INTERNATIONAL SCHOOL OF MANAGEMENT AND TECHNOLOGY

FACULTY OF COMPUTING

ASSIGNMENT COVER SHEET

This form is to be completed by students submitting assignments of level 4 and level 5. Students are required to complete all sections and attach to your assignment.

STUDENT DETAILS	
STUDENT NAME	Krishna Prasad Bajgai
STUDENT ID	

UNIT AND ASSIGNMENT DETAILS			
UNIT TITLE	Programming		
UNIT NUMBER	H/618/7388		
ASSIGNMENT TITLE	Introduction To Algorithms and Programming With an IDE		
ISSUE DATE	21/05/2023	DUE DATE	27/07/2023
ASSESSOR NAME	Binod Shah		
ESTIMATED WORD LENGTH	8000		

SUBMISSION	
HAND IN DATE	

DECLERATION AND ACKNOWLEDGEMENT

When submitting assignments, each student must sign a declaration confirming that the work is their own.

Plagiarism and Collusion
Plagiarism: to use or pass off as one's own, the writings or ideas of another without acknowledging or crediting the source from which the ideas are taken.

Collusion: submitting an assignment, project or report completed by another person and passing it off as one's.

In accordance with the *Academic Integrity and Plagiarism Policy*:

1. **I declare that:**

a) this assignment is entirely my own work, except where I have included fully-documented references to the work of others,
b) the material contained in this assignment has not previously been submitted for any other subject at the University or any other educational institution, except as otherwise permitted,

c) No part of this assignment or product has been submitted by me in another (previous or current) assessment, except where appropriately referenced, and with prior permission from the Lecturer / Tutor / Unit Coordinator for this unit.

2. I acknowledge that:

a) if required to do so, I will provide an electronic copy of this assignment to the assessor;

b) the assessor of this assignment may, for the purpose of assessing this assignment:

I. reproduce this assignment and provide a copy to another member of academic staff;

II. Communicate a copy of this assignment to a plagiarism checking service such as Plagiarism Check (which may then retain a copy of this assignment on its database for the purpose of future plagiarism checking).

I am aware of and understand that any breaches to the Academic Code of Conduct will be investigated and sanctioned in accordance with the College Policy.			
SIGNATURE	Krishna	DATE	27/07/2023

A report

Of

Himalayan Digital Solutions

Submitted By - Krishna Prasad Bajgai

Submitted To - Binod Shah

Table of Contents

Introduction:

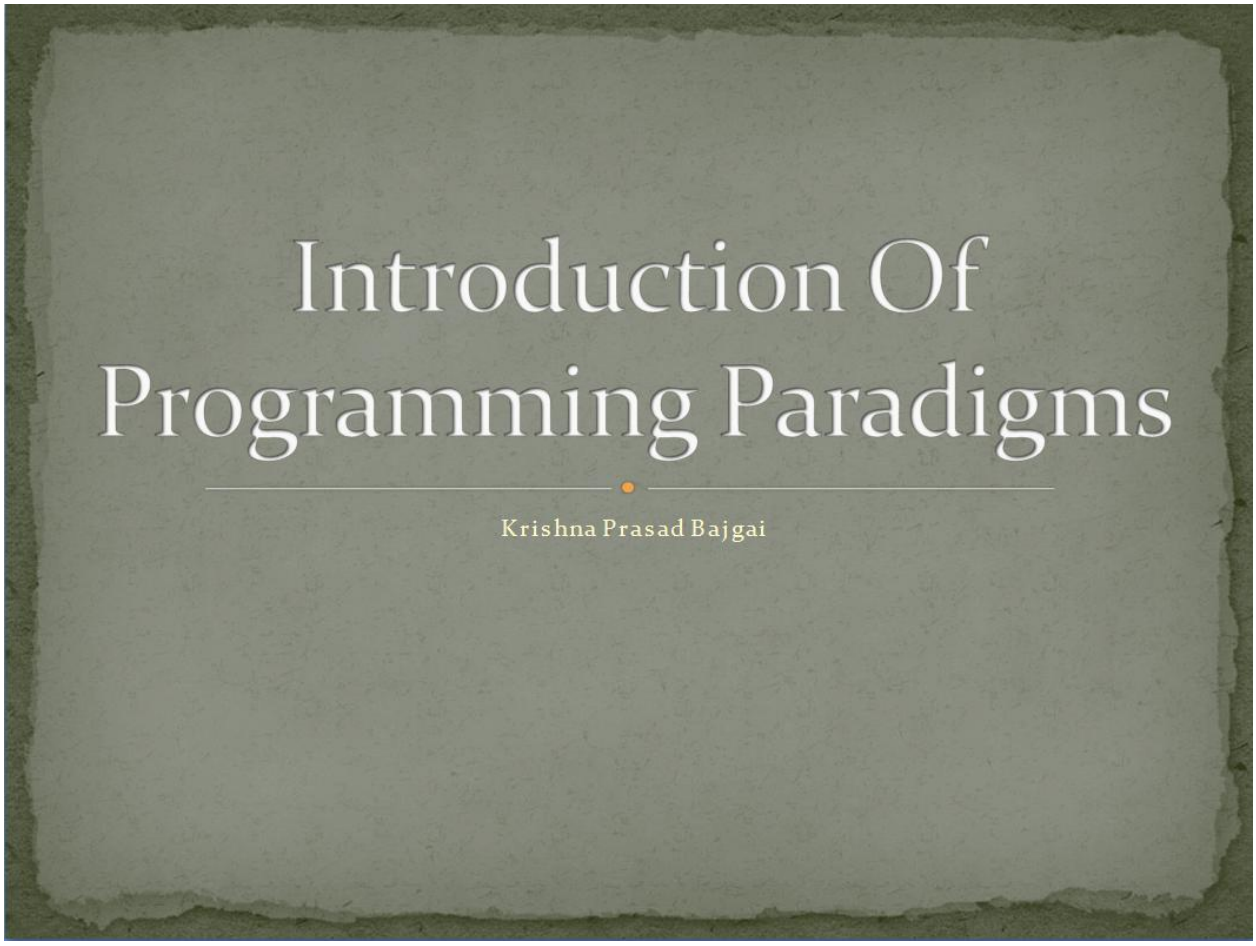
As a potential software developer employed by Himalayan Digital Solutions (HDS), I have been given the chance to collaborate with Best Fitness, a well-known local fitness training business.

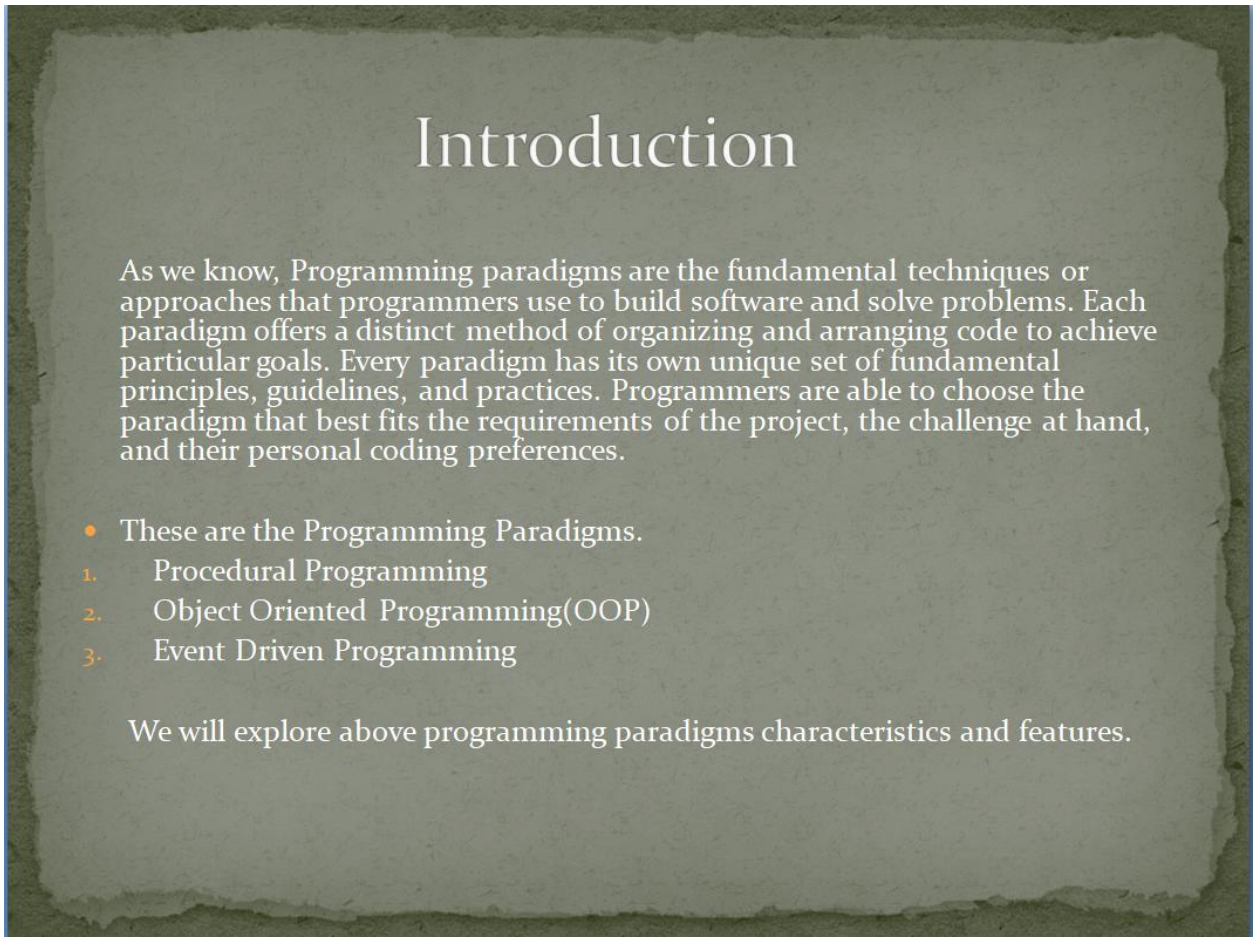
Individuals of all ages and fitness levels can count on receiving individualized training sessions from Best Fitness. To create a user-friendly tool that predicts the monthly payouts for their best athletes, they have asked HDS' help. This application must be productive, simple to use, and allow users to enter client information, training schedules, current and target weights, as well as other variable features. The final product must include a thorough cost accounting, a thorough analysis of all training expenses, and a comparison of weight objectives.

Throughout this project, my major focus will be on developing a useful algorithm, getting through any obstacles, and maintaining a smooth software development process. I would like to maintain the high standards set by HDS in addition to displaying my command of a variety of programming paradigms and my proficiency with debugging tools. My ultimate goal is to provide Best Fitness a better solution that meets their needs while showcasing the knowledge and expertise of the team at Himalayan Digital Solutions. By offering a successful and dependable program, I hope to improve HDS's standing as a trustworthy colleague for software development projects.

Activity1

Programming Paradigms comparison





(bootpootin, 2021)

Characteristics and features of Procedural programming

- Procedures or functions are used as the primary organizing element in procedural programming.
- The program is divided into smaller processes using a top-down strategy.
- For reuse and code maintenance, modularity is recommended.
- The execution of instructions happens in the same order as they are delivered.
- Global data are accessible to and under the control of any programming approach.
- In comparison with object-oriented programming, data abstraction has several limitations.
- By covering fundamental properties, procedural abstraction enables communication via interfaces.
- Compared to object-oriented programming, code reuse is limited.
- Memory use and execution speed may be successfully managed with procedural programming.
- Fortran, Pascal, and C are examples of languages used for procedural programming.

Characteristics and features of Object Oriented programming

- Abstraction is used to represent real-world objects in their simplified form.
- Classes can inherit the attributes of other individuals.
- Due to polymorphism, objects may be thought of as belonging to their super class.
- Objects communicate by exchanging messages.
- Associations serve as a representation of relationships between objects.
- Aggregation and composition define the concept of object confinement.
- OOP promotes modularity and code reuse.

Characteristics and features of Event Driven programming (EDP)

- Event-Driven Events, which are occurrences or happenings, are the center of programming.
- While awaiting events, asynchronous execution enables the software to carry out other activities.
- Functions or code blocks registered as event handlers (callbacks) respond to specified events.
- Used often in GUI development to manage user interactions with buttons and menus.
- Encourages flexible connection between components, making it simple to add new features.
- Scalable systems because of their capacity to manage several events concurrently.
- Simplifies modularity by allowing the addition of additional event handlers without altering the present code.

Use of Different types of Events in GUI-Based Programming

- Events are essential in GUI-based programming for managing user interactions and developing responsive user interfaces. The term "events" refers to acts or occurrences that a computer could observe and respond to by executing specific code. To manage various user interactions, several sorts of events are utilized. The list below includes a selection of frequently happening event kinds for GUI-based programming:

1. Button Events: Buttons are fundamental GUI elements that, when clicked, start operations. Button events are used to control how the click is handled and execute certain actions when a user clicks a button.
2. Timer Events: Actions can be launched on a timed basis by using timed events. They may be used to schedule job completion, create animations, and change UI components.
3. Menu Event: The selection of menu items, such as through a context menu or drop-down menu, is one of the menu events. Menu events are used to start certain processes or actions related to the current menu item being selected.
4. Drag and Drop Event: Using drag and drop events, the user may click, drag, and drop GUI items (such as files, photos, and text) into predetermined locations or targets. These situations make data structure and movement inside the software simpler.

