDATE | POSITION | SALARY |
|------------|-----------|--------------|------------|----------|--------|
| 13 | 48 | 4 | 04/28/2024 | Doctor | 40000 |
| 14 | 50 | 1 | 04/19/2024 | Surgeon | 450000 |

2 rows returned in 0.00 seconds <u>Download</u>

Workspace: F223440 User: F223440

Generate PDF BUTTON:

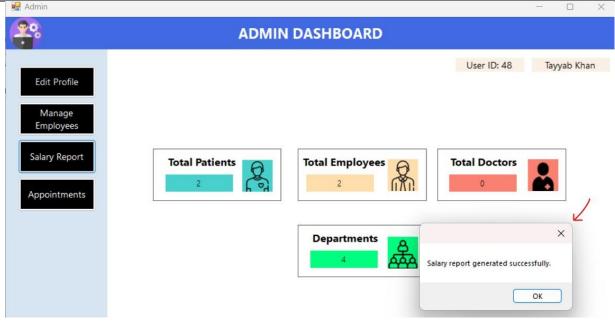
For this purpose I looked at google and found that iTextSharp is an NU Package that helps convert the data into pdfs, so what I did is I took help from documentations and other sources to convert my dataTable from oracle and put the results into the pdf. Following is the implementation of the GeneratePDF Function:

```
private void GeneratePDF(DataTable dataTable)
  {
      Document document = new Document(PageSize.A4);
      string path = Environment.GetFolderPath(Environment.SpecialFolder.Desktop) +
"\\SalaryReport.pdf";
      PdfWriter writer = PdfWriter.GetInstance(document, new
FileStream(path, FileMode.Create));
                                          document.Open();
      // Add a title to the document
      document.Add(new Paragraph("Salary Report", FontFactory.GetFont("Arial")));
      document.Add(new Paragraph(DateTime.Now.ToString("dd/MM/vyvy"),
FontFactory.GetFont("Arial", 12)));
      document.Add(Chunk.NEWLINE); // Add a new line
      // Create a table with columns
      PdfPTable table = new PdfPTable(dataTable.Columns.Count);
      // Add headers to the table
      foreach (DataColumn column in dataTable.Columns)
          PdfPCell cell = new PdfPCell(new Phrase(column.ColumnName));
cell.BackgroundColor = BaseColor.LIGHT_GRAY; // Set background color
table.AddCell(cell);
      }
      // Add data rows to the table
      foreach (DataRow row in dataTable.Rows)
          foreach (object item in row.ItemArray)
              table.AddCell(new Phrase(item.ToString()));
      }
      // Add the table to the document
document.Add(table); document.Close();
  }
```

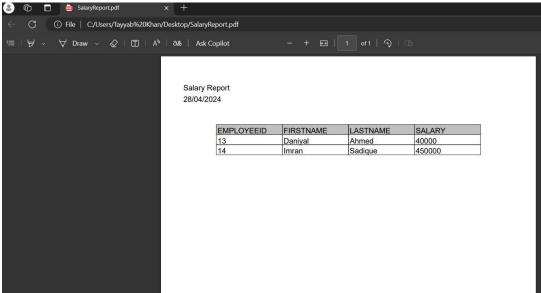
It is passed a dataTable from the outside when the "Salary" button is clicked:

```
private void GenerateSalaryReportButton_Click(object sender, EventArgs e)
{
```

```
try
{
         using (OracleConnection con = new OracleConnection(@"DATA
SOURCE=localhost:1521/xe;PASSWORD=hello;USER ID=F223440"))
         {
             con.Open();
             string query = "SELECT EmployeeID, FirstName, LastName, Salary
FROM Employees JOIN Profiles ON Employees.ProfileID = Profiles.ProfileID";
using (OracleCommand cmd = new OracleCommand(query, con))
                 OracleDataAdapter oda = new OracleDataAdapter(cmd); //
OracleDataAdapter
                 DataTable dt = new DataTable(); // DataTable to store the result
oda.Fill(dt); // Fill the DataTable with the result of the query
                 GeneratePDF(dt);
                 MessageBox.Show("Salary report generated successfully.");
             con.Close();
         }
}
     catch (Exception ex)
         MessageBox.Show("An error occurred: " + ex.Message);
     }
 }
```



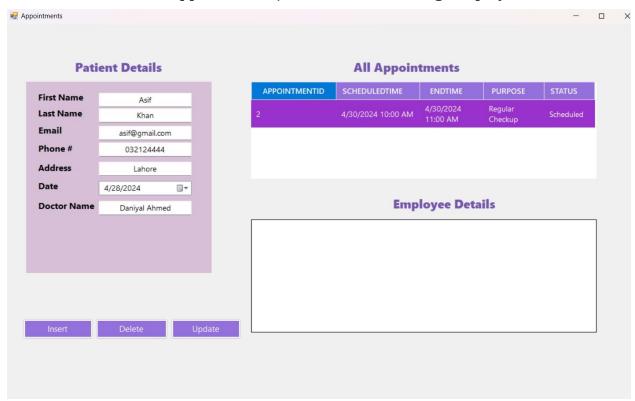




It can be adjusted with other queries as well depending on our requirements of project further in phase#3.

