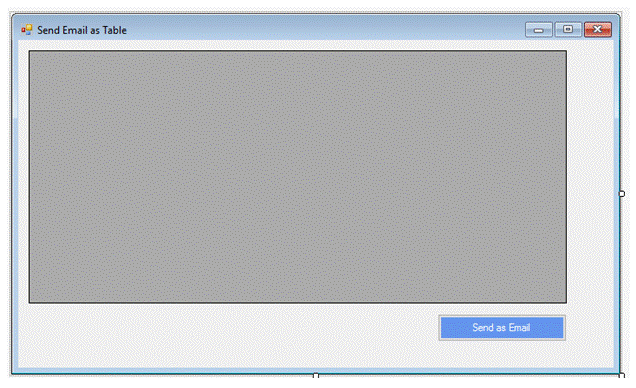
**Unit-Tests are being created for each piece of code**

When it comes to sending emails, Windows or web-based software has become the standard. Users send emails for a variety of reasons, the most common of which are to communicate, report problems, provide updates, collect user data and analytics, and send attachments. It is possible to export the data from a control and transfer it to a different person over the internet.

I'll teach you how to send a table of data from a GridView using the C# programming language in this tutorial. Desktop and web-based applications may both make use of this code. I will construct an application for our use case by writing it in the Windows Forms programming language.

Step 1. Create a Client

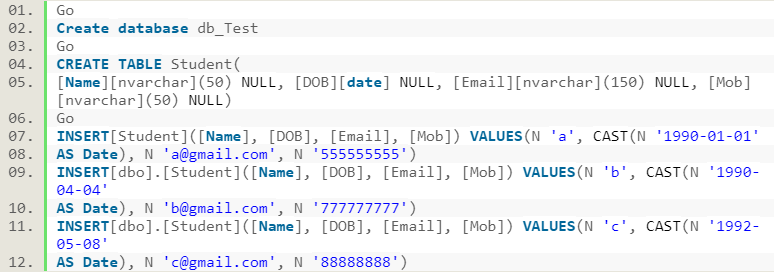
Using the C# template in Visual Studio, a Windows Forms application may be created. Give the GridView control a name that represents its purpose, such as Send Email. Adding the Button control, as seen below, is another option..



Step 2. Email Database

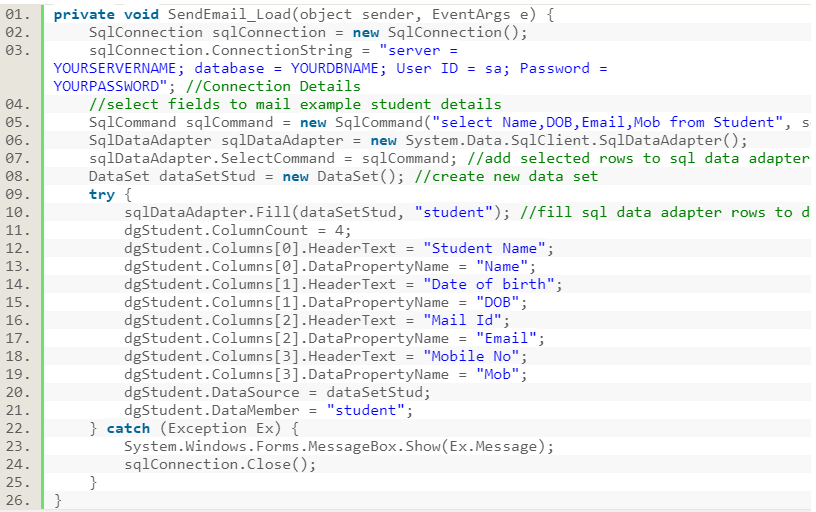
With SQL Server, we will be able to keep track of all of our email correspondence.

Using SQL Server, create a table and then begin populating it with data. In the next part, you will find the script. Student data is stored in the db Test database by use of the script below. These records include every piece of contact information we have on the students, including their first and last names, birth dates, and email addresses.



Step 3. Data should be loaded from the database.