Comparison Between Paradigms

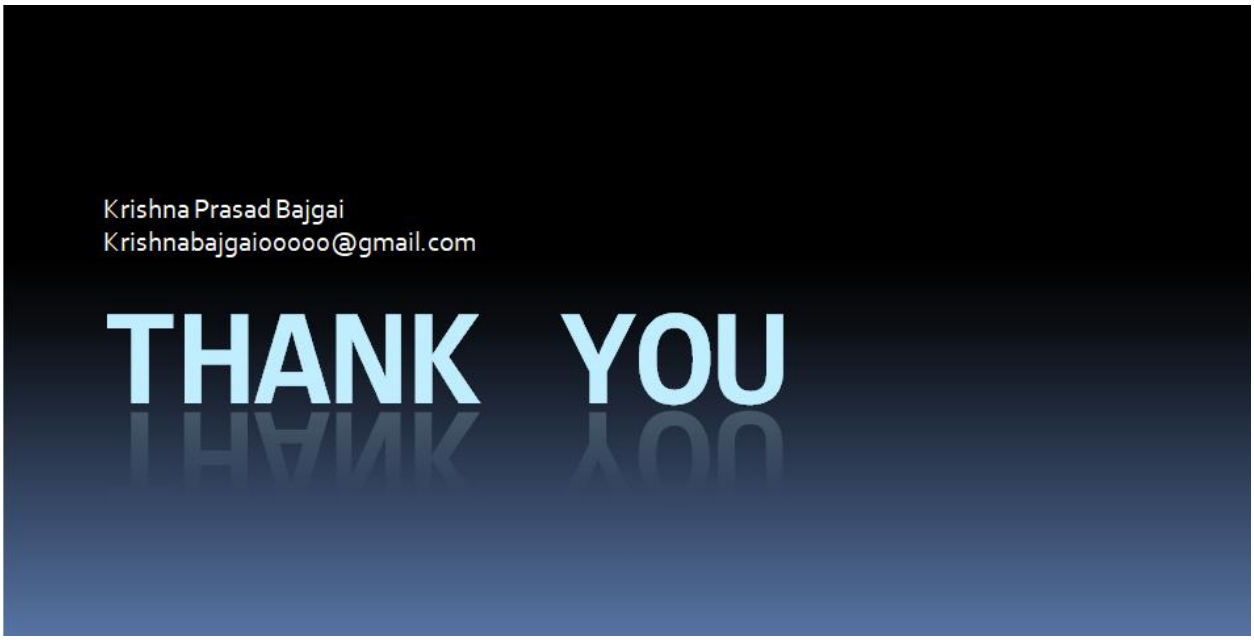
Programming Paradigms	Advantages	Disadvantages
Procedural	1. Simplicity: Straightforward, linear approach to problem-solving. 2. Efficiency: Lower memory overhead and faster execution time. 3. Code reusability: Functions can be easily reused.	1. Weak encapsulation 2. Difficult maintenance 3. Code duplication 4. Scalability issues
Object Oriented	1. Reusability and modularity. 2. Managing complexity through abstraction. 3. Code reuse and inheritance. 4. Data security and encapsulation. 5. Simple to Maintain and Understand.	1. Increasing learning curve 2. Overhead 3. Large-scale project complexity 4. Ineffective for specific jobs
Event Driven	1. Managing events and modularity. 2. Applications that are distributed and scalable. 3. Flexibility and loose coupling. 4. Performance and Parallelism.	1. Having trouble debugging 2. Concerns about scalability 3. Sequence of events 4. Resource administration

Critical Evaluation

- Procedural Programming: Clear structure, but can become unwieldy for large projects.
- Object Oriented Programming: Promotes reusability and modularity, but can lead to complex class structures.
- Event Driven Programming: Responsive for GUI and parallel systems, but complex systems can be hard to debug.

Conclusion

- In conclusion, each programming paradigm has certain benefits and disadvantages. Although procedural programming is simple and effective, it might not be the best option for managing big projects. On the other side, object-oriented programming needs careful preparation yet allows for code reuse and encapsulation. While managing asynchronous events might add complexity, event-driven programming works in GUI-based systems. I chose the event-driven programming paradigm in light of the particular project requirements.


Krishna Prasad Bajgai
Krishnabajgaiooooo@gmail.com
THANK YOU

Activity 2

Introduction

This article's major focus is on the process of developing a cutting-edge algorithm and effectively implementing it into a practical application for Best Fitness. Our ultimate goal is to use C# programming in Visual Studio and the.NET Framework to develop a high-performing piece of software that precisely fulfills the unique needs of our customer. We will make sure that the software development process is successful and effective by carefully integrating the appropriate algorithms, programming languages, and IDE tools.

Algorithm Definition

An algorithm is a set of specified, constrained steps that a computer takes to complete a task or solve a particular issue. It consists of a small amount of instructions that manage the computer's calculations and analytical processes. (Upadhyay, 2021)

The algorithm for the Best Fitness program will be described here:

Algorithm: BestFitnessProgram

Step1: Begin

Step2: Create a new Windows Forms application named "Fitness_Club_Assignment" with a main form named "Form1".

Step 3: Import the required namespaces for database access and GUI controls.

Step 4: Define the class "Form1" that inherits from the "Form" class.

Step 5: In the constructor of "Form1":

a. Initialize the form components using the "InitializeComponent ()" method.
b. Create an instance of the "SubscriptionClass1" from the "Database" namespace and assign it to the variable "sc".
c. Set the data source of the DataGridView "dGVDetail" using "getAllSubscription ()" method from the "sc" object.

Step 6: Implement event handlers for various controls and actions on the form:

a. Handle the Search Button event:
- Call the "searchSubscription ()" method from the "sc" with the value of "txt1" textbox as the search criteria.
- Update the DataGridView "dGVDetail" with the returned search results.

b. Handle the Delete Button event:
- Validate the input fields to ensure a valid sc ID is provided.
- Call the "manageSubscriptions ()" method from the "sc" with the provided sc ID to delete the subscription.
- Display a success message if the deletion is successful and update the DataGridView with the updated data.
- Display an error message if there is an issue with the input or the deletion process.

c. Handle the "btnBrowse_Click" event:
- Create an OpenFileDialog to allow the user to select an image file.
- If the user selects an image, display it in the PictureBox "picbox2".

a. Handle the radio button events for different Subscription options:
- Update "SubscriptionType" variables based on the selected radio buttons.

b. Handle the "dGVDetail_CellClick" event:
- Get the selected row from the DataGridView and extract the values of each column.
- Set the corresponding textboxes and radio buttons with the extracted values to allow editing.

c. Handle the Edit Button event:
- Validate the input fields to ensure valid data is providing for editing.
- Call manageSubscriptions method from the object with appropriate parameters to update the subscription details.
- Display a successful message if the update is successful and refresh the DataGridView with updated data.
- Display an error message if there is an issue with the input or the update process.

d. Handle the Save Button event:
- Validate all input fields to ensure complete and correct subscription information is provided.
- Call the manage Subscriptions method from the given object with appropriate parameters to add a new subscription.
- Display a success message if the addition is successful and refresh the DataGridView with updated data.
- Display an error message if there is an issue with the input or the addition process.

e. Implement the Total Cost Button event to calculate the total cost of the selected subscription plan and display it on the form.

f. Implement the Clear Data method to reset all input fields on the form and clear the selected image in the PictureBox "picbox2".

Step 7: End

Process of Building the Application

In this session, we'll take a step-by-step look at how to build an excellent fitness application with C# and Visual Studio. Let's look at how to create excellent fitness software with a range of features to meet customers' desires for health and wellness.

- **Planning:** During the planning stage, a project's broad scope, demands, and goals are defined. This process includes selecting the qualities the fitness program app should have in addition to the project's overall objectives. You'll set up the project and distribute the required resources, such as manpower, equipment, and software.

- **Requirement Analysis:** You now request an extensive list of necessities and demands for the fitness program app from the client or anyone else with an interest. Finally, the features and actions of the software have been chosen. In order to guarantee that the development team produces the right product and to avoid misunderstandings, the requirements must be carefully examined.

- **Design:** We will develop the user interface and basic system architecture for the fitness program application during the design stage.

  Using the forms and controls in Visual Studio, the user interface will be built while decisions regarding data processing, storage, and presentation are made.

- **Implementation:** The actual coding happens during the implementation phase. Programmers will create C# code based on the design to convert the algorithms and logic into usable instructions. At this level, databases must be linked to the software in order to store and access data. We will carefully manage unexpected circumstances using the appropriate error management techniques.

- **Testing:** The testing phase is important because it gives programmers the chance to find and fix any mistakes or weaknesses in the code. The application will go through a series of tests using Visual Studio's testing tools to ensure that it complies with the specifications and performs as planned. This procedure could include user acceptance testing, integration testing, and unit testing.

- **Deployment:** Once the testing procedure has been successfully completed, the fitness program application is ready for distribution. The software must be made available on the target platform, no matter whether it is a desktop application, a web application, or a mobile application. This deployment process is simplified by the publishing capabilities in Visual Studio.

- **Maintenance:** Once the application is in use, the developer team will remain involved. In this process, consumer feedback is taken into account, the program is updated to include new features or upgrades, and any bugs or other issues are fixed. The product is simple to manage and upgrade thanks to Visual Studio's integrated development environment.

(Shiverware, 2019)

Steps from Writing Code to Execution:

The published instructions for using Visual Studio to convert written C# code into a workable application contain a number of errors that need to be clarified. Here is a process description that is more precise:

- **Coding:** The programmers write human-readable C# source code using Visual Studio. They create the classes, methods, variables, and other program logic-related components.

- **Compilation:** The developers launch Visual Studio's compilation process after writing the C# code. The C# compiler (csc.exe) then converts the human-readable source code into Intermediate Language (IL) code. This IL code is also known as managed code.

- **Intermediate Language (IL):** Platform-neutral code is referred to as "IL" by the.NET framework alone. It enables Common Language Runtime (CLR)-compliant programs to run on any platform.

  - **Assembling:** During compilation, assemblies are created from the IL code. An assembly, or created unit of code, can be found in many library (.dll) and executable (.exe) files. The assembly metadata contains descriptions of the types and members that were specified in the code.

  - **Just-In-Time (JIT) Compilation:** When the program is launched, the Common Language Runtime (CLR) for the.NET platform takes over.

  The CLR manages Just-In-Time (JIT) compilation. The JIT compiler of the CLR transforms the IL code present in the assemblies into machine code unique to the target

computer's processor architecture. This ensures that the intended gear will perform at its peak.

- **Execution:** After JIT compilation, the program's machine code is immediately run on the CPU. The software runs and accomplishes the goals it was intended to.

Algorithm to Working Program Conversion

We may take use of Visual Studio's user-friendly interface and the GUI development capabilities by turning an algorithm into a practical C# application.

As a result, development is more successful. We were able to create a user-friendly application that provides users with a natural experience by employing C#'s flexible capabilities and GUI support. By bridging the gap between theoretical problem-solving and actual software execution, this conversion approach aids us in developing a trustworthy and attractive application that satisfies our unique criteria.

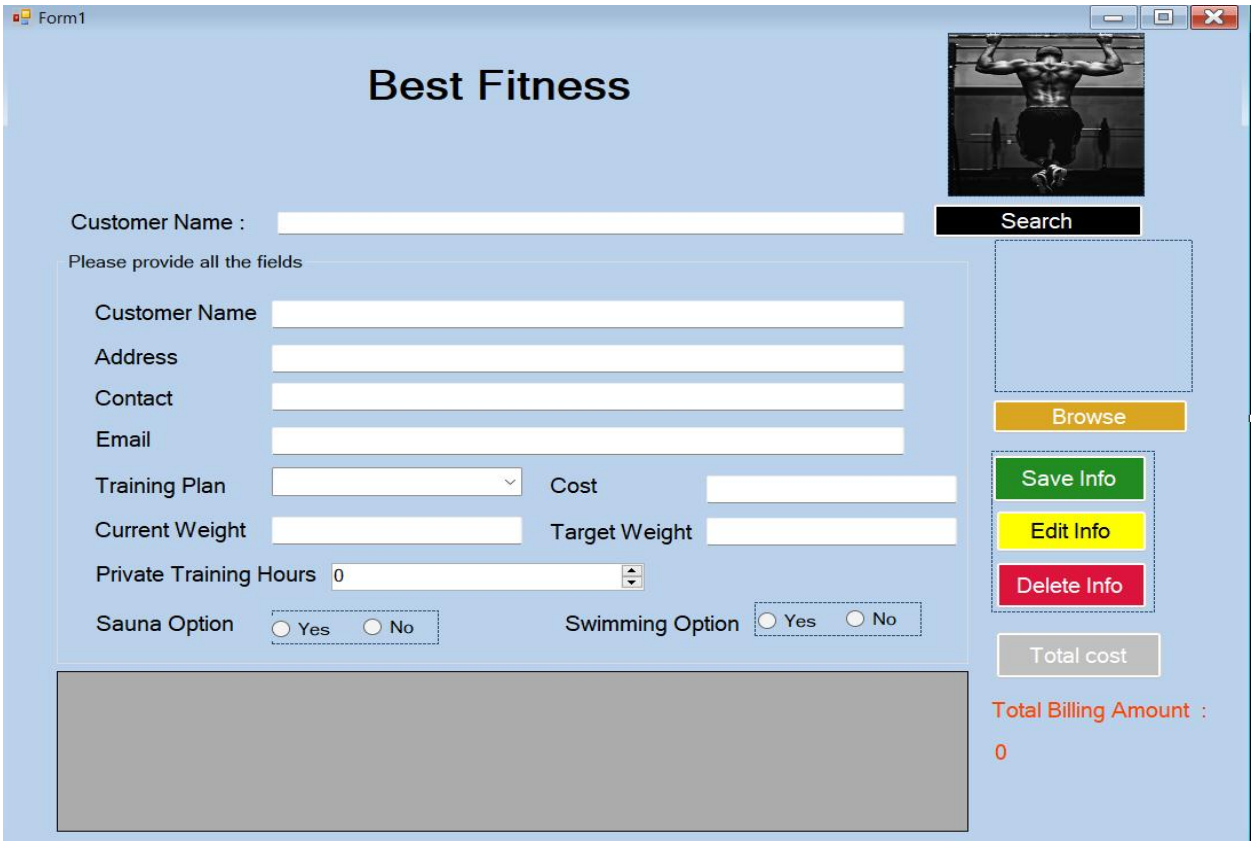
Program Code Relationship with Algorithm:

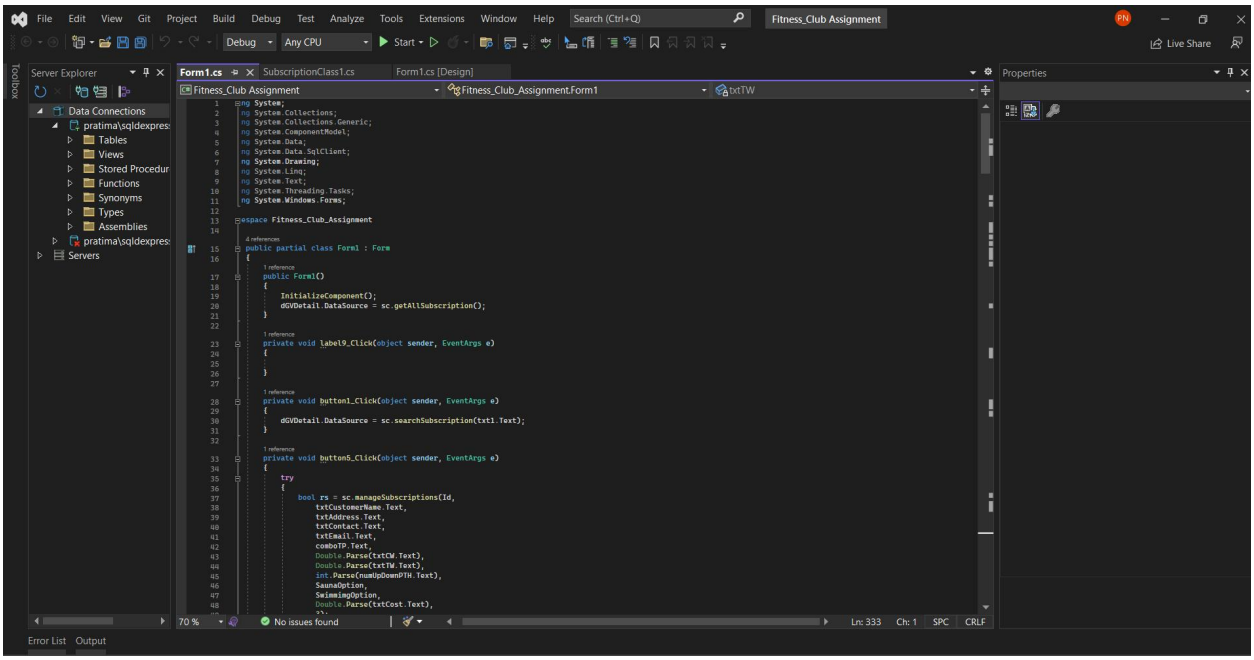
The creation of a simple and attractive user interface will be given priority when the algorithm has been implemented as usable software. To construct a dynamic main menu and user interactions, this will include fusing C# code with Visual Studio libraries. The GUI will be meticulously designed to provide consumers the best experience possible. The application will employ event handling techniques to guarantee a successful user input response. Users will engage with the application in a fun and natural way as a consequence.