Appointments.cs:

This will show all the appointments to the Admin, for now only the data is retrieved from the appointments table in Oracle, and displayed here. Later I will implement the insertion deletions of appointments just like I did in ManageEmployee.cs



CODES in C#

Form1.cs:

```
using Oracle.ManagedDataAccess.Client;
using System;
using
System.Collections.Generic;
using System.ComponentModel;
using System.Data; using
System.Drawing; using
System.Ling;
using System.Runtime.Remoting.Messaging;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using static System.Windows.Forms.VisualStyles.VisualStyleElement;
namespace WindowsFormsApp1
    public partial class Form1 : Form
        OracleConnection con;
        int loggedInUserId; // Variable to store the UserID of the logged-in user
        public Form1()
            InitializeComponent();
        }
        private void Form1_Load(object sender, EventArgs e)
            string conStr = @"DATA SOURCE=localhost:1521/xe;PASSWORD=hello;USER
ID=F223440";
            con = new OracleConnection(conStr);
        }
        private void LoginButton_Click(object sender, EventArgs e)
try
                con.Open();
                OracleCommand selectCmd = con.CreateCommand();
                selectCmd.CommandText = "SELECT UserID, RoleID FROM USERS WHERE
USERNAME = :email AND PASSWORD = :password";
                selectCmd.CommandType = CommandType.Text;
                // Bind parameters
                selectCmd.Parameters.Add(":email", OracleDbType.Varchar2).Value
username.Text;
                selectCmd.Parameters.Add(":password", OracleDbType.Varchar2).Value =
password.Text;
                OracleDataReader reader = selectCmd.ExecuteReader(); // Execute the
query and get the result
```

```
if (reader.Read())
                    loggedInUserId = reader.GetInt32(0); // Get the UserID of the
logged-in user
                   int roleId = reader.GetInt32(1); // Get the RoleID of the
logged-in user
                  MessageBox.Show("Logged in as User ID: " + loggedInUserId + "
 Role ID: " + roleId);
                     // Open corresponding form based on role
 if (roleId == 1) // Patient
                         // Hide this form and open the patient form
 this.Hide();
                         Patient patientForm = new Patient(); // Pass
 loggedInUserId to Patient form
                         patientForm.Closed +=
                                                   (s,
                                                         args)
                                                                      this.Close();
 patientForm.Show();
                     else if (roleId == 2) // Employee
                         this.Hide();
                         Employee employeeForm = new Employee( loggedInUserId); //
 Pass loggedInUserId to Employee form
                         employeeForm.Closed += (s,
                                                         args)
                                                                      this.Close();
                                                                 =>
 employeeForm.Show();
                     else if (roleId == 3) // Admin
                         this.Hide();
                         Admin adminForm = new Admin(loggedInUserId); // Pass
 loggedInUserId to Admin form
                         adminForm.Closed
                                                  (s,
                                                        args)
                                                                      this.Close();
 adminForm.Show();
                     }
 else
                     {
                         MessageBox.Show("Invalid Role!");
                   }
 else
                 {
                     MessageBox.Show("Invalid Email or Password!");
                 reader.Close();
                                    //
                                          Close
                                                   the
                                                          reader
con.Close();
           catch (Exception ex)
               MessageBox.Show("An error occurred: " + ex.Message);
           }
```

Admin.cs:

```
using System;
using
System.Collections.Generic;
using System.ComponentModel;
using System.Data; using
System.Drawing; using
System.Linq; using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Xml;
using
Oracle.ManagedDataAccess.Client;
using iTextSharp.text; using
iTextSharp.text.pdf; using System.IO;
namespace WindowsFormsApp1
   public partial class Admin : Form
        int LocalUserId;
        public Admin(int UserId)
            InitializeComponent();
            //save this UserId
            LocalUserId = UserId;
            LoadAdminProfile();
        }
        private void button1_Click(object sender, EventArgs e)
            //hide this form and open the check profiles form
this.Hide();
            CheckProfiles checkProfilesForm = new CheckProfiles(LocalUserId);
checkProfilesForm.Closed += (s, args) => this.Close();
checkProfilesForm.Show();
        }
        private void LoadAdminProfile()
try
                string conStr = @"DATA SOURCE=localhost:1521/xe;PASSWORD=hello;USER
ID=F223440";
                using (OracleConnection con = new OracleConnection(conStr))
                {
                    con.Open();
                    // Fetch admin's profile information from the database using the
LocalUserId
                    OracleCommand getAdminProfile = con.CreateCommand();
```

```
getAdminProfile.CommandText = "SELECT FirstName, LastName
FROM PROFILES WHERE UserID = :userId";
                    getAdminProfile.CommandType
                                                                 CommandType.Text;
getAdminProfile.Parameters.Add(":userId",
OracleDbType.Int32).Value = LocalUserId;
                    //get result
                                    (OracleDataReader
                                                                reader
                    using
getAdminProfile.ExecuteReader())
                        if (reader.Read())
                            string firstName = reader.GetString(0); // Get the
value of the FirstName column
                            string lastName = reader.GetString(1); // Get the
value of the LastName column
                            int LocalDoctorCount;
int LocalPatientCount:
int LocaldeptCount;
                            OracleCommand getDoctorCount = con.CreateCommand();
getDoctorCount.CommandText = "SELECT COUNT(*) FROM USERS WHERE RoleID = 4";
                            getDoctorCount.CommandType = CommandType.Text;
                            LocalDoctorCount =
Convert.ToInt32(getDoctorCount.ExecuteScalar());
                            OracleCommand
                                                    getPatientCount
con.CreateCommand();
                                                  getPatientCount.CommandText =
"SELECT COUNT(*) FROM USERS WHERE RoleID = 1";
                            getPatientCount.CommandType = CommandType.Text;
                            LocalPatientCount =
Convert.ToInt32(getPatientCount.ExecuteScalar());
                            OracleCommand
                                                    getEmployeeCount
con.CreateCommand();
                                                  getEmployeeCount.CommandText =
"SELECT COUNT(*) FROM USERS WHERE RoleID = 2";
                            getEmployeeCount.CommandType = CommandType.Text;
                            DoctorCount.Text = LocalDoctorCount.ToString();
                            PatientCount.Text = LocalPatientCount.ToString();
                            EmployeeCount.Text
getEmployeeCount.ExecuteScalar().ToString();
```

```
// departments query
                            OracleCommand getDeptCount = con.CreateCommand();
getDeptCount.CommandText = "SELECT COUNT(*) FROM DEPARTMENTS";
                            getDeptCount.CommandType = CommandType.Text;
                            LocaldeptCount =
Convert.ToInt32(getDeptCount.ExecuteScalar());
                            DeptCount.Text = LocaldeptCount.ToString();
                            // Display UserID, name, and current system time on the
screen
                            UserIdLabel.Text = "User ID: " + LocalUserId.ToString();
                            nameLabel.Text = firstName + " " + lastName;
                        }
else
                        {
                            MessageBox.Show("Admin profile not found.");
                    }
                    con.Close();
}
            catch (Exception ex)
```

```
MessageBox.Show("Error fetching admin profile information: " +
ex.Message);
}
}
        private void button1_Click_1(object sender, EventArgs e)
            //open ManageEmployee.cs form
            this.Hide();
            ManageEmployee manageEmployeeForm = new ManageEmployee(LocalUserId);
            manageEmployeeForm.Closed
                                       += (s,
                                                     args)
                                                                     this.Close();
manageEmployeeForm.Show();
}
        private void Admin_Load(object sender, EventArgs e)
}
        private void label9_Click(object sender, EventArgs e)
}
        private void GenerateSalaryReportButton_Click(object sender, EventArgs e)
try
                using (OracleConnection con = new OracleConnection(@"DATA
SOURCE=localhost:1521/xe;PASSWORD=hello;USER ID=F223440"))
                    con.Open();
                    string query = "SELECT EmployeeID, FirstName, LastName,
Salary FROM Employees JOIN Profiles ON Employees.ProfileID =
Profiles.ProfileID";
                                         using (OracleCommand cmd = new
OracleCommand(query, con))
                        OracleDataAdapter oda = new OracleDataAdapter(cmd); //
OracleDataAdapter
                        DataTable dt = new DataTable(); // Create a new DataTable
to store the result
                        oda.Fill(dt); // Fill the DataTable with the result off
query
                        GeneratePDF(dt);
                        MessageBox.Show("Salary report generated
successfully.");
                    con.Close();
                }
}
            catch (Exception ex)
                MessageBox.Show("An error occurred: " + ex.Message);
```

```
}
       private void GeneratePDF(DataTable dataTable)
           Document
                                           Document(PageSize.A4);
                      document =
                                     new
string path =
Environment.GetFolderPath(Environment.SpecialFolder.Desktop)
 "\\SalaryReport.pdf";
             PdfWriter
                                            PdfWriter.GetInstance(document,
                           writer
 FileStream(path, FileMode.Create));
             document.Open();
             // Add a title to the document
             document.Add(new
                                           Paragraph("Salary
                                                                           Report",
 FontFactory.GetFont("Arial")));
             document.Add(new
                                     Paragraph(DateTime.Now.ToString("dd/MM/yyyy"),
FontFactory.GetFont("Arial", 12)));
             document.Add(Chunk.NEWLINE); // Add a new line
             // Create a table with columns
             PdfPTable table = new PdfPTable(dataTable.Columns.Count);
             // Add headers to the table
             foreach (DataColumn column in dataTable.Columns)
                 PdfPCell cell = new PdfPCell(new Phrase(column.ColumnName));
 cell.BackgroundColor = BaseColor.LIGHT_GRAY; // Set background color
 table.AddCell(cell);
             }
             // Add data rows to the table
             foreach (DataRow row in dataTable.Rows)
                 foreach (object item in row.ItemArray)
 {
```

