

PATIENT MANAGEMENT SYSTEM

ASHFIRUNA ZAMAN

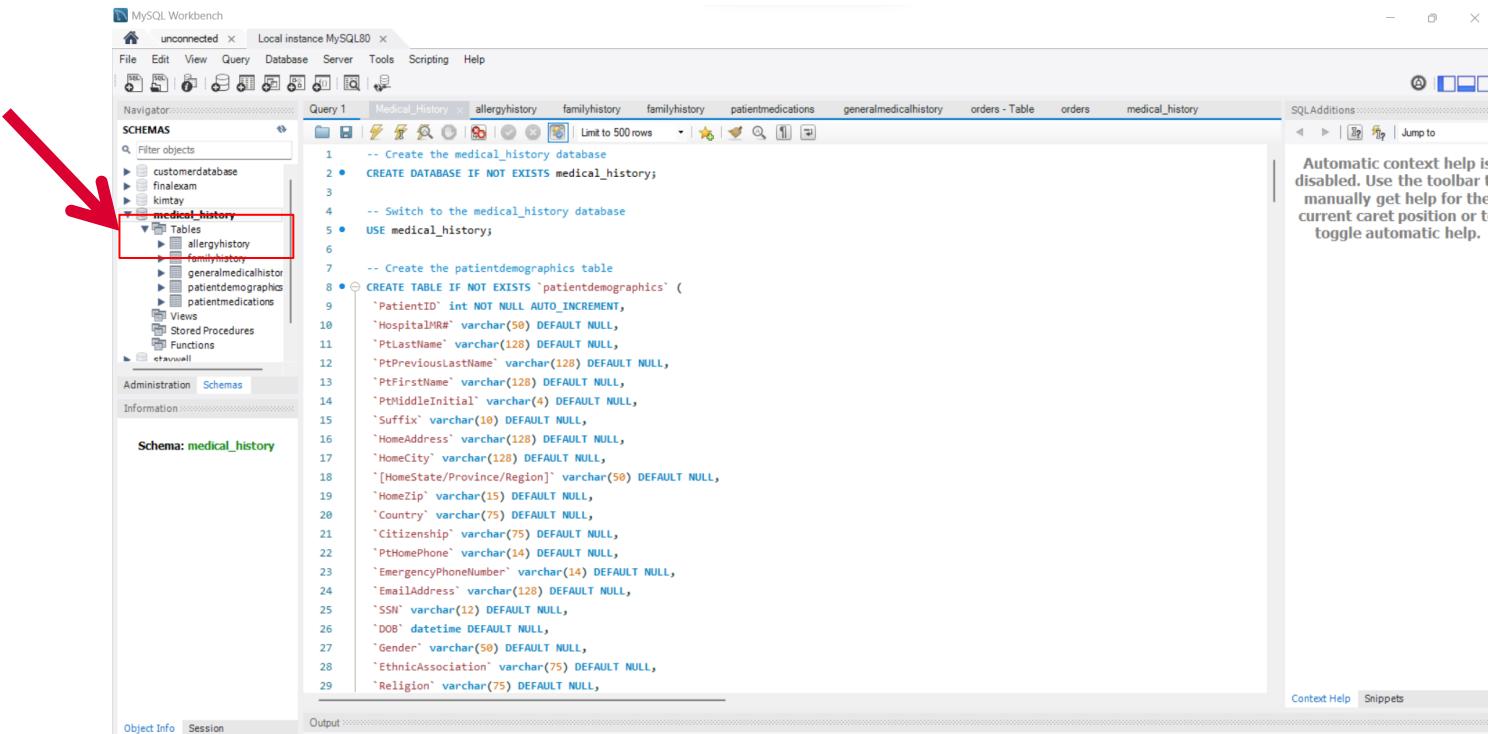
MARUF HOSSAIN



OVERVIEW

- Console Application Name “Patient Management System”
- Created Database Name “medical_history”
- Needed 6 forms including
selectpatient,demographics,allergyinformation,medicalhistory,medicationinformation and
familyhistory

MEDICAL HISTORY DATABASE



The screenshot shows the MySQL Workbench interface with the 'medical_history' database selected. The 'Tables' section in the 'Schemas' tree view is highlighted with a red box and a red arrow pointing to it. The table names listed are: allergyhistory, familyhistory, generalmedicalhistory, patientdemographics, and patientmedications.

```
-- Create the medical_history database
CREATE DATABASE IF NOT EXISTS medical_history;
USE medical_history;

-- Create the patientdemographics table
CREATE TABLE IF NOT EXISTS `patientdemographics` (
    `PatientID` int NOT NULL AUTO_INCREMENT,
    `HospitalID` varchar(50) DEFAULT NULL,
    `PtLastName` varchar(128) DEFAULT NULL,
    `PtPreviousLastName` varchar(128) DEFAULT NULL,
    `PtFirstName` varchar(128) DEFAULT NULL,
    `PtMiddleInitial` varchar(4) DEFAULT NULL,
    `Suffix` varchar(10) DEFAULT NULL,
    `HomeAddress` varchar(128) DEFAULT NULL,
    `HomeCity` varchar(128) DEFAULT NULL,
    `HomeState/Province/Region` varchar(50) DEFAULT NULL,
    `HomeZip` varchar(15) DEFAULT NULL,
    `Country` varchar(75) DEFAULT NULL,
    `Citizenship` varchar(75) DEFAULT NULL,
    `PtHomePhone` varchar(14) DEFAULT NULL,
    `EmergencyPhoneNumber` varchar(14) DEFAULT NULL,
    `EmailAddress` varchar(128) DEFAULT NULL,
    `SSN` varchar(12) DEFAULT NULL,
    `DOB` datetime DEFAULT NULL,
    `Gender` varchar(50) DEFAULT NULL,
    `EthnicAssociation` varchar(75) DEFAULT NULL,
    `Religion` varchar(75) DEFAULT NULL
)
```

- Included five tables
- Connected the database with the project

PATIENT DEMOGRAPHICS TABLE

```
INSERT INTO `patientdemographics` ( `HospitalMR#`, `PtLastName`, `PtPreviousLastName`, `PtFirstName`, `PtMiddleInitial`,  
`Suffix`, `HomeAddress`, `HomeCity`, `HomeState/Province/Region`, `HomeZip`, `Country`, `Citizenship`, `PtHomePhone`,  
`EmergencyPhoneNumber`, `EmailAddress`, `SSN`, `DOB`, `Gender`, `EthnicAssociation`, `Religion`, `MaritalStatus`,  
`EmploymentStatus`, `DateofExpire`, `Referral`, `CurrentPrimaryHCPId`, `Comments`, `DateEntered`, `NextOfKinID`,  
`NextOfKinRelationshipToPatient`, `deleted` )
```

The screenshot shows the MySQL Workbench interface with the following details:

- File Menu:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas:** Navigator pane shows the schema structure, including the **medical_history** database which contains tables like **allergyhistory**, **familyhistory**, **generalmedicalhistory**, **patientdemographics**, and **patientmedications**.
- SQL Editor:** The current query is `SELECT * FROM medical_history.patientdemographics;`. The results are displayed in a grid format.
- Result Grid:** The grid displays 13 rows of data from the **patientdemographics** table. The columns are: PatientID, HospitalMR #, PtLastName, PtPreviousLastName, PtFirstName, PtMiddleInitial, Suffix, HomeAddress, HomeCity, HomeState, HomeZip, Country, Citizenship, PtHm. The data includes entries for patients like Karim, Gaffar, Rahman, Ad, Eikram, Zulsalal, Duha, Abu, Ali, Samira, Aboud, Kareem, Hassan, Aisha, Fard, Omar, Ahmed, Fatima, Karim, Amer, Hasan, Noor, Ibrahim, Bilal, and others.
- Toolbar:** Includes icons for New, Open, Save, Print, Copy, Paste, Find, Refresh, and other database operations.
- Help:** SQL Additions, Jump to, and Automatic context help is disabled.
- Information:** Shows table details for **allergyhistory**, including columns: AllergyID, PatientID, AllergyType, AllergyStartDate, AllergyEndDate, AllergyDescription, and deleted.