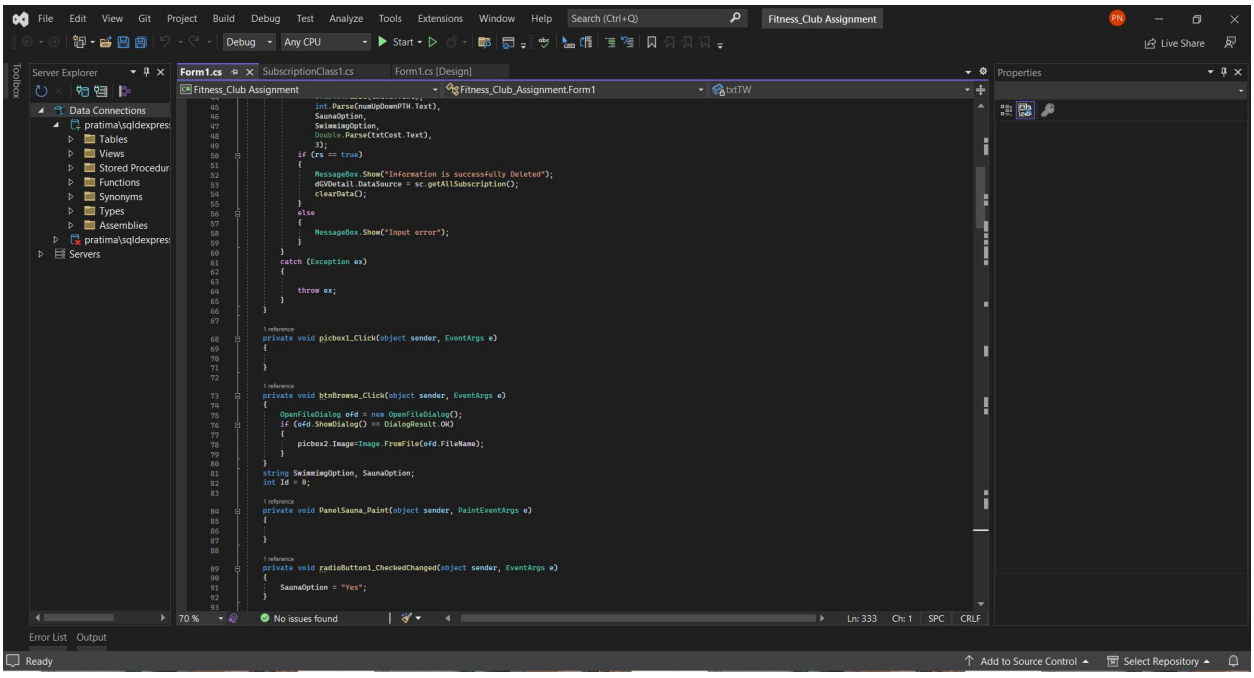
The heart of the algorithm will be developed in C#, making it possible to manage user data and training cycles effectively. Users may design and maintain personalized fitness plans that match their tastes and training goals. By combining the potent capabilities of C# with the expertise of Visual Studio, the Best Fitness program will be able to provide a wide range of user-friendly features. Users might expect an easy and enjoyable app experience as they work toward their health goals.

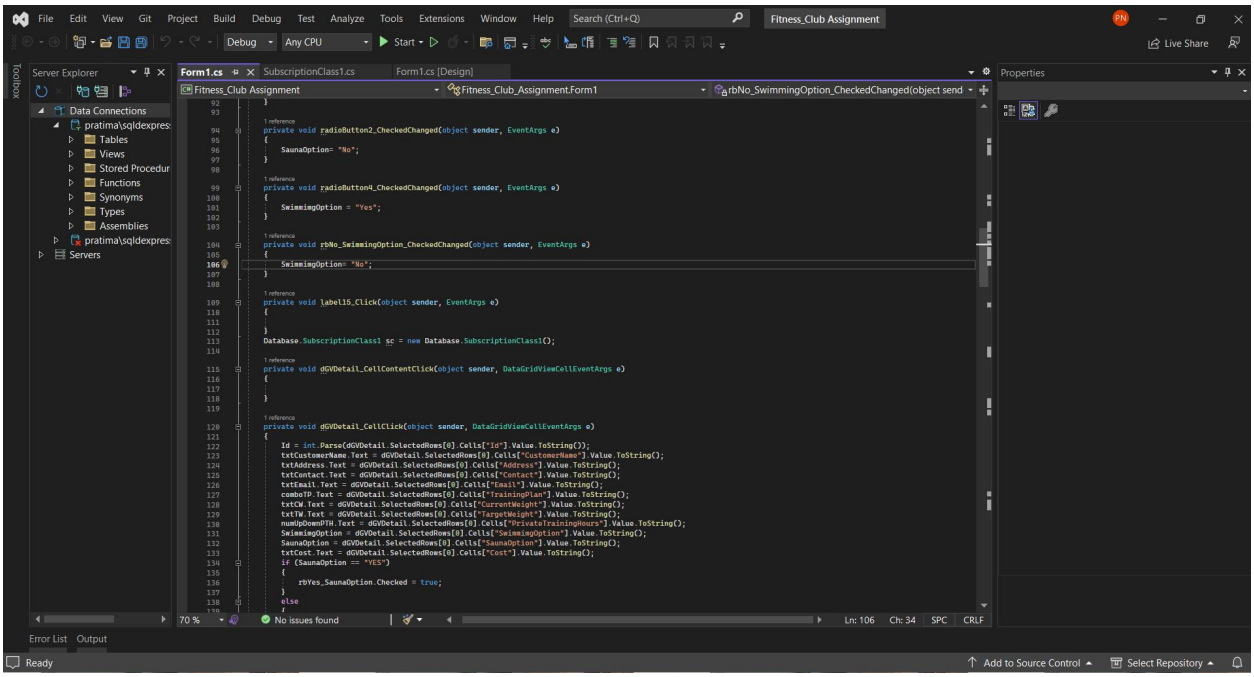
Demonstration of the Project:

Below, I'll provide a detailed analysis of the coding and design work I made for this application. Check out my work's summary below:

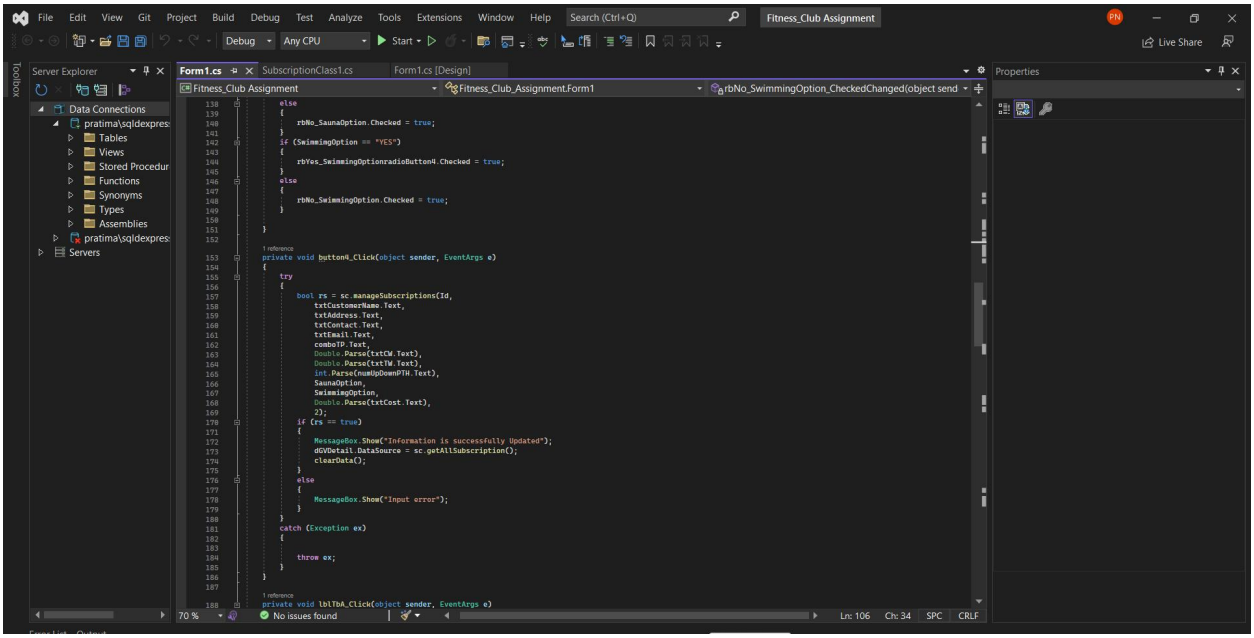

Form1
Best Fitness
Customer Name :
Search
Please provide all the fields
Customer Name
Address
Contact
Email
Training Plan
Cost
Current Weight
Target Weight
Private Training Hours
0
Sauna Option
Yes
No
Swimming Option
Yes
No
Browse
Save Info
Edit Info
Delete Info
Total cost
Total Billing Amount :
0


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
Fitness_Club Assignment
Debug
Any CPU
Start
Live Share
Server Explorer
Form1.cs
SubscriptionClass1.cs
Form1.cs [Design]
Fitness_Club Assignment
Fitness_Club_Assignment.Form1
txtTW
Properties
Data Connections
pratima\sqldexpres
Tables
Views
Stored Procedur
Functions
Synonyms
Types
Assemblies
pratima\sqldexpres
Servers
using System;
using System.Collections;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
namespace Fitness_Club_Assignment
4 references
public partial class Form1 : Form
{
1 reference
public Form1()
{
InitializeComponent();
dGVDetail.DataSource = sc.getAllSubscription();
}
1 reference
private void label9_Click(object sender, EventArgs e)
{
}
1 reference
private void button1_Click(object sender, EventArgs e)
{
dGVDetail.DataSource = sc.searchSubscription(txt1.Text);
}
1 reference
private void button5_Click(object sender, EventArgs e)
{
try
{
bool rs = sc.manageSubscriptions(Id,
txtCustomerName.Text,
txtAddress.Text,
txtContact.Text,
txtEmail.Text,
comboTP.Text,
Double.Parse(txtCW.Text),
Double.Parse(txtTW.Text),
int.Parse(numUpDownPTH.Text),
SaunaOption,
SwimmimgOption,
Double.Parse(txtCost.Text),
70 %
No issues found
Ln: 333
Ch: 1
SPC
CRLF
Error List
Output

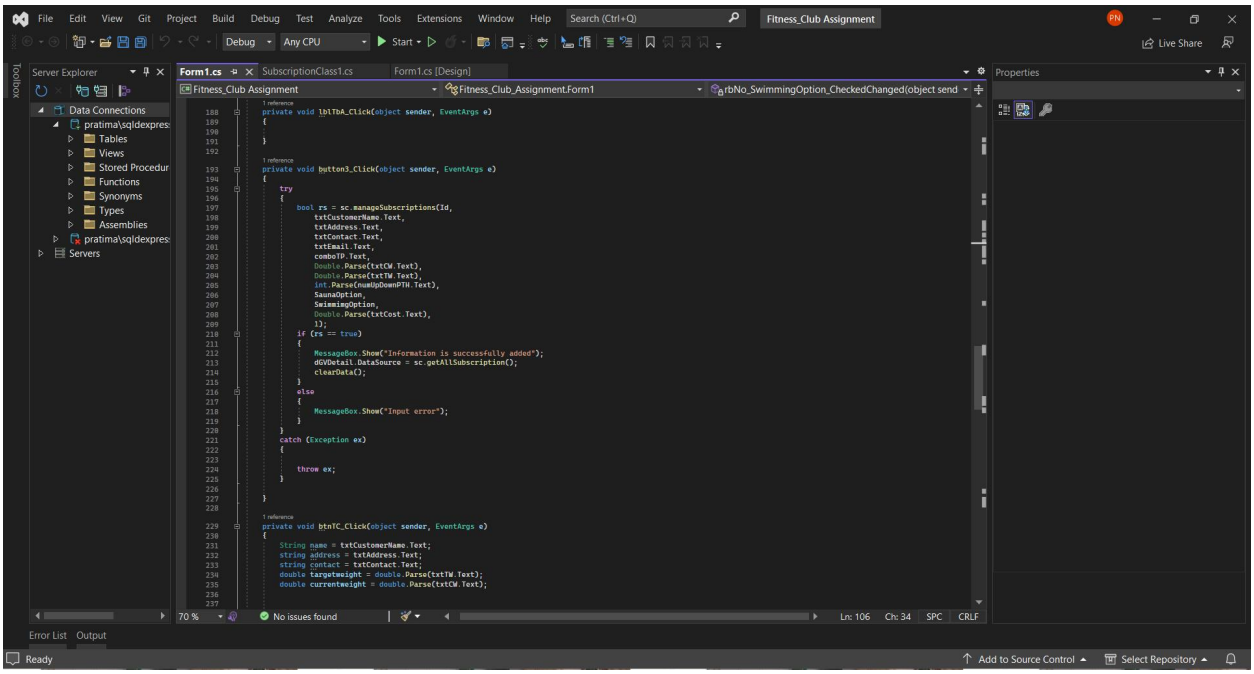

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
Fitness_Club Assignment
Debug
Any CPU
Start
Live Share
Server Explorer
Form1.cs
SubscriptionClass1.cs
Form1.cs [Design]
Fitness_Club Assignment
Fitness_Club_Assignment.Form1
txtTW
Properties
Data Connections
pratima\sqldexpres
Tables
Views
Stored Procedur
Functions
Synonyms
Types
Assemblies
pratima\sqldexpres
Servers
int.Parse(numUpDownPTH.Text),
SaunaOption,
SwimmimgOption,
Double.Parse(txtCost.Text),
);
if (rs == true)
{
MessageBox.Show("Information is successfully Deleted");
dGVDetail.DataSource = sc.getAllSubscription();
clearData();
}
else
{
MessageBox.Show("Input error");
}
}
catch (Exception ex)
{
throw ex;
}
}
1 reference
private void picbox1_Click(object sender, EventArgs e)
{
}
1 reference
private void btnBrowse_Click(object sender, EventArgs e)
{
OpenFileDialog ofd = new OpenFileDialog();
if (ofd.ShowDialog() == DialogResult.OK)
{
picbox2.Image=Image.FromFile(ofd.FileName);
}
}
string SwimmimgOption, SaunaOption;
int Id = 0;
1 reference
private void PanelSauna_Paint(object sender, PaintEventArgs e)
{
}
1 reference
private void radioButton1_CheckedChanged(object sender, EventArgs e)
{
SaunaOption = "Yes";
}
70 %
No issues found
Ln: 333
Ch: 1
SPC
CRLF
Error List
Output
Ready
Add to Source Control
Select Repository