The GridView control may be loaded with student data by choosing and obtaining it. Loading an event handler: Your Form should now include the following code. To use this code with a new database, you must change the connection string. Receiving and showing data is the job of the GridView control.



Step 4. HTML email templates are currently being created.

How to develop a function that receives GridView data and outputs an HTML table as output is shown in the following example HTML is a widely established standard for structuring and delivering email messages in order to make them easier to read for the recipient.

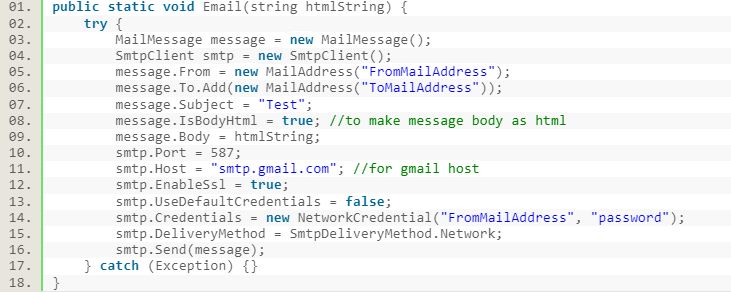
With the help of this code, you can create an HTML email template, which will be used to display your email.



Step 5To send an email; make use of SMTP (Simple Mail Transfer Protocol).

We can now send an email using this HTML format string. Before developing this function, be sure to include the following two namespaces.

The email method is provided in the section below. HTML is accepted as a string by the Email function in the following code. All of the attributes of the new message, including the From and To addresses, are established. When SmtpClien is created, the parameters for the port and host are pre-specified. Send is the last step in the process.

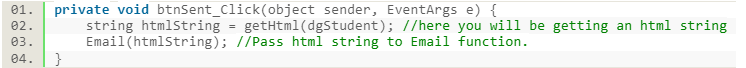


Step 6. Sending and Receiving Emails For this method to work, you must provide it a string (the email's subject line). This function only takes a Gmail host as an argument. If your "from address" is different from Gmail, you must update the server port number, host name, and SSL attribute. Here are a few samples of server names that you may like.



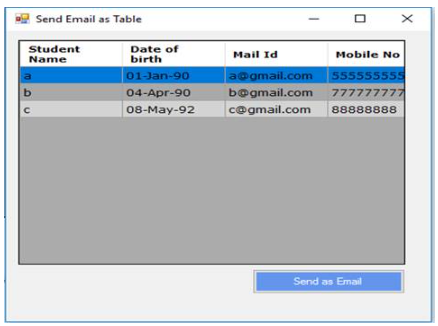
Step 7. Send email

In the event that a button is pushed, these strategies may be used to communicate GridView data to a recipient through email.



Step 8. Build and run

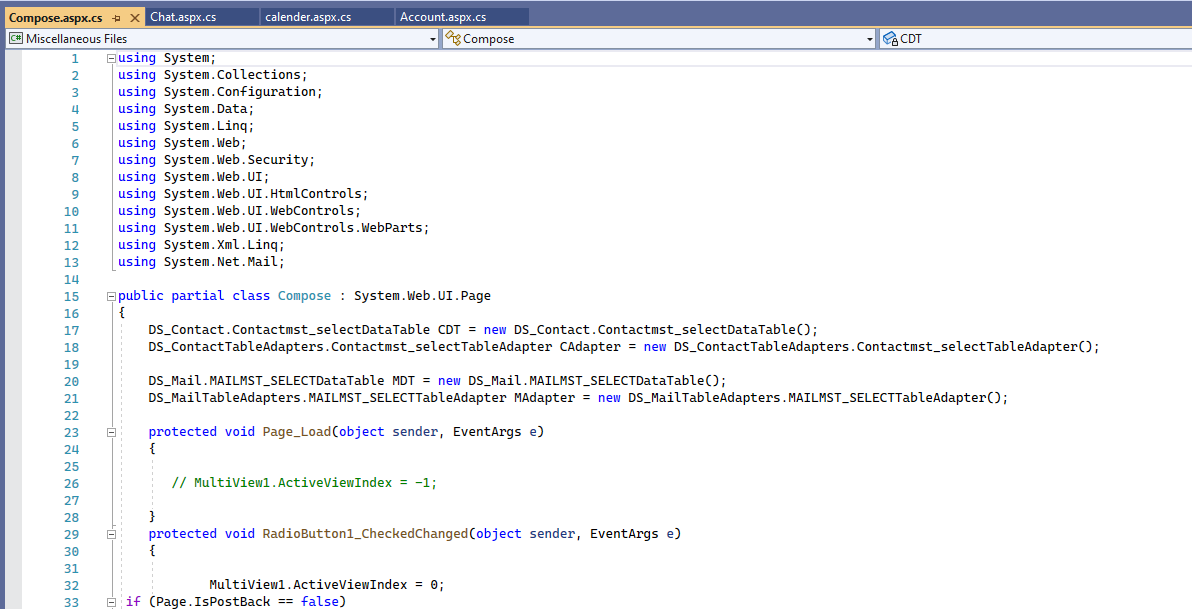
To send an email, start up the software and then click "send an email." Make sure that the "to address" you enter will get an email, as seen in the result below.



