جامعة حفر الباطن
University of Hafr Al Batin

Computer Science and Engineering (CSE)

COOP FINAL REPORT
2022

Student name: Raghad Talal Halael Inad

Student ID: [redacted]

Major: CSE

COOP Advisor: Dr. Ammar Elhadi, Dr. Ibrahim Alzahrani

Trained at:

- King Fahad Medical City at (Web Development Dep).

Acknowledgments

First and foremost, I want to express my sincere thanks to Allah for all of the benefits that he has bestowed upon us. I also want to express my gratitude to my parents and family for being by my side throughout my entire life, supporting and caring for me from the moment I was born till now.

Those who gave us the opportunity to complete this report deserve our deepest appreciation, and I would like to express it. Special gratitude to my academic advisors, Dr. Ammar Elhadi, and Dr. Ibrahim Alzahrani, for their constant support, guidance, and assistance. My sincere thanks and respect also extend to Eng. Ali Mohamed Hamidaddin in King Fahad Medical City for all of the assistance he provided, the knowledge I learned from him, and his support throughout the training program. His level of patience and ingenuity is something I will always keep aspiring to. This training program would've been difficult if it were not for their help and cooperation.

Finally, I have valued this opportunity and learned a lot that I will utilize in the future.

ABSTRACT

The training program is an internship program between the university and an external organization, and it is a great opportunity for the students to put into practice what they have learned theoretically throughout their college years, develop their skills and abilities, and prepare them for work. This report's intent is to provide the reader with a brief look at my COOP training program at King Fahad Medical City's (KFMC) Web Development Department (online). With all the things I've done and achieved during my training, which lasted from June 2022 to December 2022.

There are two key chapters in this report. The first chapter, with the heading King Fahd Medical City, is divided into 6 sub-chapters, that begin with a brief history of King Fahd Medical City and the department, Website Development Department, that was in charge of my training. It also includes a thorough overview of the projects and websites I worked on. Additionally, it includes the primary project that served as the basis of my team's and my own evaluation.

Lastly, it ends with a chapter of conclusions and recommendations for improving co-op training for any other student who will take a co-op program.

Table of Contents

LIST OF FIGURE

LIST OF TABLES

CHAPTER 1: King Fahad Medical City at (Web Development Dep)

This chapter is separated into multiple chapters with step-by-step explanations of the work I have done during the period of cooperative training in King Fahad Medical City.

CHAPTER 1.1: INTRODUCTION

1.1.1: Company Introduction:

King Fahad Medical City, also popularly abbreviated as "KFMC" is one of the largest and fastest-growing medical complexes in the Middle East, with a total capacity of 1200 beds. It has a highly qualified and efficient technical management team. King Fahad Medical City started as a pioneering idea that came true. King Fahad bin Abdulaziz accepted the idea. King Salman bin Abdulaziz, prince of Riyadh at the time, laid the foundation stone for King Fahad Medical City in 1403, which corresponds to 1983. KFMC was established under the reign of King Fahad bin Abdulaziz, the Custodian of the Two Holy Mosques, and was officially opened on August 21, 1425, H, or October 5, 2004, by King Abdullah bin Abdulaziz, the Crown Prince. (City, n.d.)

1.1.2: Work Agenda of Web Development Department:

KFMC has many different IT departments. My training was in executive management for information technology, specifically in the web development department. We were focused on a practical approach to gathering business requirements, designing, developing, and deploying a web-based application that automates business processes based on business needs.

1.1.3: Web Development Department Used:

The most significant thing in development is database design. For a high-performance application, the design must come before the real coding, to understand and create a reliable database, we used Microsoft SQL Server.

Microsoft Visual Studio, the second technology, is an IDE (integrated development environment) developed by Microsoft Corporation and utilized for various software development processes. We can work with a wide range of topics and programming languages, including HTML, C#, CSS, and Bootstrap, using Microsoft Visual Studio.

1.1.4: Work Process:

The following are the training tasks established for the training program:

Work Task
• Install development tools (MS SQL Server & Visual Studio)
• Review Database & programming concepts (DML practice)
• Working with Visual Studio, creating a new project to design a simple web app
• backup, restore, upload DB & web project to host services (project deployment)
• Business problem case discussion, analyzing requirements, designing a solution
• developing a web project, creating User interfaces, connecting to DB
• Developing classes, configuring web projects, creating middle-tier BL layer
• Creating CRUD class, applying security & configuration to project
• Creating Navigation menu, continue development
• testing application for bugs and fixing bugs
• Creating Membership, Roles, and Responsibilities, continue development
• Adding features to web applications such as Export to Excel, Word, and Pdf
• Adding email notifications to the web application by configuring the SMTP service
• Backup local Db, Restore Db to host, Create deployment packages

• Deploying and publishing a web application to host
• Working with Visual Studio, to upload files to a file system and database
• Applying server-side and client-side validation
• Learning Mail Merge using Macros
• Learning Mail Merge using Syncfusion
• Sending Emails with attachments using Syncfusion
• Overview of Git

Table 1 Work Tasks

CHAPTER 1.2: TRAINING PREPARATION

Since the KFMC's training was conducted online, this chapter gives a quick overview of how we communicate with the trainer and attend meetings.

1.2.1: Preparation of a room in Zoom App:

Following my acceptance, I received an email with a link to the Zoom meeting room that we use to hold daily meetings. The trainer took care of how to write everyone's names so that it was consistent. It starts with the trainee's file number, followed by the university code, a trainee or visitor, inside or online training, the trainee's name, the term, and the year. By "university code" we mean the first and last letters of the university name.

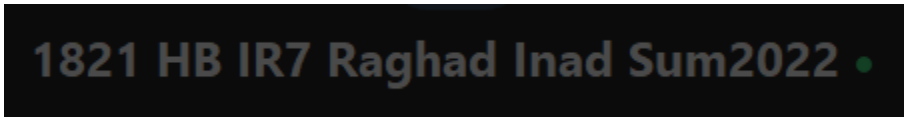


Figure 1 Naming in Zoom

1.2.2: Telegram Groups:

A link to two Telegram groups was also included in the email I received after being accepted.

1.2.2.1: WebAppDev_Tasks:

Only the trainer will be giving instructions and directives to this group. Sometimes the trainer will provide links that will help enhance our learning and may even include some of the city-hosted courses, like introduction and employment qualification programs. This group reports on attendance records, assignments, and necessary documents.

1.2.2.2: WebAppDev_FAQ:

The purpose of this group is to discuss the difficulties someone faces and identify potential solutions while interacting with group members.

1.2.3: HARDWARE AND SOFTWARE REQUIREMENT:

1.2.3.1: Hardware requirement:

Hardware requirement
• Intel i5 processor or higher
• 8 RAM or higher

Table 2 Hardware requirement

1.2.3.2: Software requirement:

Software requirement
• MS SQL Server Management Studio (SSMS)
• MS Visual Studio

Table 3 Software requirement

CHAPTER 1.3: MAIN PROGRAMS

This chapter will include a brief introduction to each program we used daily and worked on during this course, as well as a summary of the steps we took to complete it.

1.3.1: Microsoft SQL Server:

The database is the basic seed for building websites, and it feeds the website with information. A database is a planned grouping of material that has been arranged and is often kept electronically in a computer system. For writing and querying data, most databases employ structured query language (SQL). Almost all relational databases employ SQL, a programming language, to query, manage, and define data as well as to provide access control. Since a well-designed database ensures data accuracy, we have been working with **Microsoft SQL Server** to learn and develop one. As a result, we had to learn how to design a database that only saves relevant and useful data. The first task assigned to us was to install **Microsoft SQL Server**.

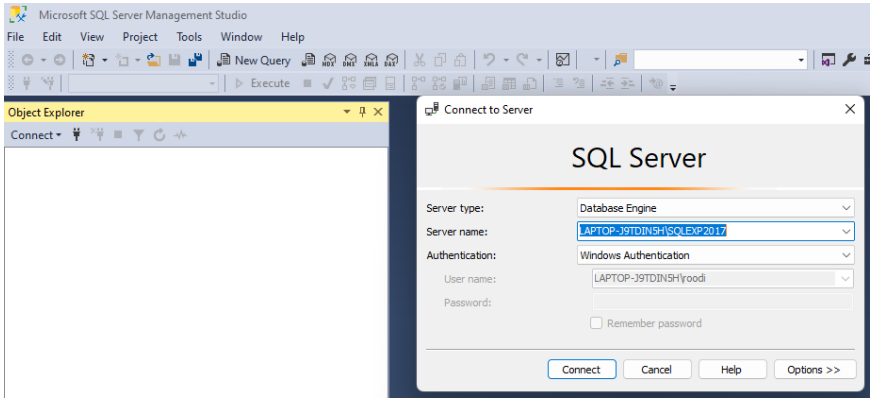


Figure 2 SQL Server Interface

After choosing your laptop as a local server name and then clicking on connect, the main window will show up and all the databases in the system will be available to work with. The second task was to learn how to back up and restore a database in order to use it on another device or to exchange database information between group members.

- To back up the database we follow the steps shown in the figures below:

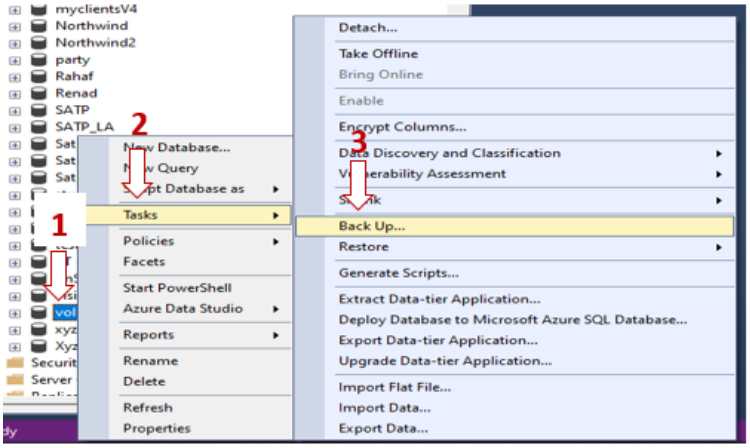


Figure 3: Backup a Database

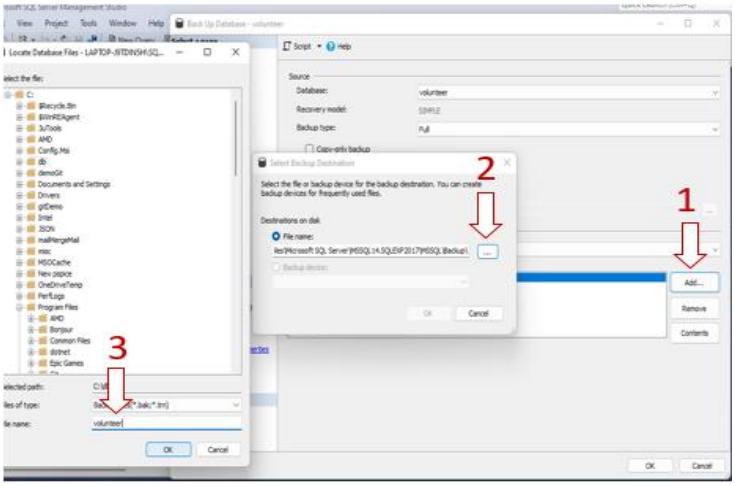


Figure 4: Database Backup successfully

- If I receive a backup of a previously prepared database I need to restore it first, by following the steps shown in the figures below:

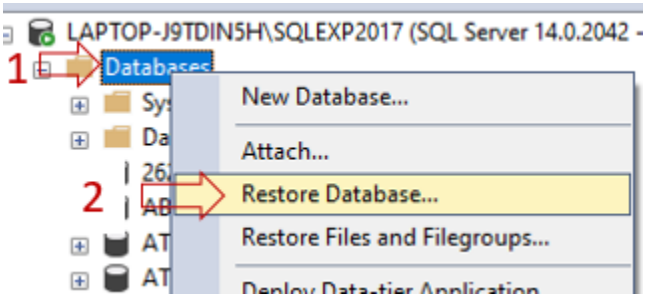


Figure 5: Restore a database

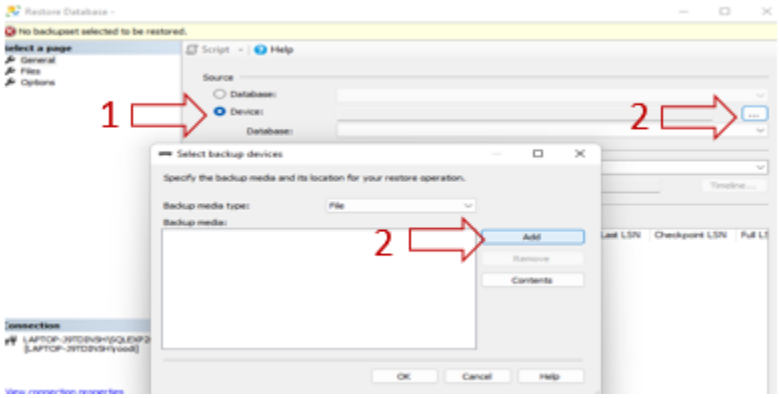


Figure 6: steps to restore a database

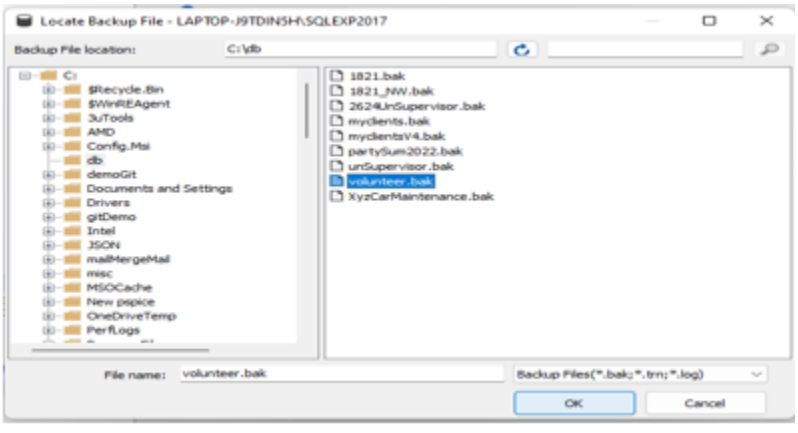


Figure 7: database restored successfully

After that, we got deeper and deeper with SQL and learned many uses of MS SQL Server that were used to support our training, such as:

- Create table

The tables are the main things in the database since they contain the data, to create one, from Tables → New → Table, as shown in the figure below:

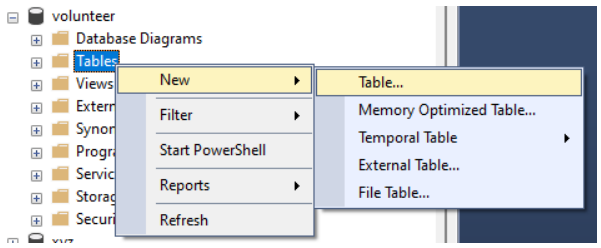


Figure 8: Create table

After creating a table, we must choose the fields names, and datatype and set a primary key.

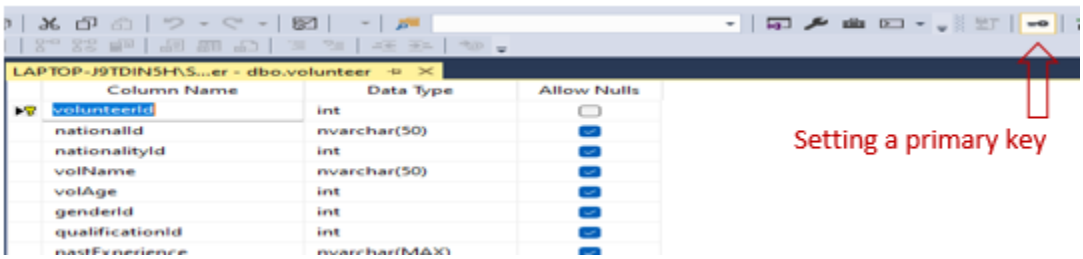


Figure 9: Setting a PK, fields, datatype

Then we go on to the query; there are four major queries that each database platform supports:

- Insert Statement

```
15
16  insert into volunteer (nationalId,nationalityId,volName,volAge,genderId,qualificationId,pastExperience,note,contact)
17  values (1185356754, 3 , 'Raghad', 23 , 2 , 4 , 'no past experience' , 'nothing' , 'phone # 0557766787' )
```

Messages

```
(1 row affected)

Completion time: 2022-11-26T04:44:48.1404097+03:00
```

Figure 10: Insert Statement

- Delete Statement

```
19  delete from volunteer where volunteerId = 31
```

Messages

```
(1 row affected)

Completion time: 2022-11-26T04:46:19.3468612+03:00
```

Figure 11: Delete Statement

- Select Statement

```
20
21  select volName , volAge from volunteer
```

Results | Messages

	volName	volAge
1	Raghad Talal	34
2	Ahmed Ali	30
3	Asma Hammad	32
4	Manar Omar	26
5	Omar Turky	17
6	Azam Alhrbi	18
7	Mariah Almunea	19
8	nawal saad	23

Figure 12: Select Statement

- Update Statement

```
23  update volunteer set volAge = 21 where volunteerId = 10
```

Messages

```
(1 row affected)

Completion time: 2022-11-26T04:49:40.5687913+03:00
```

Figure 13: Update Statement

Then we had two important concepts in SQL which are DB views, and stored procedures.

- View is simple showcasing data stored in the database tables

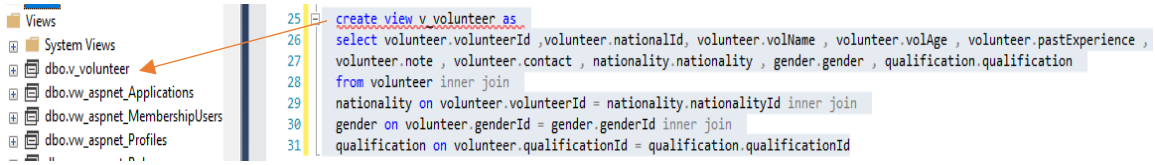


Figure 14: Create View

- Stored procedure is a group of statements that can be executed

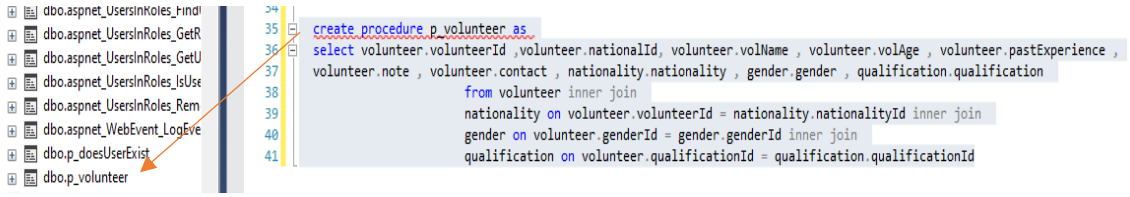


Figure 15: Stored procedure

1.3.2: Microsoft Visual Studio:

Visual Studio is a Microsoft Integrated Development Environment (IDE) used to create GUIs, consoles, web applications, web apps, mobile apps, cloud services, and web services, among other things. With the help of this IDE. Compilers, completion tools, and other features are included to make the software development process easier. In Microsoft Visual Studio, we have worked with ASP.NET Web Forms, which is a part of the ASP.NET web application framework, for building great websites and web applications using HTML, CSS, and JavaScript, which we used in our training. (Microsoft® Visual Studio 2017, n.d.) (microsoft, n.d.)

- Interface of Microsoft Visual Studio:

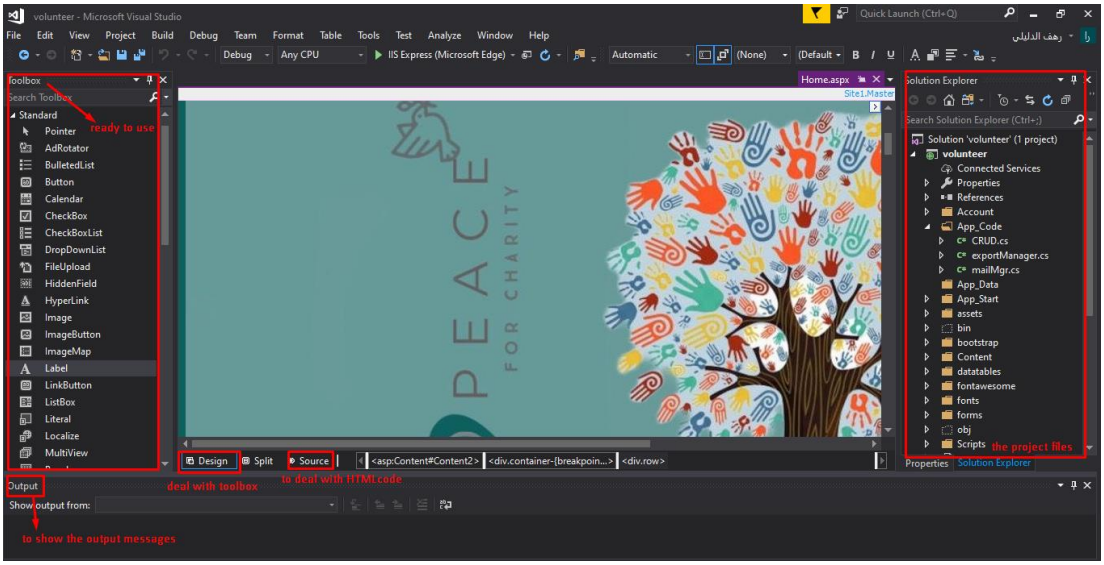


Figure 16: Microsoft Visual Studio Interface

- Creating new pages:

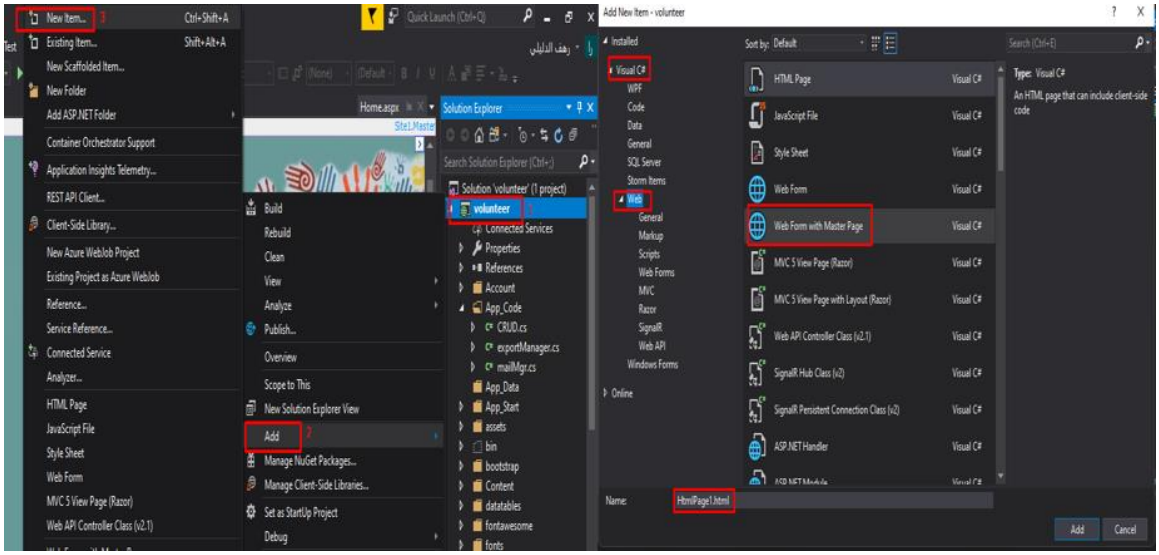


Figure 17: Adding New Page steps

- Most Asp controls used:

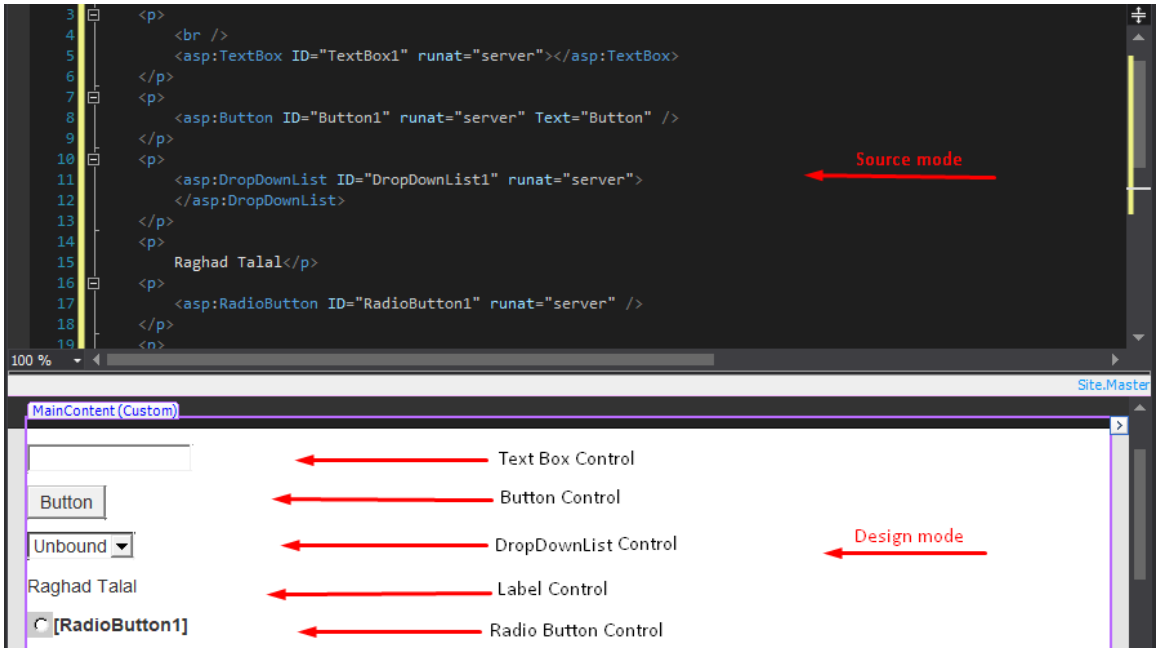


Figure 18: Asp Controls

- Switching into programming mode (using C# language):

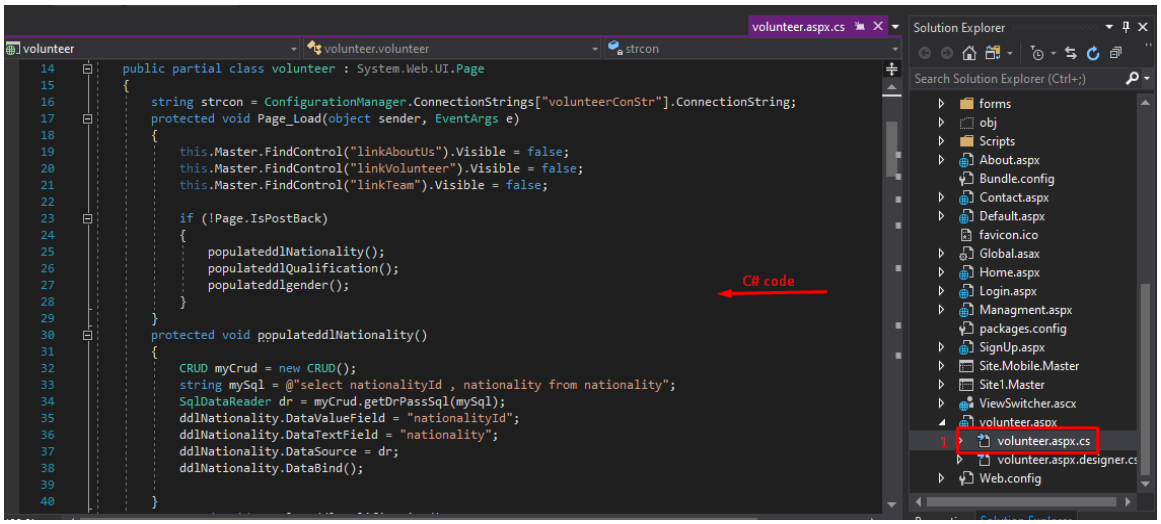


Figure 19: C# Code

1.3.3: Smarter Asp Server:

SmarterASP.NET is a service provider that has data centers in both United States and Europe. It gives you the option to place your site in the region that best suits you and your customers. During my training in King Fahad Medical City, Asp Server was used at the end of the completion of our project. Our database and website have been uploaded to the internet using this server. (SmarterASP.NET, n.d.)

Figure 20: Interface of Smarter Asp Server

- The first step was to create an account, then upload our database

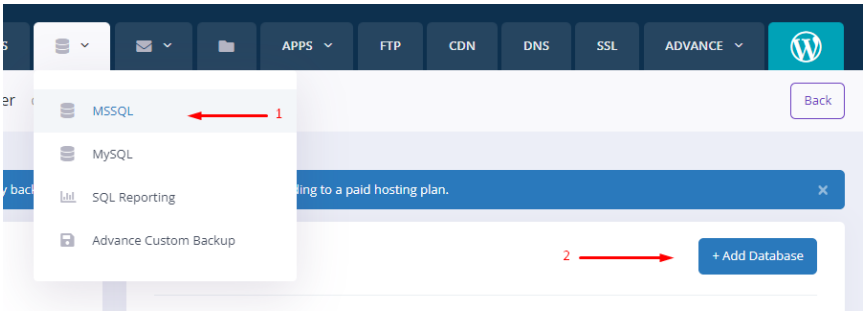


Figure 21: Upload a Database

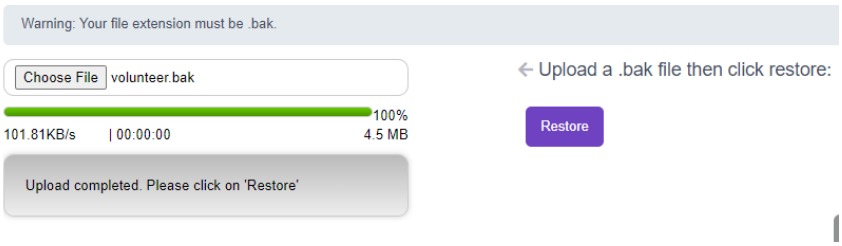


Figure 22: Database Uploaded Successfully

- To publish our application, we start by changing the Connection String from local to host

Figure 23: Connection String host

- Then, install the publishing packet provided by SmarterAsp.

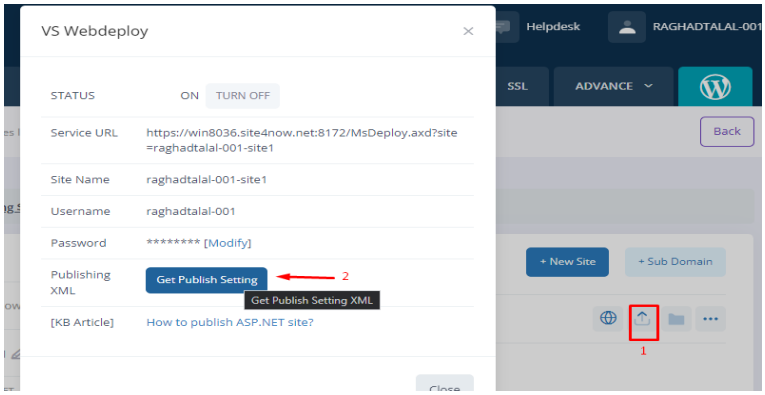


Figure 24: Publishing Packet

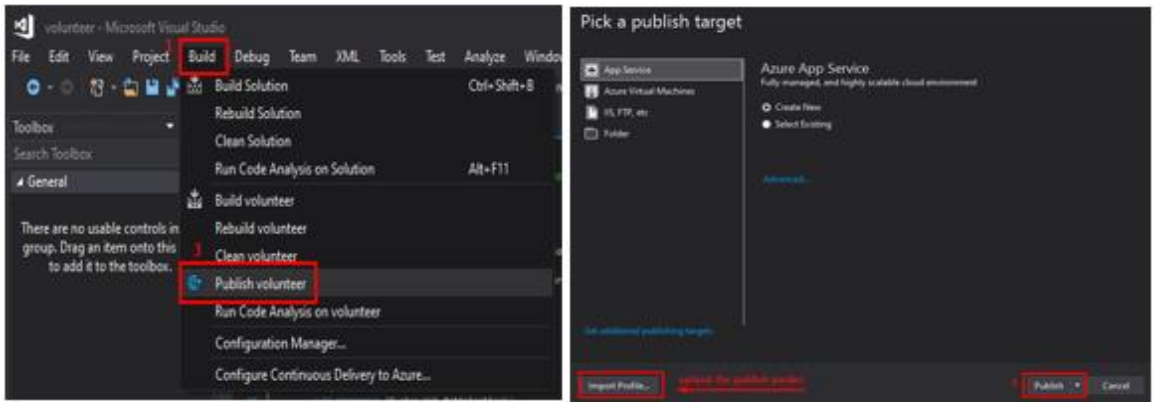


Figure 25: Application Published

CHAPTER 1.4: MAIN PROJECT

This chapter details the work that my team and I did in creating our website **(Peace For Charity).**

1.4.1: Introduction:

The main idea of this project is to create an organization dedicated to promoting peace that facilitates access to volunteer work while fostering the culture of volunteerism throughout the Kingdom by providing the tools and resources needed. My team and I tried to make it easy to use and easy for volunteers to access volunteer opportunities. We organized the pages using Bootstrap and have worked hard to show you the perfect application that fulfills all the requirements.

1.4.2: Requirement of the Main Project:

There are some evaluation criteria that the trainer asks for in the final evaluation of the project, and we had to apply them to our project to get 100 out of 100.

	evaluation criteira	grade
1	CRUD ops	10
2	Navigation	10
3	email notification	10
4	export to excel , word, pdf	10
5	Security	10
6	client side validation	10
7	serverside validation	10
8	DB backup & recovery	10
9	Naming convention & ERD diagram (pk fk relationship)	10
10	deployment package (webhost) & hosting your application	10
		100

Figure 26: Evaluation Criteria

1.4.3: Peace For Charity database:

The first step in creating this application was to create the database, which we did using the Microsoft SQL app, where we created six tables, each with its own primary key and other attributes.

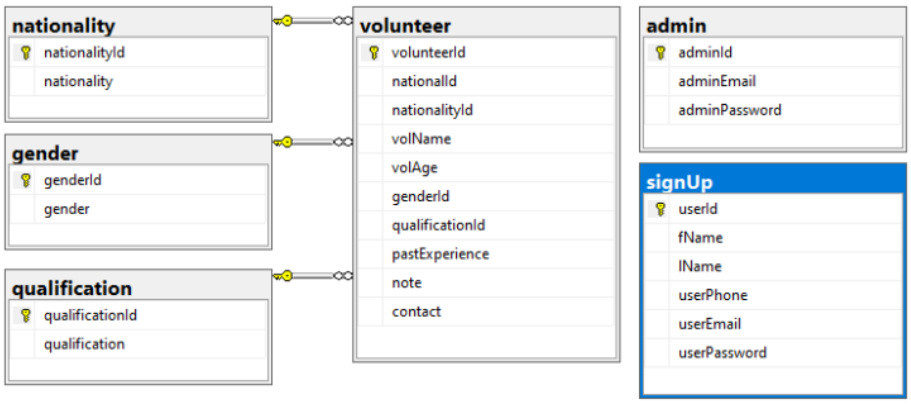


Figure 27: Peace For Charity ER diagram

Volunteer is the main table in the project that holds the information of the volunteers, as you can see in the ER diagram, it has three look-up tables (nationality, gender, and qualification) because these data are always static. They have been connected to the volunteer table by using their primary key as a foreign key.