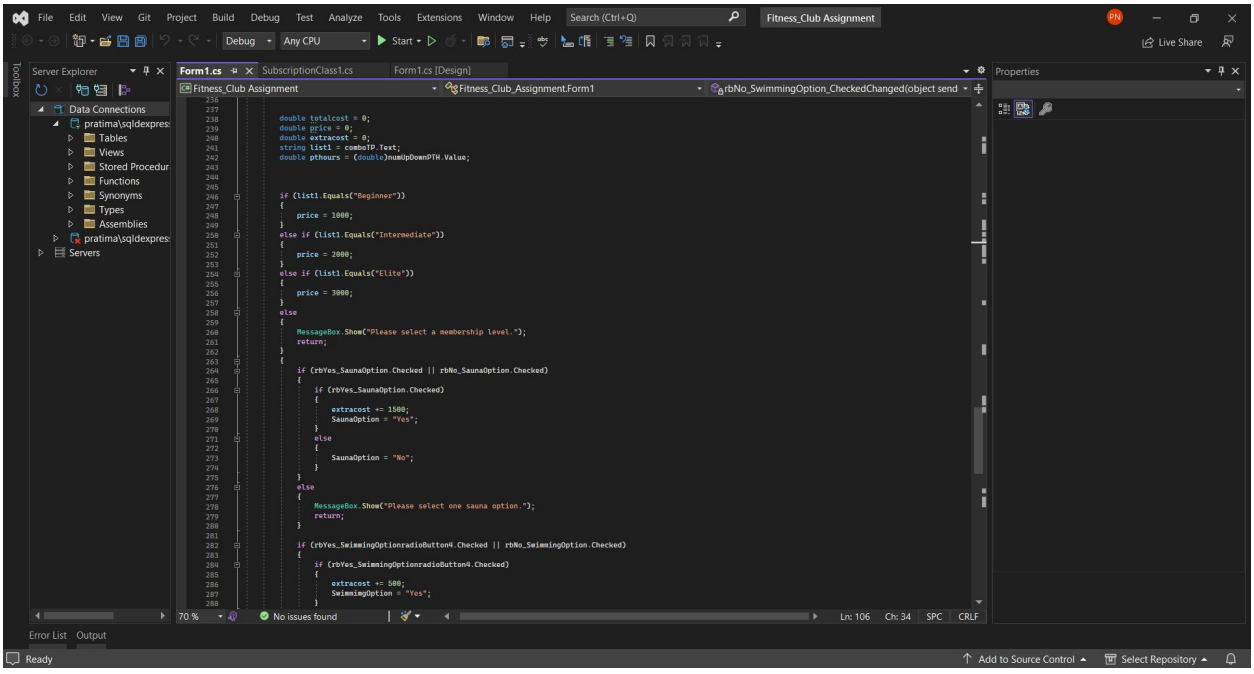
```
        }

        1 reference
        private void radioButton2_CheckedChanged(object sender, EventArgs e)
        {
            SaunaOption= "No";
        }

        1 reference
        private void radioButton4_CheckedChanged(object sender, EventArgs e)
        {
            SwimmingOption = "Yes";
        }

        1 reference
        private void rbNo_SwimmingOption_CheckedChanged(object sender, EventArgs e)
        {
            SwimmingOption= "No";
        }

        1 reference
        private void label15_Click(object sender, EventArgs e)
        {

        }
        Database.SubscriptionClass1 sc = new Database.SubscriptionClass1();

        1 reference
        private void dGVDetail_CellContentClick(object sender, DataGridViewCellEventArgs e)
        {

        }

        1 reference
        private void dGVDetail_CellClick(object sender, DataGridViewCellEventArgs e)
        {
            Id = int.Parse(dGVDetail.SelectedRows[0].Cells["Id"].Value.ToString());
            txtCustomerName.Text = dGVDetail.SelectedRows[0].Cells["CustomerName"].Value.ToString();
            txtAddress.Text = dGVDetail.SelectedRows[0].Cells["Address"].Value.ToString();
            txtContact.Text = dGVDetail.SelectedRows[0].Cells["Contact"].Value.ToString();
            txtEmail.Text = dGVDetail.SelectedRows[0].Cells["Email"].Value.ToString();
            comboTP.Text = dGVDetail.SelectedRows[0].Cells["TrainingPlan"].Value.ToString();
            txtCW.Text = dGVDetail.SelectedRows[0].Cells["CurrentWeight"].Value.ToString();
            txtTW.Text = dGVDetail.SelectedRows[0].Cells["TargetWeight"].Value.ToString();
            numUpDownPTH.Text = dGVDetail.SelectedRows[0].Cells["PrivateTrainingHours"].Value.ToString();
            SwimmingOption = dGVDetail.SelectedRows[0].Cells["SwimmingOption"].Value.ToString();
            SaunaOption = dGVDetail.SelectedRows[0].Cells["SaunaOption"].Value.ToString();
            txtCost.Text = dGVDetail.SelectedRows[0].Cells["Cost"].Value.ToString();
            if (SaunaOption == "YES")
            {
                rbYes_SaunaOption.Checked = true;
            }
            else
            {
```



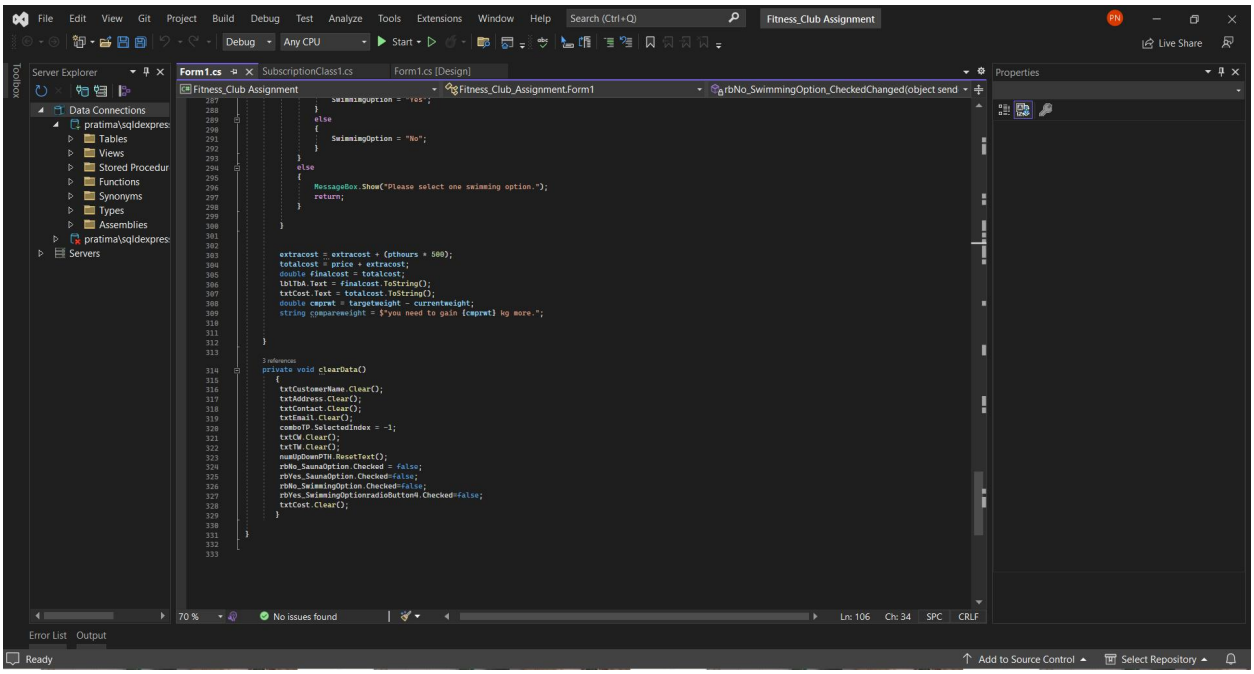
```
            else
            {
                rbNo_SaunaOption.Checked = true;
            }
            if (SwimmingOption == "YES")
            {
                rbYes_SwimmingOptionradioButton4.Checked = true;
            }
            else
            {
                rbNo_SwimmingOption.Checked = true;
            }

        }

        1 reference
        private void button4_Click(object sender, EventArgs e)
        {
            try
            {
                bool rs = sc.manageSubscriptions(Id,
                    txtCustomerName.Text,
                    txtAddress.Text,
                    txtContact.Text,
                    txtEmail.Text,
                    comboTP.Text,
                    Double.Parse(txtCW.Text),
                    Double.Parse(txtTW.Text),
                    int.Parse(numUpDownPTH.Text),
                    SaunaOption,
                    SwimmingOption,
                    Double.Parse(txtCost.Text),
                    2);
                if (rs == true)
                {
                    MessageBox.Show("Information is successfully Updated");
                    dGVDetail.DataSource = sc.getAllSubscription();
                    clearData();
                }
                else
                {
                    MessageBox.Show("Input error");
                }
            }
            catch (Exception ex)
            {

                throw ex;
            }
        }

        1 reference
        private void lblTbA_Click(object sender, EventArgs e)
```



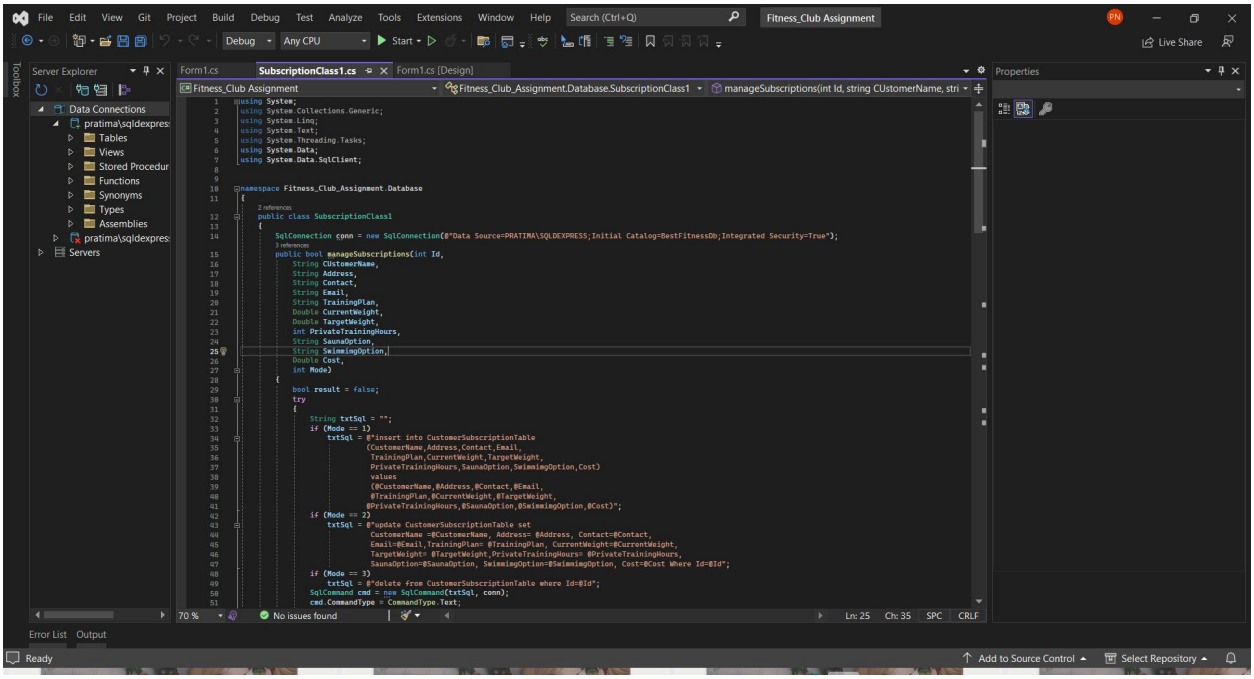
```
1 reference
private void lblTbA_Click(object sender, EventArgs e)
{

}

1 reference
private void button3_Click(object sender, EventArgs e)
{
    try
    {
        bool rs = sc.manageSubscriptions(Id,
            txtCustomerName.Text,
            txtAddress.Text,
            txtContact.Text,
            txtEmail.Text,
            comboTP.Text,
            Double.Parse(txtCW.Text),
            Double.Parse(txtTW.Text),
            int.Parse(numUpDownPTH.Text),
            SaunaOption,
            SwimmingOption,
            Double.Parse(txtCost.Text),
            1);
        if (rs == true)
        {
            MessageBox.Show("Information is successfully added");
            dGVDetail.DataSource = sc.getAllSubscription();
            clearData();
        }
        else
        {
            MessageBox.Show("Input error");
        }
    }
    catch (Exception ex)
    {

        throw ex;
    }

}

1 reference
private void btnTC_Click(object sender, EventArgs e)
{
    String name = txtCustomerName.Text;
    string address = txtAddress.Text;
    string contact = txtContact.Text;
    double targetweight = double.Parse(txtTW.Text);
    double currentweight = double.Parse(txtCW.Text);
```



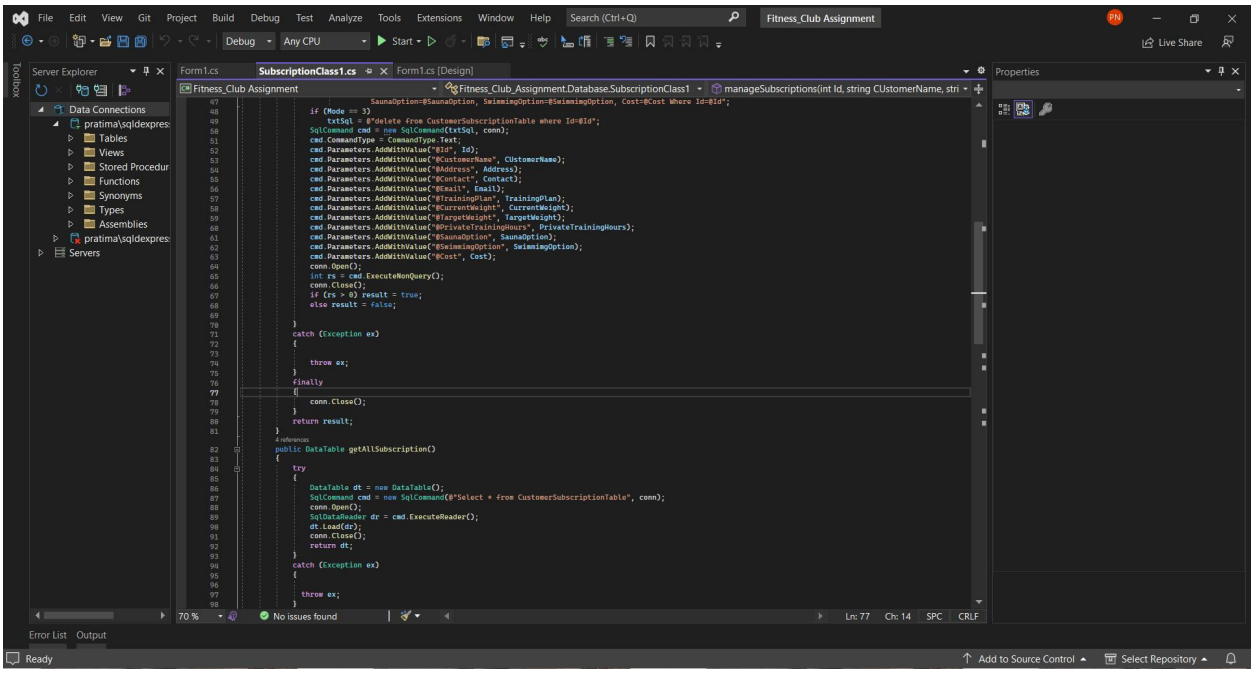
```
    double tptalcost = 0;
    double price = 0;
    double extracost = 0;
    string list1 = comboTP.Text;
    double pthours = (double)numUpDownPTH.Value;


    if (list1.Equals("Beginner"))
    {
        price = 1000;
    }
    else if (list1.Equals("Intermediate"))
    {
        price = 2000;
    }
    else if (list1.Equals("Elite"))
    {
        price = 3000;
    }
    else
    {
        MessageBox.Show("Please select a membership level.");
        return;
    }
    {
        if (rbYes_SaunaOption.Checked || rbNo_SaunaOption.Checked)
        {
            if (rbYes_SaunaOption.Checked)
            {
                extracost += 1500;
                SaunaOption = "Yes";
            }
            else
            {
                SaunaOption = "No";
            }
        }
        else
        {
            MessageBox.Show("Please select one sauna option.");
            return;
        }

        if (rbYes_SwimmingOptionradioButton4.Checked || rbNo_SwimmingOption.Checked)
        {
            if (rbYes_SwimmingOptionradioButton4.Checked)
            {
                extracost += 500;
                SwimmingOption = "Yes";
            }
```