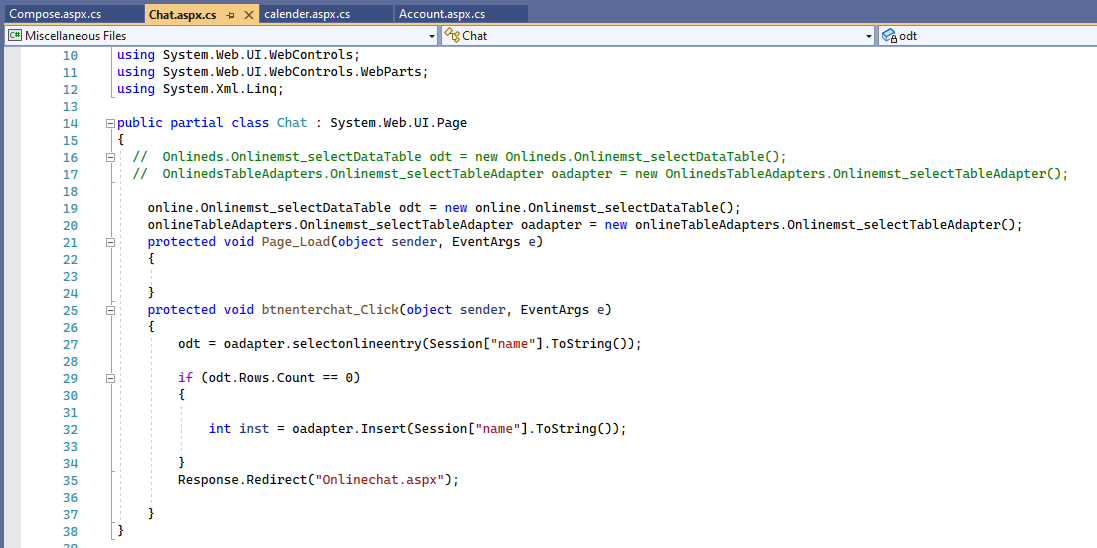
**A library project is present and classes are being created**

The different classes have been created for the email management system. Like Account, calendar, chat, and compose mail. Below some screenshot has been put.

