**Project Proposal & Report**

DEVELOPMENT OF ELECTRONIC TICKET BOOKING APPLICATION FOR RUGBY MATCHES USING ASP.NET AND WEATHER API- WEBSERVICE



|  |  |
| --- | --- |
| **Student Name** | SWAPNADEEP DAS |
| **Student Number** | 10364254 |
| **Course Tittle** | MSc |
| **Course Stream** | Information Systems with Computing |
| **Moodle Name** | Enterprise Information Systems |
| **Moodle Code** | B9IS104 |
| **Submitted to** | Paul Laird |
| **No. of words** | 999 words excluding Table of Contents, References, Appendix and Ethical Information |

[MSC IN INFORMATION SYSTEMS WITH COMPUTING]

[SWAPNADEEP DAS]

Dublin Business School

**Table of Contents**

[1. 1 BACKGROUND OF THE PROJECT 2](#_Toc13582748)

[1. 2 THE ROLE OF WEB SERVICES IN THIS PROJECT 3](#_Toc13582749)

[1.3 THE ROLE OF SOURCE CONTROL MANAGEMENT IN THIS PROJECT 3](#_Toc13582750)

[2.1 TECHNOLOGIES ARE GOING TO IMPLEMENT 4](#_Toc13582751)

[2.2 ASP.NET 4](#_Toc13582752)

[2.3 GITHUB 4](#_Toc13582753)

[3.1 ARTIFACT DESIGN OF THE PROJECT 5](#_Toc13582754)

[4.1 REFERENCES 6](#_Toc13582755)

**List of Tables and Figures**

[Figure 1: Artefact Design of The Project 6](#_Toc3558906)

# 1. 1 BACKGROUND OF THE PROJECT

It has been surveyed that most of the projects have been conducted on cricket, football and popular sports in Europe. **The project is related to application development of rugby match ticket booking** because rugby has been the source of games spirit and been very popular in Ireland. So, it is a good idea to make a project on rugby match ticket booking application.

**The focus of the project is to allow users to book tickets beforehand of the rugby matches going to happen in the next month or year.** In this project number of rugby matches will be shown to the users in which they can select the matches they want to book their tickets for and then they can proceed to the next step. In the third step, users can make the payment directly to the stadium by using their debit cards, credit cards or cash.

# 1. 2 THE ROLE OF WEB SERVICES IN THIS PROJECT

The Web service will be used in this project is Weather API. This API will show the climate during matches (*Home - Weather Unlocked*, no date).

# 1.3 THE ROLE OF SOURCE CONTROL MANAGEMENT IN THIS PROJECT

**The software for source control management will be used in this project is Git.** It is also known as a revision control system (*GIT Fast Version Control*, no date). **GitHub will act as a hosting service for Git repositories** (*Githubflow Online*, no date)**.** Every day when the project code is finished the code has been uploaded to GitHub and if any other person has to acknowledge the code, he or she can directly look into the public GitHub repository and in this way daily code has been uploaded to the GitHub public repository and the success of the project can be tracked by looking at the GitHub code. One other way GitHub is useful as when any other person has to contribute to the project then the source control management will be helped in this process and code will be updated after the help.

# 2.1 TECHNOLOGIES ARE GOING TO IMPLEMENT

Two technologies will be used in this project. The first technology will be used to make front end and backend of the application and the other technology will be used to update the daily code.

# 2.2 ASP.NET

The frontend will be made using languages such as HTML, CSS. In the backend of the project, we will be using databases like MS Access as well as the programming language C#. This will be made using a database and all the tickets information and stadium information will be stored in the database (*asp.net\_tutorial.pdf*, no date).

# 2.3 GITHUB

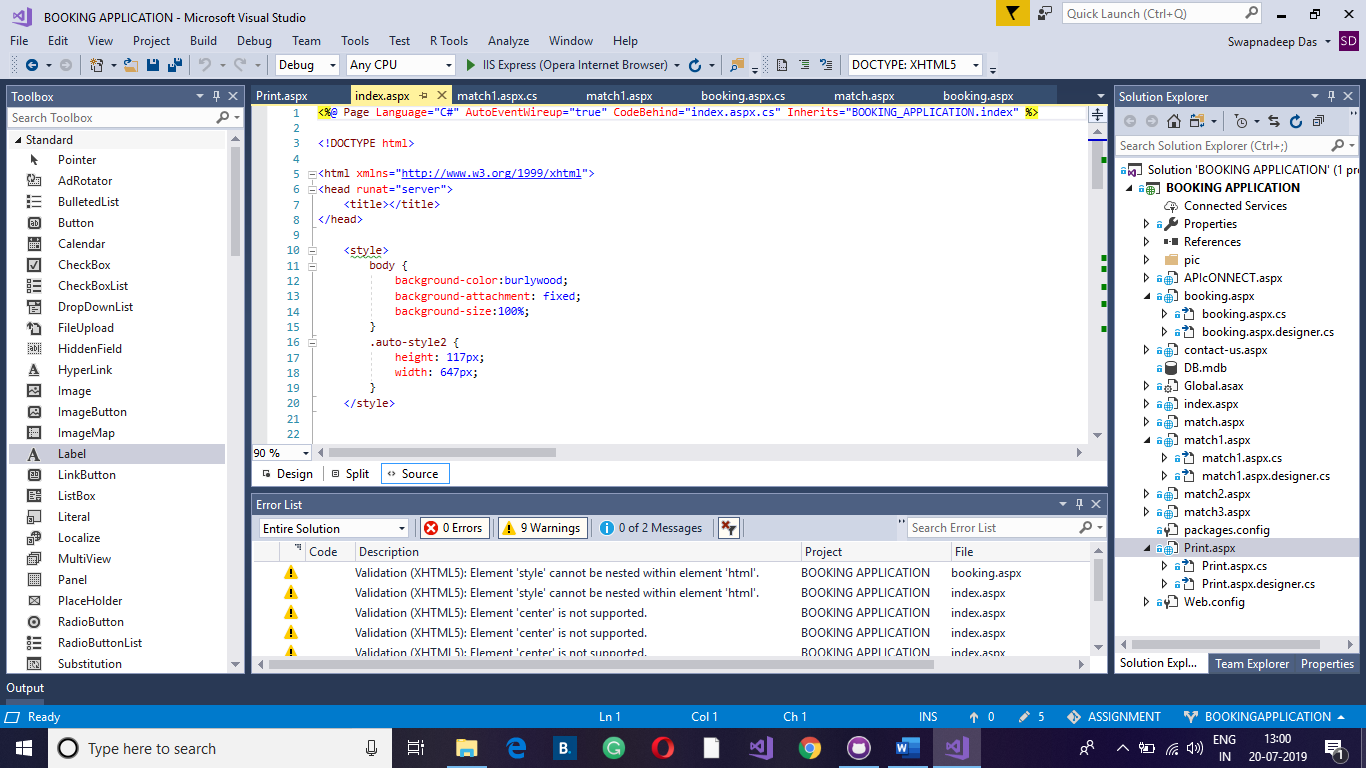
The second technology will be used in this application for version source control management is GitHub. It will also be used to track day to day code revisions and if there is some change in the code it will be very useful to restore the previous versions of the code using GitHub version source control management (Li, no date, p. 9).

