A
Project Report
On
"Resource Management"

(IT346 – Summer Internship-1)

Prepared by

Kashyap Nirmal (16IT059)

Under the Supervision of
Prof. Kamlesh Makvana

Submitted to

Charotar University of Science & Technology (CHARUSAT)
for the Partial Fulfillment of the Requirements for the
Degree of Bachelor of Technology (B.Tech.)
in Information Technology (IT)
for 5th semester B.Tech

Submitted at

CHARUSAT
CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY
Accredited with Grade A by NAAC
Accredited with Grade A by KCG

DEPARTMENT OF INFORMATION TECHNOLOGY
(NBA Accredited)
Chandubhai S. Patel Institute of Technology (CSPIT)
Faculty of Technology & Engineering (FTE), CHARUSAT
At: Changa, Dist: Anand, Pin: 388421.
June, 2018

DECLARATION BY THE CANDIDATE

I hereby declare that the project report entitled "**Resource Management**" submitted by me to Chandubhai S. Patel Institute of Technology, Changa in partial fulfilment of the requirement for the award of the degree of **B. Tech** in Information Technology, from Department of Information Technology, CSPIT/FTE, is a record of bonafide **IT346 Summer Intership-1** carried out by me under the guidance of **Prof Kamlesh Makvana**. I further declare that the work carried out and documented in this project report has not been submitted anywhere else either in part or in full and it is the original work, for the award of any other degree or diploma in this institute or any other institute or university.

Kashyap Nirmal 16IT059

Mr. Kamlesh Makvana,
Assistant Professor,

Department of Information Technology,

CHARUSAT, Changa, Gujarat.

CHARUSAT
CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY

Accredited with Grade A by NAAC
Accredited with Grade A by KCG

CERTIFICATE

This is to certify that the report entitled "**Resource Management**" is a bonafied work carried out by **Kashyap Nirmal (16IT059)** under the guidance and supervision of **Prof. Kamlesh Makvana** for the subject **Summer Intership-1 (IT346)** of 5th Semester of Bachelor of Technology in **Information Technology** at Chandubhai S. Patel Institute of Technology (CSPIT), Faculty of Technology & Engineering (FTE) – CHARUSAT, Gujarat.

To the best of my knowledge and belief, this work embodies the work of candidate himself has duly been completed, and fulfills the requirement of the ordinance relating to the B.Tech. Degree of the University and is up to the standard in respect of content, presentation and language for being referred by the examiner(s).

Under the supervision of,

Mr. Kamlesh Makvana
Assistant Professor,
Department of Information Technology,
CHARUSAT, Changa , Gujarat.

Dr. Parth Shah
Head -Department of Information Technology,
CHARUSAT, Changa, Gujarat.

Chandubhai S. Patel Institute of Technology (CSPIT)

Faculty of Technology & Engineering (FTE), CHARUSAT

At: Changa, Ta. Petlad, Dist. Anand, Pin:388421. Gujarat.

ACKNOWLEDGEMENT

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many hands. I would like to extend my sincere thanks to all of them.

I am grateful to my guide **Mr. Kamlesh Makvana**, for giving me the support and encouragement that was necessary for the completion of this project. He has always been willingly present whenever I needed the slightest support from him. I also thank him for the time that he spared for me, from his extreme busy schedule. I would also like to show my gratitude to my guide for sharing his pearls of wisdom with me during the course of this project.

I am highly indebted to **Mr. Nirav Bhatt** for his guidance and constant supervision as well as for providing necessary encouragement and support.

I would like to express my gratitude to **H.O.D. Mr. Parth Shah** and I am also grateful to all my faculty members of Chandubhai S Patel Institute of Technology for their kind co-operation and encouragement which help me in completion of this project and preparing the report.

Last but not the least, I would also like to thank my colleagues, who have co - operated during the preparation of my report and without them this project has not been possible. Their ideas helped me a lot to improve my project report.

-Kashyap Nirmal(16IT059)