ManageEmployee.cs

```
using Oracle.ManagedDataAccess.Client;
using System; using
System.Collections.Generic; using
System.ComponentModel;
using System.Data; using
System.Drawing; using
System.Globalization; using
System.Linq; using
System.Text; using
System.Threading.Tasks;
using System.Windows.Forms;
namespace WindowsFormsApp1
{
 public partial class ManageEmployee : Form
   int loggedInUserId;
   public ManageEmployee(int a)
     loggedInUserId = a;
InitializeComponent();
     LoadData();
     LoadDepartments();
     EmployeesGrid.SelectionChanged += EmployeesGrid_SelectionChanged;
CustomizeDataGridView();
   }
```

```
private void dataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs e)
   private void LoadData()
     using (OracleConnection con = new OracleConnection(@"DATA
SOURCE=localhost:1521/xe;PASSWORD=hello;USER ID=F223440"))
       con.Open();
       OracleDataAdapter adapter = new OracleDataAdapter("SELECT * FROM Employees",
con);
        DataTable dt = new DataTable();
       adapter.Fill(dt);
        EmployeesGrid.DataSource = dt;
con.Close();
     }
   }
   private void EmployeesGrid SelectionChanged(object sender, EventArgs e)
     if (EmployeesGrid.SelectedRows.Count > 0)
       int selectedRowIndex = EmployeesGrid.SelectedRows[0].Index;
if (selectedRowIndex >= 0)
       {
         DataGridViewRow row = EmployeesGrid.Rows[selectedRowIndex];
int profileID = Convert.ToInt32(row.Cells["ProfileID"].Value);
LoadProfileData(profileID);
string employeeID;
                            int
deptID;
         //extract the id of the selected employee and update the empId label
         employeeID = row.Cells["EmployeeID"].Value.ToString();
         EmpId.Text = employeeID;
```

```
//extract the department id of the selected employee and update the deptId combobox
with the department name it should also show all other departments as well upon expansion
// Update the Departments ComboBox to reflect the selected employee's department
deptID = Convert.ToInt32(row.Cells["DepartmentID"].Value);
         Departments.SelectedValue = deptID;
         //extract the hiring date now into dateTimePicker1
         if (DateTime.TryParseExact(row.Cells["HireDate"].Value.ToString(), "dd/MM/yyyy",
CultureInfo.InvariantCulture, DateTimeStyles.None, out DateTime hiringDate))
           dateTimePicker1.Value = hiringDate.Date; // Extracting only the date part
         }
         string position = row.Cells["Position"].Value.ToString();
employeePosition.Text = position;
         string salary = row.Cells["Salary"].Value.ToString();
         employeeSalary.Text = salary;
     }
   private void LoadProfileData(int profileID)
     using (OracleConnection con = new OracleConnection(@"DATA
SOURCE=localhost:1521/xe;PASSWORD=hello;USER ID=F223440"))
     {
       con.Open();
       OracleDataAdapter adapter = new OracleDataAdapter($"SELECT ProfileID, UserID,
FirstName, LastName, Email, PhoneNumber, Address, DateOfBirth FROM Profiles WHERE ProfileID
 = {profileID}", con);
         DataTable dt = new DataTable();
         adapter.Fill(dt);
         ProfileGrid.DataSource = dt;
         //hide the profileId and employeeId columns
         ProfileGrid.Columns["ProfileID"].Visible = false;
         ProfileGrid.Columns["UserID"].Visible = false;
```

```
//now we extract the data from the profile grid and update the textboxes
if (ProfileGrid.Rows.Count > 0)
        {
          DataGridViewRow row = ProfileGrid.Rows[0];
string firstName = row.Cells["FirstName"].Value.ToString();
string lastName = row.Cells["LastName"].Value.ToString();
string email = row.Cells["Email"].Value.ToString();
          string phoneNumber = row.Cells["PhoneNumber"].Value.ToString();
          string address = row.Cells["Address"].Value.ToString();
string dateOfBirth = row.Cells["DateOfBirth"].Value.ToString();
          //find the username and passsword from USERS table of the associated employee
from the userID
          OracleCommand selectCmd = con.CreateCommand();
          selectCmd.CommandText = $"SELECT Username, Password FROM USERS WHERE
UserID
= {row.Cells["UserID"].Value}";
          selectCmd.CommandType = CommandType.Text;
OracleDataReader reader = selectCmd.ExecuteReader();
if (reader.Read())
          {
            string username = reader.GetString(0);
string password = reader.GetString(1);
            // Update the textboxes with the extracted data
            Username.Text = username:
            UserPassword.Text = password;
          }
          // Update the textboxes with the extracted data
          FirstName.Text = firstName:
          LastName.Text = lastName:
          Email.Text = email;
          PhoneNumber.Text = phoneNumber;
          Address.Text = address:
          //extract only the date part of the date of birth
          if (DateTime.TryParseExact(dateOfBirth, "dd/MM/yyyy", CultureInfo.InvariantCulture,
DateTimeStyles.None, out DateTime dob))
```

```
{
           DateofBirth.Value = dob.Date; // Extracting only the date part
       }
       con.Close();
    }
    private void LoadDepartments()
     using (OracleConnection con = new OracleConnection(@"DATA
SOURCE=localhost:1521/xe;PASSWORD=hello;USER ID=F223440"))
     {
       con.Open();
       OracleDataAdapter adapter = new OracleDataAdapter("SELECT DepartmentID,
DepartmentName FROM Departments ORDER BY DepartmentName", con);
       DataTable dt = new DataTable();
       adapter.Fill(dt);
       // Set up the ComboBox
       Departments.DisplayMember = "DepartmentName";
       Departments.ValueMember = "DepartmentID";
       Departments.DataSource = dt;
       con.Close();
     }
   private void button3_Click(object sender, EventArgs e)
```

```
private void InsertButton_Click(object sender, EventArgs e)
     using (OracleConnection con = new OracleConnection(@"DATA
SOURCE=localhost:1521/xe;PASSWORD=hello;USER ID=F223440"))
       con.Open();
       OracleCommand insertCommand = con.CreateCommand();
       insertCommand.CommandText =
@"
       DECLARE
                      v UserID
NUMBER:
               v_ProfileID NUMBER;
    BEGIN
     -- Insert into Users table to generate UserID
      INSERT INTO Users (UserID, Username, Password, RoleID)
     VALUES (USER_ID_SEQ.NEXTVAL, :p_Username, :p_Password, :p_RoleID)
RETURNING UserID INTO v_UserID;
     -- Insert into Profiles table using the generated UserID
      INSERT INTO Profiles (ProfileID, UserID, FirstName, LastName, Email, PhoneNumber,
Address, DateOfBirth)
     VALUES (PROFILE_ID_SEQ.NEXTVAL, v_UserID, :p_FirstName, :p_LastName, :p_Email,
:p_PhoneNumber, :p_Address, :p_DateOfBirth)
     RETURNING ProfileID INTO v_ProfileID;
     -- Insert into Employees table using the generated ProfileID
     INSERT INTO Employees (EmployeeID, ProfileID, DepartmentID, HireDate, Position,
            VALUES (EMPLOYEE_ID_SEQ.NEXTVAL,v_ProfileID, :p_DepartmentID, :p_HireDate,
:p_Position, :p_Salary);
     COMMIT;
    END;";
       insertCommand.Parameters.Add("p_Username", OracleDbType.Varchar2).Value =
Username.Text:
       insertCommand.Parameters.Add("p_Password", OracleDbType.Varchar2).Value
                            insertCommand.Parameters.Add("p_RoleID",
= UserPassword.Text;
```

```
OracleDbType.Int32).Value = 2;
insertCommand.Parameters.Add("p_FirstName", OracleDbType.Varchar2).Value =
FirstName.Text:
       insertCommand.Parameters.Add("p_LastName", OracleDbType.Varchar2).Value =
                       insertCommand.Parameters.Add("p_Email",
LastName.Text:
OracleDbType.Varchar2).Value = Email.Text;
insertCommand.Parameters.Add("p_PhoneNumber", OracleDbType.Varchar2).Value =
PhoneNumber.Text; // PhoneNumber TextBox
       insertCommand.Parameters.Add("p_Address", OracleDbType.Varchar2).Value =
Address.Text; // Address TextBox
       insertCommand.Parameters.Add("p_DateOfBirth", OracleDbType.Date).Value =
DateofBirth.Value; // DateOfBirth DateTimePicker
       insertCommand.Parameters.Add("p DepartmentID", OracleDbType.Int32).Value =
Convert.ToInt32(Departments.SelectedValue); // Departments
       insertCommand.Parameters.Add("p_HireDate", OracleDbType.Date).Value =
dateTimePicker1.Value; // HireDate DateTimePicker
       insertCommand.Parameters.Add("p_Position", OracleDbType.Varchar2).Value =
employeePosition.Text; // Position TextBox
       insertCommand.Parameters.Add("p_Salary", OracleDbType.Decimal).Value =
Convert.ToDecimal(employeeSalary.Text); // Salary TextBox
try
       {
         insertCommand.ExecuteNonQuery();
         MessageBox.Show("New employee inserted successfully.");
         LoadData(); // Refresh DataGridView
       catch (OracleException ex)
         MessageBox.Show("An error occurred: " + ex.Message);
finally
         con.Close();
     }
```

```
private void DeleteButton_Click(object sender, EventArgs e)
     if (MessageBox.Show("Are you sure you want to delete this employee?", "Confirm Deletion",
MessageBoxButtons.YesNo) == DialogResult.Yes)
       if (EmployeesGrid.SelectedRows.Count > 0)
         int selectedRowIndex = EmployeesGrid.SelectedRows[0].Index;
if (selectedRowIndex >= 0)
           int employeeID =
Convert.ToInt32(EmployeesGrid.Rows[selectedRowIndex].Cells["EmployeeID"].Value);
           DeleteEmployee(employeeID);
         }
     }
   private void DeleteEmployee(int employeeID)
     using (OracleConnection con = new OracleConnection(@"DATA
SOURCE=localhost:1521/xe;PASSWORD=hello;USER ID=F223440"))
       con.Open();
       OracleCommand cmd = new OracleCommand("DELETE_EMPLOYEE", con);
cmd.CommandType = CommandType.StoredProcedure;
       cmd.Parameters.Add("p_EmployeeID", OracleDbType.Int32).Value = employeeID;
try
         cmd.ExecuteNonQuery();
         MessageBox.Show("Employee and associated records deleted successfully.");
LoadData(); // Refresh the DataGridView to reflect the changes
       catch (OracleException ex)
         MessageBox.Show("An error occurred: " + ex.Message);
```

```
}
finally
         con.Close();
   private void UpdateButton_Click(object sender, EventArgs e)
     if (EmployeesGrid.SelectedRows.Count > 0)
       int selectedRowIndex = EmployeesGrid.SelectedRows[0].Index;
       DataGridViewRow selectedRow = EmployeesGrid.Rows[selectedRowIndex];
       // Retrieve the current employee's ID, profile ID, and user ID
                                                                          int
employeeID = Convert.ToInt32(selectedRow.Cells["EmployeeID"].Value);
int profileID = Convert.ToInt32(selectedRow.Cells["ProfileID"].Value);
                                                                           int
userID; // You'll need to fetch this from the database using the profileID
       using (OracleConnection con = new OracleConnection(@"DATA
SOURCE=localhost:1521/xe;PASSWORD=hello;USER ID=F223440"))
         con.Open();
         // Fetch the userID from Profiles table using the profileID
         OracleCommand getUserIdCmd = new OracleCommand("SELECT UserID FROM
Profiles
WHERE ProfileID = :profileId", con);
         getUserIdCmd.Parameters.Add(new OracleParameter("profileId", profileID));
userID = Convert.ToInt32(getUserIdCmd.ExecuteScalar());
         // Begin a transaction to ensure atomicity
         OracleTransaction transaction = con.BeginTransaction();
```

```
trv
           // Update the Users table
           OracleCommand updateUsersCmd = new OracleCommand("UPDATE Users SET
Username = :username, Password = :password WHERE UserID = :userId", con);
           updateUsersCmd.Parameters.Add(new OracleParameter("username",
Username.Text));
           updateUsersCmd.Parameters.Add(new OracleParameter("password",
UserPassword.Text));
           updateUsersCmd.Parameters.Add(new OracleParameter("userId", userID));
updateUsersCmd.ExecuteNonQuery();
           // Update the Profiles table
           OracleCommand updateProfilesCmd = new OracleCommand("UPDATE Profiles SET
FirstName = :firstName, LastName = :lastName, Email = :email, PhoneNumber = :phoneNumber,
Address = :address, DateOfBirth = :dateOfBirth WHERE ProfileID = :profileId", con);
updateProfilesCmd.Parameters.Add(new OracleParameter("firstName", FirstName.Text));
           updateProfilesCmd.Parameters.Add(new
                                                        OracleParameter("lastName".
                                              updateProfilesCmd.Parameters.Add(new
LastName.Text));
OracleParameter("email", Email.Text));
                                              updateProfilesCmd.Parameters.Add(new
OracleParameter("phoneNumber", PhoneNumber.Text));
           updateProfilesCmd.Parameters.Add(new OracleParameter("address",
Address.Text));
           updateProfilesCmd.Parameters.Add(new OracleParameter("dateOfBirth",
DateofBirth.Value));
           updateProfilesCmd.Parameters.Add(new OracleParameter("profileId", profileID));
updateProfilesCmd.ExecuteNonQuery();
           // Update the Employees table
           OracleCommand updateEmployeesCmd = new OracleCommand("UPDATE
Employees
SET DepartmentID = :departmentId, HireDate = :hireDate, Position = :position, Salary = :salary
WHERE EmployeeID = :employeeId", con);
           updateEmployeesCmd.Parameters.Add(new OracleParameter("departmentId",
Convert.ToInt32(Departments.SelectedValue)));
           updateEmployeesCmd.Parameters.Add(new OracleParameter("hireDate",
dateTimePicker1.Value));
           updateEmployeesCmd.Parameters.Add(new OracleParameter("position",
employeePosition.Text));
           updateEmployeesCmd.Parameters.Add(new OracleParameter("salary",
Convert.ToDecimal(employeeSalary.Text)));
           updateEmployeesCmd.Parameters.Add(new OracleParameter("employeeId",
employeeID));
           updateEmployeesCmd.ExecuteNonQuery();
```

```
// Commit the transaction
 transaction.Commit();
           MessageBox.Show("Employee updated successfully.");
           // Refresh the DataGridView to show the updated data
            LoadData();
          catch (OracleException ex)
            // Attempt to roll back the transaction
transaction.Rollback();
            MessageBox.Show("An error occurred while updating the employee. " + ex.Message);
          finally
            con.Close();
else
        MessageBox.Show("No employee selected for updating.");
    }
    private void ManageEmployee_Load(object sender, EventArgs e)
    }
```