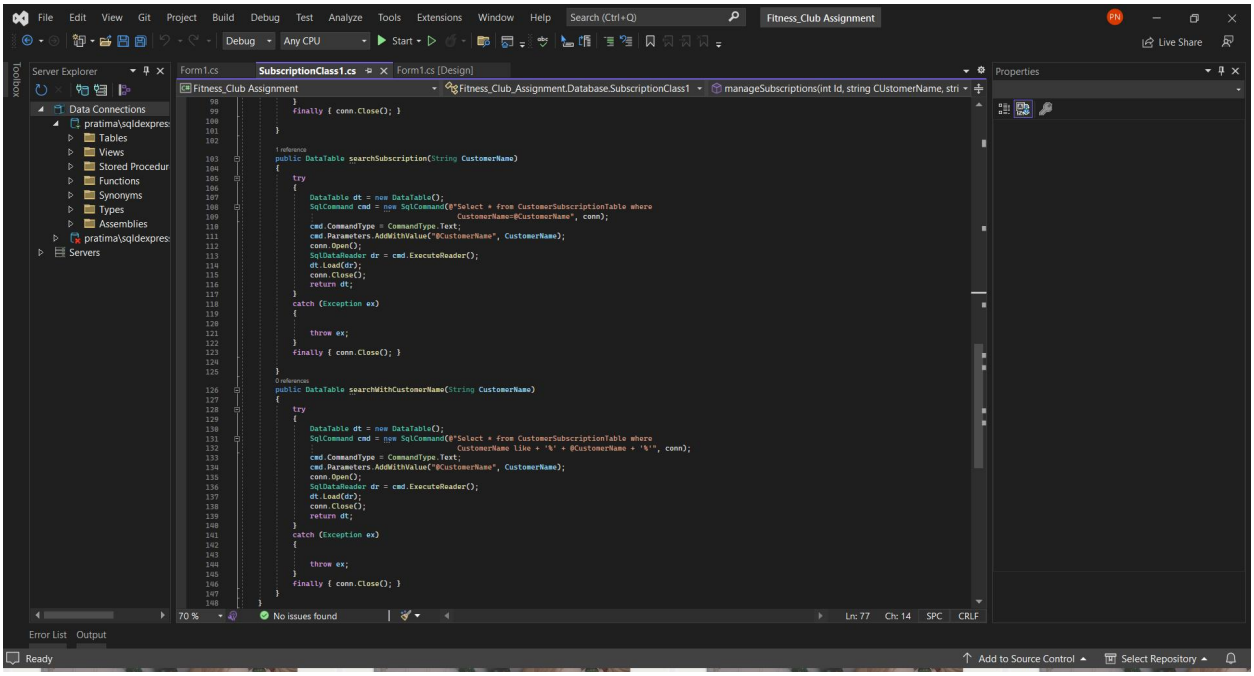
```
            SwimmingOption = "Yes";
        }
        else
        {
            SwimmingOption = "No";
        }
    }
    else
    {
        MessageBox.Show("Please select one swimming option.");
        return;
    }
}

extracost = extracost + (pthours * 500);
totalcost = price + extracost;
double finalcost = totalcost;
lblTbA.Text = finalcost.ToString();
txtCost.Text = totalcost.ToString();
double cmprwt = targetweight - currentweight;
string compareweight = $"you need to gain {cmprwt} kg more.";


}

3 references
private void clearData()
    {
     txtCustomerName.Clear();
     txtAddress.Clear();
     txtContact.Clear();
     txtEmail.Clear();
     comboTP.SelectedIndex = -1;
     txtCW.Clear();
     txtTW.Clear();
     numUpDownPTH.ResetText();
     rbNo_SaunaOption.Checked = false;
     rbYes_SaunaOption.Checked=false;
     rbNo_SwimmingOption.Checked=false;
     rbYes_SwimmingOptionradioButton4.Checked=false;
     txtCost.Clear();
    }
}
```



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Data;
using System.Data.SqlClient;


namespace Fitness_Club_Assignment.Database
{
    2 references
    public class SubscriptionClass1
    {
        SqlConnection conn = new SqlConnection(@"Data Source=PRATIMA\SQLDEXPRESS;Initial Catalog=BestFitnessDb;Integrated Security=True");
        3 references
        public bool manageSubscriptions(int Id,
            String CUstomerName,
            String Address,
            String Contact,
            String Email,
            String TrainingPlan,
            Double CurrentWeight,
            Double TargetWeight,
            int PrivateTrainingHours,
            String SaunaOption,
            String SwimmingOption,
            Double Cost,
            int Mode)
        {
            bool result = false;
            try
            {
                String txtSql = "";
                if (Mode == 1)
                    txtSql = @"insert into CustomerSubscriptionTable
                               (CustomerName,Address,Contact,Email,
                                TrainingPlan,CurrentWeight,TargetWeight,
                               PrivateTrainingHours,SaunaOption,SwimmingOption,Cost)
                               values
                               (@CustomerName,@Address,@Contact,@Email,
                               @TrainingPlan,@CurrentWeight,@TargetWeight,
                              @PrivateTrainingHours,@SaunaOption,@SwimmingOption,@Cost)";
                if (Mode == 2)
                    txtSql = @"update CustomerSubscriptionTable set
                               CustomerName =@CustomerName, Address= @Address, Contact=@Contact,
                               Email=@Email,TrainingPlan= @TrainingPlan, CurrentWeight=@CurrentWeight,
                               TargetWeight= @TargetWeight,PrivateTrainingHours= @PrivateTrainingHours,
                               SaunaOption=@SaunaOption, SwimmingOption=@SwimmingOption, Cost=@Cost Where Id=@Id";
                if (Mode == 3)
                    txtSql = @"delete from CustomerSubscriptionTable where Id=@Id";
                SqlCommand cmd = new SqlCommand(txtSql, conn);
                cmd.CommandType = CommandType.Text;
```

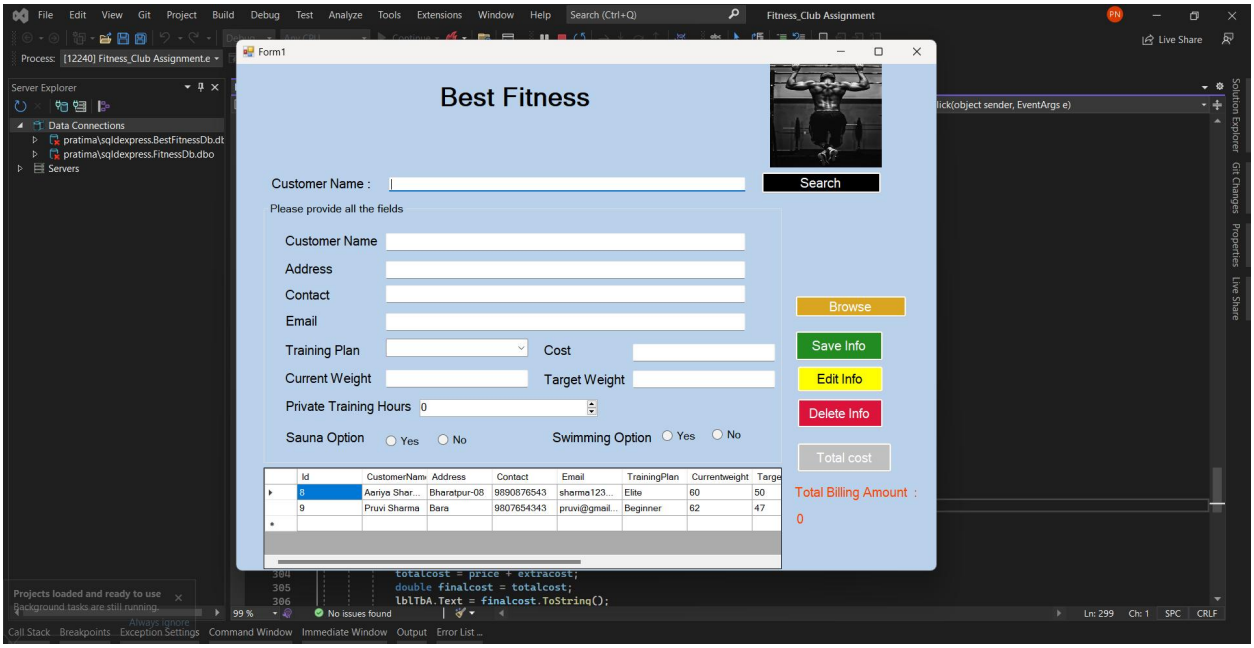


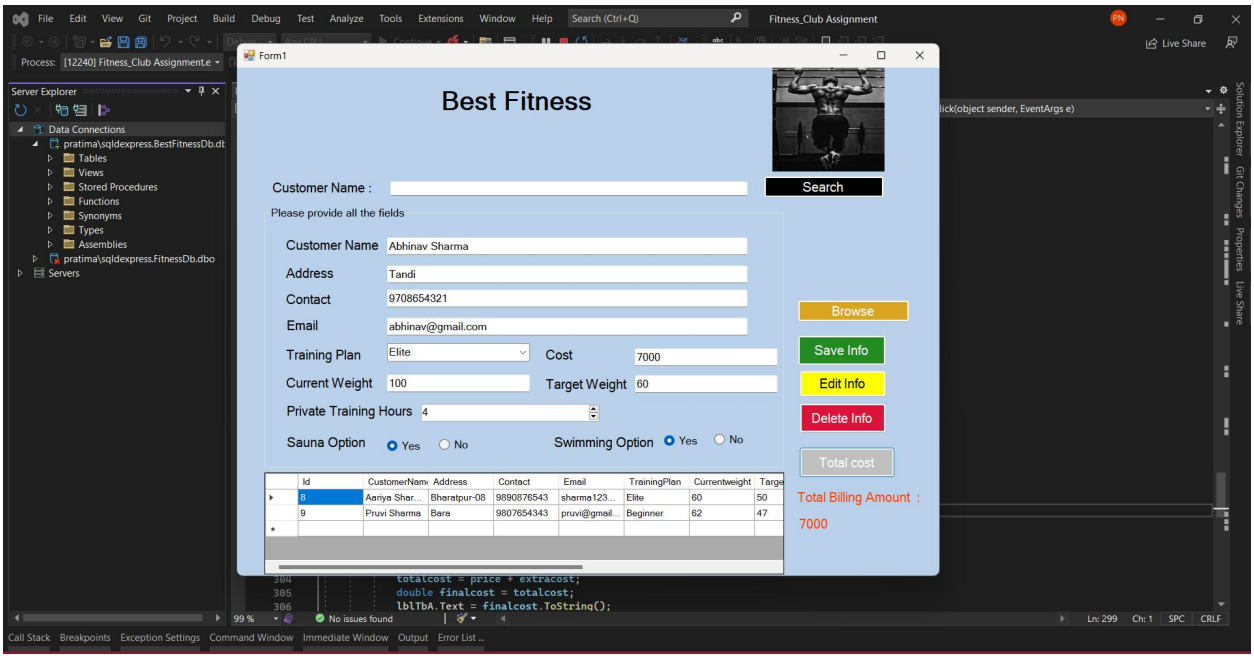
```
                if (Mode == 3)
                    txtSql = @"delete from CustomerSubscriptionTable where Id=@Id";
                SqlCommand cmd = new SqlCommand(txtSql, conn);
                cmd.CommandType = CommandType.Text;
                cmd.Parameters.AddWithValue("@Id", Id);
                cmd.Parameters.AddWithValue("@CustomerName", CUstomerName);
                cmd.Parameters.AddWithValue("@Address", Address);
                cmd.Parameters.AddWithValue("@Contact", Contact);
                cmd.Parameters.AddWithValue("@Email", Email);
                cmd.Parameters.AddWithValue("@TrainingPlan", TrainingPlan);
                cmd.Parameters.AddWithValue("@CurrentWeight", CurrentWeight);
                cmd.Parameters.AddWithValue("@TargetWeight", TargetWeight);
                cmd.Parameters.AddWithValue("@PrivateTrainingHours", PrivateTrainingHours);
                cmd.Parameters.AddWithValue("@SaunaOption", SaunaOption);
                cmd.Parameters.AddWithValue("@SwimmingOption", SwimmingOption);
                cmd.Parameters.AddWithValue("@Cost", Cost);
                conn.Open();
                int rs = cmd.ExecuteNonQuery();
                conn.Close();
                if (rs > 0) result = true;
                else result = false;