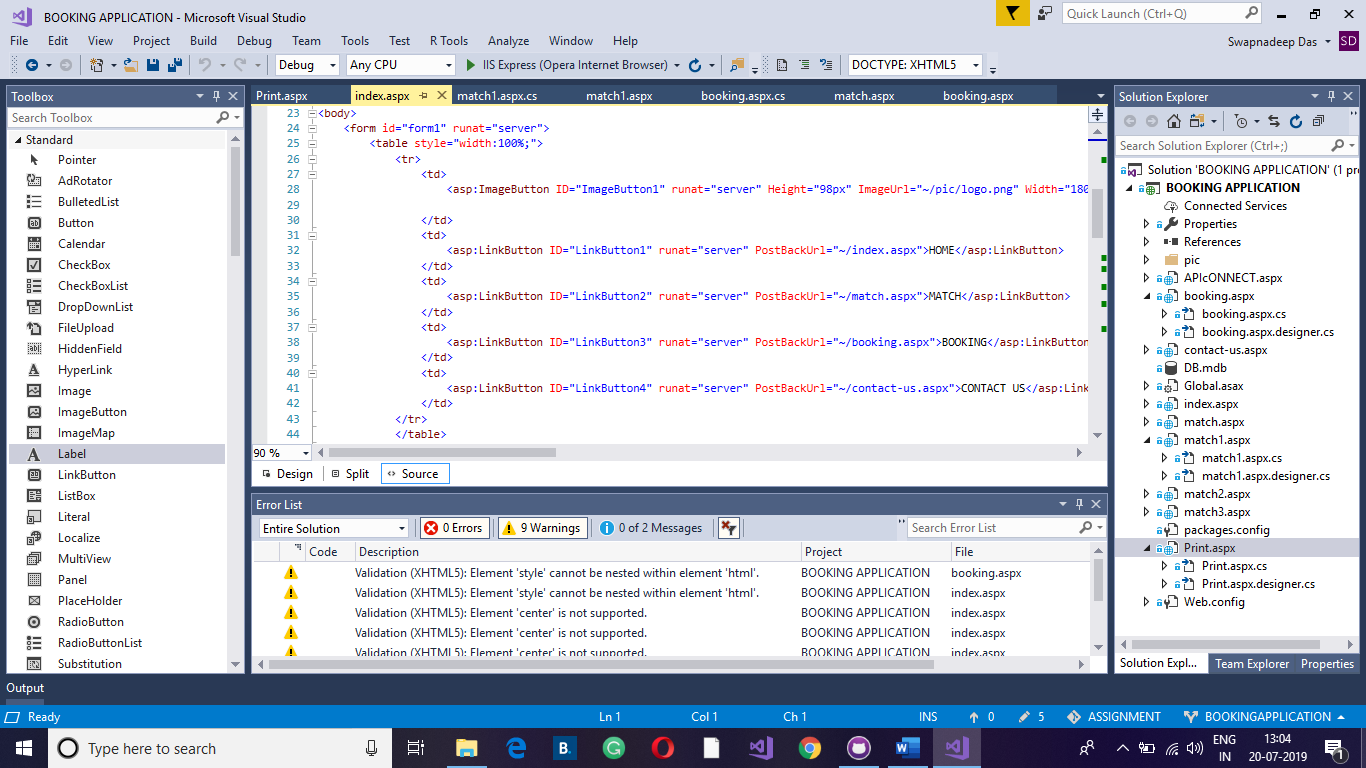
# 3.1 ARTIFACT DESIGN OF THE PROJECT

Figure 1: Artefact Design of The Project

# 3.2 CODES SCREENSHOTS AND EXPLANATION:

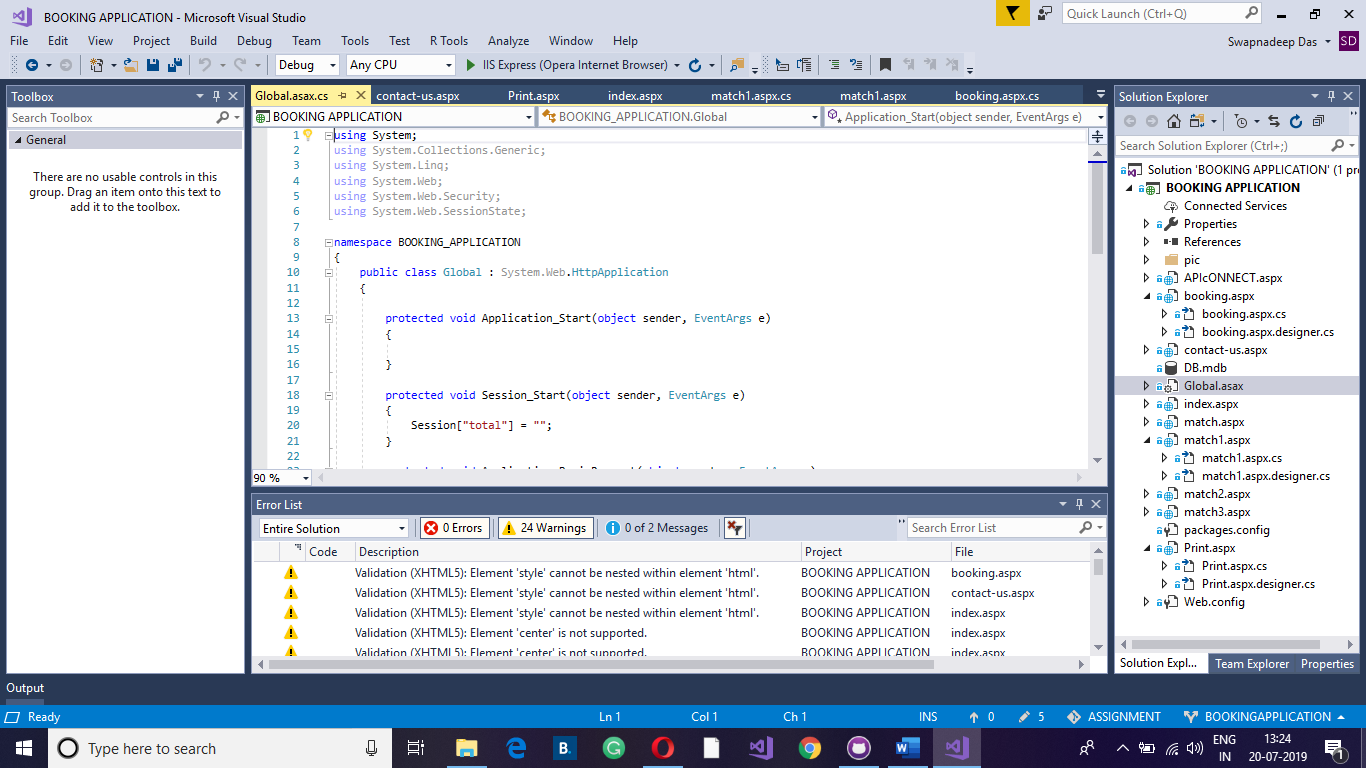
**Screenshot 1:**

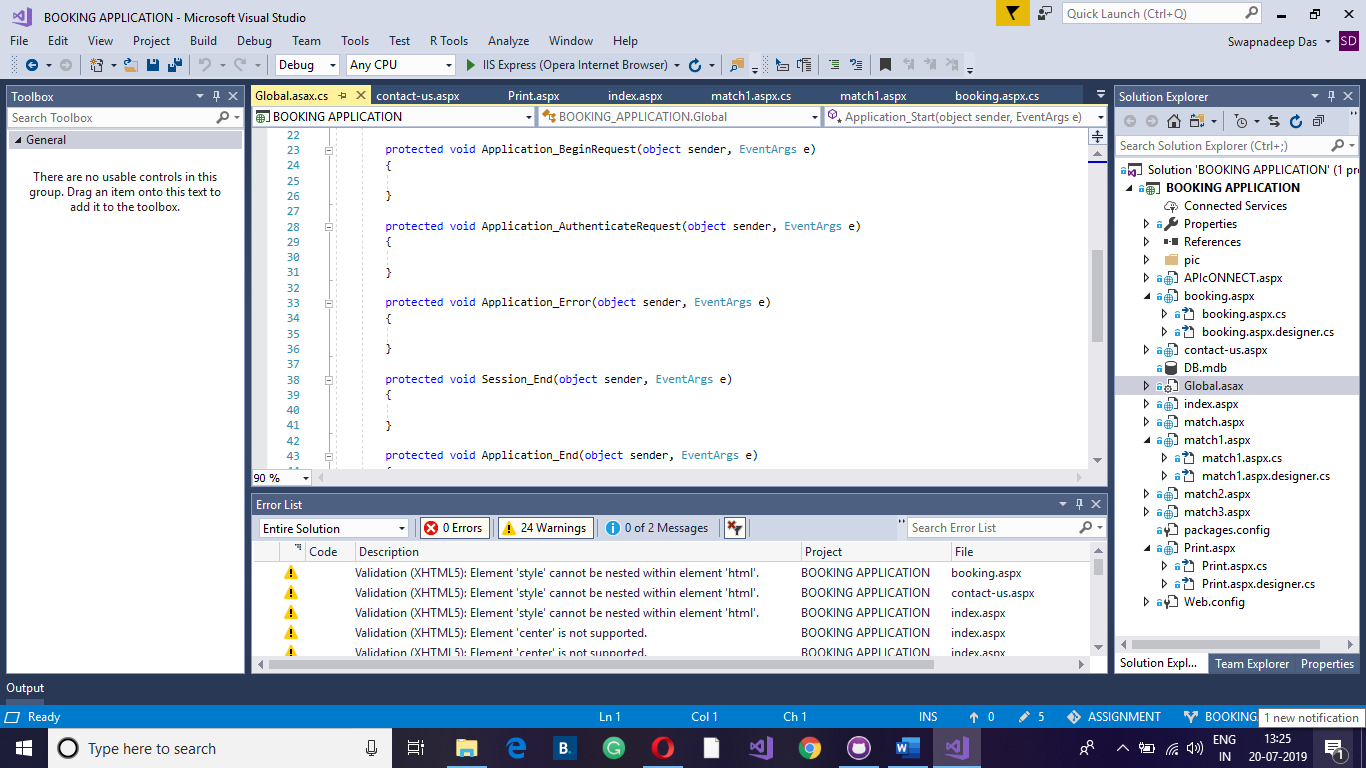




This is the concept of Table, Tr and Td. Tr means table row and Td means table column. The above codes are basic HTML codes. The same concept has been applied for all the pages.