Appointments.cs

```
using System; using
System.Collections.Generic; using
System.ComponentModel;
using System.Data; using
System.Drawing; using
System.Ling; using
System.Text; using
System.Threading.Tasks; using
System.Windows.Forms;
using Oracle.ManagedDataAccess.Client;
namespace WindowsFormsApp1
 public partial class Appointments: Form
 {
   int LocalUserId;
   public Appointments(int userId)
     InitializeComponent();
     LocalUserId = userId;
     AppointmentsGrid.SelectionChanged += AppointmentsGrid_SelectionChanged;
CustomizeDataGridView();
   }
   private void Appointments_Load(object sender, EventArgs e)
   {
     // Load the appointments from the database
     string conStr = @"DATA SOURCE=localhost:1521/xe;PASSWORD=hello;USER ID=F223440";
using (OracleConnection con = new OracleConnection(conStr))
       con.Open();
         //load data from database from appointments table
         OracleDataAdapter adapter = new OracleDataAdapter($"SELECT * FROM
APPOINTMENTS", con);
         DataTable dt = new DataTable();
adapter.Fill(dt);
         AppointmentsGrid.DataSource = dt;
       //hide patient id and employee id columns
       AppointmentsGrid.Columns["PATIENTID"].Visible = false;
```

```
AppointmentsGrid.Columns["EMPLOYEEID"].Visible = false;
       //extract the patient profile information and display in the textboxes, also extarct the
employee name from the employee table and display name of doctor in the textbox
foreach (DataGridViewRow row in AppointmentsGrid.Rows)
{
         int patientId = Convert.ToInt32(row.Cells["PATIENTID"].Value);
int employeeId = Convert.ToInt32(row.Cells["EMPLOYEEID"].Value);
         //get the patient profile information
         OracleCommand getPatientProfile = con.CreateCommand();
         getPatientProfile.CommandText = "SELECT FirstName, LastName FROM PROFILES
WHERE UserID = :userId";
         getPatientProfile.CommandType = CommandType.Text;
         getPatientProfile.Parameters.Add(":userId", OracleDbType.Int32).Value = patientId;
         OracleDataReader reader = getPatientProfile.ExecuteReader();
if (reader.Read())
         {
           row.Cells["PATIENT"].Value = reader.GetString(0) + " " + reader.GetString(1);
         reader.Close();
         //get the employee name
         OracleCommand getEmployeeName = con.CreateCommand();
getEmployeeName.CommandText = "SELECT FirstName, LastName FROM PROFILES WHERE
UserID = :userId";
         getEmployeeName.CommandType = CommandType.Text;
         getEmployeeName.Parameters.Add(":userId", OracleDbType.Int32).Value =
                      reader = getEmployeeName.ExecuteReader();
employeeId;
                                                                           if
(reader.Read())
           row.Cells["DOCTOR"].Value = reader.GetString(0) + " " + reader.GetString(1);
         reader.Close();
       }
    //appointmentsGrid_CellContentClick
    private void AppointmentsGrid_CellContentClick(object sender, EventArgs e) { }
    private void AppointmentsGrid_SelectionChanged(object sender, EventArgs e)
```