            }
            catch (Exception ex)
            {

                throw ex;
            }
            finally
            {
                conn.Close();
            }
            return result;
        }
        public DataTable getAllSubscription()
        {
            try
            {
                DataTable dt = new DataTable();
                SqlCommand cmd = new SqlCommand(@"Select * from CustomerSubscriptionTable", conn);
                conn.Open();
                SqlDataReader dr = cmd.ExecuteReader();
                dt.Load(dr);
                conn.Close();
                return dt;
            }
            catch (Exception ex)
            {

                throw ex;
            }
```

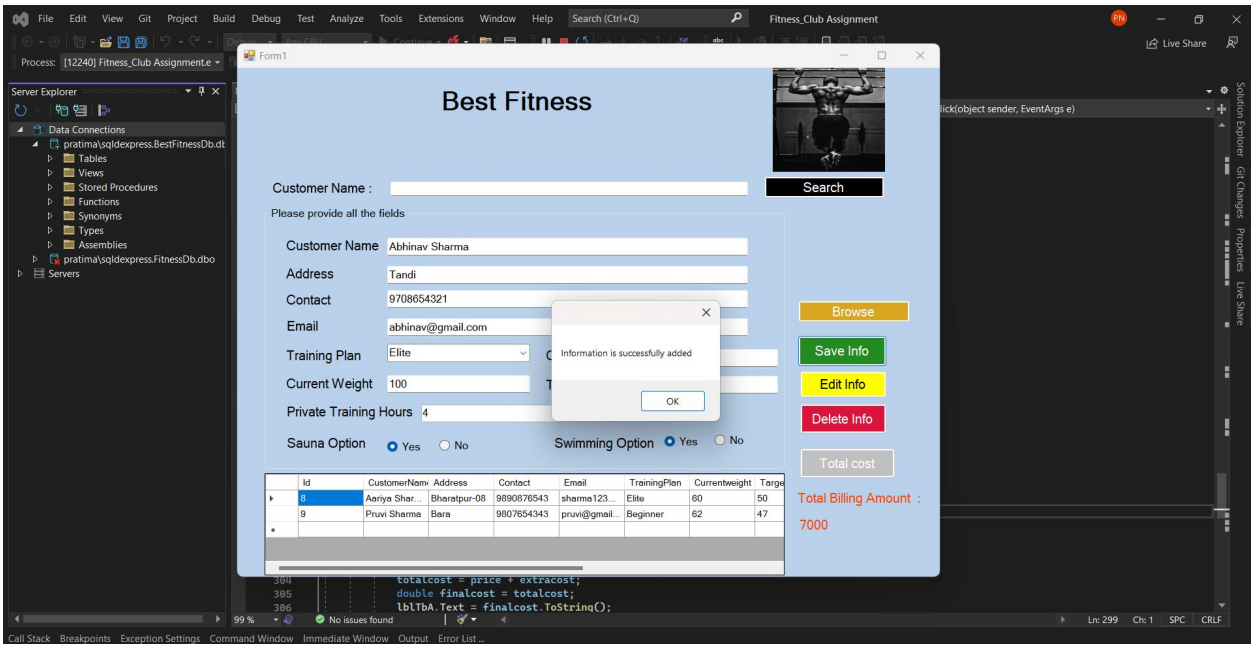


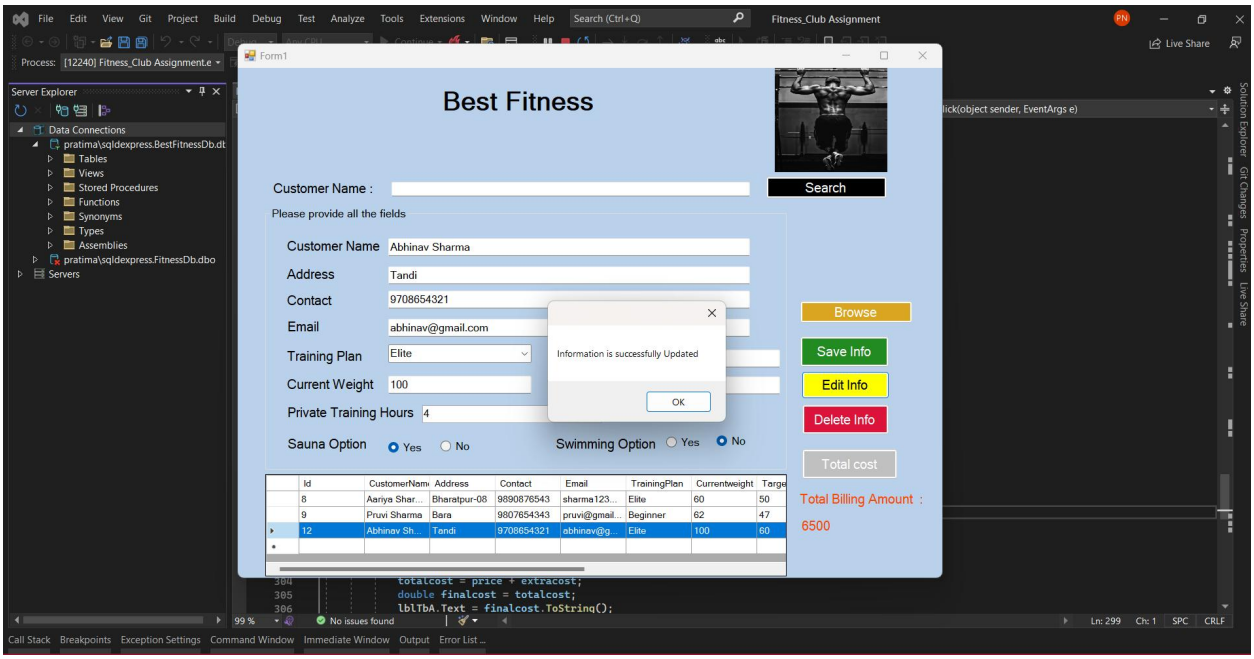
```
            finally { conn.Close(); }

        }

        public DataTable searchSubscription(string CustomerName)
        {
            try
            {
                DataTable dt = new DataTable();
                SqlCommand cmd = new SqlCommand(@"Select * from CustomerSubscriptionTable where
                                                  CustomerName=@CustomerName", conn);
                cmd.CommandType = CommandType.Text;
                cmd.Parameters.AddWithValue("@CustomerName", CustomerName);
                conn.Open();
                SqlDataReader dr = cmd.ExecuteReader();
                dt.Load(dr);
                conn.Close();
                return dt;
            }
            catch (Exception ex)
            {

                throw ex;
            }
            finally { conn.Close(); }

        }
        public DataTable searchWithCustomerName(string CustomerName)
        {
            try
            {
                DataTable dt = new DataTable();
                SqlCommand cmd = new SqlCommand(@"Select * from CustomerSubscriptionTable where
                                                  CustomerName like + '%' + @CustomerName + '%'", conn);
                cmd.CommandType = CommandType.Text;
                cmd.Parameters.AddWithValue("@CustomerName", CustomerName);
                conn.Open();
                SqlDataReader dr = cmd.ExecuteReader();
                dt.Load(dr);
                conn.Close();
                return dt;
            }
            catch (Exception ex)
            {