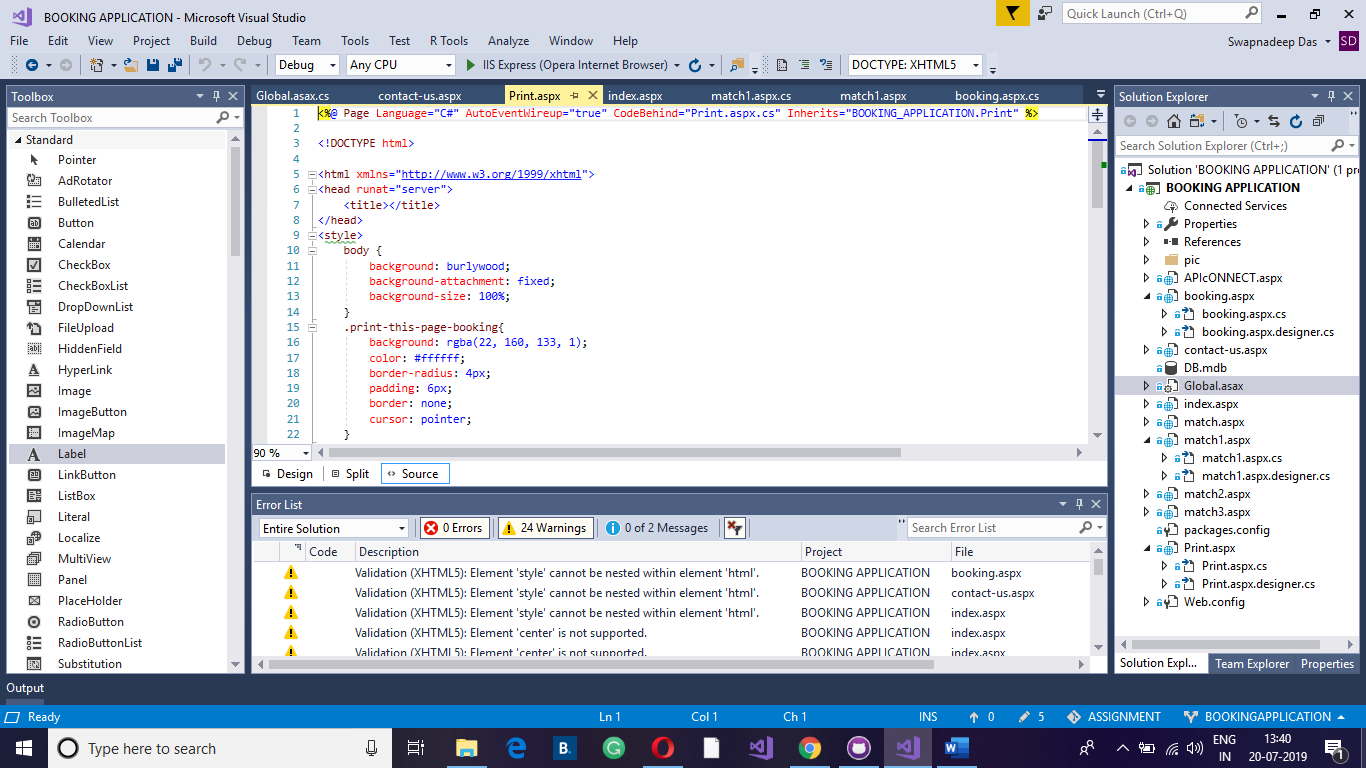
**Screenshot 2:**

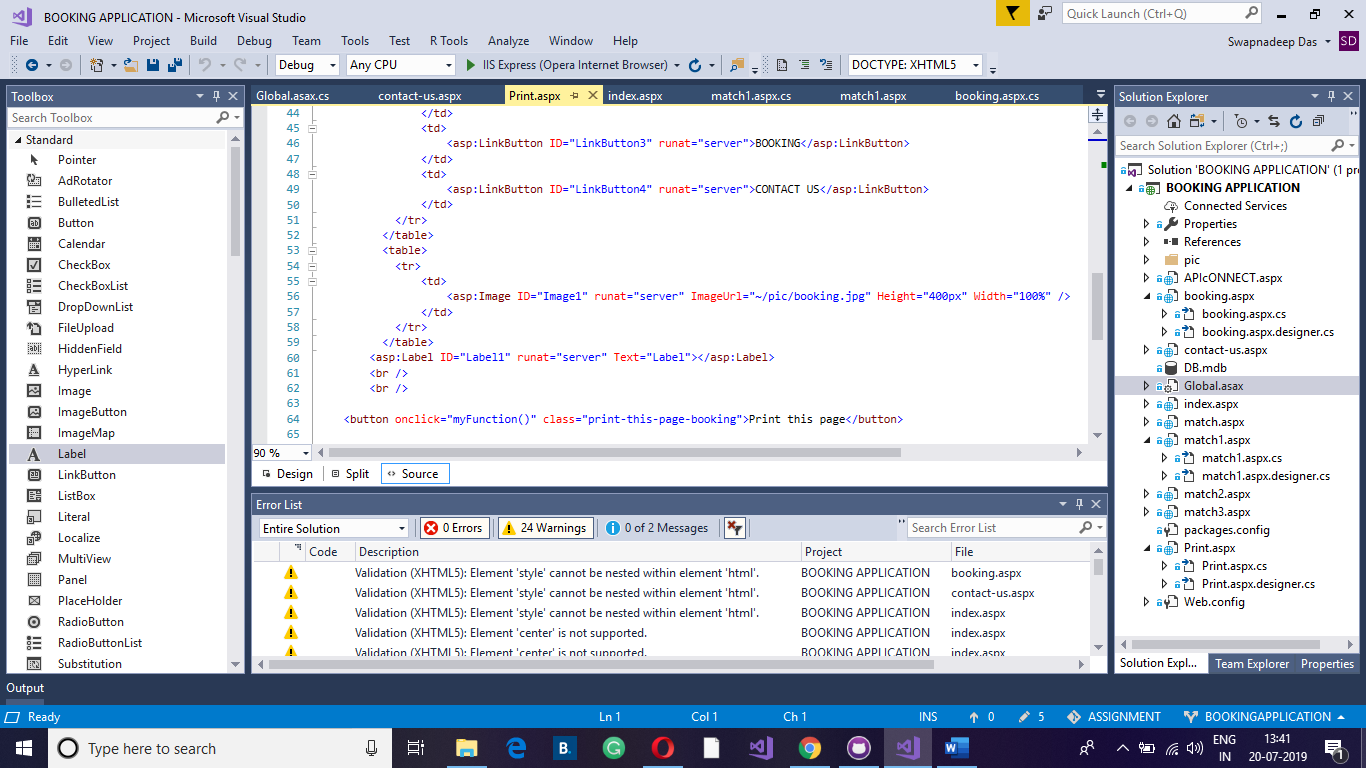




Global.asax is the unique file in the entire project. This is called as session variable. When the app is opened then it some functions start working examples include, Application\_Start, Session\_Start etc... The session variable will work how long the application is running on the browser. The session variable will stop working when the user will close the browser. All the booking page data is stored in session variable.