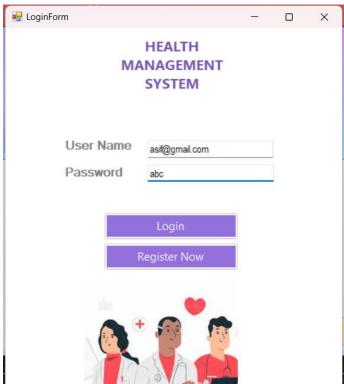
```
if (AppointmentsGrid.SelectedRows.Count > 0)
        DataGridViewRow selectedRow = AppointmentsGrid.SelectedRows[0];
       int patientId = Convert.ToInt32(selectedRow.Cells["PatientID"].Value);
int employeeId = Convert.ToInt32(selectedRow.Cells["EmployeeID"].Value);
       string conStr = @"DATA SOURCE=localhost:1521/xe;PASSWORD=hello;USER
ID=F223440";
                      using (OracleConnection con = new OracleConnection(conStr))
         con.Open();
         // Fetch patient ProfileID
int patientProfileId;
         OracleCommand getPatientProfileId = new OracleCommand("SELECT ProfileID FROM
PATIENTS WHERE PatientID = :patientId", con);
         getPatientProfileId.Parameters.Add(":patientId", OracleDbType.Int32).Value =
patientId;
                   patientProfileId = Convert.ToInt32(getPatientProfileId.ExecuteScalar());
         // Fetch patient details from Profiles
         OracleCommand getPatientDetails = new OracleCommand("SELECT FirstName,
LastName, Email, PhoneNumber, Address FROM PROFILES WHERE ProfileID = :profileId", con);
getPatientDetails.Parameters.Add(":profileId", OracleDbType.Int32).Value = patientProfileId;
         using (OracleDataReader reader = getPatientDetails.ExecuteReader())
         {
           if (reader.Read())
             // Assign the details to textboxes here.
```