                throw ex;
            }
            finally { conn.Close(); }
        }
    }
```

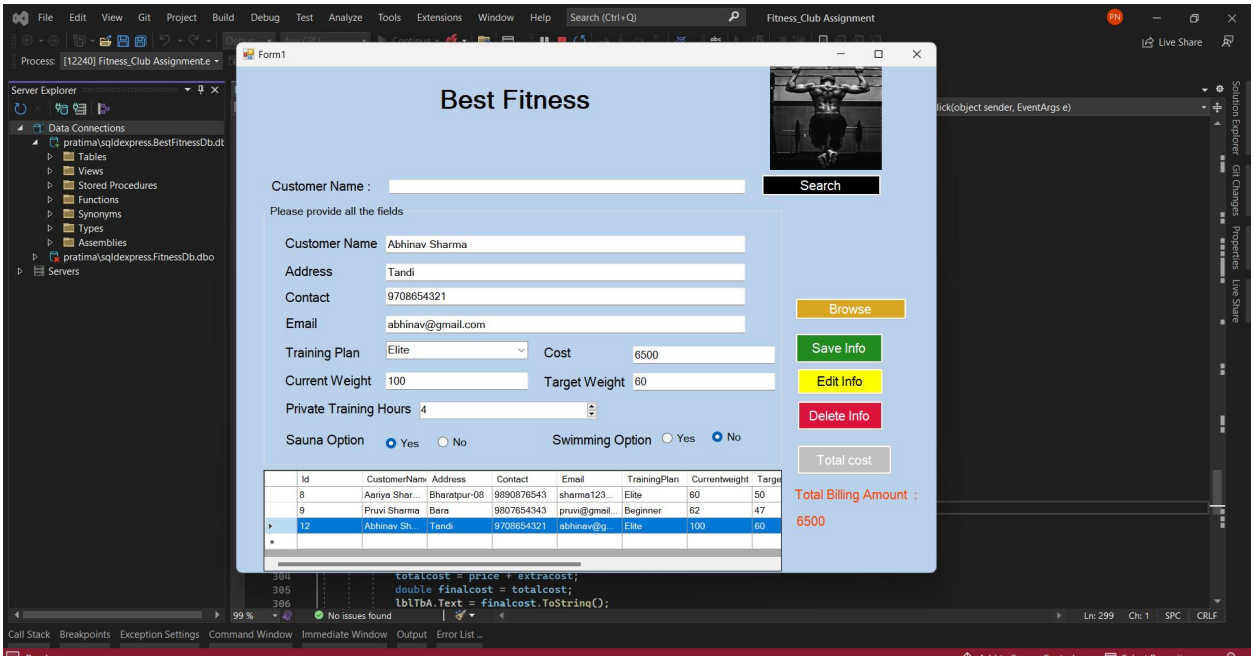
Execution of Application

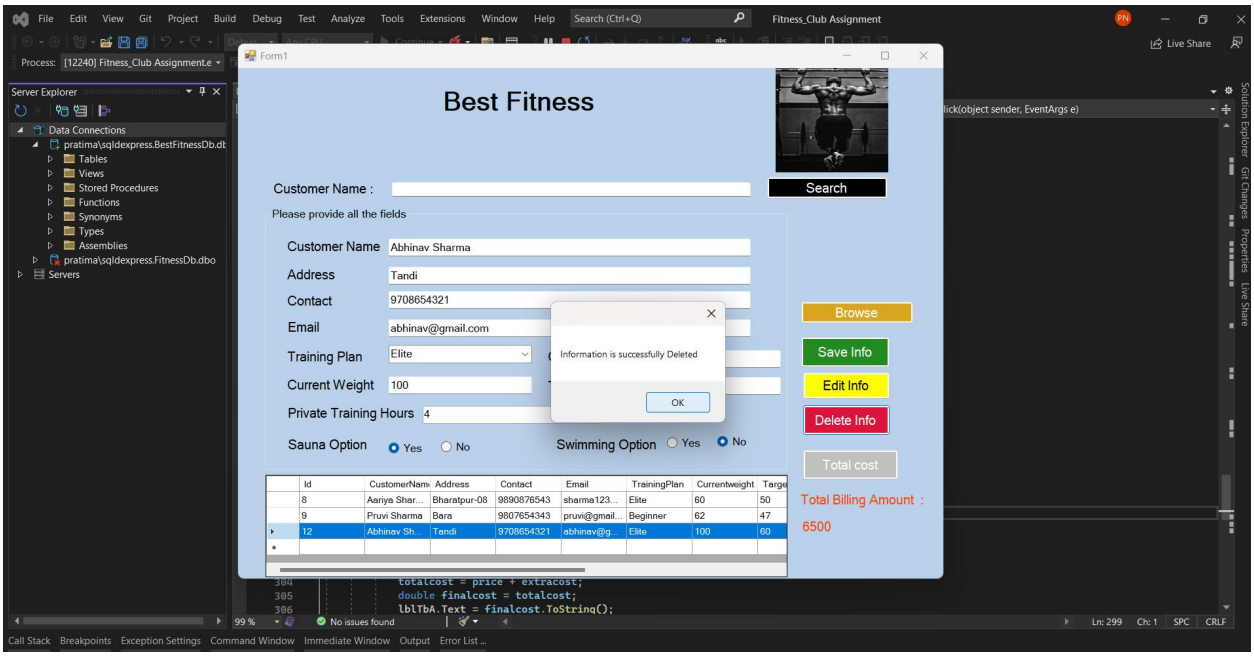



Form1
Best Fitness
Customer Name :
Search
Please provide all the fields
Customer Name
Abhinav Sharma
Address
Tandi
Contact
9708654321
Email
abhinav@gmail.com
Training Plan
Elite
Cost
7000
Current Weight
100
Target Weight
60
Private Training Hours
4
Sauna Option
Yes
No
Swimming Option
Yes
No
Browse
Save Info
Edit Info
Delete Info
Total cost
Total Billing Amount :
7000


Form1
Best Fitness
Customer Name :
Search
Please provide all the fields
Customer Name
Abhinav Sharma
Address
Tandi
Contact
9708654321
Email
abhinav@gmail.com
Training Plan
Elite
Information is successfully added
OK
Current Weight
100
Private Training Hours
4
Sauna Option
Yes
No
Swimming Option
Yes
No
Browse
Save Info
Edit Info
Delete Info
Total cost
Total Billing Amount :
7000


Form1
Best Fitness
Customer Name :
Search
Please provide all the fields
Customer Name Abhinav Sharma
Address Tandi
Contact 9708654321
Email abhinav@gmail.com
Training Plan Elite
Current Weight 100
Private Training Hours 4
Sauna Option Yes No
Swimming Option Yes No
Information is successfully Updated
OK
Browse
Save Info
Edit Info
Delete Info
Total cost
Total Billing Amount :
6500


Form1
Best Fitness
Customer Name :
Search
Please provide all the fields
Customer Name Abhinav Sharma
Address Tandi
Contact 9708654321
Email abhinav@gmail.com
Training Plan Elite
Cost 6500
Current Weight 100
Target Weight 60
Private Training Hours 4
Sauna Option Yes No
Swimming Option Yes No
Browse
Save Info
Edit Info
Delete Info
Total cost
Total Billing Amount :
6500



Yes, our application is working perfectly and without any problems.

Explanation of Features Utilized in Visual Studio 2022:

I'll list the features we used in the section below:

- **Server Explorer:** Managing connections to databases and other data-related resources is made simpler by the Server Explorer tool window in Visual Studio. Developers may connect to a range of data sources using Server Explorer, including SQL Server databases, web-based services, and more. The Visual Studio IDE makes browsing and interacting with the data straightforward, so developers can work with databases and services from within the IDE.

- **Version Control Integration:** It's easy to connect Git, a well-known version management system, with Visual Studio. Version control promotes collaboration by allowing developers to keep track of changes made to their code over time.

By allowing for the creation of branches, the merging of changes, and the ability to return to previous rounds, version control speeds and organizes the development process, especially when working in teams.

- **Toolbox Controls:** The controls in the Toolbox can be used by developers to design the user interface for their apps. They consist of buttons, text boxes, labels, drop-down menus, and other control kinds. Developers may speed up UI design by simply dropping these elements into forms instead of manually writing the code for each component.

- **Code Editor:** The Code Editor in Visual Studio is where they spend the most of their time creating and editing code. It provides a feature-rich environment with syntax highlighting, completion of code (IntelliSense), code formatting, and other productivity-enhancing features. The code editor accelerates and enhances code development and supports a variety of programming languages.

- **Debugger:** To find and fix bugs in their code, programmers can utilize the Visual Studio Debugger and other effective debugging tools.

Programmers may interrupt the running of their code at certain lines by establishing breakpoints, which enables them to analyze variable values and monitor the program's development in real time. Developers can understand the reasoning of the program and

follow the process of execution by moving around the code (step into, step over, step out), which makes it simpler to find and repair problems.

- **Solution Explorer:** A crucial element of Visual Studio, the Solution Explorer, presents a hierarchical view of the project's documents and directories.

It's simple administration of resources and navigation may be used by developers to set up their project. The Solution Explorer, which allows for the addition, deletion, and reorganization of files and folders, serves as the primary administrative center for a project's document and folder hierarchy.

Challenges in Converting Algorithm into Program Code

As we translated the plan into code that could be executed, we faced a number of difficulties that put our ability to solve problems to the test. Among the challenges are:

- **Data Structure:** The proper data types must be used to manage user inputs and calculation outputs, such as integers for the time period count and floating-point for the cost.
- **Control Structures:** Regulatory structures use conditional expressions (such as if-else statements) to apply discounts based on client demands.
- **GUI Design:** Use Visual Studio's toolkit and extensive selection of choices to create a visually beautiful and user-friendly user interface.

- **User Input Validation:** To avoid application problems brought on by incorrect data or input mistakes, user input is validated.
- **Database Integration:** The program will keep user profiles and training schedules in a database.
- **Algorithm Efficiency:** Upgrades are being made to the algorithm and software to improve efficiency and response times.
- **Exception Handling:** Strong error handling guarantees that unexpected occurrences are kindly handled in your code.

Coding Standards

Coding standards must be used in order to ensure consistency and quality throughout a software development process. They serve as a collection of guidelines and recommendations that all programmers working on the project should follow. Consistency in coding practices and style is beneficial in group projects like the Best Fitness program because it makes it easier for team members to read and understand one another's code. (Bratty, 2023)

- **Consistent Formatting:** We will maintain consistent indentation and follow a clear code structure.
- **Comments:** Adding details to challenging code fragments or important algorithms.
- **Modularity:** Software can be described as modular if it has been divided up into smaller, well defined functions.
- **Error Handling:** Correct exception management is a key component of error control to prevent unexpected program termination.

- **Code Reusability:** The usage of libraries and functions that encourage the reuse of code components.

IDE Features and Enhancement

By utilizing cutting-edge technology and a wide range of useful features, the Visual Studio IDE was essential in completely redesigning the Best Fitness application. The success of the Best Fitness project was significantly influenced by these cutting-edge improvements and IDE capabilities. Some of the enhancement and IDE features are:

- **Visual Form Designer:** The development of a user-friendly and attractive graphical user interface that sped up the whole design process proved its significance.

- **Visual Studio's Debugging Tools:** As a result, our ability to spot and correct any logical mistakes that appeared during coding significantly increased, resulting in a quicker and more effective development process.
- **Unit Testing:** Each component has been tested in Visual Studio to ensure dependability and effectiveness. We can be sure that each component works as expected and produces the desired effects by carefully testing it.

- **Database Integration:** We made the most of SQL Server Explorer in Visual Studio to manage database integration quickly and easily.

Advantages and Challenges of Using an IDE

Advantages:

- **Integrated build tools:** To speed up the development process, integrated build tools handle compilation of code and deployment. IDEs typically come with these tools.
- **Error Highlighting and Suggestions:** To increase code quality, IDEs may find and highlight grammatical mistakes, faults, and code smells.
- **Debugging Tools:** Since powerful debugging tools are included as standard on IDEs, developers can simply set breakpoints, evaluate variables, go through through code, and identify and solve mistakes.
- **Syntax Highlighting:** Different parts of the code are color-coded in IDEs using syntax highlighting, making it simpler to visually identify between strings, variables, and keywords.

Challenges:

- **Resource Intensive:** Some IDEs could be resource-intensive and demand a lot of memory and processing power, which might impact how effectively they run on PCs with slow CPU speeds.
- **Learning Curve:** Depending on how complicated and use of the IDE, there may be a learning curve for new developers to understand all the features and capabilities.
- **Dependency:** In contexts lacking these tools, developers who rely too much on IDE features run the danger of losing hold of the basic code and language.

- **Performance and Stability:** Despite the fact that IDEs have much improved, some may still now and then have performance or stability problems that might result in surprise crashes.

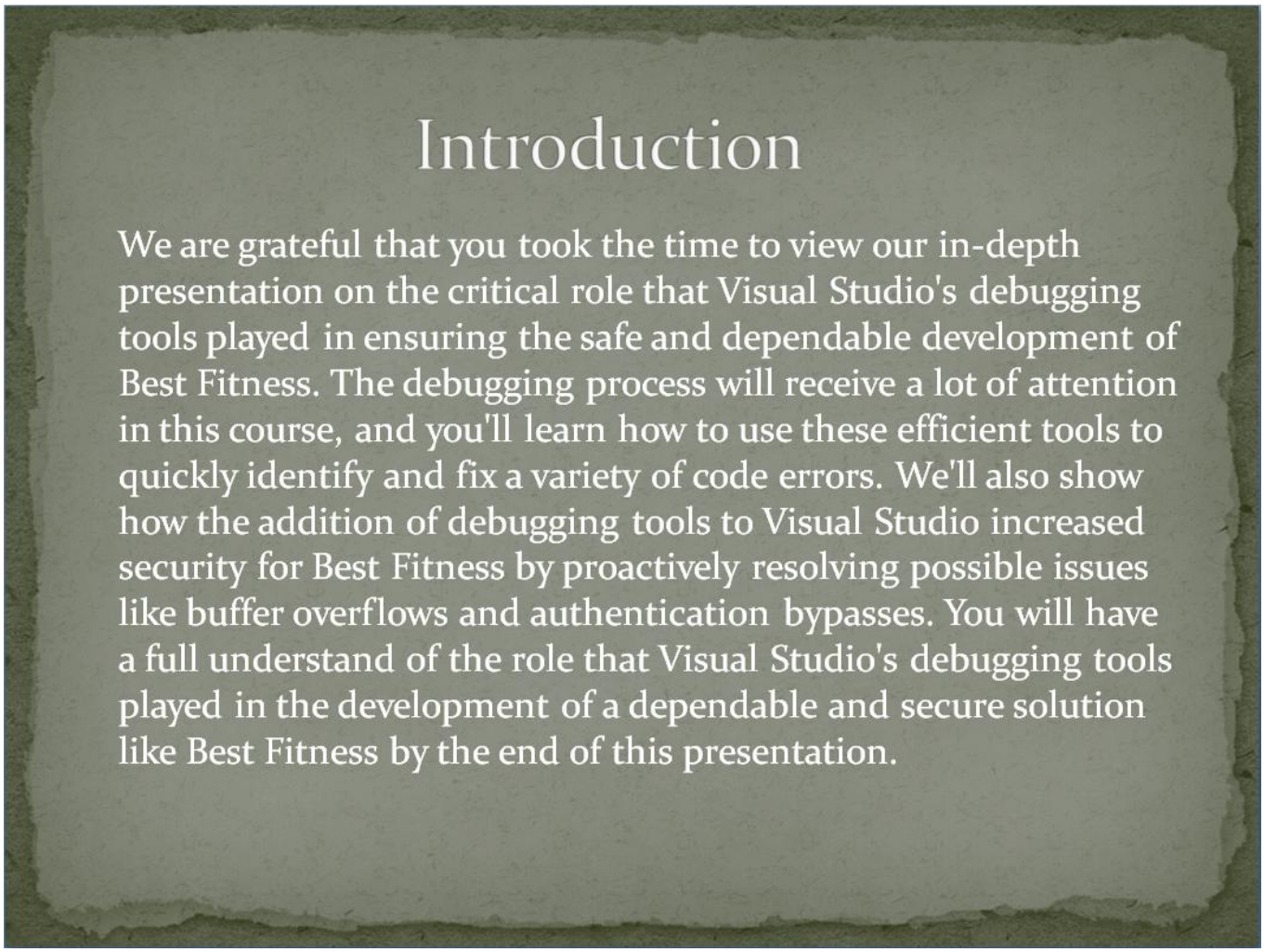
Despite these difficulties, adopting an IDE has more benefits than drawbacks, particularly for teams working on difficult projects and lengthy software development. It is crucial to consider the characteristics of the finished product as well as the needs and preferences of the development team when selecting an IDE.

Conclusion

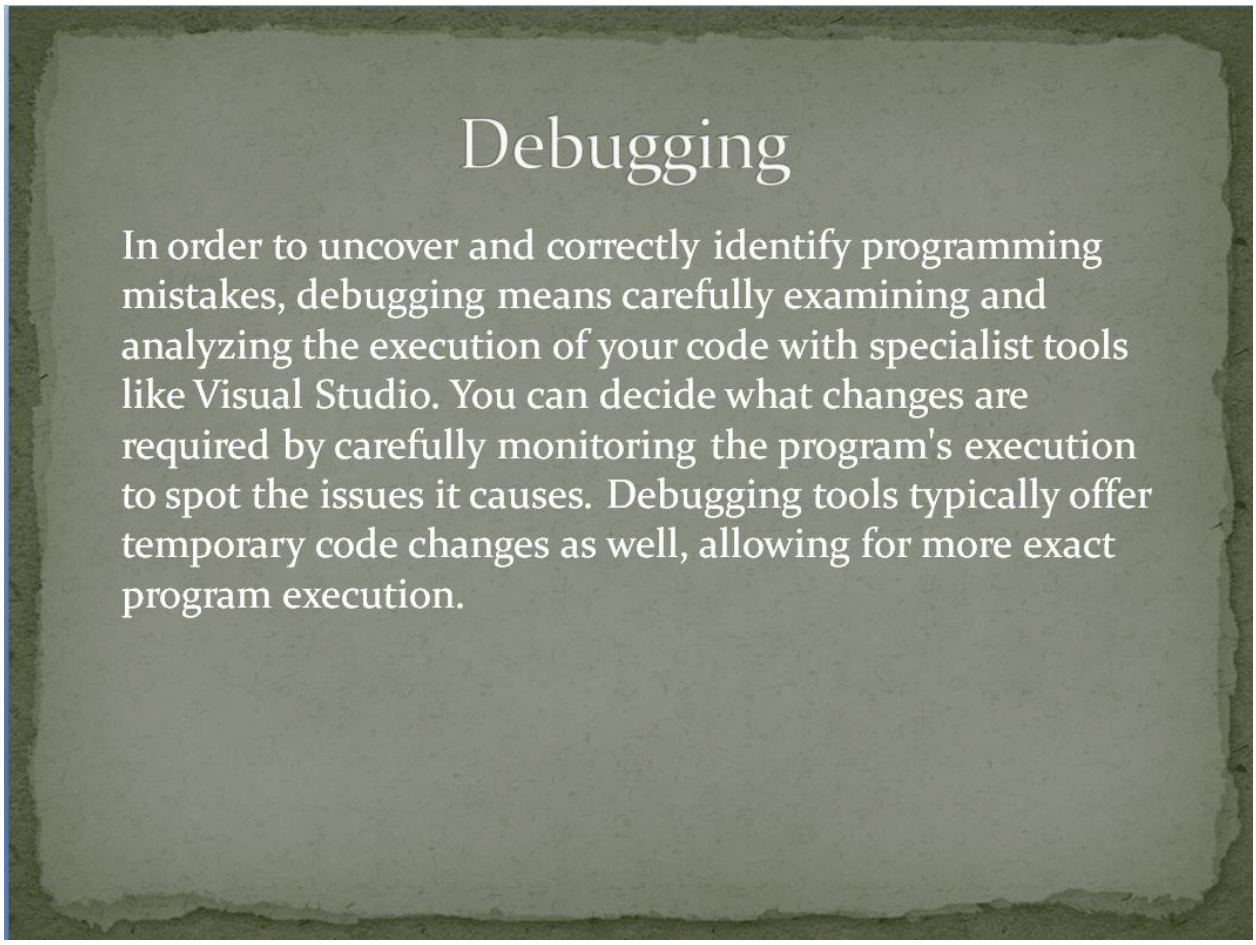
The development of the Best Fitness application with Visual Studio and the C# programming language is thoroughly covered in this in-depth study. Through the use of the IDE's extraordinary features and capabilities, the idea was effectively translated into usable code. The seamless integration of Visual Studio resulted in a remarkable speed in the development and debugging processes. Programmers should exercise caution and find a balance between focusing on IDE features and understanding the foundations of programming, despite the many advantages that IDEs provide. The incorporation of these elements ensures the successful development of complete software.

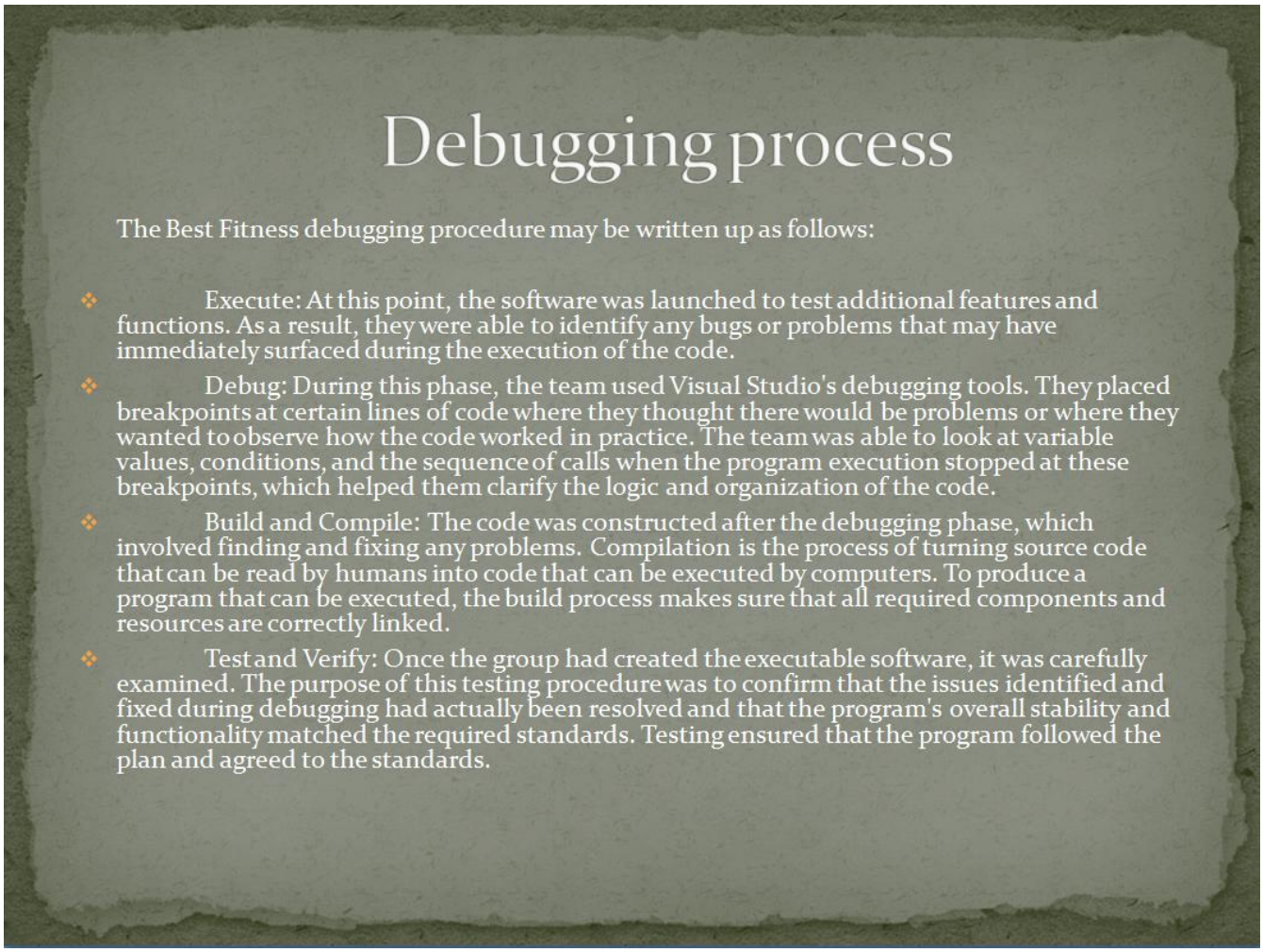
Activity 3

Debugging Features in Visual Studio for Best Fitness Development


Introduction
We are grateful that you took the time to view our in-depth presentation on the critical role that Visual Studio's debugging tools played in ensuring the safe and dependable development of Best Fitness. The debugging process will receive a lot of attention in this course, and you'll learn how to use these efficient tools to quickly identify and fix a variety of code errors. We'll also show how the addition of debugging tools to Visual Studio increased security for Best Fitness by proactively resolving possible issues like buffer overflows and authentication bypasses. You will have a full understand of the role that Visual Studio's debugging tools played in the development of a dependable and secure solution like Best Fitness by the end of this presentation.

(anandmeg, 2023)


Debugging
In order to uncover and correctly identify programming mistakes, debugging means carefully examining and analyzing the execution of your code with specialist tools like Visual Studio. You can decide what changes are required by carefully monitoring the program's execution to spot the issues it causes. Debugging tools typically offer temporary code changes as well, allowing for more exact program execution.



(Mikejo5000, 2023)

Debugging features in Visual Studio

It's nice to learn that the debugging capabilities of Visual Studio were crucial in locating and resolving difficulties when Best Fitness was being developed. Software development requires debugging, and using reliable tools may speed up the process considerably. Let's examine each of the debugging tools you listed in greater detail:

- Breakpoints: In order to prevent programs from executing at certain points in the code, breakpoints must be used. Developers may carefully examine the state of variables while understanding how the program is processing data at crucial places by properly establishing breakpoints. This is especially useful for verifying user input during login processes to make sure that data is accurately verified and to find any possible errors.
- Call Stack: The call stack is a useful tool for viewing the arrangement of function calls. It provides an easy-to-understand explanation of the sequence of calls to functions that brought the code to its current location. When trying to find mistakes or issues with nested function calls, this is quite helpful. The Call Stack allows developers to quickly figure out the sequence of activities that led to a given problem.