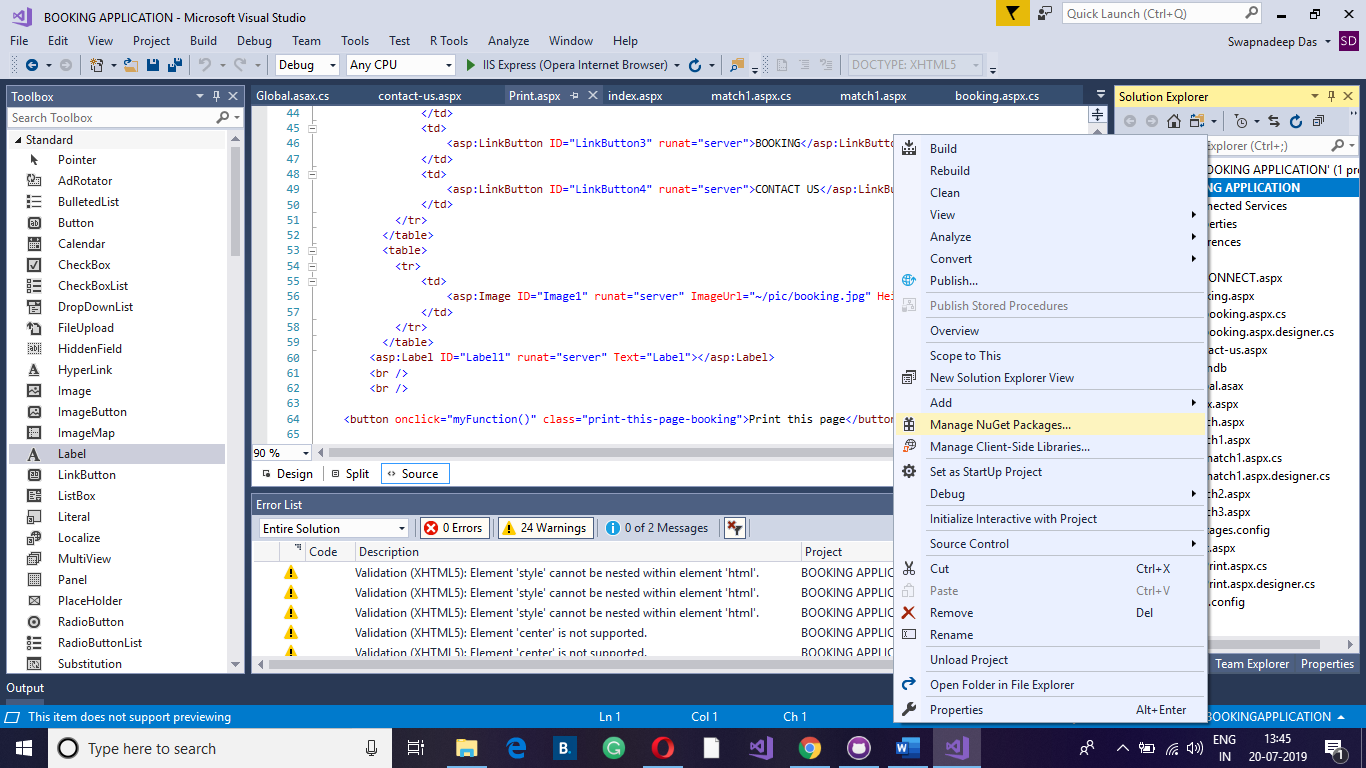
**Screenshot 3:**

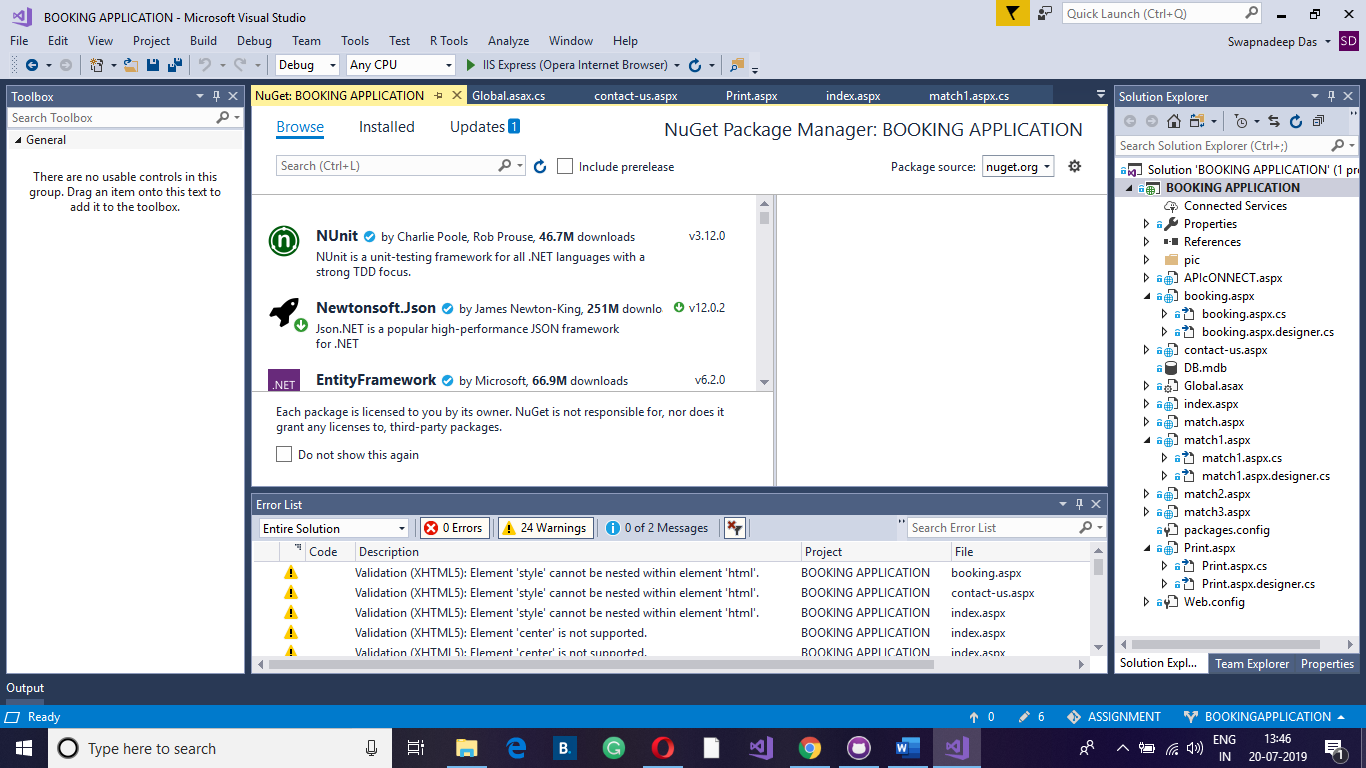




All the data of variable is printed in Print.aspx page.