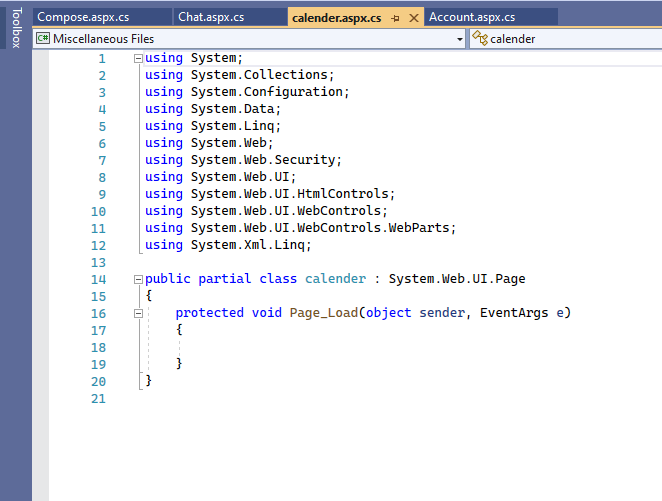
**Compose Email:**



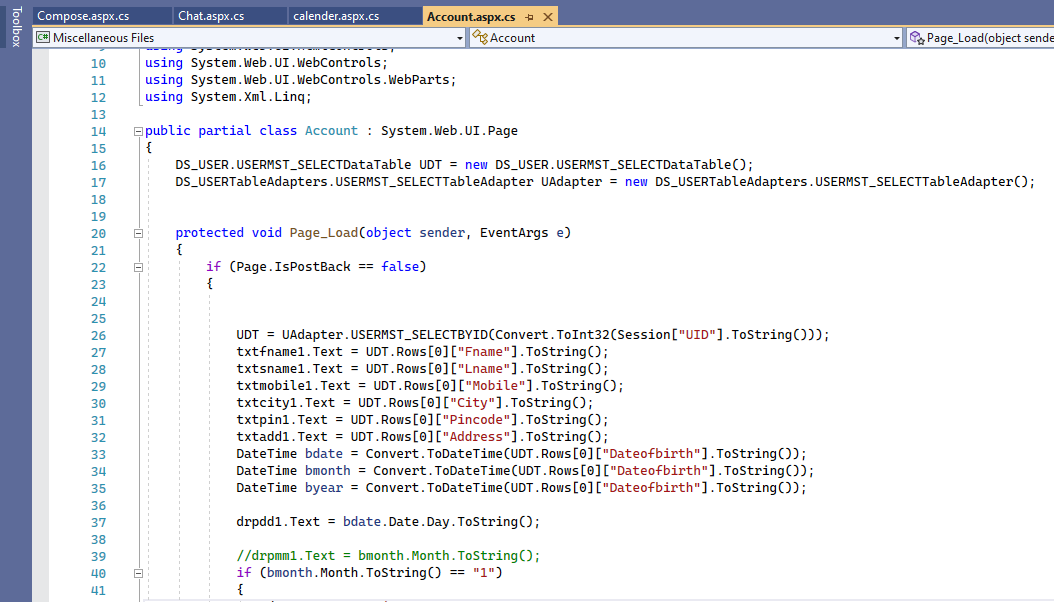
**Chat:**



**Calendar**



**Accounts:**

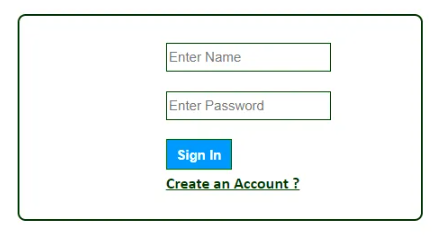


**Ui is being created and refined**

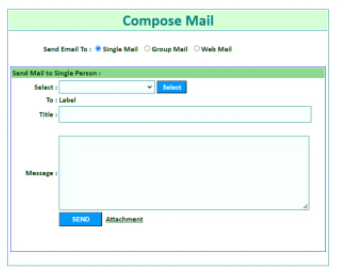
**Account Create**



**Password Set:**



**Compose Email:**



Working on other GUI as well.

**Data Attributes**

* Login
* Drafts
* Outbox
* Themes
* Trash

**SQL Queries**

**Integration Tests**

As we will design the project we will do the integration tests for created pages. The integration test for the login page would be.

* Ensure that users are sent to the homepage after registering and entering their password in order to offer a smooth transition between the login page and the home page.
* Before a user may read the profile of another, the user's homepage and profile page must be connected together via an interface.
* To see an invitation on your connection page, just click on the "welcome" button on the system's Invitation page (found on the Invitation page)
* You should see a new notification window appear when you click on the UI connection between the Notifications and Say Congratulations pages, which is the Say Congratulations button.