GENERAL MEDICAL HISTORY

```
INSERT INTO `generalmedicalhistory`
(`PatientID`, `MaritalStatus`, `Education`,
`BehavioralHistory`,
`Tobacco`, `TobaccoQuantity`, `TobaccoDuration`,
`Alcohol`,
`AlcoholQuantity`, `AlcoholDuration`, `Drug`,
`DrugType`,
`DrugDuration`, `Dietary`, `BloodType`, `Rh`,
`NumberOfChildren`,
`LMPStatus`, `MensesMonthlyYes`,
`MensesMonthlyNo`,
`MensesFreq`, `MedicalHistoryNotes`,
`HxObtainedBy`, `deleted`
```

The screenshot shows the MySQL Workbench interface with the 'generalmedicalhistory' table selected in the 'generalmedicalhistory' schema. The table has the following columns and data:

GeneralMedicalHistoryID	PatientID	MaritalStatus	Education	BehavioralHistory	Tobacco	TobaccoQuantity	TobaccoDuration	Alcohol	AlcoholQuantity
1	1	Single	College Graduate	Non-smoker	No	N/A	N/A	No	N/A
2	2	Married	High School Graduate	Ex-smoker	No	N/A	N/A	Yes	2-3 drinks per week
3	3	Divorced	Some College	Social smoker	Yes	One pack per day	10 years	Yes	2-3 drinks per week
4	4	Single	College Graduate	No significant behavioral history	No	N/A	N/A	Occasional	1-2 drinks per week
5	5	Married	Some College	Non-smoker	No	N/A	N/A	Yes	3-5 drinks per week
6	6	Divorced	Master's Degree	Non-smoker	YES	Half a pack per day	8 years	NO	N/A
7	7	Widowed	High School Graduate	Social smoker	Yes	One pack per day	15 years	Yes	2-3 drinks per week
8	8	Married	Edward College	Ex-smoker	NO	N/A	N/A	YES	2-4 drinks per week
9	9	Widowed	High School Graduate	Non-smoker	No	N/A	N/A	Yes	2-4 drinks per week
10	10	Married	Some College	Non-smoker	No	N/A	N/A	Yes	2-4 drinks per week
11	6	Divorced	Master's Degree	Non-smoker	YES	Half a pack per day	8 years	NO	N/A
12	5	Married	Some College	Non-smoker	NO	N/A	N/A	YES	3-5 drinks per week

ALLERGY HISTORY

```
INSERT INTO `allergyhistory` (
`PatientID`, `Allergen`, `AllergyStartDate`,
`AllergyEndDate`, `AllergyDescription`,
`deleted`
```

The screenshot shows the MySQL Workbench interface with the 'Medical_History' schema selected. A query window displays the following SQL command:

```
1 • SELECT * FROM medical_history.allergyhistory;
```

The results grid shows the data from the 'allergyhistory' table:

AllergyID	PatientID	Allergen	AllergyStartDate	AllergyEndDate	AllergyDescription	deleted
1	1	Pollen	2010-01-01	2022-12-09	Seasonal allergy to pollen	0
2	2	Penillin	2015-05-10	2016-05-10	Allergic reaction, discontinued medication	0
3	3	Shelfish	2012-03-15 00:00:00	2022-12-09 00:00:00	Shelfish allergy, avoids seafoods	0
4	4	Pollen	2022-12-09	2022-12-09	No known allergies	0
5	5	Dust mites	2018-08-20 00:00:00	2022-12-09 00:00:00	Slight allergy, managed by regular cleaning	0
6	6	Penillin	2022-12-09	2022-12-09	Penut Allergies	0
7	7	Peanuts	2014-02-01	2022-12-09	Peanut allergy, carries an epinephrine injector	0
8	8	Peanuts	2022-12-09	2022-12-09	No known allergies	0
9	9	Gluten	2022-12-09	2022-12-09	Allergic reaction, discontinued medication	0
10	10	Sugar	2022-12-09	2022-12-09	No known allergies	0
11	4	Pollen	2022-12-09 00:00:00	2022-12-09 00:00:00	No known allergies	0
12	4	Pollen	2022-12-09 00:00:00	2022-12-09 00:00:00	No known allergies	0
13	1	Pollen	2010-01-01 00:00:00	2022-12-09 00:00:00	Seasonal allergy to pollen	0

FAMILY HISTORY

```
INSERT INTO `familyhistory` (`PatientID`, `Name`, `Relation`, `Alive`, `Lives with patient`,  
`MajorDisorder`, `SpecificTypeDisorder`, `deleted`)
```

The screenshot shows the MySQL Workbench interface with a query results grid. The grid displays data from the 'familyhistory' table, which has columns: FamilyID, PatientID, Name, Relation, Alive, Lives with patient, MajorDisorder, SpecificTypeDisorder, and deleted. The data includes various family members (Abdul Rahman, Muhammad Gaffar, etc.) with their relations (Father, Mother, Sister, Brother, Son, Daughter) and health conditions (Heart Disease, Coronary Artery Disease, Cancer, Diabetes, Hypertension, Asthma, Allergies, etc.).

FamilyID	PatientID	Name	Relation	Alive	Lives with patient	MajorDisorder	SpecificTypeDisorder	deleted
1	1	Abdul Rahman	Father	0	0	Heart Disease	Coronary Artery Disease	0
4	2	Muhammad Gaffar	Father	0	0	Cancer	Lung Cancer	0
7	3	Ahmed Duha	Father	1	0	Diabetes	Type 1 Diabetes	0
8	3	Layla Duha	Mother	0	0	Cancer	Breast Cancer	0
9	4	Ibrahim Ebram	Father	1	0	Hypertension	N/A	0
10	4	Sofia Ebram	Mother	1	0	Asthma	N/A	0
11	4	Amina Ebram	Sister	1	0	Allergies	Pollen Allergy	0
12	5	Yusuf Ad	Father	0	0	Heart Disease	Coronary Artery Disease	0
13	5	Khadja Ad	Mother	0	0	Hypertension	N/A	0
14	6	Amina Ali	Mother	1	0	Cancer	Breast Cancer	0
15	7	Hassan Aboud	Father	1	0	Heart Disease	Coronary Artery Disease	0
16	7	Sara Aboud	Mother	1	0	Hypertension	N/A	0
17	8	Sofia Hassan	Mother	0	0	Diabetes	Type 2 Diabetes	0
18	9	Ahmed Farid	Father	1	0	Hypertension	N/A	0
19	10	Ali Omar	Father	1	0	Asthma	N/A	0
20	10	Layla Omar	Mother	1	0	Heart Disease	Coronary Artery Disease	0
21	10	Yusuf Omar	Brother	1	0	Diabetes	Type 2 Diabetes	0
22	11	Fatima Ali	Mother	1	0	Hypertension	N/A	0
23	12	Muhammad Karimi	Father	0	0	Heart Disease	Coronary Artery Disease	0
24	13	Asha Hasan	Mother	0	0	Cancer	Breast Cancer	0

PATIENT MEDICATIONS

- INSERT INTO `patientmedications` (`PatientID`, `Medication`, `MedicationAmt`, `MedicationUnit`, `Instructions`, `MedicationStartDate`, `MedicationEndDate`, `PrescriptionHCP`, `deleted`)

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the database structure under the schema `medical_history`, including tables like `allergyhistory`, `familyhistory`, `generalmedicalhistory`, `patientdemographics`, and `patientmedications`.
- Query Editor:** Displays the query `SELECT * FROM medical_history.patientmedications;`
- Result Grid:** Shows the data for the `patientmedications` table.
- Table Definition:** Shows the columns and their definitions for the `allergyhistory` table.