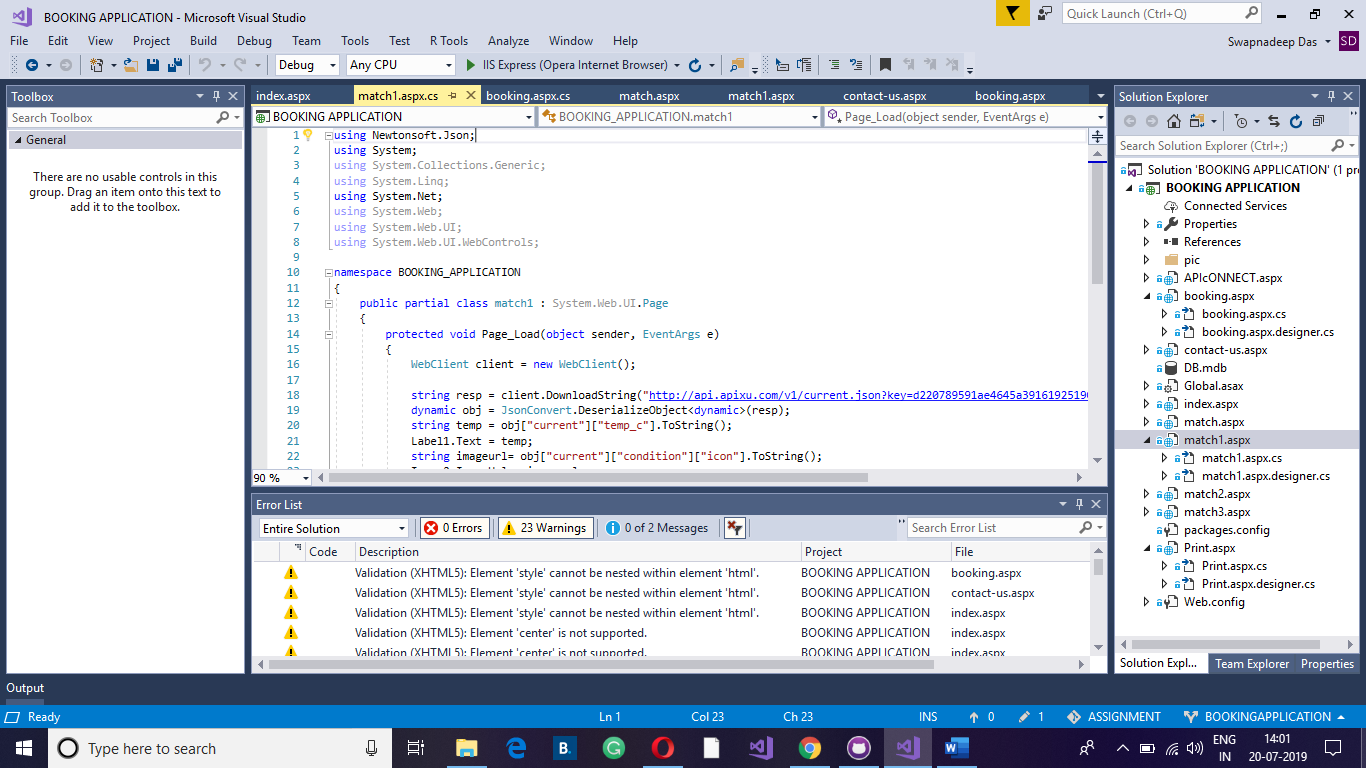
**Screenshot 4:**

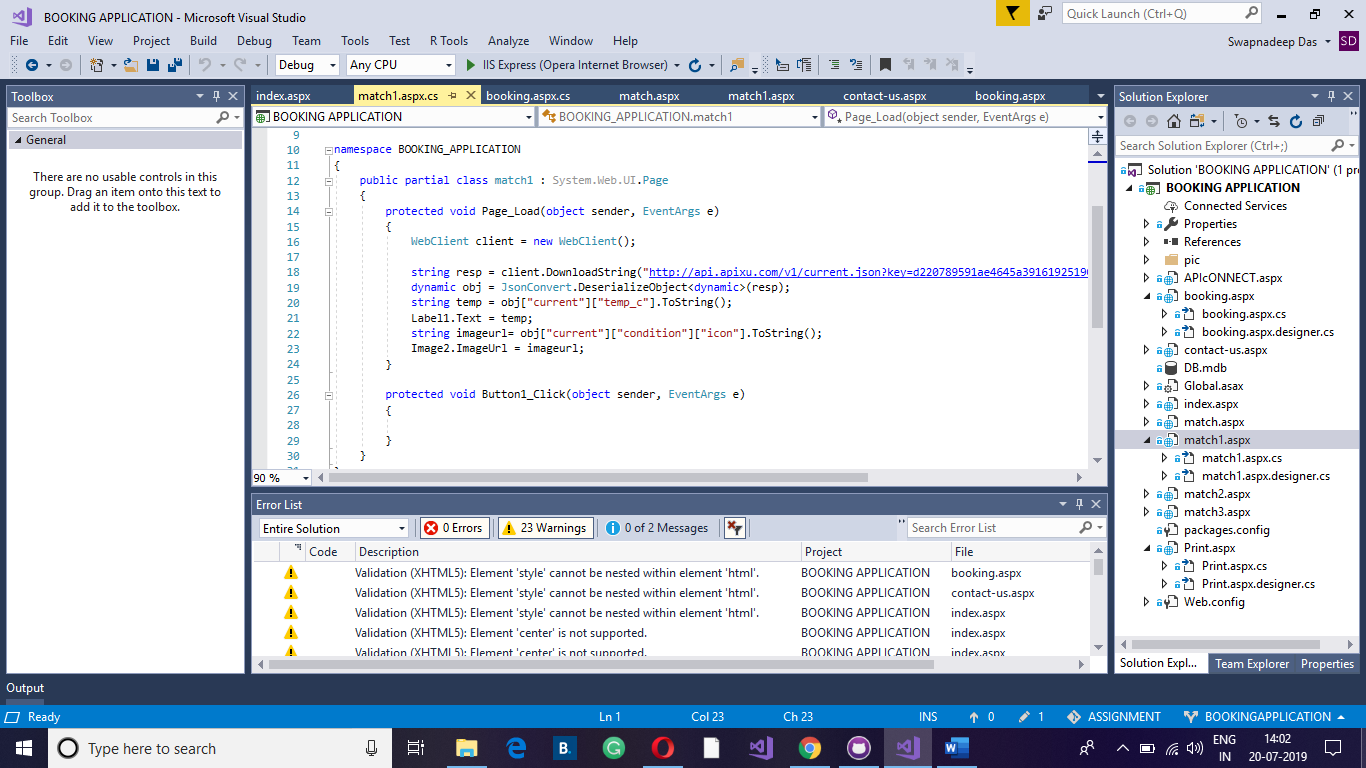




The above screenshots are the basic concepts of API connection. After right click on the application one option is displayed which is Manage NuGet. After clicking on Manage NuGet one API Json option displays. As I have used it for one time that is why this option is not displaying on the screenshot. After double click on Json the API connection activates.

**Screenshot 5:**





Object sender and EventArgs e are the parameters of C#. EventArgs e contains event data and Object sender is a parameter which is known as sender and it contains a reference to the object that raised the event.

**WebClient client = new WebClient();**

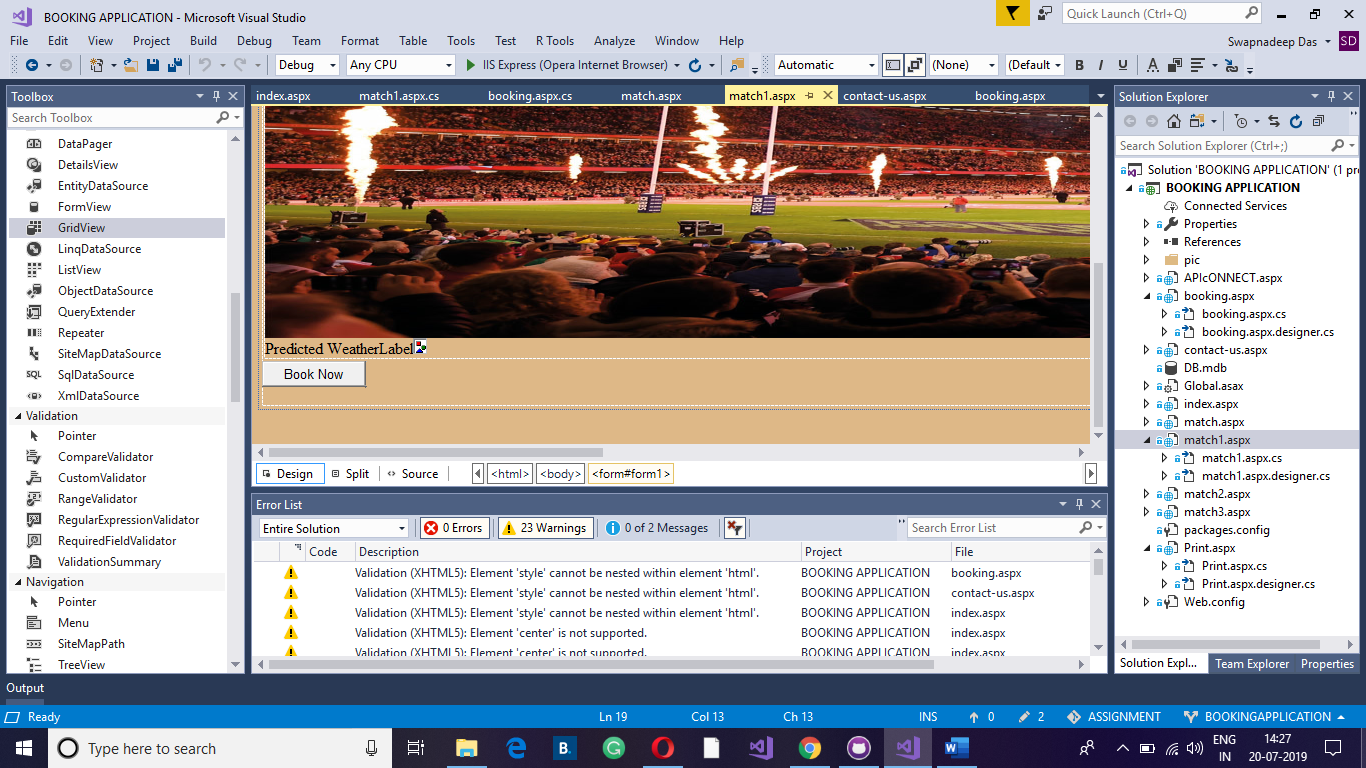
The above code is to send the request to confirm the condition of the weather and the response will come in a Json file. The Json file contains all the details.

**string resp = client.DownloadString("http://api.apixu.com/v1/current.json?key=d220789591ae4645a39161925190307&q=Paris");**

The above request has been sent which is http://api.apixu.com/v1/current.json?key=d220789591ae4645a39161925190307&q=Paris. The request is the application key. The application key has been received after site registration. The weather is about Paris city. They will send the response and the response will come in a string format. The string is Json string.

**dynamic obj = JsonConvert.DeserializeObject<dynamic>(resp);**

The above code is written to break Json string which is kept inside the object.



Predicted WeatherLebel icon are the levels.

**string imageurl= obj["current"]["condition"]["icon"].ToString();**

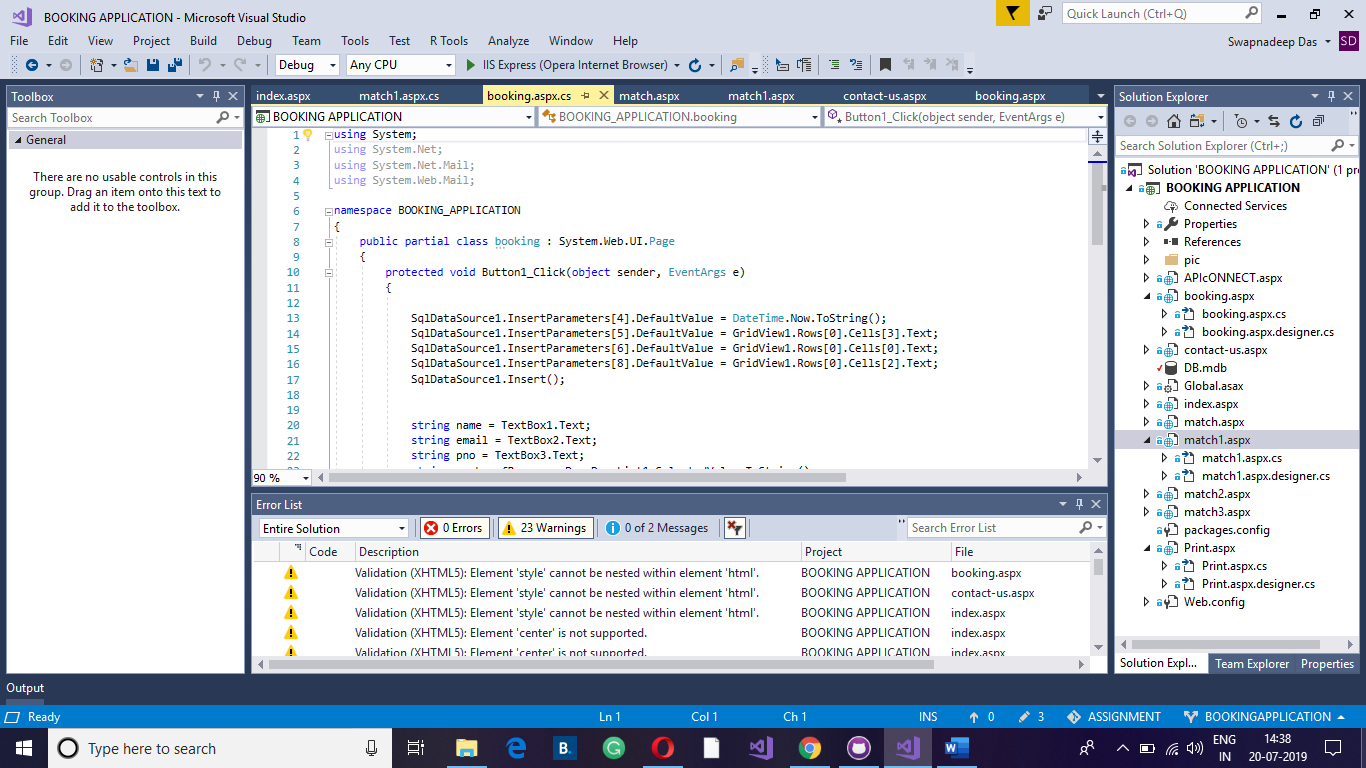
The above code is written to work on weather level.