TABLE OF CONTENTS

Acknowledgement i

Abstract 4

List Of Figures 3

Chapter 1 Introduction 5

Chapter 2 Problem Statement 6

Chapter 3 Solution 7

Chapter 4 Levels of Users 8

Chapter 5 Normal User 9

Resource Book 9

Resource Availability 10

Resource Search 11

Chapter 6 Admin User 13

Authentication and Authorization 13

Request Approval 13

Classroom /Lecture Entry 15

Chapter 7 Master Admin User 16

Authorized User 16

Room Type 18

Building 19

Rooms 20

AddRooms 21

Forgot Password 22

Sending Mail 24

Chapter 8 Design 25

Database Design 25

Page Design 25

Chapter 9 Conclusions 26

Chapter 10 Future Works 27

References 28

LIST OF FIGURES AND TABLES

1. Snippet Resource Book(days) ……………………………………….……………..9
2. Snippet Resource Book(Availability) …………………………….……….10
3. Snippet Resource Book(Availability) …………………………….……….10
4. Resource Book ……………………………………………………...……11
5. Snippet Resource Search……..……………………………………....……12
6. Resource Search……..……..…………………………………………....…..12
7. Login Page ……………………………………………………….……..13
8. Snippet Resource Approval..……………………………………..……14
9. Resource Approval ……………………………………………………14
10. Snippet Resource Book(days)………………………………….…………15
11. Lecture Entry………..…..………………………………………………...15
12. Snippet Authorised User… ……………………………………………16
13. Snippet Authorised User(btnupdate)……………………………………16
14. Authorised User ……….………………………………………………..17
15. Authorised User(Updated)……………………….………………….....17
16. Snippet Room_Type ……….……………………………………………...18
17. Room_Type ……..…………………………………………………...18
18. Snippet Building………..……………………………………………19
19. Building………..………………………………………………………..19
20. Snippet Rooms..………..…………………………………………………20
21. Rooms……...………...………………………………………………20
22. Rooms(Updated)………..…………………………………………………21
23. Snippet AddRooms…….…………………………………………………21
24. AddRooms………..…………………………………………………….22
25. AddRooms(User Pop up)...……………………………………………….22
26. Snippet Forgot_Password..…………………………………………….23
27. Forgot_Password……....…………………………………………….23
28. Snippet Sending Mails ……………………………………………………..24
29. BootStrap (Modal Popup User)..……………………………………………25

ABSTRACT

We have created an ASP.NET website. Alongside the ASP.NET We have used ADO.NET, and C# as the main language. The website is to overcome the few of the issues for managing the available resources within the college campus i.e. Seminar halls, Conference Rooms, Classrooms, Laboratories etc. The managing basically means having that specific resource registered and retrieved through the provided interface. So the website lets the user input the required conditions either may be for registering the resource for any pre-planned workshop or Expert Talk and many more. The website also lets the user retrieve and check the availability of the desired resource may be while registering itself and also according to a timespan of the user input dates. The website lets the authorized user to check all the requests and approve them as per the requirements and the authorized user is also notified by the mail while registering. The requested user is also notified by the mail of approval. There is also a provision for entering the classroom schedule according to the timetable by the authorized user for the convenience of the end user to receive the proper details of the classroom lectures too and also for avoiding the clashes during any Expert sessions, if any. So, ultimately the website includes almost all the functionalities for managing the resource which was manually handled previously.

1. INTRODUCTION

Purpose: -

To manage the resource according to the need basis. Managing resource may include requesting for the available resources, approving and rejecting the requests, adding resources, managing the resources, managing the users etc.

Reason: -

Because till present date all the task of managing the resources was done manually. So to reduce all this manual processes and having a centralised system for managing the resource was very much important.

Work: -

For any institutes many sessions like Expert talks, or Workshops etc. are ongoing. So it is very much important for them to manage the resources accordingly. Not only managing but making proper utilization of the resources is also too much important. The situation that may arise here may be like:

- Resource may be pre-booked.
- Resource may be occupied for routines.
- Resource may be registered by some other departments, etc.

This may sometimes lead to any clash like situations etc. So in order to avoid these kind of situations we can have a centralized application which can manage all the resources and all such situations.

Thus a centralized application keeping in mind the present scenario can be developed. Which reduces the complexities, makes it easier and manages the resource in a well-organized manner.

2. PROBLEM STATEMENT

Whenever we are supposed to conduct a workshop, or expert session, or meeting in the campus then it was very much a hard situation due to following reasons:

- You may have to call the person in-charge.
- You were supposed to write mail in order to notify the person about your requirements and request.
- The records maintained were manual.
- Not only that almost all the procedures included have to be performed manually.
- So maintaining the record was difficult. Any change in the event needs to be modified and again seek the permission of the user. And updating the details in the record of person in-charge was also to be manually be made.

3. SOLUTION

We can create a centralized application. Now this application can provide us with the major functionalities including almost all that are manually done till date. The functionalities can be extended as per the requirements too. Through this application almost all the tasks can be made computerized.

We have to create the application which maintains the data entries for the requests. The requests are displayed to the Admin user. Then he will grant or reject the request. Now, till date for all these process mails were transferred and the user may be contacted by some other means. But in our application we decided to use the mails that are written through the program itself. These mails are sent for the request to the Admin user and for the grant or reject to the user who made the request.