Table: allergyhistory

Columns:	AlergyID	PatientID	Allergen	AllergyStartDate	AllergyEndDate	AllergyDescription	deleted	
int AI PK	1	N/A	Aspirin	2022-01-15	2022-01-15	Take 1 tablet daily in the morning	0	
int	2	N/A	Mettformin	500 mg	2022-03-10	2022-03-10	Take 1 tablet with breakfast and 1 tablet with d...	0
varchar(255)	3	N/A	Albuterol Inhaler	N/A	2022-03-20	N/A	Use as needed for asthma symptoms	0
varchar(255)	4	N/A	None	N/A	N/A	N/A	No current medications	0
varchar(255)	5	N/A	Fluticasone Nasal Spray	50 mcg	2022-04-05	2022-04-05	Use 1 spray in each nostril once daily	0
tinyint(1)	6	N/A	None	N/A	N/A	N/A	No current medications	0
tinyint(1)	7	N/A	Epinephrine Injector	N/A	2022-05-10	2022-05-10	Use in case of severe allergic reaction to peanuts	0
tinyint(1)	8	N/A	None	N/A	N/A	N/A	No current medications	0
tinyint(1)	9	20	Insulin	units	2022-06-01	2022-06-01	Take as directed for diabetes	0
tinyint(1)	10	N/A	None	N/A	N/A	N/A	No current medications	0

Select Patient Form

Patient General Medical History

Select Patient

Patient Information

First Name: Last Name: Contact:

Patient List

PatientID	HospitalMR#	PtLastName	PtPreviousLastName	PtFirstName	PtMiddleInitial	Suffix	HomeAddress
1	MR67890	Karim		Haider			456 Oak
2	MR54321	Gaffar		Rahman			789 Pine
3	MR22222	Duha		Ad			202 Map
4	MR11111	Ek		Zulzalal			101 Oak
5	MR22222	Duha		Abu			202 Map
6	MR33333	Ali		Samira			303 Ced
7	MR44444	Aboud		Kareem			404 Elm
8	MR55555	Hassan		Aisha			505 Pine
9	MR66666	Farid		Omar			606 Map
10	MR77777	Ahmed		Fatima			707 Oak
11	MR88888	Karimi		Amir			808 Pine
12	MR99999	Hasan		Noor			909 Ced
13	MR10101	Ibrahim		Bilal			1010 Elm
*							

Patient ID or Last Name: Search

Forms.Button

Demographic

Medical History

Allergy History

Family History

Medication History

SELECT PATIENT FORM

The screenshot shows the Microsoft Visual Studio IDE interface with the following details:

- File Menu:** File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help.
- Search Bar:** Search dropdown.
- Toolbox:** Search Toolbox, MySQL.Data.Core, Standard, Data, Common WPF Controls, Common XAML Controls, Validation, All WPF Controls, All XAML Controls, Navigation, Login, WebParts, AJAX Extensions, Dynamic Data, All Windows Forms, Common Controls, Containers, Menus & Toolbars, Data, Components, Printing, Dialogs, WPF Interoperability, HTML, Dialog Editor, Ribbon Editor, DataSet, Entity Framework, General, Pointer.
- Solution Explorer:** Solution 'Final Project' (1 of 1) contains Final Project, Properties, References, Resources, AllergyForm.cs, App.config, DemographicsForm.cs, FamilyHistoryForm.cs, MedicalHistoryForm.cs, MedicationForm.cs, packages.config, Program.cs, and SelectPatient.cs.
- Code Editor:** The main editor window displays the code for `SelectPatient.cs` (Design). The code initializes MySQL connections and handles various UI events like Click for picture boxes and CellClick for data grids.
- Output Window:** Shows the following messages:
 - No issues found
 - The thread 0x188 has exited with code 0 (0x0).
 - The thread 0x2f48 has exited with code 0 (0x0).
 - The program '[28984] Final Project.exe' has exited with code 4294967295 (0xffffffff).
- Status Bar:** Ln: 104 Ch: 33 SPC CRLF.

Project Build Debug Test Analyze Tools Extensions Window Help | Search ▾ Final Project

Debug Any CPU Start Align by... abc abc abc abc abc abc abc

SelectPatient.cs* SelectPatient.cs [Design]* MedicalHistory...rm.cs [Design] MedicalHistoryForm.cs AllergyForm.cs [Design]

C# Final Project Final_Project.SelectPatient LoadPatientList(string searchCriteria)

```
58     }
59     2 references
60     private void LoadPatientList(string searchCriteria)
61     {
62         try
63         {
64             // Open the database connection
65             connection.Open();
66
67             // SQL query to search for patients based on criteria
68             string query = "SELECT * FROM PatientDemographics WHERE " +
69             "PTFirstName LIKE @SearchCriteria OR " +
70             "PTLastName LIKE @SearchCriteria OR " +
71             "PatientID = @SearchCriteriaInt";
72
73             // Create a MySqlCommand with the query and connection
74             using (MySqlCommand command = new MySqlCommand(query, connection))
75             {
76                 // Add a parameter for searching patient names
77                 command.Parameters.AddWithValue("@SearchCriteria", "%" + searchCriteria + "%");
78
79                 int searchCriteriaInt;
80                 if (int.TryParse(searchCriteria, out searchCriteriaInt))
81                 {
82                     // Add a parameter for searching patient ID (if a valid integer)
83                     command.Parameters.AddWithValue("@SearchCriteriaInt", searchCriteriaInt);
84                 }
85                 else
86                 {
87                     // Add a default value for searching patient ID (if searchCriteria is not a valid integer)
88                     command.Parameters.AddWithValue("@SearchCriteriaInt", -1);
89                 }
90
91                 // Create a MySqlDataAdapter to retrieve data
92                 using (MySqlDataAdapter adapter = new MySqlDataAdapter(command))
93                 {
94                     // Create a DataTable to store patient data
95                     DataTable patientData = new DataTable();
96
97                     // Fill the DataTable with data from the database
98                     adapter.Fill(patientData);
99
100                    // Set the DataGridView's data source to the patient data
101                    dataGridView1.DataSource = patientData;
102                }
103            }
104        catch (Exception ex)
105        {
106            // Display an error message if an exception occurs
107            MessageBox.Show("Error: " + ex.Message);
108        }
109    }
110 }
```

57 % No issues found

Output

Show output from: Debug

The thread 0xt88 has exited with code 0 (0x0).
The thread 0x2f48 has exited with code 0 (0x0).