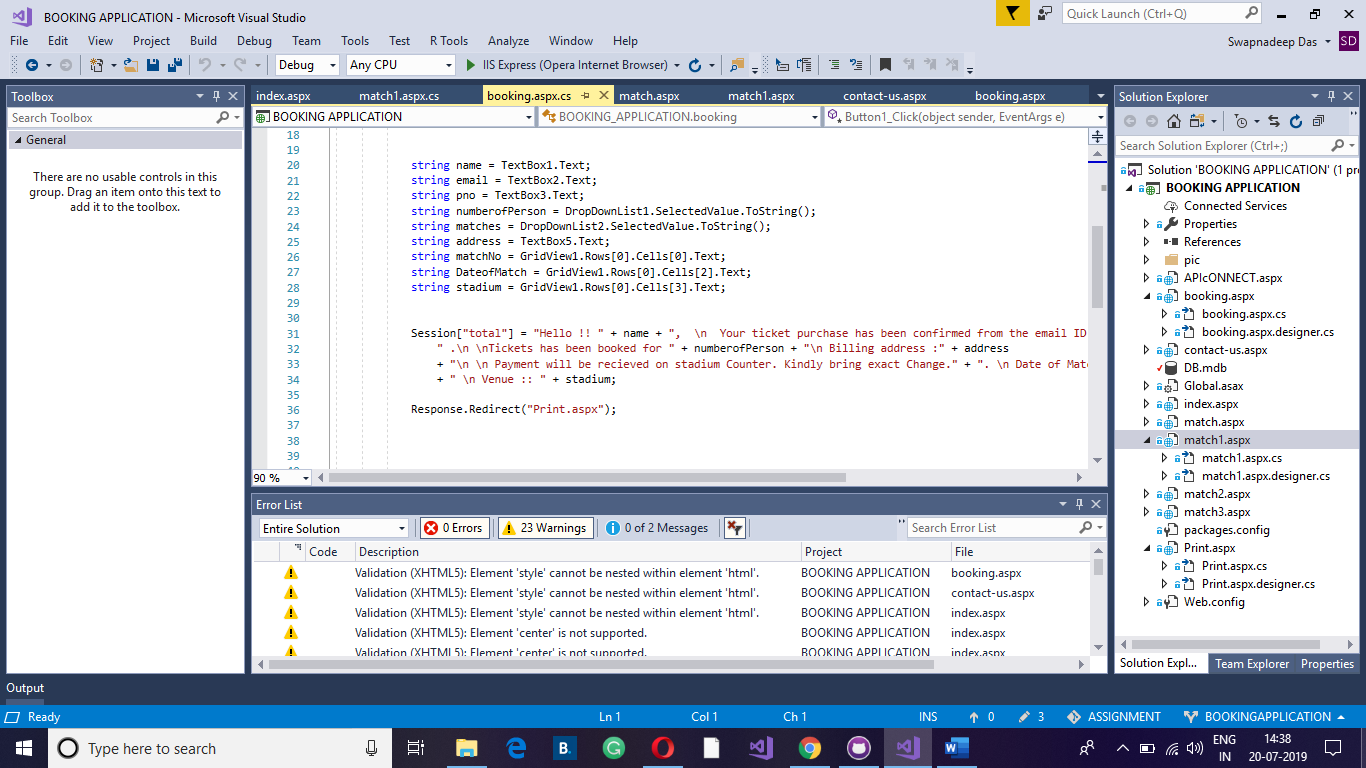
The word “Predicted” stands on [“current”].

The word “WeatherLevel” stands on [“condition”].

The word “icon” stands on [“icon”].

**Screenshot 6:**





Session["total"] = "Hello !! " + name + ", \n Your ticket purchase has been confirmed from the email ID : " + TextBox2.Text +

" .\n \nTickets has been booked for " + numberofPerson + "\n Billing address :" + address

+ "\n \n Payment will be recieved on stadium Counter. Kindly bring exact Change." + ". \n Date of Match :: " + DateofMatch

+ " \n Venue :: " + stadium;

Response.Redirect("Print.aspx");

The above text is stored within a Session variable and is confined within a string. The above text is the concatenation function.

string name = TextBox1.Text;

string email = TextBox2.Text;

string pno = TextBox3.Text;

string numberofPerson = DropDownList1.SelectedValue.ToString();

string matches = DropDownList2.SelectedValue.ToString();

string address = TextBox5.Text;

string matchNo = GridView1.Rows[0].Cells[0].Text;

string DateofMatch = GridView1.Rows[0].Cells[2].Text;

string stadium = GridView1.Rows[0].Cells[3].Text;

The name is taken TextBox1.

The email is taken TextBox2.

The pno is taken TextBox3.

The same concept is applied for all other things.

SqlDataSource1.InsertParameters[4].DefaultValue = DateTime.Now.ToString();

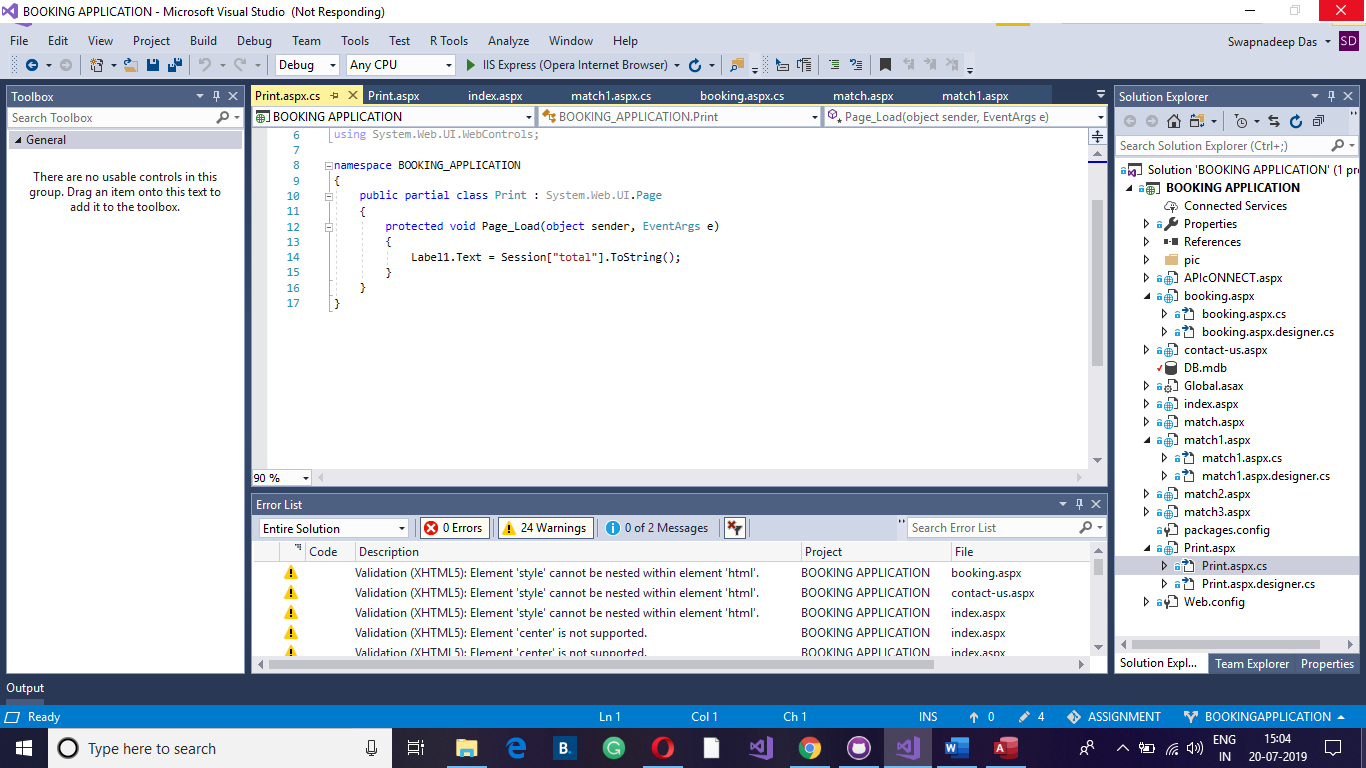
SqlDataSource1.InsertParameters[5].DefaultValue = GridView1.Rows[0].Cells[3].Text;

SqlDataSource1.InsertParameters[6].DefaultValue = GridView1.Rows[0].Cells[0].Text;

SqlDataSource1.InsertParameters[8].DefaultValue = GridView1.Rows[0].Cells[2].Text;

SqlDataSource1.Insert();

The above code is to collect data and insert.