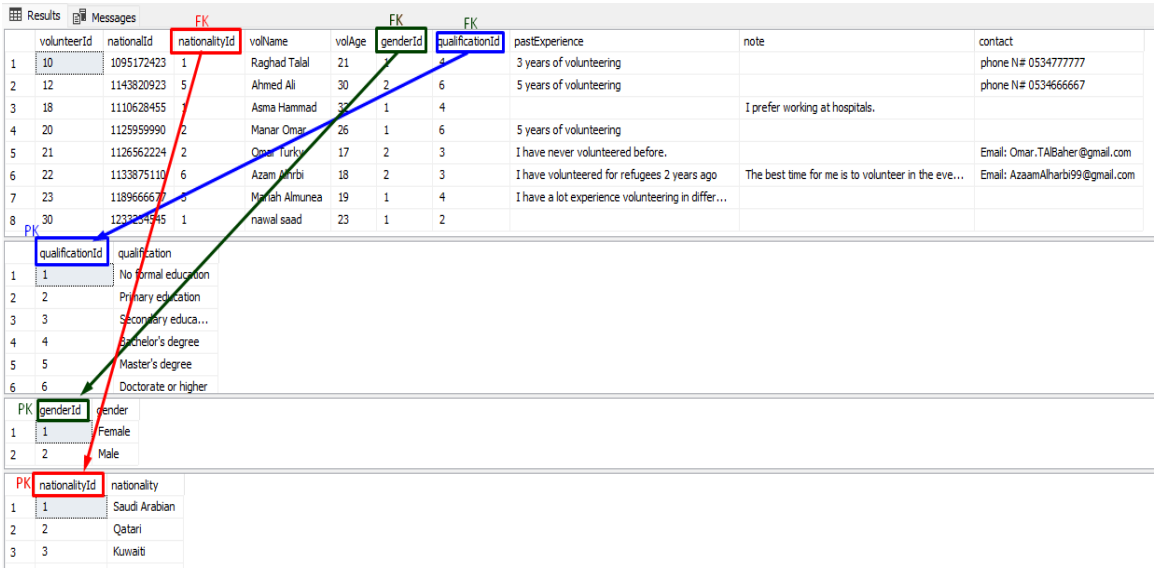


Figure 28: look-up tables

For each application, there is an admin for the site. Five administrators who are members of our team are listed on our application. The administrator is the person who has the authority to modify, delete, and add to volunteers' data.

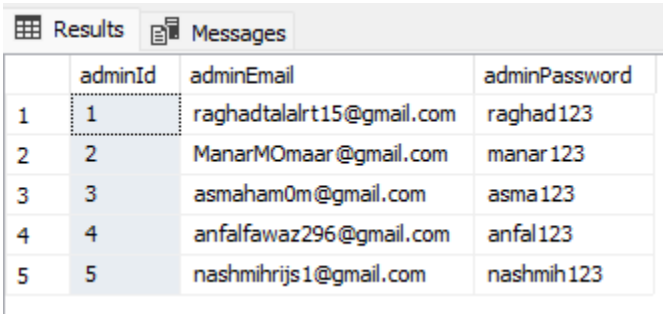


Figure 29: Table of Admin

Every website on the Internet is visited by many customers, including visitors and some of them who register on the site. This is why one of the most important tables that should be in the database is a table that holds the customer data, which is in our project called the "signup" table.

	userId	fName	lName	userPhone	userEmail	userPassword
1	9	Raghad	Talal	0123456789	raghad151315@gmail.com	r123
2	10	Ahmed	Ali	0534666666	tarqnwd@gmail.com	a123
3	11	Asma	Hammad	0534888887	iiiiii4sm@gmail.com	i123
4	12	Latifah	Mohammed	0555543595	iamaroundyo@gmail.com	la123
5	13	Ziyad	Aziz	0533248951	sdfga.131@hotmail.com	ziyad123

Figure 30: SignUp table

1.4.4: Peace For Charity Visual Studio:

The first step in Microsoft Visual Studio was to learn how we can connect our project with the database. This step can be completed within the web config page, where we established the connection string code that is present by default in any project. There are two types of connection strings one is the local connection (containing information about the database that is already in our device). The other connection is related to the database that is uploaded in SmarterAsp, both are shown in the figure below.

```
<configuration>
  <connectionStrings>
    <add name="volunteerConStrxx" connectionString="Data Source=LAPTOP-J9TDIN5H\SQLEXP2017;Initial Catalog=volunteer;Integrated Security=True"
      providerName="System.Data.SqlClient" />
    <add name="volunteerConStr" connectionString="Data Source=SQL8002.site4now.net;Initial Catalog=db_a9092b_raghadtalal15;User
        Id=db_a9092b_raghadtalal15_admin;Password=123raghad"
      providerName="System.Data.SqlClient" />
  </connectionStrings>
```

Figure 31: ConnectionString Code

After that, we started to design the pages:

- Master Page:

It is a nonprinting page that you can use as the template for the rest of the pages in your manuscript. It can contain text and graphic items that will appear on all pages. Within the master page, we have written the navigation menu because we want it to appear on all pages, as well as selected a new logo and chosen the appropriate bootstrap design.

Figure 32: Navigation menu view

- Home Page:

The Home page contains the navigation bar, which we designed as a link button in the site master page. The navigation bar's contents (Peace for Charity Logo, Home, About Us, Team, Login, and Sign Up).

Figure 33: Navigation Bar source code

The home button which moves to the home page, and About us, contains who are we, our vision, and our message.

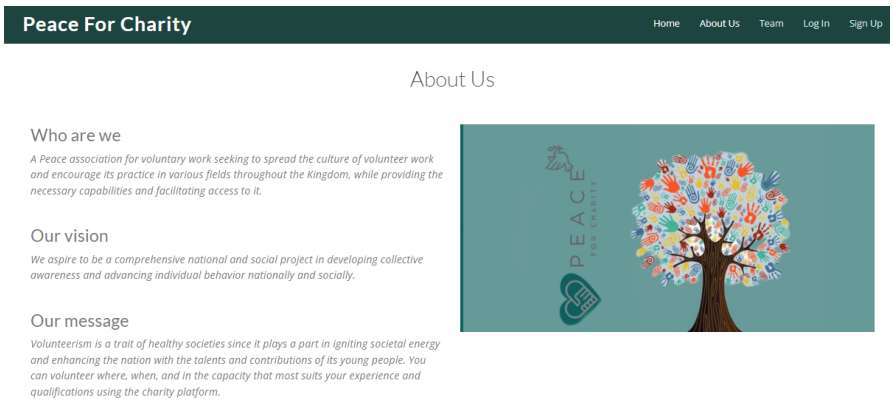


Figure 34: About us

Team, which contains our team's members each member has (Image, Name, Small Introduction, file number in KFMC, and icon of our social accounts).

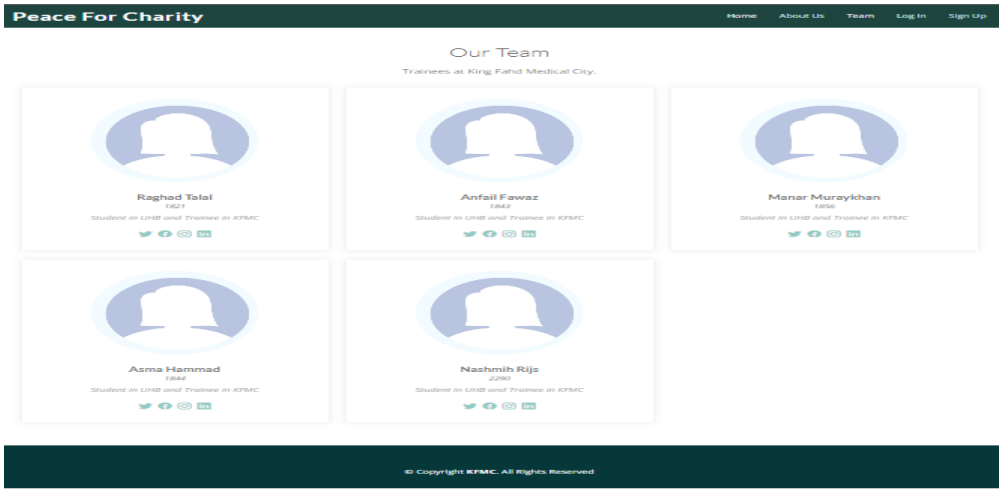


Figure 35: Team

- Security:

It is impossible to deny the significance of security on websites because it affects the safety of users' and administrators' data. Because of this, one of the benefits of our website is the use of security, which enables the website to receive three roles, each with unique features.

1- Visitor role:

The visitor has access to only three pages (Home, Login, and Signup), and it is not possible for the visitor to enter and fill out the volunteer form until he\she signs up for the website.

```
public partial class Site1 : System.Web.UI.MasterPage
{
    protected void Page_Load(object sender, EventArgs e)
    {


        try
        {
            if (Session["role"] == null)
            {
                linkLogIn.Visible = true; // user login link button
                linkSignup.Visible = true; // sign up link button

                linkLogout.Visible = false; // logout link button


                //lbtnAdminLogin.Visible = true; // admin login link button
            }
```

Figure 36: Visitor Role code

Figure 37: Visitor View

2- Sign Up role:

The visitor can choose from the menu to either log in if they already have an account or sign up as a new member of our website.

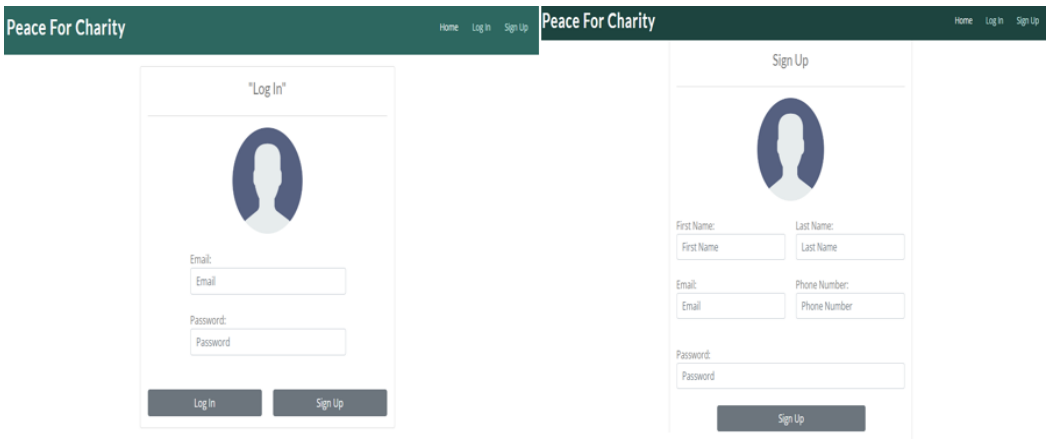


Figure 38: sign-up and log-in pages

After the visitor signs up successfully, they are now considered to be in the Sign-Up role, and they will immediately receive an email with a welcoming message.

```
MailMessage ms = new MailMessage();
ms.From = new MailAddress("appdev4y@gmail.com");
ms.To.Add(txtEmail.Text);
ms.Subject = "Welcome to Peace For Charity!";
ms.Body = "Welcome, We are excites you have joined Peace For Charity family!";
SmtpClient sc = new SmtpClient("smtp.gmail.com", 587);
sc.Port = 587;
sc.Credentials = new NetworkCredential("appdev4y@gmail.com", "aowhaqeyqiarghyr");
sc.EnableSsl = true;
sc.Send(ms);
signUpNewMember();
}
```

Figure 39: Welcome Message (code)

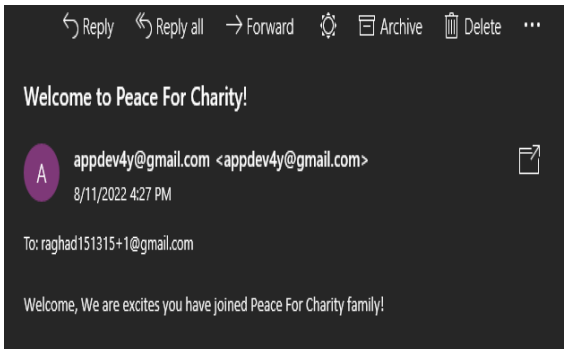


Figure 40: Welcome Message (Email sent)

```
}
else if (Session["role"].Equals("SignUp"))
{
    linkLogIn.Visible = false; // user login link button
    linkSignup.Visible = false; // sign up link button

    linkVolunteer.Visible = true;


    linkLogout.Visible = true; // logout link button

}
```

Figure 41: sing-up Role Code

After successfully logging in, an additional menu called "Volunteer" appears, along with a log-out option.