- This code retrieves patient data based on search criteria and displays it
- Parameterized queries to protect against SQL injection
- Error handling to gracefully handle exceptions

Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help | Search Final Project

Sign in  - 

[22424] Final Project.exe | Lifecycle Events | Thread: [18796] Main Thread | Stack Frame: Final_Project.MedicalHistoryForm.button2

Select Patient Form

Patient General Medical History

Patient Information

First Name: Last Name: Contact:

Patient List

PatientID	HospitalMR#	PtLastName	PtPreviousLastName	PtFirstName	PtMiddleInitial	Suffix	HomeAddress
1	MR67890	Karim		Haider			456 Oak
2	MR54321	Gaffar		Rahman			789 Pine
3	MR22222	Duha		Ad			202 Map
4	MR11111	Ekram		Zulzalal			101 Oak
5	MR22222	Duha		Abu			202 Map
6	MR33333	Ali		Samira			303 Cedar
7	MR44444	Aboud		Kareem			404 Elm
8	MR55555	Hassan		Aisha			505 Pine
9	MR66666	Farid		Omar			606 Map
10	MR77777	Ahmed		Fatima			707 Oak
11	MR88888	Karimi		Amir			808 Pine
12	MR99999	Hasan		Noor			909 Cedar
13	MR10101	Ibrahim		Bilal			1010 Elm

Properties

Search

Demographic

Medical History

Allergy History

Family History

Medication History

SELECT PATIENT FORM

Field 'AllergyForm.isAddMode' is never assigned to, and will always have its default value false

Final Project AllergyForm.cs 23 Active

Call Stack Breakpoints Exception Settings Command Window Immediate Window Output Error List



Patient General Medical History

Select Patient

Patient ID or Last Name: 1

Search

Patient Information

First Name: Rahman

Last Name: Gaffar

Contact: 111-222-3333

Patient List



Demographic



Medical History



Allergy History



Example 11.1

DemographicsForm.cs Demographics....cs [Design] MedicalHistory...rm.cs [Design] MedicalHistoryForm.cs AllergyForm.cs [Design]

Patient Demographics Form

Patient General Medical History

Demographic Information

Patient ID:	Country:	CurrHCPId:
Hospital MR#:	Citizen:	Comment:
Last Name:	Home Phone#:	Date Entered:
Prev. Last Name:	Emerg. Phone#:	11/27/2023
First Name:	Email:	Next of Kin ID:
Prev. First Name:	SSN:	Kin Relationship:
Middle Initial:	Date of Birth:	Add
Suffix:	Gender:	Modify
Home Address:	Ethnicity:	Save
Home City:	Religion:	Undo
Home State:	Martial Status:	Delete
Home Zip:	Employ. Status:	Demographic
	Date of Expire:	Medical History
	Referral:	Allergy History
		Family History

Demographic Table

Output

Show output from: Debug

The thread 0xb53c has exited with code 0 (0x0).
The thread 0x57ac has exited with code 0 (0x0).
The program '[22424] Final Project.exe' has exited with code 4294967295 (0xffffffff).

Solution Explorer

Search Solution Explorer (Ctrl+F)

Solution 'Final Project' (1)

- Final Project
 - Properties
 - References
 - Resources
 - AllergyForm.cs
 - App.config
 - DemographicsForm.cs
 - FamilyHistoryForm.cs
 - MedicalHistoryForm.cs
 - MedicationForm.cs
 - packages.config
 - Program.cs
 - SelectPatient.cs

Properties

DemographicsForm System.Windows.Forms.Form

- RightToLeft
- RightToLeftLayout
- ShowIcon
- ShowInTaskbar
- Size
- SizeGripStyle
- StartPosition
- Tag
- Text
- TopMost

Text

The text associated with the control.

The screenshot shows a Windows application window titled "Demographic Form". The main area contains a search interface with fields for "First Name", "Last Name", and "Patient ID", each with a dropdown arrow button. Below these fields is a "Search" button. To the right of the search fields is a "Clear All" button. At the bottom of the window is a "Close" button.

Demographic Form

First Name:

Last Name:

Patient ID:

Search

Clear All

Close

DEMOGRAPHIC FORM

Final Project.exe Lifecycle Events Thread: [18576] Main Thread Stack Frame: Final_Project.DemographicsForm.dataGrid

Patient General Medical History

Demographic Information

Patient ID:	1	Country:	Palestine	CurrHCPId:	PrimaryHCP567
Hospital MR#:	MR67890	Citizen:	Citizen	Comment:	No specific comments
Last Name:	Karim	Home Phone#:	555-123-4567	Date Entered:	11/27/2023
Prev. Last Name:		Emerg. Phone#:	555-987-6543	Next of Kin ID:	KinID789
First Name:	Haider	Email:	haider.karim@email.com	Kin Relationship:	Friend
Prev. First Name:		SSN:	987-65-4321	Add	Modify
Middle Initial:		Date of Birth:	11/27/2023	Save	Undo
Suffix:		Gender:		Delete	
Home Address:	456 Oak St	Ethnicity:	EthnicGroup2		
Home City:	Townsville	Religion:	Islam		
Home State:	Provinceville	Martial Status:	Single		
Home Zip:	56789	Employ. Status:	Employed		
		Date of Expire:	11/27/2023		
		Referral:	Friend Referral		

Demographic Table

PatientID	HospitalMR#	PtLastName	PtPreviousLastName	PtFirstName	PtMiddleInitial	Suffix	HomeAddress
1	MR67890	Karim		Haider			456 Oak
2	MR54321	Gaffar		Rahman			789 Pine
3	MR22222	Duha		Ad			202 Map

Field 'AllergyForm.isAddMode' is never assigned to, and will always have its default value false.

DEMOGRAPHIC FORM



Demographic



Medical History



Allergy History



Family History



Medication History

DemographicsForm.cs* Demographics...m.cs [Design]* MedicalHistor...m.cs [Design] * MedicalHistoryForm.cs AllergyForm.cs [Design]

Medical History Form

Patient General Medical History

Select Patient

Medical Information

Patient ID: Drug: Medical History Note:

Medical History ID: Drug Type:

Martial Status: Drug Duration:

Education: Dietary:

Behavioral History: Blood Type:

Tobacco: RH (+) or (-): Number of Children:

Tobacco Quantity: LMP Status:

Tobacco Duration: Menses Monthly:

Alcohol: Menses Freq:

Alcohol Quantity: Hx Obtained By:

Add Modify Save Undo Delete

Medical History Table

MEDICAL HISTORY FORM

Patient Name:
Patient Age:

Demographic 

Medical History 

Allergy History 

Family History 

Medication History 

Properties MedicalHistoryForm.cs File Properties

Build Action Copy to Output Directory Custom Tool Custom Tool Namespace File Name Full Path

The screenshot shows the Microsoft Visual Studio IDE interface. The top menu bar includes File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, and Search. The toolbar has icons for Undo, Redo, Cut, Copy, Paste, Find, and Replace. The status bar at the bottom shows 'Ready'.