The total string is kept within Global.aspx. The booking page is printed on printing page.

Response.Redirect("Print.aspx");

The above code is used to print the page in printing page.

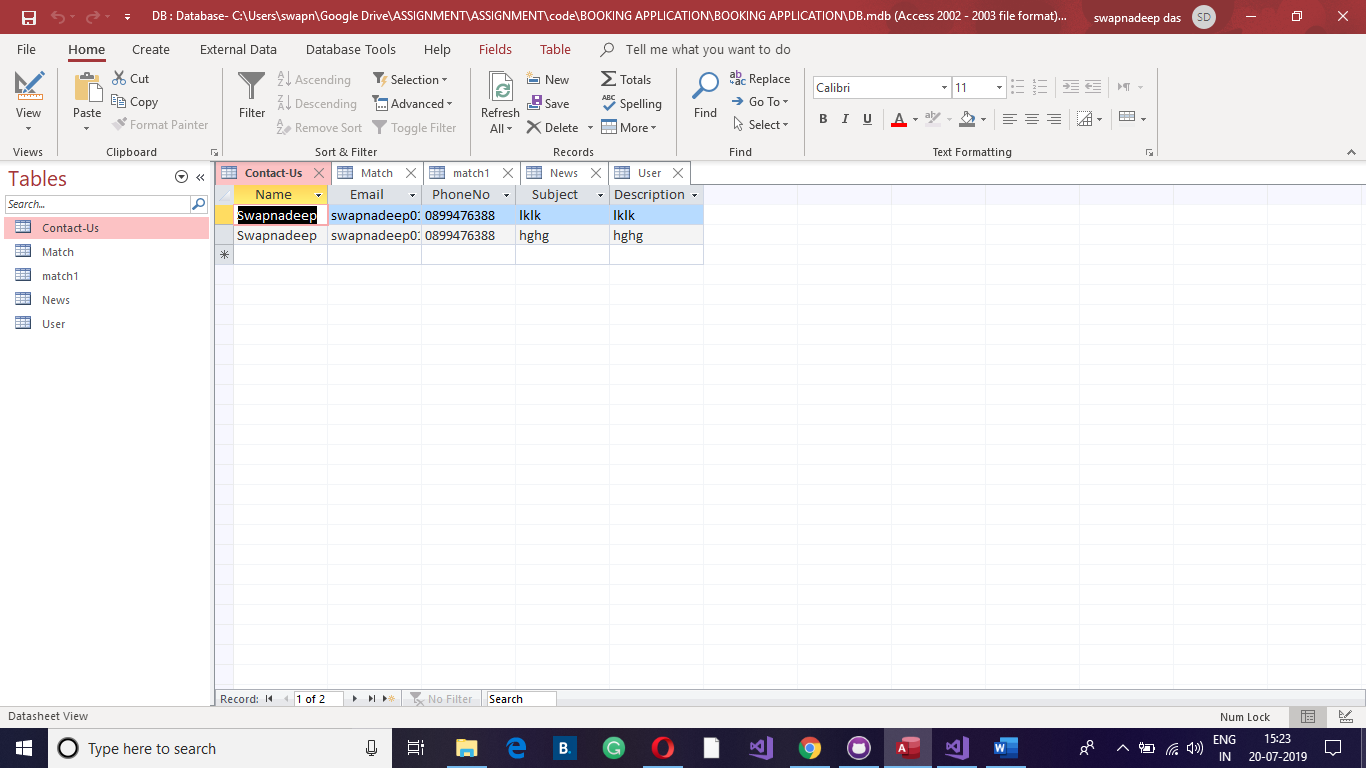
After loading printing page the total string is stored in Level1.Text.

Label1.Text = Session["total"].ToString();

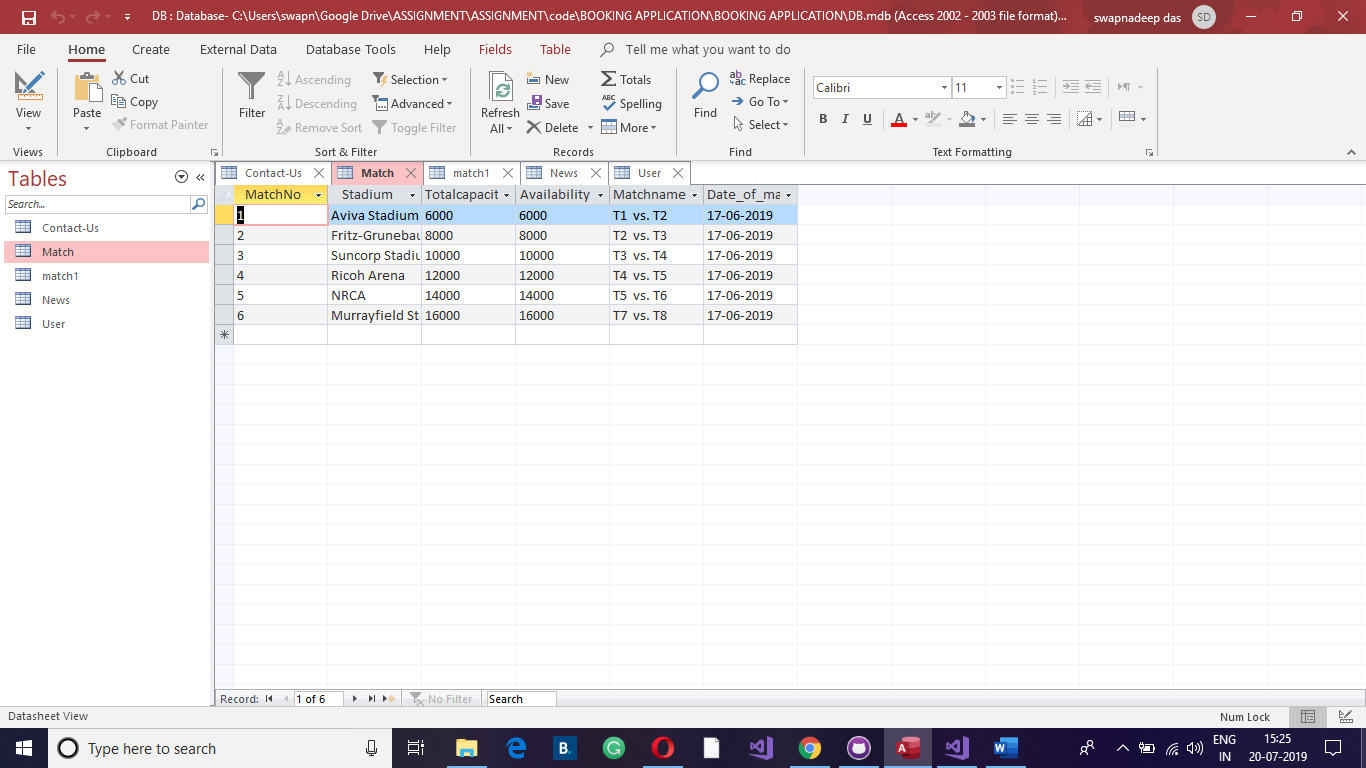
# 3.3 DATABASE SCREENSHOTS:

All the details are storing in database.