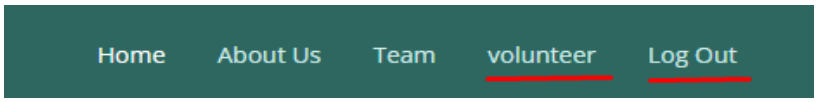


Figure 42: Available menu after logging in

The user who is looking for volunteer work will choose the "Volunteer" menu and complete the Volunteering Form.

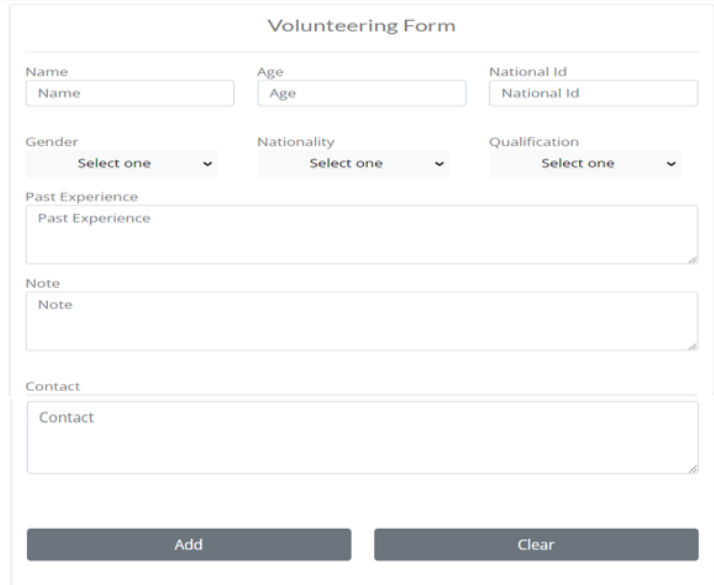


Figure 43: Volunteering Form

3- Admin role:

We established a role for an admin who has more features because any website needs to manage or modify its information. A new volunteer could be added by the admin. Without any SQL interference, he could simply delete any volunteer and update any existing data. Administrators can also view all of the volunteers' data in a grid view, and the data can be exported to Word or Excel.

```
else if (Session["role"].Equals("admin"))
{
    linkLogIn.Visible = false; // user login link button
    linkSignup.Visible = false; // sign up link button

    linkVolunteer.Visible = false;


    linkLogout.Visible = true; // logout link button



    linkManage.Visible = true; // admin management link button
    linkManagment.Visible = true;
}
```

Figure 44: Admin Role code

If the user is logged in as an admin an additional menu called "Manage" will appear in the navigation bar.

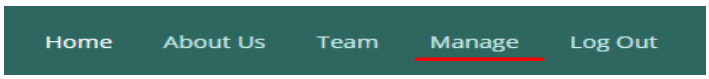


Figure 45: Available menu for admin

The following screen will appear if the administrator selects "manage," allowing him to add new volunteers, edit previous volunteers' information, and delete existing volunteers. As you pointed out, the design of the qualifications, nationality, and gender options is all a dropdown list with static data. They were included in the database as a lookup table primarily for this reason.

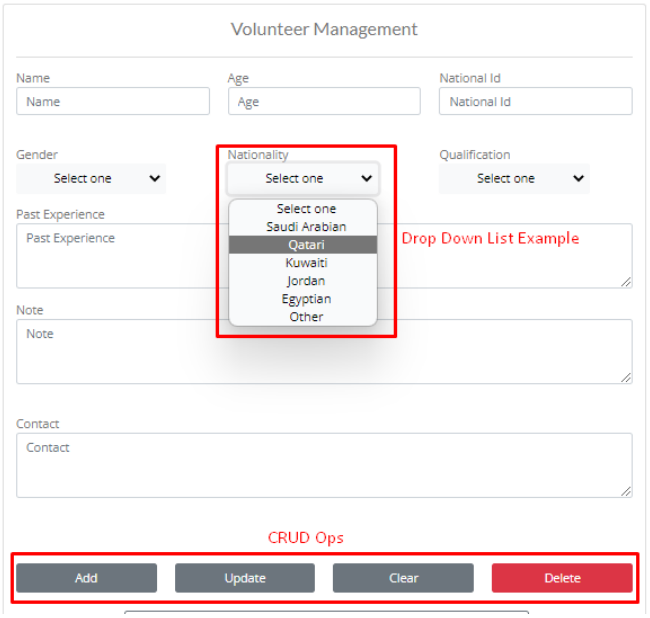


Figure 46: volunteer management page

One of the most crucial components of the bootstrap is a table called the data table, and the grid view button is in charge of showing the information in it. The administrator could then export this data to Word or Excel.

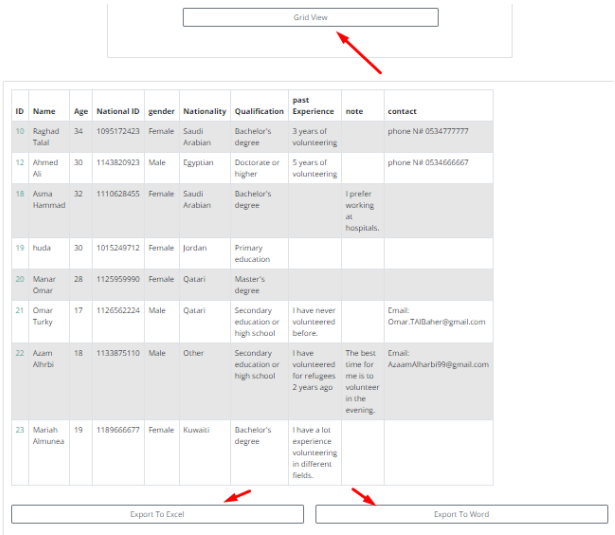


Figure 47: Data Table

In the Grid View, there is a link button (ID) this button is used in the update and delete operations to fill the information in the up text boxes.

```
protected void GetvolunteerData(object sender, EventArgs e)
{
    LinkButton mySender = sender as LinkButton;
    int intvolunteerId = int.Parse(mySender.Text);
    lblOutput.Text = intvolunteerId.ToString();

    CRUD myCrud = new CRUD();
    string mySql = @" select * from volunteer where volunteerId = @volunteerId";
    Dictionary<string, object> myPara = new Dictionary<string, object>();
    myPara.Add("@volunteerId ", intvolunteerId);
    myPara.Add("@volName ", txtvolName.Text);
    myPara.Add("@volAge", txtvolAge.Text);
    myPara.Add("@nationalId", txtNationalId.Text);
    myPara.Add("@genderId", ddlgender.SelectedValue);
    myPara.Add("@nationalityId", ddlNationality.SelectedValue);
    myPara.Add("@qualificationId", ddlQualification.SelectedValue);
    myPara.Add("@pastExperience", txtpastExperience.Text);
    myPara.Add("@note", txtnote.Text);
    myPara.Add("@contact", txtcontact.Text);

    SqlDataReader dr = myCrud.getDrPassSqlDic(mySql, myPara);
    if (dr.HasRows)
    {
        while (dr.Read())
        {
            Session["volunteerId"] = int.Parse(dr["volunteerId"].ToString());
            txtvolName.Text = dr["volName"].ToString();
            txtvolAge.Text = dr["volAge"].ToString();
            txtNationalId.Text = dr["nationalId"].ToString();
            ddlgender.SelectedValue = dr["genderId"].ToString();
            ddlNationality.SelectedValue = dr["nationalityId"].ToString();
            ddlQualification.SelectedValue = dr["qualificationId"].ToString();
            txtpastExperience.Text = dr["pastExperience"].ToString();
            txtnote.Text = dr["note"].ToString();
            txtcontact.Text = dr["contact"].ToString();

        }
    }
```

Figure 48: Code of Link button (ID)

- CRUD Operations :

Add: This button inserts and saves the data of volunteers entered into the website by the end-user into the databases using the insert statement.

```
CRUD myCrud = new CRUD();
string mySql = @"insert into volunteer (volName , volAge , pastExperience , note , contact
        ,nationalId, nationalityId , genderId , qualificationId)
        values (@volName , @volAge , @pastExperience , @note , @contact , @nationalId ,
        @nationalityId, @genderId , @qualificationId) ";
Dictionary<string, object> myPara = new Dictionary<string, object>();
myPara.Add("@volName ", txtvolName.Text);
myPara.Add("@volAge", txtvolAge.Text);
myPara.Add("@nationalId", txtNationalId.Text);
myPara.Add("@genderId", ddlgender.SelectedValue);
myPara.Add("@nationalityId", ddlNationality.SelectedValue);
myPara.Add("@qualificationId", ddlQualification.SelectedValue);
myPara.Add("@pastExperience", txtpastExperience.Text);
myPara.Add("@note", txtnote.Text);
myPara.Add("@contact", txtcontact.Text);

int rtn = myCrud.InsertUpdateDeleteViaSqlDic(mySql, myPara);
if (rtn >= 1)
{
    lblOutput.Text = " ops seccesful ";

}
else
{ lblOutput.Text = " ops faild ! "; }
```

Figure 49: Add code

Update: This button works to update and save the change in DB, after a click on the link button (ID), the data will refill and you can change the desired information using the update statement.

```
protected void btnUpdate_Click(object sender, EventArgs e)
{
    CRUD myCrud = new CRUD();
    string mySql = @" update volunteer set volName=@volName , volAge=@volAge , pastExperience=@pastExperience ,
                      note=@note,contact=@contact, nationalId=@nationalId , nationalityId=@nationalityId,
                      genderId=@genderId, qualificationId=@qualificationId
                      where volunteerId = @volunteerId";
    Dictionary<string, object> myPara = new Dictionary<string, object>();
    myPara.Add("@volunteerId ", Session["volunteerId"]);
    myPara.Add("@volName ", txtvolName.Text);
    myPara.Add("@volAge", txtvolAge.Text);
    myPara.Add("@nationalId", txtNationalId.Text);
    myPara.Add("@genderId", ddlgender.SelectedValue);
    myPara.Add("@nationalityId", ddlNationality.SelectedValue);
    myPara.Add("@qualificationId", ddlQualification.SelectedValue);
    myPara.Add("@pastExperience", txtpastExperience.Text);
    myPara.Add("@note", txtnote.Text);
    myPara.Add("@contact", txtcontact.Text);

    int rtn = myCrud.InsertUpdateDeleteViaSqlDic(mySql, myPara);
    if (rtn >= 1)
    {
        lblOutput.Text = " ops seccesful ";
        populategvVolunteer();

    }
    else
    { lblOutput.Text = " ops faild ! "; }
}
```

Figure 50: Update code

Delete: This button works to delete the data, after a click on the link button (ID), the data will refill, and then you click delete to remove this data from the DB using the delete statement.

```
protected void btnDelete_Click(object sender, EventArgs e)
{
    CRUD myCrud = new CRUD();
    string mySql = @" delete from volunteer
                      where volunteerId = @volunteerId ";
    Dictionary<string, object> myPara = new Dictionary<string, object>();
    myPara.Add("@volunteerId ", Session["volunteerId"]);

    int rtn = myCrud.InsertUpdateDeleteViaSqlDic(mySql, myPara);
    if (rtn >= 1)
    {
        lblOutput.Text = " ops seccesful ";
        populategvVolunteer();

    }
}
```

Figure 51: Delete code

Clear: This button removes information only from the entry page to fill in information for new data using a clear statement.

```
protected void btnClear_Click(object sender, EventArgs e)
{
    lblOutput.Text = "";
    txtvolName.Text = "";
    txtvolAge.Text = "";
    txtNationalId.Text = "";
    ddlgender.SelectedIndex = 0;
    ddlNationality.SelectedIndex = 0;
    ddlQualification.SelectedIndex = 0;
    txtpastExperience.Text = "";
    txtnote.Text = "";
    txtcontact.Text = "";
}
```

Figure 52: Clear code

- Validation:

When we designed our website, we applied two types of validation: one is called client-side validation, and the other is server-side validation.

Client-side validation:

The client-side validation is the software that runs on the user's web browser. We applied it on the sign-up page by using a control in Visual Studio called Regular Expression Validator.