The main area displays the code for the `LoadGeneralMedicalHistory()` method in the `MedicalHistoryForm.cs` file. The code uses MySQL commands to retrieve medical history data from a database, handles exceptions, and populates various UI controls like TextBoxes and ComboBoxes with the retrieved data.

```
private void LoadGeneralMedicalHistory()
{
    try
    {
        // Open the database connection
        connection.Open();

        // SQL query to retrieve general medical history for a specific patient
        string query = "SELECT * FROM medical_history_general WHERE PatientID = @PatientID";
        using (MySqlCommand command = new MySqlCommand(query, connection))
        {
            // Provide the patient ID as a parameter to filter the results
            command.Parameters.AddWithValue("@PatientID", selectedPatientID);

            using (MySqlDataAdapter adapter = new MySqlDataAdapter(command))
            {
                // Create a DataTable to store the general medical history data
                DataTable medicalHistoryData = new DataTable();
                adapter.Fill(medicalHistoryData);
                if (medicalHistoryData.Rows.Count > 0)
                {
                    // Set TextBox and ComboBox values based on the retrieved data
                    textBox1.Text = medicalHistoryData.Rows[0]["PatientID"].ToString();
                    textBox1.Text = medicalHistoryData.Rows[0]["GeneralMedicalHistoryID"].ToString();
                    textBox2.Text = medicalHistoryData.Rows[0]["MaritalStatus"].ToString();
                    textBox2.Text = medicalHistoryData.Rows[0]["ReligiousSect"].ToString();
                    comboBox1.Text = medicalHistoryData.Rows[0]["Ethnicity"].ToString();
                    textBox3.Text = medicalHistoryData.Rows[0]["AlcoholQuantity"].ToString();
                    textBox4.Text = medicalHistoryData.Rows[0]["AlcoholType"].ToString();
                    comboBox2.Text = medicalHistoryData.Rows[0]["AlcoholReason"].ToString();
                    textBox5.Text = medicalHistoryData.Rows[0]["AlcoholHealthImpact"].ToString();
                    comboBox3.Text = medicalHistoryData.Rows[0]["DrugsList"].ToString();
                    textBox6.Text = medicalHistoryData.Rows[0]["DrugsQuantity"].ToString();
                    textBox7.Text = medicalHistoryData.Rows[0]["DrugsType"].ToString();
                    textBox8.Text = medicalHistoryData.Rows[0]["DrugsReason"].ToString();
                    comboBox4.Text = medicalHistoryData.Rows[0]["MedicationID"].ToString();
                    textBox9.Text = medicalHistoryData.Rows[0]["MedicationName"].ToString();
                    textBox10.Text = medicalHistoryData.Rows[0]["MedicationType"].ToString();
                    textBox11.Text = medicalHistoryData.Rows[0]["MedicationHealthImpact"].ToString();
                    textBox12.Text = medicalHistoryData.Rows[0]["MedicationReason"].ToString();
                }
            }
        }
    catch (Exception ex)
    {
        // Display an error message if an exception occurs during data retrieval
        MessageBox.Show("Error: " + ex.Message);
    }
    finally
    {
        // Close the database connection to release resources
        connection.Close();
    }
}
```

Patient Data Retrieval:

- This code retrieves comprehensive general medical history data for a selected patient from a database.

User Interface Integration:

- It seamlessly populates various TextBoxes and ComboBoxes with the retrieved data, enhancing the user interface's functionality and usability.

Error Handling and Database Connection Management:

- The code incorporates robust error handling to gracefully handle exceptions during data retrieval.
- It ensures proper opening and closing of the database connection, preventing resource leaks and maintaining system stability.

MEDICAL HISTORY FORM

File Edit View Git Project Build Debug Format Test Analyze Tools Extensions Window Help | Search ▾ Final Project

Process: [13292] Final Project.exe | Lifecycle Events ▾ Thread: [18576] Main Thread | Stack Frame: Final_Project.DemographicsForm.dataGrid

Application Insights ▾ Live Share

Medical History Form

Patient General Medical History

Select Patient

Medical Information

Patient ID:	2	Drug:	Prescription medication
Medical History ID:	2	Drug Duration:	Long-term
Marital Status:	Married	Dietary:	Balanced diet
Education:	High School Graduate	Blood Type:	A+
Behavioral History:	Ex-smoker	RH (+) or (-):	
Tobacco:		Number of Children:	2
Tobacco Quantity:	N/A	LMP Status:	Regular
Tobacco Duration:	N/A	Menses Monthly:	
Alcohol:		Menses Freq:	Monthly
Alcohol Quantity:	2-3 drinks per week	Hx Obtained By:	Physician-reported
Alcohol Duration:	Social drinker		