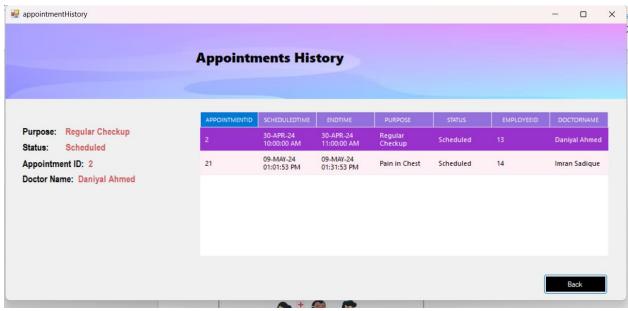
```
FirstName.Text = reader.GetString(0);
             LastName.Text = reader.GetString(1);
             Email.Text = reader.GetString(2);
             PhoneNumber.Text = reader.GetString(3);
Address.Text = reader.GetString(4);
           }
         }
         // Fetch employee ProfileID
int employeeProfileId;
         OracleCommand getEmployeeProfileId = new OracleCommand("SELECT ProfileID FROM
EMPLOYEES WHERE EmployeeID = :employeeId", con);
         getEmployeeProfileId.Parameters.Add(":employeeId",
                                                               OracleDbType.Int32).Value
employeeId;
         employeeProfileId = Convert.ToInt32(getEmployeeProfileId.ExecuteScalar());
         // Fetch employee name from Profiles
         OracleCommand getEmployeeName = new OracleCommand("SELECT FirstName,
LastName FROM PROFILES WHERE ProfileID = :profileId", con);
         getEmployeeName.Parameters.Add(":profileId", OracleDbType.Int32).Value =
employeeProfileId;
         using (OracleDataReader reader = getEmployeeName.ExecuteReader())
           if (reader.Read())
             // Assign the doctor's name to the textbox here...
             DocName.Text = reader.GetString(0) + " " + reader.GetString(1);
         }
         con.Close();
       }
     }
```