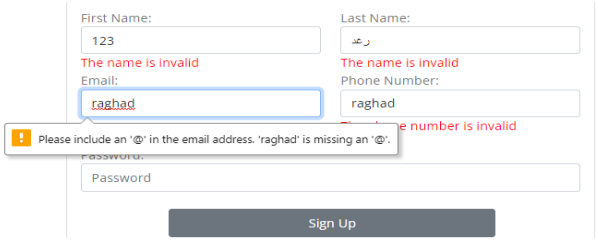


Figure 53: Clint-Side Validation on Sign-Up page

Server-side validation:

The Server-side validation is the system that runs on the server. We applied server-side validation on the sign-up page by using a method called ValidateEmpty in the code behind, which enforces the user to fill in all the blanks by showing a message to the user if he clicks on "Sign Up" before filling in all the blanks.

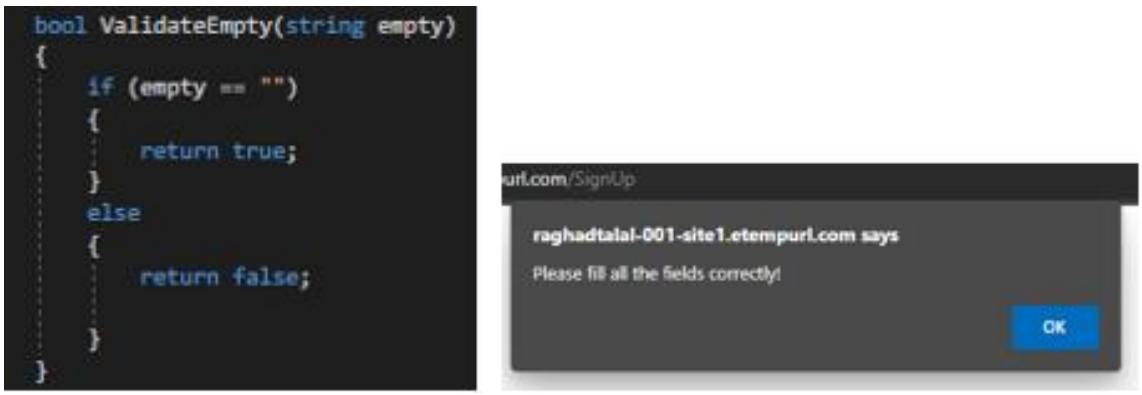


Figure 54: Server-Side Validation on Sign-Up page

- Check Member Exists:

We applied the verification of the account on the sign-up page, and the way it works is as follows: if the customer registers again with the same email that he used to register before, an error message will appear telling him that this account already exists. This feature allows us to avoid duplicating accounts in which the same person is registered.

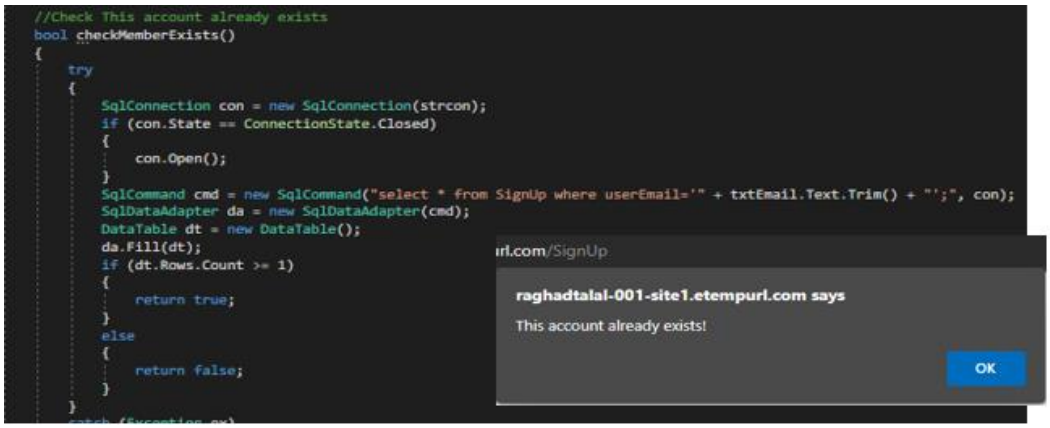


Figure 55: Check Member Exists

CHAPTER 1.5: Additional Tasks

This chapter details the tasks that were assigned to us after building the website, these were more advanced features that help make any program more professional and help us establish a well-designed and more secure website.

1.5.1: Mail Merge:

Mail merge is used to customize a batch of documents for each recipient. For example, sending a form letter where each recipient is addressed with specific information, such as

his\her name, and a data source is linked to the document, such as a list, spreadsheet, or database. There are many resources and ways for applying the mail merge, and through our training, we have learned two of them: one by using Macros on the Microsoft Word app and the other by using Syncfusion on the Visual Studio app.

1.5.1.1: Macros:

Macro was one of the simplest ways to use mail merge because it only required the Word app and an Excel data source; there was no complexity or requirement for a third party. To start using mail merge, we prepared the data source in Excel and filled in the necessary fields, including the paths and names for both Word and PDF documents.

	A	B	C	D	E	F	G
1	**FullName**	**KnownAs**	**Complement**	**DocFolderPath**	**DocFileName**	**PdfFolderPath**	**PdfFileName**
2	Raghad Talal	Raghad	Great	C:\Users\roodi\OneDrive\	end of year letter to Raghad Talal	C:\Users\roodi\One	end of year letter to Raghad Talal
3	Asmaa Hamad	Asmaa	excellent	C:\Users\roodi\OneDrive\	end of year letter to Asmaa Hamad	C:\Users\roodi\One	end of year letter to Asmaa Hamad
4	Manar Omar	Manar	good	C:\Users\roodi\OneDrive\	end of year letter to Manar Omar	C:\Users\roodi\One	end of year letter to Manar Omar
5							
6							
7							
8							

Figure 56: Excel Data Source

After that, we linked this data source to the word template from the Mailings tab and customized it with the desired personalized fields like fullName, knownAs and complement.

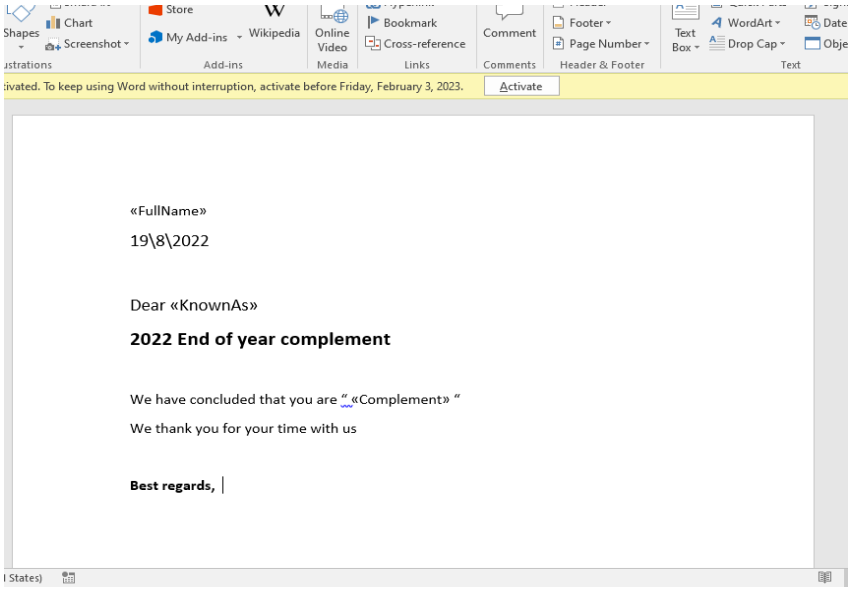


Figure 57: Word template

Now, in order to perform the mail merge, we clicked on the Visual Basic button in the Developer tab to open the Visual Basic editor, where we added our macro.

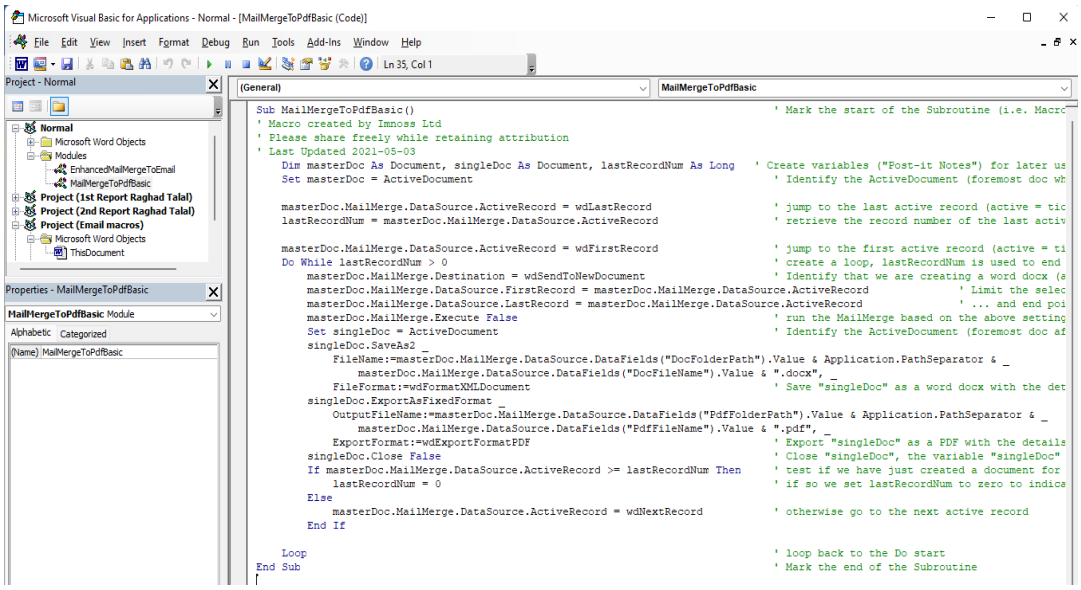


Figure 58: the Macro added to Visual Basic

For the final step to run the mail merge to pdf, we clicked on the Macros button on the developer tab, selected the macro "MailMergeToPdf" and clicked Run.

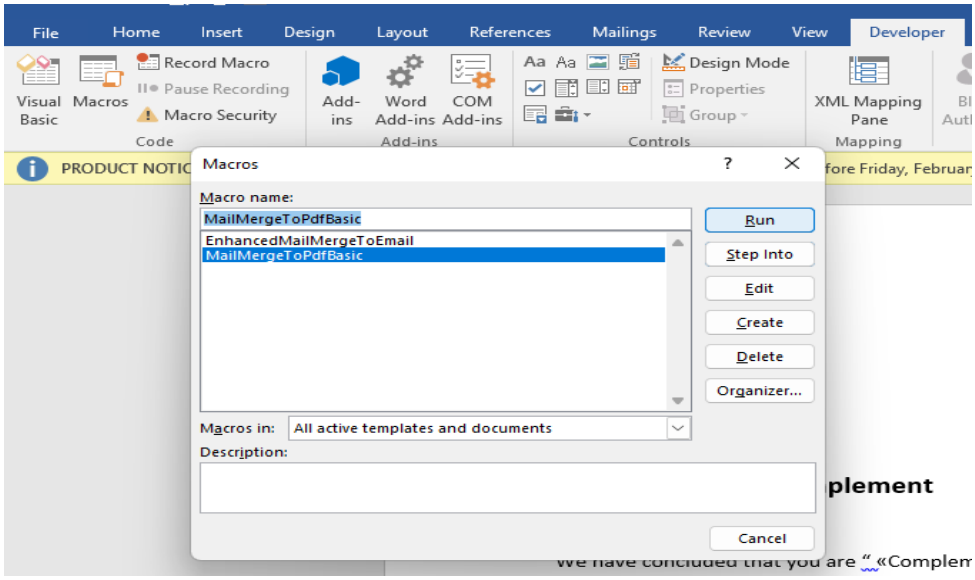


Figure 59: Running the mail merge

And just with these simple steps, we were able to run the macro and save a personalized documents with a specific name and folder path chosen by the user. The figures below show two samples from the above task.