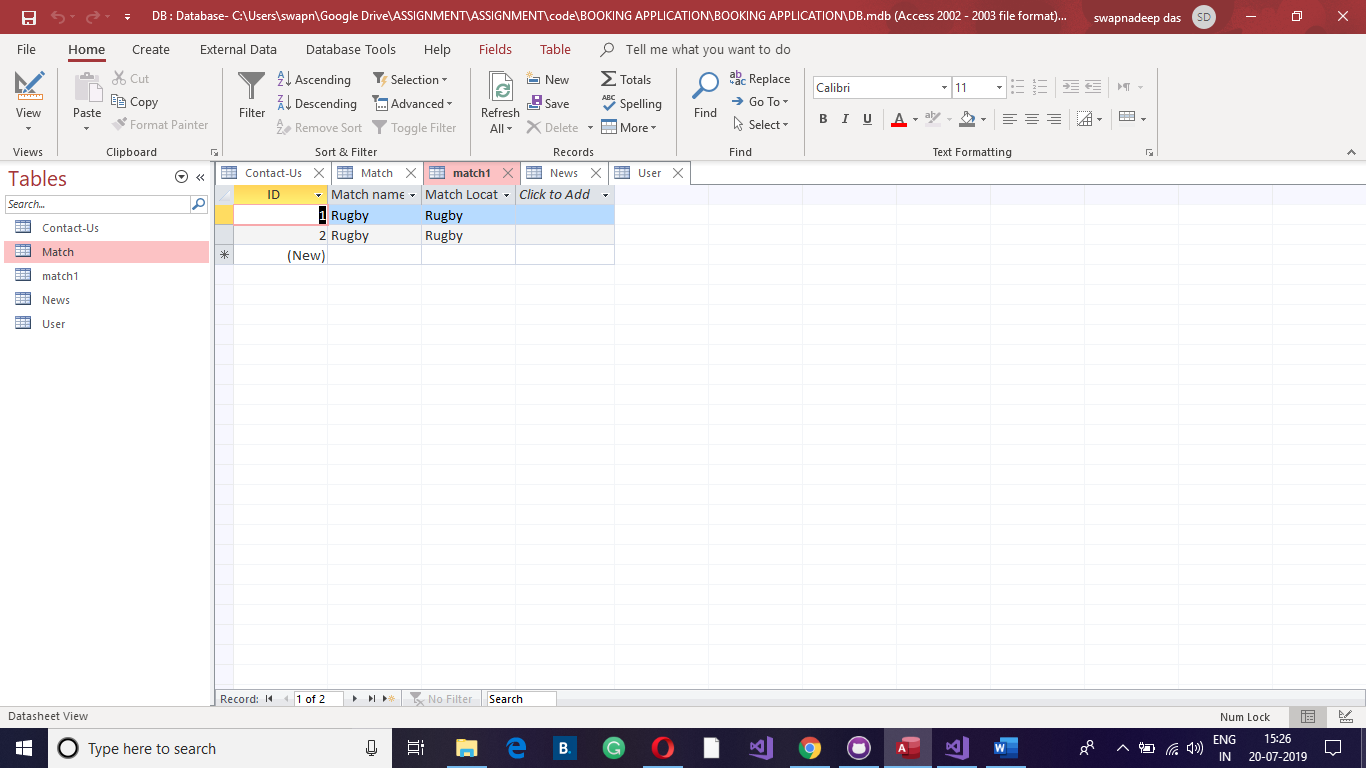
**Contact Us Page:**



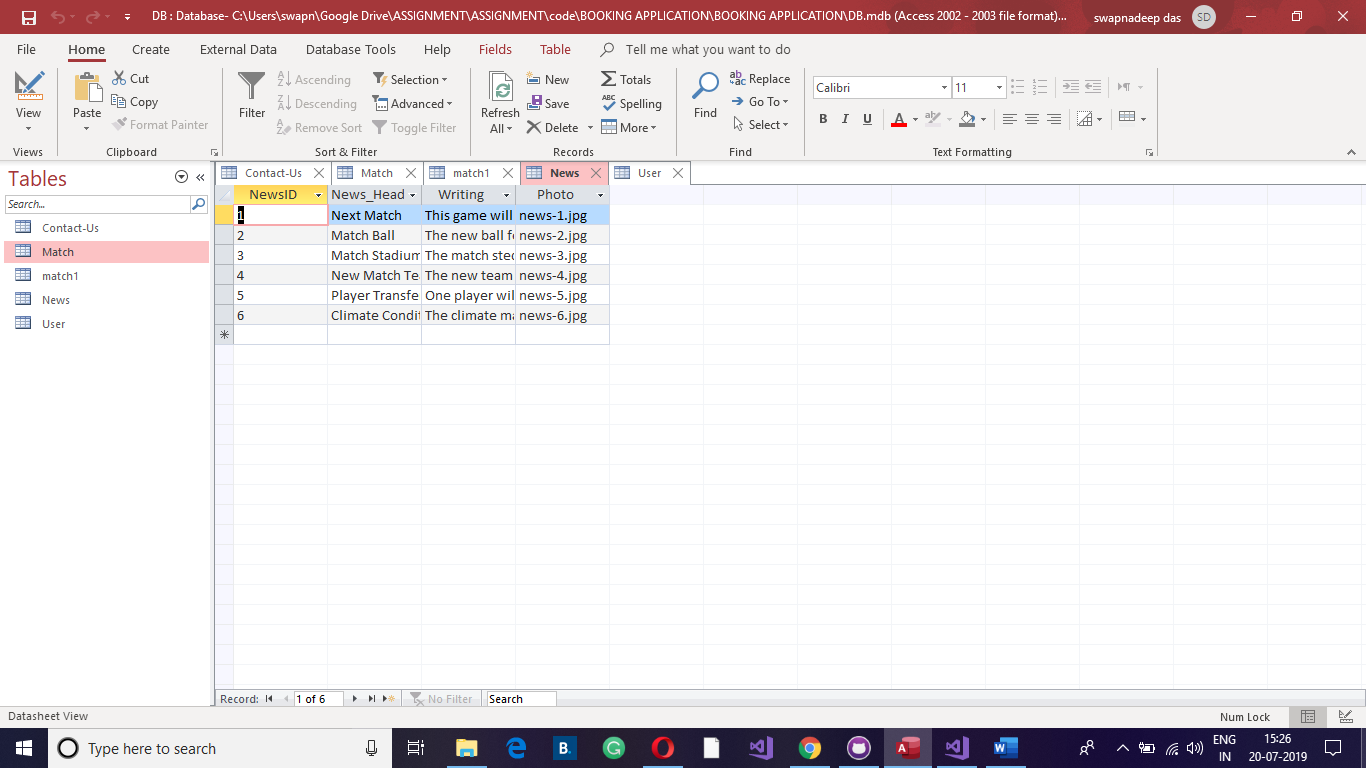
**Match Page:**



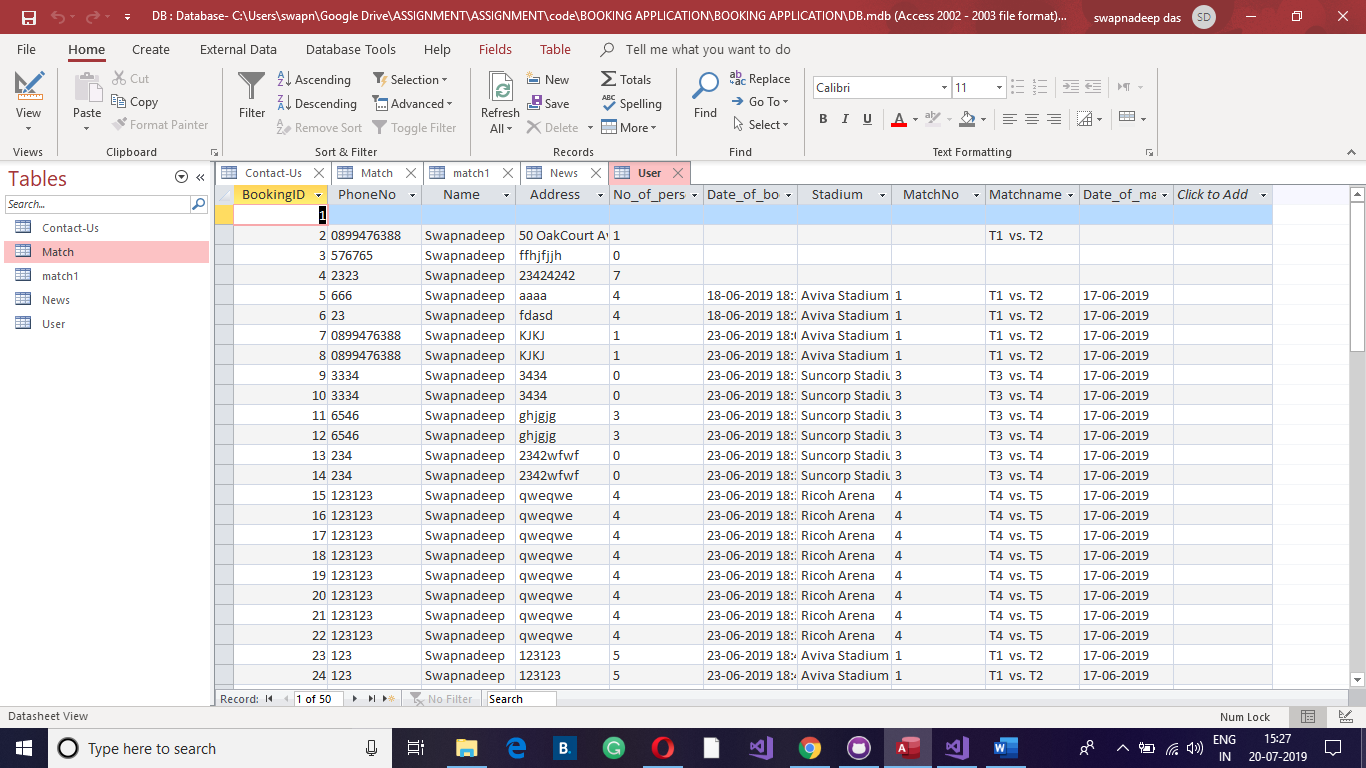
**Match1 Page:**



**News Page:**



**User Page:**



# 

# 4.1 REFERENCES

*asp.net\_tutorial.pdf* (no date). Available at: https://www.tutorialspoint.com/asp.net/asp.net\_tutorial.pdf (Accessed: 9 July 2019).

*Githubflow Online* (no date). Available at: https://guides.github.com/pdfs/githubflow-online.pdf (Accessed: 15 March 2019).

*Home - Weather Unlocked* (no date). Available at: http://www.weatherunlocked.com/ (Accessed: 17 April 2019).

Li, X. (no date) ‘A Tutorial for Git and GitHub’, p. 31.

Tutorialspoint (no date) *GIT Fast Version Control*. Available at: https://www.tutorialspoint.com/git/git\_tutorial.pdf (Accessed: 15 March 2019).