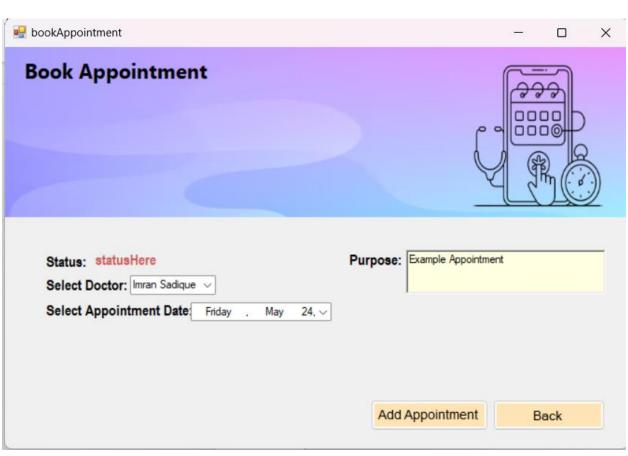
Rest of code working:

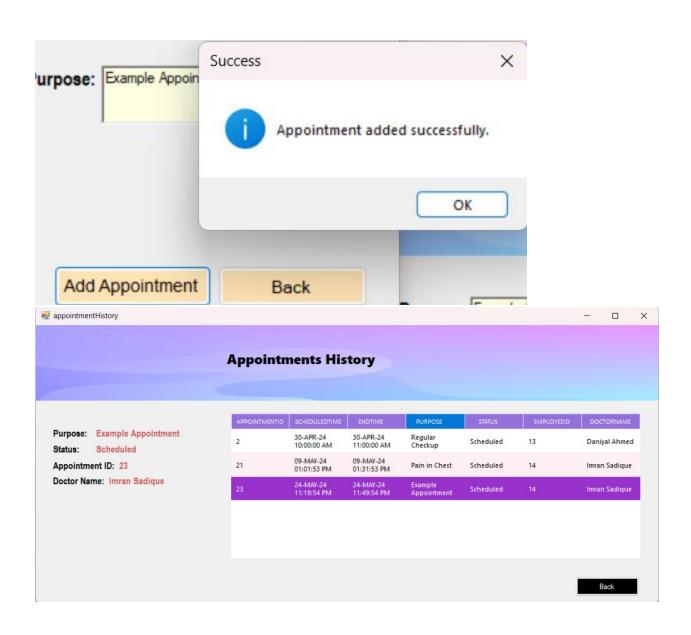
Patient Form:

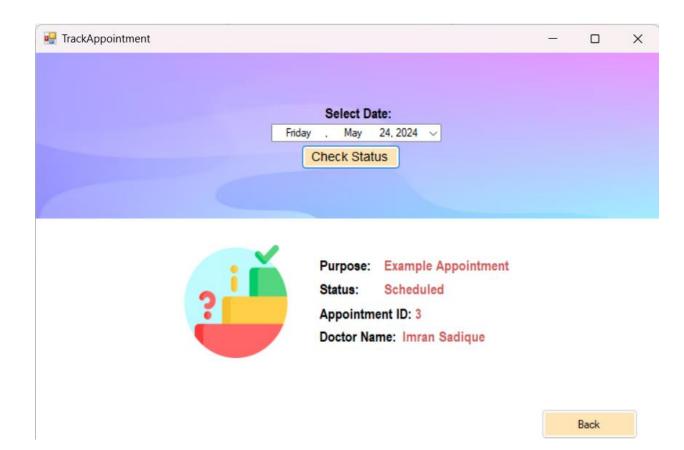


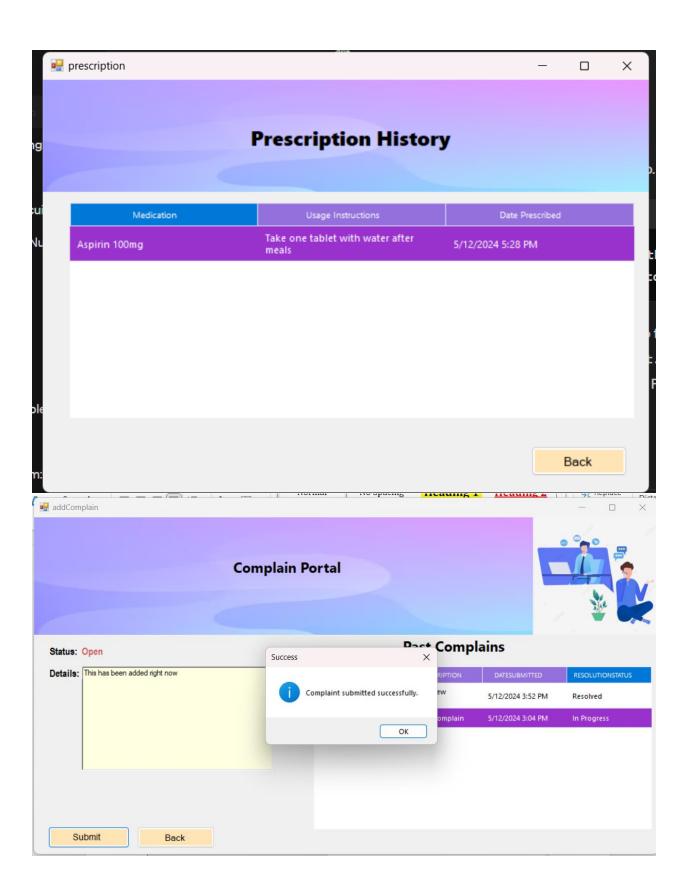


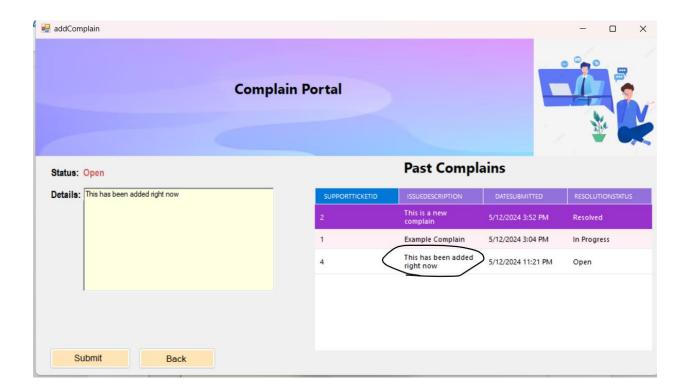




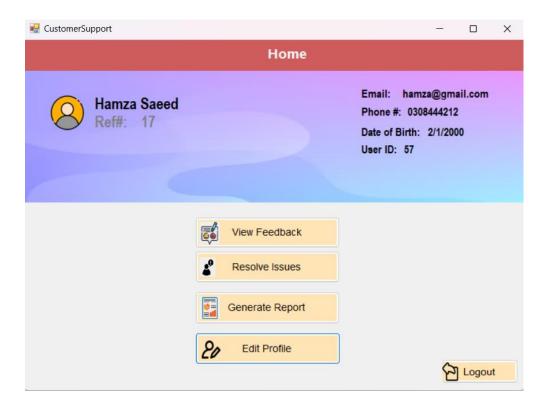


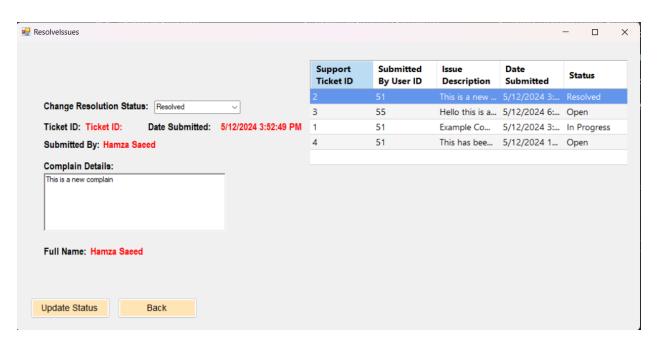


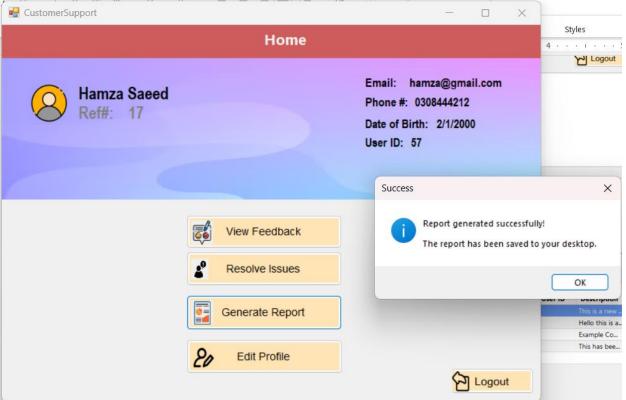


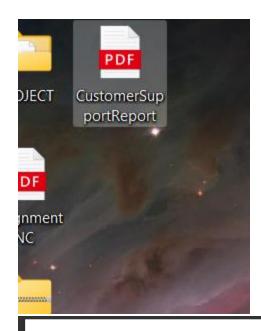


Customer Support Form:









Customer Support Report

| SUPPORTTICKETI D | SUBMITTEDBYUS ERID | ISSUEDESCRIPTI ON | DATESUBMITTED | RESOLUTIONSTA TUS |
|---------------------|-----------------------|------------------------------------|-----------------------|----------------------|
| 2 | 51 | This is a new complain | 5/12/2024 3:52:49 PM | Resolved |
| 3 | 55 | Hello this is a complain from huma | 5/12/2024 6:37:57 PM | Open |
| 1 | 51 | Example Complain | 5/12/2024 3:04:24 PM | In Progress |
| 4 | 51 | This has been added right now | 5/12/2024 11:21:40 PM | Open |