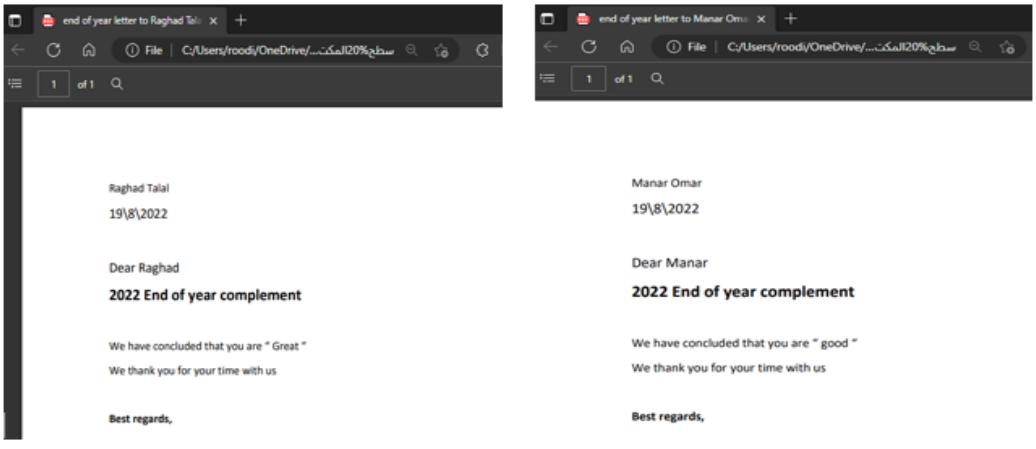


Figure 60: Mail merge output samples

1.5.1.2: Syncfusion:

Syncfusion Essential Studio is a .NET-based software product that provides a variety of components and frameworks for use in Visual Studio to create creative apps. To perform mail merge using it, we were assigned to do a task called "university template" consisting of one page shown below:

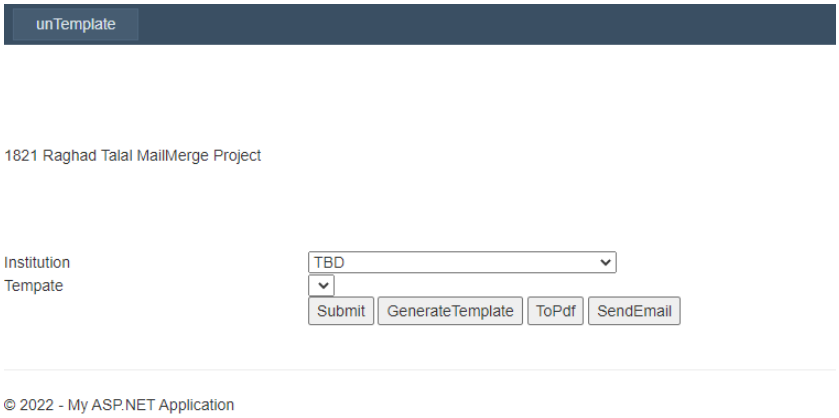


Figure 61: university template project

The project consists of two drop-down lists and four buttons. First drop-down list has the names of institutions; the second drop-down list depends on the choice of the first one. As shown in the figure, it only displays templates related to the institution selected in the first drop-down list:

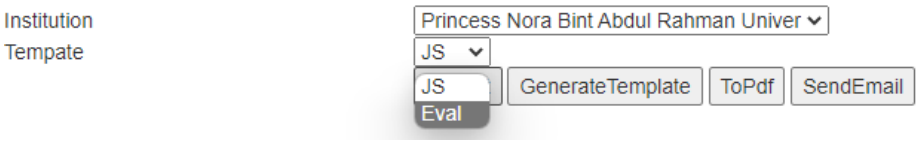


Figure 62: depending drop down list

After selecting the desired institution and template, we use the four buttons. The first button in the project (submit) will show a grid view of interns' data:

Institution Princess Nora Bint Abdul Rahman Univer

Tempate JS

Submit | GenerateTemplate | ToPdf | SendEmail

internId	institutionId	institution	supervisorId	fName	mName	gFName	lName	cell	email	active
1	2	Princess Nora Bint Abdul Rahman University		ali	m	alhussein	hamid	2222	roodi1513@gmail.com	☑
2	2	Princess Nora Bint Abdul Rahman University		ahmed	y	mubarak	alonazi	333	raghad151315@gmail.com	☑

Figure 63: Submit Button

When you click the "Generate Template" button, the data from the selected grid view will be filled into a prepared Word template and saved on the path specified in the code.

```
protected void btnGenerateTemplate_Click(object sender, EventArgs e)
{
    wordT();
}
public void wordT()
{
    for (int i = 0; i <= gvData.Rows.Count - 1; i++)
    {
        String sid = gvData.Rows[i].Cells[0].Text;
        String sname = gvData.Rows[i].Cells[4].Text;
        String uname = gvData.Rows[i].Cells[2].Text;
        String phone = gvData.Rows[i].Cells[8].Text;
        String email = gvData.Rows[i].Cells[9].Text;
        Syncfusion.DocIO.DLS.WordDocument document = new Syncfusion.DocIO.DLS.WordDocument(Server.MapPath("~/mytemplate/testTemplate.dotx"));
        document.MailMerge.RemoveEmptyParagraphs = true;
        string[] fieldNames = new string[] { "sname", "sid", "email", "uname", "phone" };
        string[] fieldValues = new string[] { sname, sid, email, uname, phone };
        // mail merge
        document.MailMerge.Execute(fieldNames, fieldValues);
        //Saves
        document.Save(Server.MapPath("~/myDoc/" + email + ".docx"));
        document.Close();
    }
    // pass message to user notifying of successfull operation
    lblOutput.Text = "Word Doc output generated successfully!";
}
```

Figure 64: Button (Generate Template) Code

Clicking on (ToPdf) button will just convert the Word documents into PDFs and save them on the path assigned for PDFs.

```
protected void ToPdf_Click(object sender, EventArgs e)
{
    pdfT();
}
public void pdfT()
{
    //https://help.syncfusion.com/file-formats/docio/word-to-pdf

    for (int i = 0; i <= gvData.Rows.Count - 1; i++)
    {
        String sname = gvData.Rows[i].Cells[4].Text;
        String email = gvData.Rows[i].Cells[9].Text;
        WordDocument wordDocument = new WordDocument(Server.MapPath("~/myDoc/" + email + ".dotx"), FormatType.Docx);
        wordDocument.ChartToImageConverter = new ChartToImageConverter();
        wordDocument.ChartToImageConverter.ScalingMode = Syncfusion.OfficeChart.ScalingMode.Normal;
        DocToPDFConverter converter = new DocToPDFConverter();
        converter.Settings.EmbedFonts = true;
        PdfDocument pdfDocument = converter.ConvertToPDF(wordDocument);
        pdfDocument.Save(Server.MapPath("~/myPdf/" + email + ".pdf"));
        pdfDocument.Close(true);
        wordDocument.Close();
    }
    // pass message to user notifying of successfull operation
    lblOutput.Text = "Pdf output generated successfully!";
}
```

Figure 65: Button (ToPdf) Code

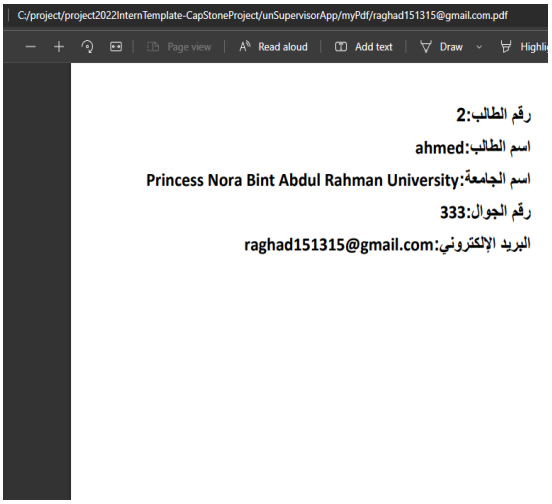
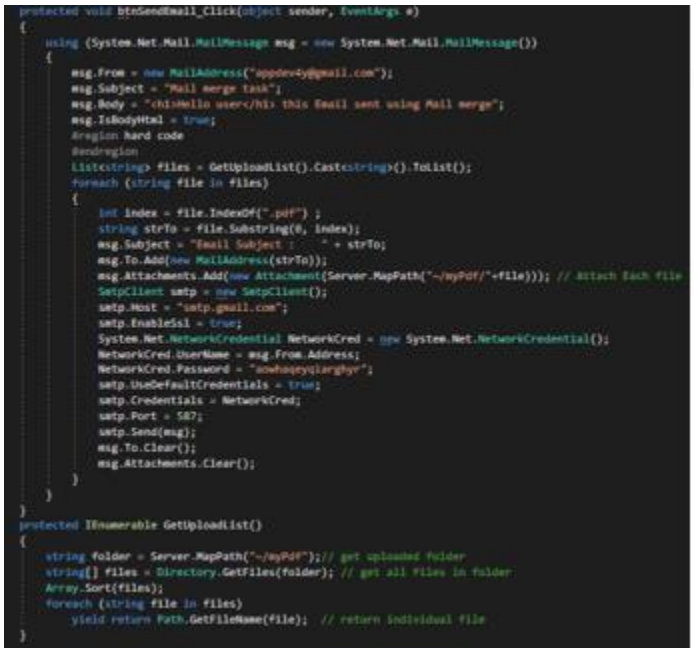


Figure 66: Sample of the generated output

As for the last button (Send Email), clicking on it will automatically send an email with a PDF file for each intern in the database, each with only the attachment containing his\her personal information using the code shown below.



```
protected void btnSendEmail_Click(object sender, EventArgs e)
{
    using (System.Net.Mail.MailMessage msg = new System.Net.Mail.MailMessage())
    {
        msg.From = new MailAddress("appdev4y@gmail.com");
        msg.Subject = "Mail merge task";
        msg.Body = "<h1>Hello user</h1> this Email sent using Mail merge";
        msg.IsBodyHtml = true;
        #region hard code
        #endregion
        List<string> files = GetUploadList().Cast<string>().ToList();
        foreach (string file in files)
        {
            int index = file.IndexOf(".pdf") ;
            string strTo = file.Substring(0, index);
            msg.Subject = "Email Subject :    " + strTo;
            msg.To.Add(new MailAddress(strTo));
            msg.Attachments.Add(new Attachment(Server.MapPath("~/myPdf/"+file))); // Attach Each file
            SmtpClient smtp = new SmtpClient();
            smtp.Host = "smtp.gmail.com";
            smtp.EnableSsl = true;
            System.Net.NetworkCredential NetworkCred = new System.Net.NetworkCredential();
            NetworkCred.UserName = msg.From.Address;
            NetworkCred.Password = "aowhaqeyqlargbyr";
            smtp.UseDefaultCredentials = true;
            smtp.Credentials = NetworkCred;
            smtp.Port = 587;
            smtp.Send(msg);
            msg.To.Clear();
            msg.Attachments.Clear();
        }
    }
}
protected IEnumerable GetUploadList()
{
    string folder = Server.MapPath("~/myPdf");// get uploaded folder
    string[] files = Directory.GetFiles(folder); // get all files in folder
    Array.Sort(files);
    foreach (string file in files)
        yield return Path.GetFileName(file);  // return individual file
}
```

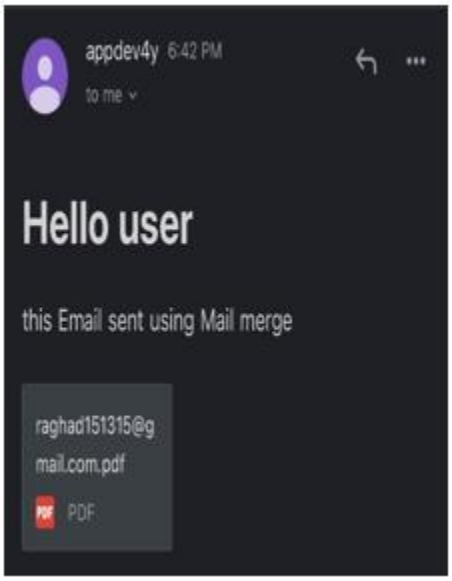


Figure 67: button (Send Email)

1.5.2: Overview of Git:

Losing files, codes, or projects can cause a lot of trouble, especially for companies with large databases, where a simple mistake can lead to massive technical issues. So we took an overview of Git and learned how to use it to create an identical version of our projects and files to retrieve them whenever needed. We were given a task to initialize and save a folder using it. First of all, we create a new user and user email using this code line:

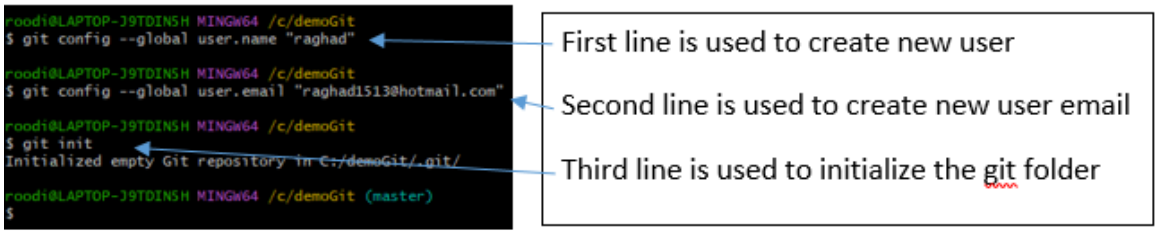


Figure 68: Git initialize

Then we used the (git status) command to show the status of the files, as we can see in the figure, the name of the file is written in red because we have not added the file from the working area to the staging area yet.

```
roodi@LAPTOP-J9TDIN5H MINGW64 /c/demoGit (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        raghad git demo.txt

nothing added to commit but untracked files present (use "git add" to track)
```

Figure 69: (git status) command

In order to add it to the staging area, we used the (git add.) command.