- Step over/into: The Step Over and Step Into options are excellent for step-by-step debugging, which enables you to carefully examine the application's execution sequence. Developers may follow complex code pathways, such as those involved in payment and registration processing, by examining the code line by line. This makes it simpler to spot any problems or logical errors that could occur during execution and helps to guarantee that the software works as planned.
- Exception Handling: To write dependable programs, one must be able to manage exceptions. Programmers may respond quickly to unexpected circumstances with special case handling features in Visual Studio. By providing efficient handling of errors in crucial components like connection to databases and file operations, the team can ensure that the application handles issues effectively and avoids shutting down unexpectedly.
- Watch Window: While the application is running, programmers may keep an eye on variable values using the Watch Window, a real-time monitoring tool. For recognizing unanticipated actions and hiding issues, it's crucial to understand how particular variables change throughout the course of the application's runtime. By providing developers with immediate knowledge of the condition of those variables in their program, it facilitates faster debugging.

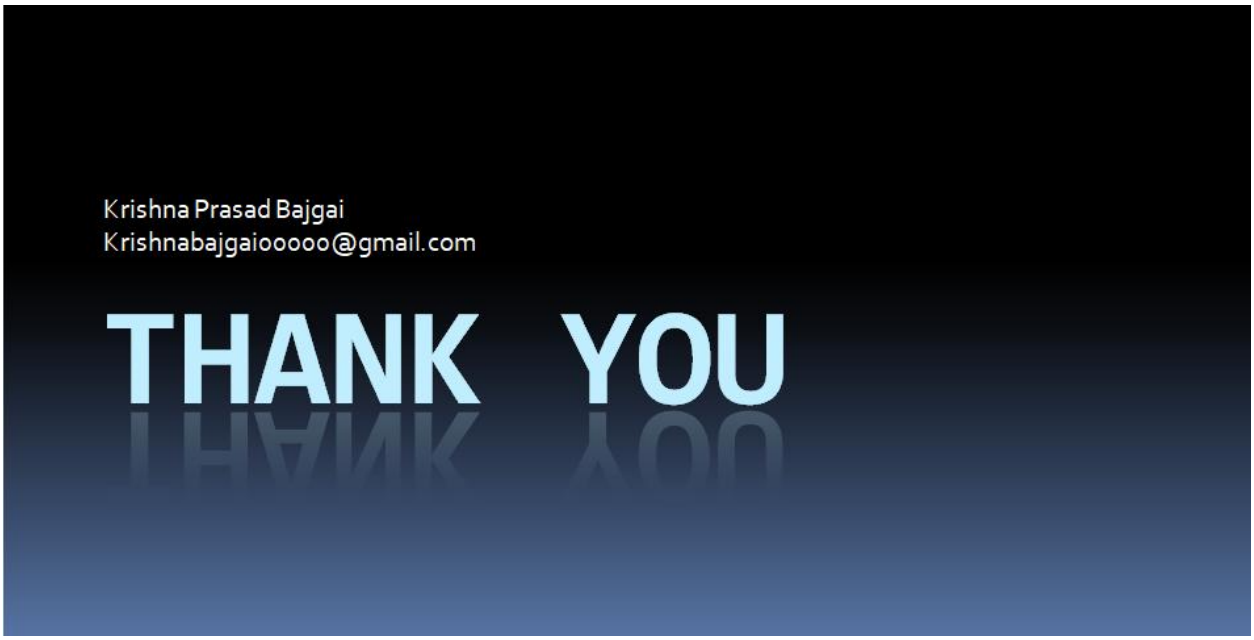
What advantages does debugging have for application security?

Increasing application security requires debugging since it makes it easier to find and solve software bugs. The examples below show how debugging boosted the app's security and benefited Best Fitness:

- Buffer Overflow Vulnerability: When software attempts to write more data into a buffer than it can contain, nearby areas of memory are overwritten. In this circumstance, a buffer overflow occurs. Attackers can utilize this to insert malicious code into the program. The team was able to examine memory consumption and data tampering throughout the file upload process using debugging tools. As a consequence, they were able to locate the buffer overflow problem and quickly solve it. They improved the application's security by fixing the problem and thwarting attempts to introduce malicious code.
- Data Validation: To guarantee that user input is correct and safe for use, data validation is important. Attackers may use weaknesses in security like SQL injection, in which they insert unauthorized SQL code into input fields to change the program's database, if there is insufficient validation. When troubleshooting, the team was able to extensively investigate user input thanks to Visual Studio's debugging features. They were able to find the data validation weakness that would allow for SQL injection attacks as a result. In light of their discoveries, they improved the data validation procedures, which boosted the app's overall security by boosting its defenses against similar attackers.
- Authentication Vulnerabilities: Authentication, which ensures that only those with permission may access certain data or functionalities, is a need for application security. The security of the software might be endangered by an authentication fault that permits unapproved use or authentication bypass. Through debugging, the team was able to identify the validation code issue that resulted in the authentication bypass vulnerability. By identifying the underlying cause and putting stronger authentication checks in place, they were able to shut the security gap and stop illegal access to user data.

Conclusion

The role of Visual Studio's debugging tools in the development of Best Fitness cannot be underestimated. Because of the power of these technologies, the development team was able to quickly recognize and fix a range of errors and potential security gaps. By carefully establishing breakpoints, carefully examining variable values, and carefully going through the call stack, they were able to figure out the behavior of the program in real time. As a result, the team was able to deal with issues quickly, ensuring Best Fitness' entire reliability and safety. We were able to progress Best Fitness to a level of perfection where clients may benefit from quick and secure software with the help of the deep and feature-rich Visual Studio programming environment.


Krishna Prasad Bajgai
Krishnabajgaiooooo@gmail.com
THANK YOU

Activity 4

Introduction

This in-depth study provides a complete analysis of the techniques utilized at Himalayan Digital Solution (HDS) to develop and implement the most recent version of the Optimal Fitness algorithm, which uses Visual Studio as the Integrated Development Environment (IDE).

It is also recognized how crucial it is for software development teams and individual programmers to apply coding standards.

Designing and implementing algorithms

As part of the algorithm design process for the Best Fitness program, the client's requirements are carefully analyzed, and the work is broken into more manageable, smaller sections. I created algorithms while I was a young software developer at HDS with the intention of making them efficient, flexible, and simple to maintain. I worked with my team leader to make the fundamental algorithm more user-friendly.

The intended functionality was delivered by the technique's final implementation, which used data structures and programming concepts. I used conditional statements, loops, and data manipulation techniques to determine the cost of training based on the needs of each customer. With the help of code reviews and conversations with leading HDS experts, the code was continually enhanced and optimized for performance.

Algorithm and source code relationships

The real source code and the written algorithm have a close relationship. The logical parts of the process were translated into source code using a specific programming language. The logical

steps needed to determine the training expenses were defined by the algorithm. The software's structure and functions were laid out in the algorithm, which served as the code's design.

Consistency between the algorithm and the source code was crucial for easy comprehension, debugging, and potential future improvements. The software continued to meet the client's requirements since any algorithmic changes made during development were reflected in the source code.

Integrated Development Environment (IDE) evaluation

I used the HDS-recommended IDE to construct the Best Fitness application since it offered a wide range of tools and capabilities. The development process was significantly more efficient and effective as a result of the IDE. The primary helpful IDE elements were as follows:

- **Code editor:** The heart of an Integrated Development Environment (IDE) is the Code Editor (CE). This useful application looks like a typical text editor on the surface, but it actually hides a number of features designed to speed up the software development process. The Code Editor's ability to accurately predict a programmer's goals as they interact with it and skillfully complete instructions using its auto-completion function makes the writing process surprisingly quick and simple.
- **The compiling engine:** Compilers included in Integrated Development Environments (IDEs) rapidly assess the performance of the code. Since the compilation results are frequently shown in a different interface that is still a part of the IDE, the transition between modifying code and the compiler's feedback is typically smooth and free of any visible lag.
- **Debugging tools**: Integrated Development Environments (IDEs) occasionally include testing tools with the goal of identifying application bugs in the source code. These tools

are quite good at spotting other programming mistakes, such incorrect syntax, missing variables, misplaced instructions, and more, but they could miss logical mistakes. A major advantage of these tools is their capacity to accurately detect these issues, which facilitates programmers' debugging.

- **Class and object browser:** An integrated development environment (IDE) that is capable and full of features is commonly available to OOP programmers. One of these tools enables flexible exploration of program objects and classes for assessment and analysis. The IDE goes a step further by providing programmers with a clear understanding of the class structure, allowing them to maximize object reuse and enhance overall coding efficiency.
- **Build automation:** Automated build methods make it easier to prepare software code. Many integrated development environments (IDEs) offer built-in build automation features, even if some developers prefer to use separate build environments. These useful tools aid in packaging the code for straightforward implementation after development is complete. (Spiceworks, 2022)

Making progress without an IDE

Creating the Best Fitness program would have been much more difficult and time-consuming if I hadn't had access to the IDE. I would have had to use a simple text editor, extra debugging tools, and manually manage version control if the IDE's new features hadn't been available.

- **Time Spent on Development:** The time spent on development would have taken longer if productivity aids like code auto completion hadn't been available.
- **Debugging Challenges:** Without an integrated debugger, debugging would have been time-consuming and ineffective, making it more difficult to identify and fix faults.

- **Risk of mistakes:** Team member arguments and code disagreements are more likely when version control is managed manually.
- **Code inconsistency:** The IDE's ability to format code was essential for maintaining a consistent technique for development. It made following code standards easier than it would have been otherwise.

Role and Purpose of Coding Standards Evaluation

Coding standards provide a set of guidelines for solo programmers and teams developing software that ensure accuracy, readability, and maintainability of the code base. Coding guidelines are necessary for a variety of reasons, such as:

- **Team Collaboration:** By ensuring that all team members write code consistently and with a sense of shared ownership for the code base, coding standards help to promote unity among teammates.
- **Maintainability:** It is simpler to maintain standard-compliant code updated and bug-free since developers can more readily understand its logic and structure.
- **Readability:** The code base is made easier to read and understand by employing similar name and formatting rules, which promotes cooperation.
- **Reduced mistakes:** Coding standards aid in lowering frequent errors and improving overall code quality by detecting potential issues early in the development process.
- **Scalability:** Since new developers may easily catch up to the established standards thanks to consistent code, developing a program as it expands is made easier.
- **Coding Standards:** Coding standards offer a structure for code reviews, enabling constructive criticism and ensuring that best practices are followed to.

Conclusion

The creation of the Best Fitness algorithm was beneficial since the final source code mostly followed the original concept. The team was convinced to select Visual Studio since it accelerated work and promoted productive cooperation. By emphasizing the importance of complying to coding standards, the high level of quality and clarity of the code was significantly maintained. The end result was a strong and successful Best Fitness program for Himalayan Digital Solutions (HDS), which was made possible by a challenging algorithm, the incredible power of Visual Studio, and a strict dedication to code standards.

References

bootpootin (2021). *Procedural, Object-oriented and event driven Programming Languages (Visual Basic). - Boot Poot*. [online] Boot Poot. Available at: https://bootpoot.tech/procedural-object-oriented-and-event-driven-programming-languages-visual-basic/ [Accessed 27 Jul. 2023].

Upadhyay, S. (2021). *What Is An Algorithm? Definition, Types, Characteristics*. [online] Simplilearn.com. Available at: https://www.simplilearn.com/tutorials/data-structure-tutorial/what-is-an-algorithm# [Accessed 26 Jul. 2023].

Shiverware (2019). *7 Steps of App Development*. [online] Shiverware.com. Available at: https://shiverware.com/app-development/7-steps-app-development.html [Accessed 26 Jul. 2023].

Bratty, L. (2023). *What are coding standards? Do we really need them? - iLearn Engineering®*. [online] iLearn Engineering®. Available at: https://www.ilearnengineering.com/computer/what-are-coding-standards-do-we-really-need-them [Accessed 26 Jul. 2023].

anandmeg (2023). *What is Visual Studio?* [online] Microsoft.com. Available at: https://learn.microsoft.com/en-us/visualstudio/get-started/visual-studio-ide?view=vs-2022 [Accessed 27 Jul. 2023].

Mikejo5000 (2023). *Debugging code for absolute beginners - Visual Studio (Windows)*. [online] Microsoft.com. Available at: https://learn.microsoft.com/en-us/visualstudio/debugger/debugging-absolute-beginners?view=vs-2022&tabs=csharp [Accessed 27 Jul. 2023].

Spiceworks. (2022). *Why is Integrated Development Environment (IDE) Important?* [online] Available at: https://www.spiceworks.com/tech/devops/articles/what-is-ide/ [Accessed 26 Jul. 2023].