Add Modify Save Undo Delete

Rahman Gaff

Patient Age

Demographic

Medical History

Allergy History

Properties

calHistoryForm System.Windows.Forms.Form

mBorderStyle Sizable

ipButton False

eMode NoC

idiContainer False

/Preview (Def

nguage False

talizable False

itation 0, 0

cked False

inMenuStrip (non

ximizeBox True

ximumSize 0, 0

himizeBox True

himizeSize 0, 0

acity 100%

lding 0, 0,

htToLeft No

Toolbox

Search Toolbox

MySQL.Data.Core

Standard

Pointer

AdRotator

BulletedList

Button

Calendar

CheckBox

CheckBoxList

ContentPlaceHolder

DropDownList

FileUpload

HiddenField

HyperLink

Image

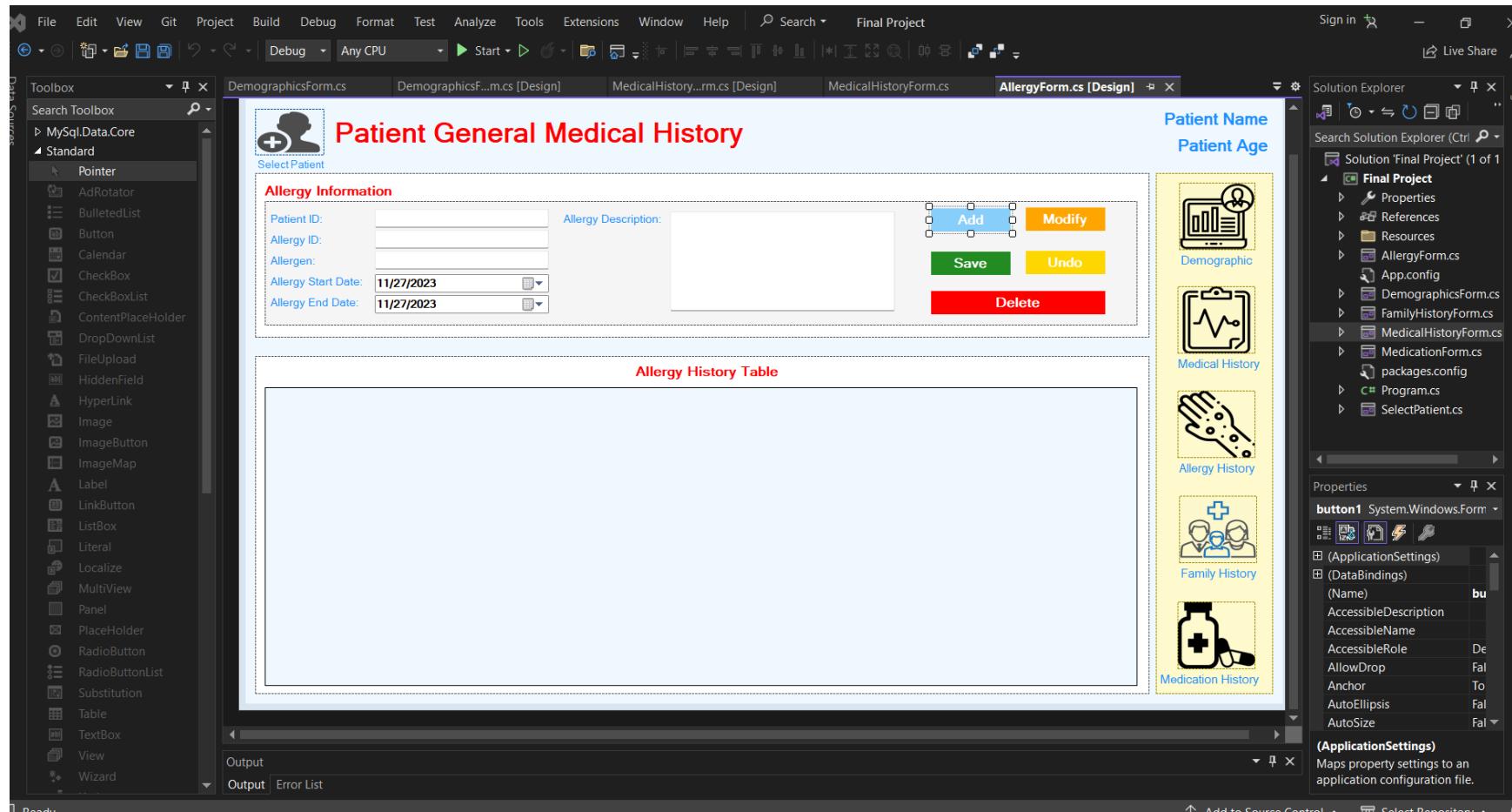
ImageButton

ImageMap

Solution Explorer

Git Changes

MEDICAL HISTORY FORM



ALLERGY FORM

The screenshot shows the Microsoft Visual Studio IDE interface. The top menu bar includes 'Git', 'Project', 'Build', 'Debug', 'Test', 'Analyze', 'Tools', 'Extensions', 'Window', 'Help', and 'Search'. Below the menu is a toolbar with icons for file operations like Open, Save, and Print. The main window displays the 'AllergyForm.cs' code under the 'Final Project' solution. The code is part of the 'AllergyForm' class, which inherits from 'Form'. It includes methods for initializing connections, loading specific patient data, and setting field readability. The code uses namespaces such as 'System', 'System.Collections.Generic', 'System.Data', 'System.Linq', 'System.Windows', and 'System.Windows.Forms'. The 'Final Project' tab is selected in the tabs at the top. The status bar at the bottom shows '52 %' completion, 'No issues found', 'Ln: 288 Ch: 13 SPC', and 'CRLF'.

```
1 using MySql.Data.MySqlClient;
2 using System;
3 using System.Collections.Generic;
4 using System.ComponentModel;
5 using System.Data;
6 using System.Drawing;
7 using System.Linq;
8 using System.Windows;
9 using System.Windows.Interservices;
10 using System.Text;
11 using System.Threading.Tasks;
12 using System.Windows.Forms;
13 using System.Windows.Forms.VisualStyles;
14 using System.Windows.Forms.VisualStyles.VisualStyleElement;
15
16 namespace Final_Project
17 {
18     public partial class AllergyForm : Form
19     {
20         private MySqlConnection connection;
21         private string selectedPatientID;
22         private string selectedPatientName;
23         private string selectedPatientLastName;
24         private string selectedAllergy;
25
26         public AllergyForm(string patientID = null)
27         {
28             InitializeComponent();
29             InitializeDatabaseConnection();
30             selectedPatientID = patientID;
31             selectedPatientName = null;
32             selectedPatientLastName = null;
33             selectedAllergy = null;
34
35             setFieldsReadOnly();
36
37             if (string.IsNullOrEmpty(selectedPatientID))
38             {
39                 LoadPatientSpecificData(selectedPatientID);
40             }
41             else
42             {
43                 LoadAllergyData();
44             }
45         }
46
47         // References
48         private void SetFieldsReadOnly()
49         {
50             // Set flag to read-only
51             textBox1.ReadOnly = true;
52             textBox2.ReadOnly = true;
53             textBox3.ReadOnly = true;
54             textBox4.ReadOnly = true;
55
56             // If you have a DateTimePicker and want to prevent editing
57             dateTimePicker1.ReadOnly = true;
58
59             // Setting up DataTimeCheckers
60             dateTimePicker1.Enabled = false;
61             dateTimePicker2.Enabled = false;
62         }
63
64         // References
65         private void LoadAllergyData()
66         {
67             try
68             {
69                 // Implementation of LoadAllergyData
70             }
71             catch (Exception ex)
72             {
73                 MessageBox.Show(ex.Message);
74             }
75         }
76     }
77 }
```

ALLERGY FORM

1. Allergy Management Interface:

1. This code defines a user interface (UI) for managing allergy-related data in a healthcare application.
 2. It allows users to view, edit, add, and delete allergy records, providing comprehensive control over patient allergy information.

2. Database Integration:

1. The code establishes a connection to a MySQL database, retrieves allergy data, and populates a DataGridView for easy data visualization.
 2. It includes methods for loading all allergy data, loading data specific to a selected patient, and saving edited or new allergy records back to the database.



Patient General Medical History

Select Patient

Allergy Information

Patient ID:	6
Allergy ID:	6
Allergen:	Penicillin
Allergy Start Date:	12/ 9/2022 <input type="button" value="▼"/>
Allergy End Date:	12/ 9/2022 <input type="button" value="▼"/>

Allergy Description:	Penut Allergies
----------------------	-----------------

<input type="button" value="Add"/>	<input type="button" value="Modify"/>
<input type="button" value="Save"/>	<input type="button" value="Undo"/>
<input type="button" value="Delete"/>	



Demographic



Medical History



Allergy History Table

	AllergyID	PatientID	Allergen	AllergyStartDate	AllergyEndDate	AllergyDescription	deleted	
▶	6	6	Penicillin	2022-12-09	2022-12-09	Penut Allergies	<input type="checkbox"/>	
*							<input type="checkbox"/>	

ALLERGY FORM

MedicationForm.cs MedicationForm.cs [Design] FamilyHistory...m.cs [Design] X

FamilyHistoryForm Patient General Medical History

Patient Name
Patient Age

Select Patient

Family Information

Patient ID:

Family ID:

Name:

Relation:

Alive: (Select One)

Lives with patient:

Add Modify

Major Disorder:

Specific Type of Disorder:

Save Undo

Delete

Family History Table

FAMILY HISTORY FORM

Output Output Error List ...

Solution Explorer

Search Solution Explorer (Ctrl+F)

Solution 'Final Project' (Final Project)

- Properties
- References
- Resources
- AllergyForm.cs
- App.config
- DemographicsForm.cs
- FamilyHistoryForm.cs
- MedicalHistoryForm.cs
- MedicationForm.cs
- packages.config
- Program.cs
- SelectPatient.cs

Properties

FamilyHistoryForm System.Windows.Forms.Form

- RightToLeft
- RightToLeftLayout
- ShowIcon
- ShowInTaskbar
- Size
- SizeGripStyle
- StartPosition
- Tag
- Text
- TopMost

Text

The text associated with the control.