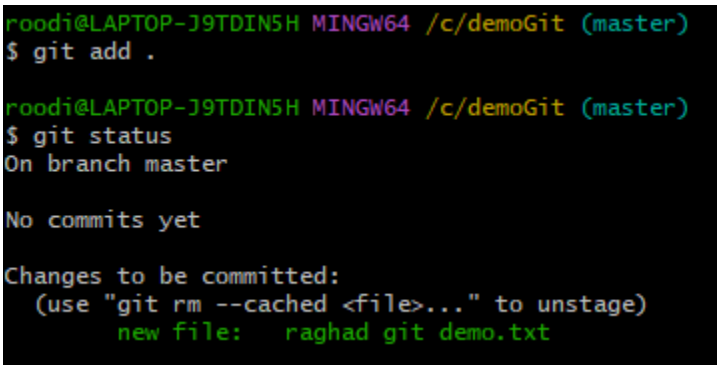


Figure 70: (git add.) command

There are other commands used in git, such as

- (git commit --m"comment") which is used to save all of the work from staging to the repository (DB) final destination for the files.
- (git log --oneline) displays all command history.
- (git checkout code + file name) which is used to restore deleted files from the repository to the working area.

CHAPTER 1.6: CASE STUDIES

This chapter describes a few case studies I faced during my training at KFMC and their solutions.

1.6.1: Case Study 1: Incorrect Register Entries:

1.6.1.1: Introduction:

On the sign-up page, there are three textboxes for the first name, second name, and phone number.

1.6.1.2: Problem statement:

We have encountered some issues because interns are required to provide accurate information because it is a formal registration form. Some have substituted numbers or symbols for their names or substituted letters for the phone number.

1.6.1.3: Solution:

To solve this issue, we must only allow the end user to provide string values in the name text boxes and digital values in the phone number text box, so we added a control in Microsoft Visual Studio called a Regular ExpressionValidator. First, we identified which textboxes need validation, and then from the toolbox, we dragged and dropped the Regular Expression Validator Control, in its proprieties sets your text box ID at the "control to validate" field, and at the "validation expression" field, we must write either:

1- "^[A-Za-z]*$" for the names, which means accepting capital and small letters from A to Z.

2- "\d+" for the phone number, which means accepting digital numbers only

At "Error Message" write the message for the end-user when he but wrong information.

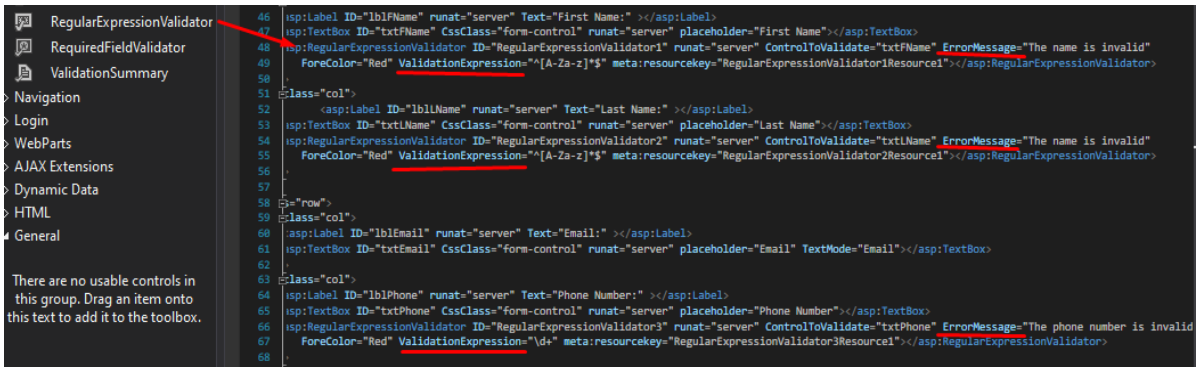


Figure 71: Regular Expression Validator

1.6.2: Case Study 2: register with empty information:

1.6.2.1: Introduction:

In both the sign-up and volunteering forms, the end-user must provide extensive information.

1.6.2.2: Problem statement:

Some users click on the submit buttons by mistake before completing all the required information or forget to fill in one of the text boxes.

1.6.2.3: Solution:

In order to forbid this issue from happening, we used a method called ValidateEmpty in the code behind, which prompts the user to complete all of the blanks by displaying a message if the user presses the submit buttons before doing so.

Figure 72: ValidateEmpty

1.6.3: Case Study 3: user role after signing up

1.6.3.1: Introduction:

The new users of the website need to sign up in order to be able to fill out the volunteering form.

1.6.3.2: Problem statement:

We have noticed that after signing up successfully, the user still needs to log-in in order to change his role to "sign up" and the "Volunteer" menu to show on the home page.

1.6.3.3: Solution:

We solved this problem by adding this code in the code behind of the "sign up" page so that after signing up successfully, it will automatically set the session role to the "sign up" role as well as redirect the user to the home page.

```
{
    Session["role"] = "SignUp";
    ScriptManager.RegisterStartupScript(this, this.GetType(), "alert",
    "alert('Thank you for Signing Up with Peace For Charity family');window.location ='Home.aspx';", true);

}
```

Figure 73: setting the user role

CHAPTER 2: CONCLUSIONS AND RECOMMENDATIONS

In this chapter, I draw conclusions from my training as a COOP trainee. Additionally, it provides recommendations for improving future COOP trainees' training experiences.

2.1: CONCLUSIONS:

The co-op training gave me the chance to enhance my university experience by connecting classroom learning to actual practical situations, I also was able to evaluate my professional interests, which gave me the opportunity to gain new skills. I was able to practice my problem-solving and decision-making skills at that time, and I was also able to relate the theoretical material I had learned in college with the practical aspects of real-world

applications. I have also gained patience during the training session by handling all the new tasks and information and completing the tasks in a timely manner. Training at King Fahad Medical City gave me the possibility to create my website and learn about the other side of the web application.

In addition to the technical knowledge, I have been able to improve my interpersonal and communication skills, also Determine where my strength is, as well as work to improve my weakness.

2.2: RECOMMENDATIONS:

Based on my experience with the coop program, I would like to provide the following advice to future computer science and engineering students who will participate in the co-op program:

- Make a good choice of company.
- When you run across an issue, don't be shy to ask questions.
- If you have a problem, be sure you fully comprehend it before seeking a solution.
- Write reports day by day to ensure you don't forget any details.
- I advise every CSE student who could be concerned about programming-related subjects to train at KFMC because they offer a complete learning environment for it.

References

bootstrap. (n.d.). Retrieved from https://getbootstrap.com/

City, K. F. (n.d.). Retrieved from https://www.kfmc.med.sa/

department, W. D. (n.d.). *Internship Training Program*. Retrieved from Application Practical Training: http://kfmc.wdbcs.com/index

Media, O. (n.d.). Retrieved from https://www.oreilly.com/library/view/web-database-applications/0596005431/ch01.html

microsoft. (n.d.). *ASP.NET*. Retrieved from https://dotnet.microsoft.com/en-us/apps/aspnet

Microsoft® Visual Studio 2017. (n.d.). Retrieved from https://visualstudio.microsoft.com/vs/older-downloads/

SmarterASP.NET. (n.d.). Retrieved from https://www.smarterasp.net/

W3Schools. (1999-2021). *web developer site*. Retrieved from https://www.w3schools.com/default.asp

- Visual Studio Download link
- Microsoft SQL Server Download link
- Peace For Charity website

APPENDIX A –

Definitions

1- **SQL**: Stands for Structured Query Language, a language for manipulating and talking about data in databases.

2- **HTML:** Hypertext markup language, a formatting system for displaying material retrieved over the Internet. HTML markup tags specify document elements such as headings, paragraphs, and tables.

3- **IDE:** Integrated Development Environment, developed by Microsoft to develop GUI.

4- **GUI:** Graphical User Interface.

APPENDIX B –

Differences between View and Stored Procedure

View	Stored Procedure
Does not accepts parameters	Accept parameters
Can be used as a building block in large query	Can not be used as a building block in large queries
Can contain only one single Select query.	Can contain several statements like if, else, loop, etc.
Can not perform modification to any table.	Can perform modification to one or several tables.
Can be used (sometimes) as the target for Insert, update, and delete queries.	Can not be used as the target for Insert, update, and delete queries.

APPENDIX C –

COOP Training Certificates

مدينة الملك فهد الطبية
King Fahad Medical City

تجمع الرياض الصحي الثاني
Riyadh Second Health Cluster

Certificate of Training

Administrative & Technical Training Department at
Executive Administration of Human Capital

awarded this certificate to:

Name of Trainee

Raghad Talal Inad

ID Number: [redacted]

Upon completion of cooperative training at

King Fahad Medical City

From: 19/06/2022

To: 01/12/2022

شهـــادة تدريب

يمنح قسم التدريب الإداري والفني

بالإدارة التنفيذية لرأس المال البشري هذه الشهادة لـ:

اسم المتدرب

رغد طلال عناد

رقم الهوية: [redacted]

إتمام التدريب التعاوني بمدينة الملك فهد الطبية

من: 19/06/2022

الى: 01/12/2022

Mr. Hammad AlRashidi
Executive Director,
Executive Administration of Human Capital

King Fahad Medical City kfmc_riyadh Kfmc Riyadh

مدينة الملك فهد الطبية
King Fahad Medical City

تجمع الرياض الصحي الثاني
Riyadh Second Health Cluster

Certificate of Attendance

Learning Institute
at Executive Administration of Academic & Training Affairs
awarded this certificate to:

Raghad Talal Halael Inad

Upon Completion of Training Course:

(General Intern Orientation)

At King Fahd Medical City

From:23\06\2022

To:23\06\2022

شهـــادة حضور

يمنح مركز التعلم

بالإدارة التنفيذية للشئون الأكاديمية والتدريب هذه الشهادة لـ:

رغد طلال هليل عناد

لإتمام الدورة التدريبية :

(البرنامج التعريفي للمتدربين)

بمدينة الملك فهد الطبية

من: 1443\11\24

الى: 1443\11\24

Ms. Afnan M. Al-Wohaibi
Chairperson, Administrative and Technical
Training Department

Dr. Adnan D. AlWadie
Associate Executive Director of Learning
Institute in the Riyadh Second Health Cluster

VITAE

Name: Raghad Talal Halael Inad.

Major: Bachelor of Computer Science and Engineering (CSE).

University: University of Hafr Al-Batin (UHB).

Email: raghad.inadd@gmail.com

Phone: 0[redacted]