The screenshot shows the Microsoft Visual Studio IDE interface. The main window displays the code editor for `FamilyHistoryForm.cs`, which contains C# code for loading family history details from a MySQL database. The code includes a try-catch block, database connection handling, and UI updates for text boxes and dropdowns. The Solution Explorer on the right shows the project structure with files like `AllergyForm.cs`, `App.config`, `DemographicsForm.cs`, `FamilyHistoryForm.cs`, `MedicalHistoryForm.cs`, `MedicationForm.cs`, `packages.config`, `Program.cs`, and `SelectPatient.cs`. The Properties window is also visible.

```
Final Project
private void LoadFamilyHistoryDetails(string patientID)
{
    try
    {
        connection.Open();

        string query = "SELECT * FROM medical_history.familyhistory WHERE PatientID = @patientID";
        MySqlCommand cmd = new MySqlCommand(query, connection);
        cmd.Parameters.AddWithValue("@patientID", patientID);

        MySqlDataAdapter adapter = new MySqlDataAdapter(cmd);
        DataTable dataTable = new DataTable();
        adapter.Fill(dataTable);

        if (dataTable.Rows.Count > 0)
        {
            dataGridView1.DataSource = dataTable;
            DataRow row = dataTable.Rows[0];

            textBox10.Text = row["PatientID"].ToString();
            textBox11.Text = row["FamilyID"].ToString();
            textBox12.Text = row["Name"].ToString();
            textBox13.Text = row["Relation"].ToString();

            // Assuming 'Alive' and 'Lives with patient' columns are tinyint(1)
            comboBox3.SelectedIndex = row["Alive"] != DBNull.Value ? Convert.ToInt32(row["Alive"]) : -1;
            comboBox1.SelectedIndex = row["Lives with patient"] != DBNull.Value ? Convert.ToInt32(row["Lives with patient"]) : -1;

            textBox13.Text = row["MajorDisorder"].ToString();
            textBox24.Text = row["SpecificTypeDisorder"].ToString();
        }
        else
        {
            MessageBox.Show("No family history data found for the selected patient.", "No Data", MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("Error: " + ex.Message);
    }
    finally
    {
        connection.Close();
    }
}

private void dataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs e)
{
}
```

Patient Name
Patient Age



Patient General Medical History

Select Patient

Family Information

Patient ID:	6
Family ID:	14
Name:	Amina Ali
Relation:	Mother
Alive:	

Lives with patient:

Major Disorder:

Specific Type of
Disorder:

Cancer

Breast Cancer

Add

Modify

Save

Undo

Delete

FAMILY HISTORY FORM



Demographic



Medical History



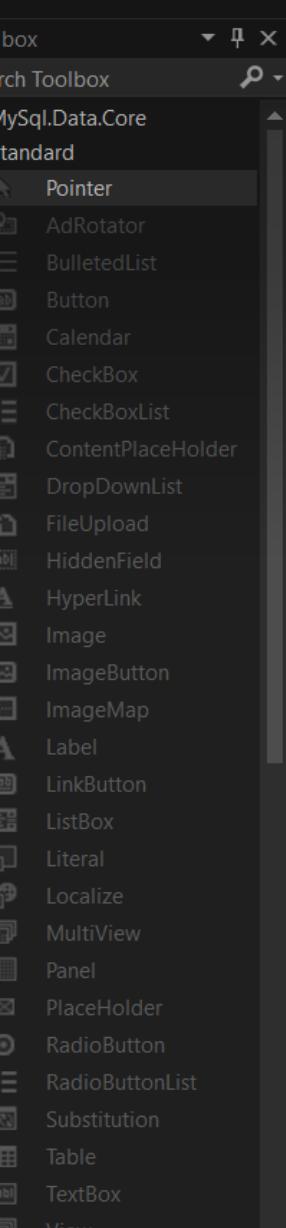
Allergy History



Family History

Family History Table

	FamilyID	PatientID	Name	Relation	Alive	Lives with patient	MajorDisorder	SpecificTyp
▶	14	6	Amina Ali	Mother	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Cancer	Breast Canc
*					<input type="checkbox"/>	<input type="checkbox"/>		



MedicationForm.cs [Design]

FamilyHistoryForm.cs [Design]

MedicationForm

Patient General Medical History

Select Patient

Medication Information

Patient ID:

Medication ID:

Medication:

Medication Amount:

Medication Unit:

Prescription HCP:

Medication Start Date:

Medication End Date:

Instruction:

Add Modify Save Undo Delete

Medication History Table

MEDICATION FORM

Patient Name
Patient Age

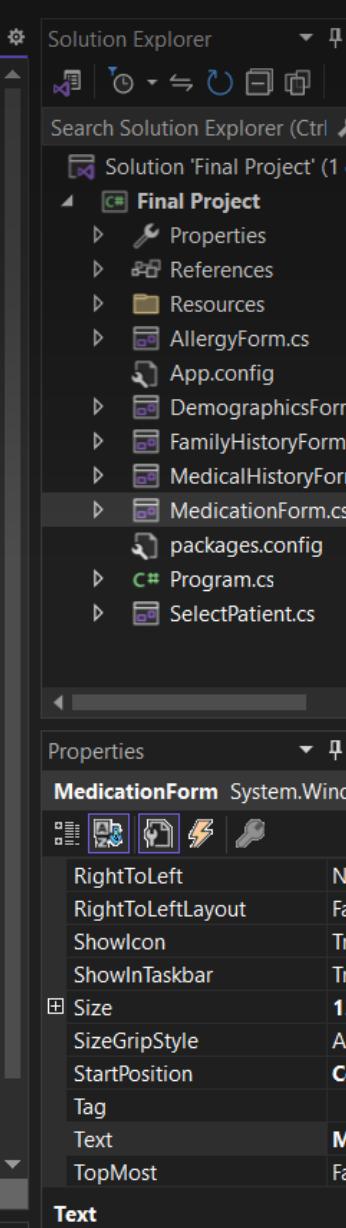
Demographic

Medical History

Allergy History

Family History

Medication History

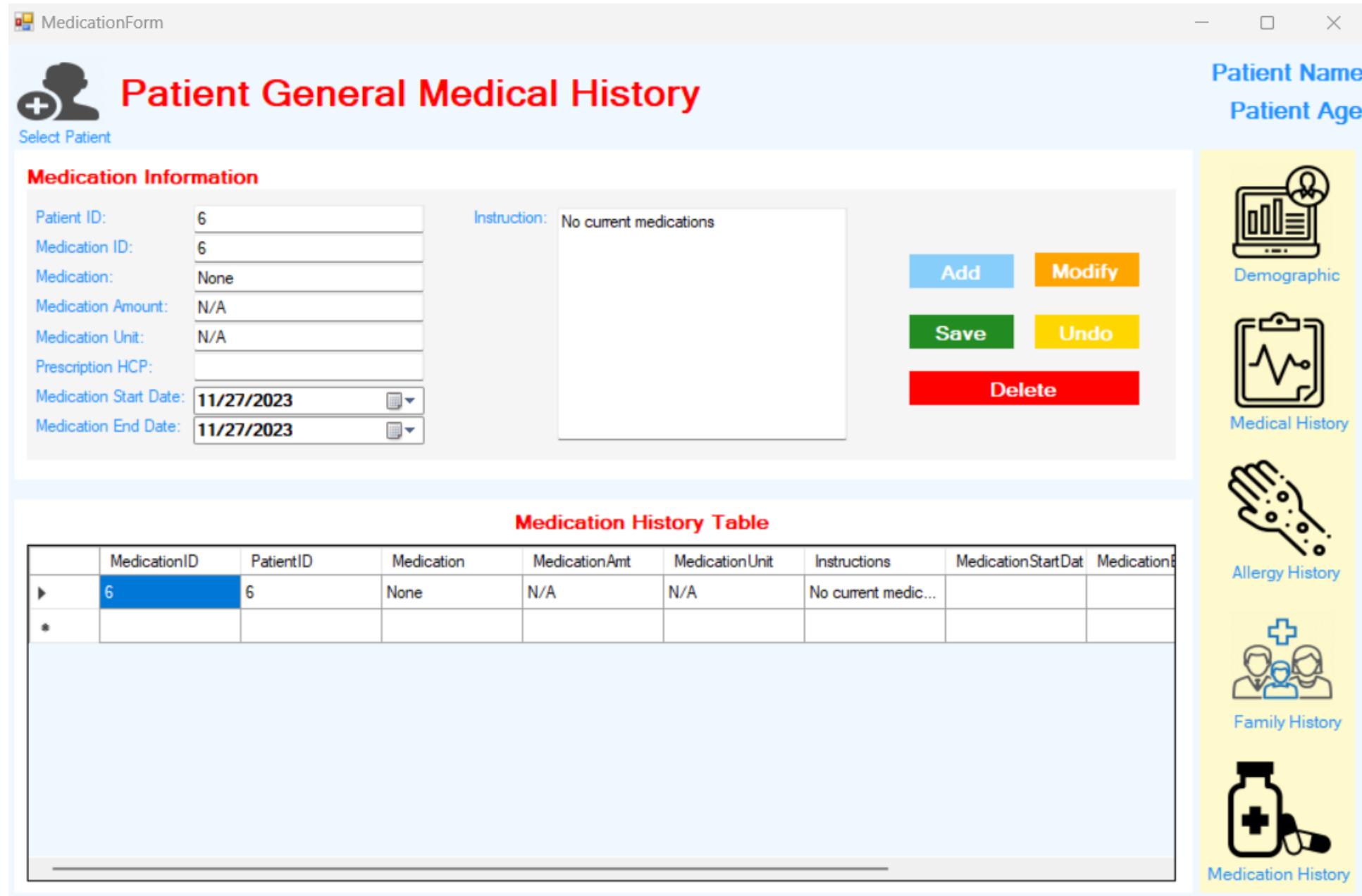


The screenshot shows the Microsoft Visual Studio IDE interface. The top menu bar includes File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, Search, and Final Project. The toolbar below has icons for file operations like Open, Save, and Print. The main window displays two tabs: MedicationForm.cs [Design] and FamilyHistoryForm.cs [Design]. The left sidebar is the Toolbox, showing categories like Data Sources, Standard, Data, etc., with the General tab selected. The right sidebar contains the Solution Explorer, Properties, and Task List. The central code editor shows C# code for a Windows application. The code connects to a MySQL database named 'medical_history' and retrieves patient medication information. It uses MySqlCommand, MySqlDataAdapter, and DataTable to handle the data. The UI is populated with text boxes (textBox10, textBox11, etc.) and a DataGridView. Error and output windows at the bottom indicate no issues found.

```
34    private void InitializeDatabaseConnection()
35    {
36        string connectionString = "Server=localhost;UserId=root;Password=toor;Database=medical_history";
37        connection = new MySqlConnection(connectionString);
38    }
39
40    private void LoadPatientInformation(string patientID)
41    {
42        try
43        {
44            connection.Open();
45
46            string query = "SELECT * FROM medical_history.patientmedications WHERE PatientID = @patientID";
47            MySqlCommand cmd = new MySqlCommand(query, connection);
48            cmd.Parameters.AddWithValue("@patientID", patientID);
49
50            MySqlDataAdapter adapter = new MySqlDataAdapter(cmd);
51            DataTable dataTable = new DataTable();
52            adapter.Fill(dataTable);
53
54            if (dataTable.Rows.Count > 0)
55            {
56                dataGridView1.DataSource = dataTable;
57                DataRow row = dataTable.Rows[0];
58
59                textBox10.Text = row["PatientID"].ToString();
60                textBox11.Text = row["MedicationID"].ToString();
61                textBox12.Text = row["MedicationName"].ToString();
62                textBox13.Text = row["MedicationStrength"].ToString();
63                textBox14.Text = row["MedicationUnit"] ? TextBox14 : null;
64                textBox15.Text = row["PrescriptionID"].ToString();
65                textBox16.Text = row["Instructions"].ToString();
66
67            }
68            else
69            {
70                MessageBox.Show("No family history data found for the selected patient.", "No Data", MessageBoxButtons.OK, MessageBoxIcon.Information);
71            }
72        }
73        catch (Exception ex)
74        {
75            MessageBox.Show("Error: " + ex.Message);
76        }
77        finally
78        {
79            connection.Close();
80        }
81    }
82
83    private void dataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs e)
84    {
85    }
86
87    private void InitializeDatabaseConnection()
88    {
89    }
```

- This code represents a user interface (UI) for managing family history records within a medical application.
- It establishes a connection to a MySQL database and retrieves family history details for a selected patient, populating the UI for easy viewing and editing
- The UI includes navigation buttons that allow users to seamlessly switch between different sections of the medical application, such as demographics, medical history, allergies, and medication

MEDICATION HISTORY FORM



File Edit View Git Project Build Debug Format Test Analyze Tools Extensions Window Help | Search ▾ Final Project

Process: [32100] Final Project.exe Lifecycle Events ▾ Thread: [18576] Main Thread Stack Frame: Final_Project.DemographicsForm.dataGrid

Toolbox ▾ Search Toolbox MySQL.Data.Core Standard Pointer AdRotator BulletedList Button Calendar CheckBox CheckBoxList ContentPlaceHolder DropDownList FileUpload HiddenField HyperLink Image ImageButton ImageMap Label LinkButton ListBox Literal Localize MultiView Panel Placeholder RadioButton RadioButtonList Substitution Table TextBox View

FamilyHistoryForm.cs MedicationForm.cs MedicationForm.cs [Design] ▾ FamilyHistoryForm.cs [Design]

MedicationForm

Patient General Medical History

Select Patient

Medication Information

Patient ID:

Medication ID:

Medication:

Medication Amount:

Medication Unit:

Prescription HCP:

Medication Start Date:

Medication End Date:

Instruction:

Add Modify Save Undo Delete

Medication History Table

Error List ...

Entire Solution 0 Errors 3 Warnings 0 of 215 Messages Build + IntelliSense Search Error List

Code	Description	Project	File	Line	Suppress
CS0169	The field 'SelectPatient.medicalHistoryForm' is never used	Final Project	SelectPatient.cs	27	Active
CS0169	The field 'SelectPatient.allergyForm' is never used	Final Project	SelectPatient.cs	28	Active
CS0169	The field 'SelectPatient.familyHistoryForm' is never used	Final Project	SelectPatient.cs	29	Active

Call Stack Breakpoints Exception Settings Command Window Immediate Window Output Error List ...

Properties MedicationForm System.Windows.Forms.Form

FormBorderStyle	Size
HelpButton	False
Icon	NoIcon
ImeMode	NoControl
IsMdiContainer	False
KeyPreview	False
Language	(Default)
Localizable	False
Location	0, 0
Locked	False
MainMenuStrip	(none)
MaximizeBox	True
MaximumSize	0, 0
MinimizeBox	True
MinimumSize	0, 0
Opacity	100%
Padding	0, 0, 0, 0
RightToLeft	No
RightToLeftLayout	False
ShowIcon	True
ShowInTaskbar	True
Size	131, 111
SizeGripStyle	Auto
StartPosition	CenterScreen
Tag	Med
Text	MedicationForm
TopMost	False
TransparencyKey	None
UseWaitCursor	False
WindowState	Normal

Solution Explorer ▾ Git Changes

Ready Add to Source Control Select Repository

```
void appreciated() {  
    if (appreciated = true) {  
        pressFollowingButton();  
    }  
  
    println("THANK YOU");  
}
```