Through this application many clashes for any resources, like for the workshop or for the others can be avoided, due to the pre-planning. Through this application the Admin user can have an watchful eye on the resources.

4. Levels of Users

For this centralized application we have various 3 levels of users. They are as listed below:

a.) Normal User
b.) Admin User
c.) Master Admin User

Here is the brief roles and rights assigned to these users as below:

1. Normal User:

The normal user is the one who can make requests for the resource. They can check the availability of the resource, whilst making the request. They can check the occupancy of any particular resource. They can search for the entire occupancy in any user specified range of days. The normal user does not have any login id or anything else unlike the other two users.

2. Admin User:

The Admin users are the assigned with the task of eyeing and approving the request made by normal users. They are supposed to make the data entry of the regular lecture/classroom schedules. They can also perform every single task performed by normal user. But the scenario here is they are not supposed to request to anyone else.

3. Master Admin User:

The users are at top level of the hierarchy. The y are the ones who have th e rights to manage the resources. Managing here means adding, updating and deleting the resources. They also manage the authorised users i.e. adding, updating or deleting users. One of the other important responsibility of this user is to manage all these roles of the existing users i.e. he can assign or update the roles of the users.

5.NORMAL USER

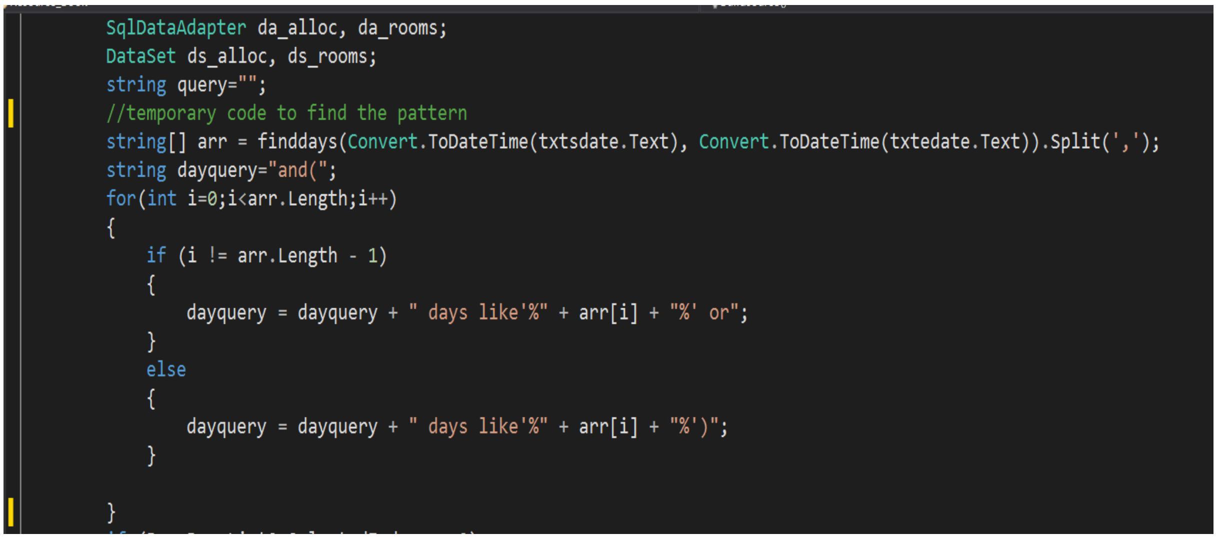
➢ Resource Book:

This is the basic web form which takes the input from the end user for registering the resource for future purposes like Workshops, Expert Talks etc. The user inputs the required details and then requests for checking the availability. So he gets the output in terms of whether the specific resource is allocated or is available in that particular mentioned conditions including the dates and the time slots. The main advantage here is as soon as the user request the available resource the authorised user is notified about the request through an email which makes keeping the track records very much easier.

So, here we have designed the efficient query for processing the user request. The request processing mainly includes retrieving the data from the database table and comparing the user request with the previously made requests made by other users for the same resource. Now we do have a scenario where the resource may be occupied for some regular routine lectures or something. These routines are defined in terms of days i.e. Mon, Wed, Thu etc. in a specific time period. So I was to add the 'days' column in database table for the same. Now here comparing dates was particularly something easier. But to work with comparing days which was added later on as I came to know about the situation was a bit trickier. So I used something like 'Pattern' with the 'like' clause for retrieving and comparing with the allocated database entries.

We are planning to add some proper AJAX into the web form as per the requirements. The ADO.NET architecture I used here for processing the database table is 'Disconnected' architecture which includes DataSet and a few more.

Code Snippets:



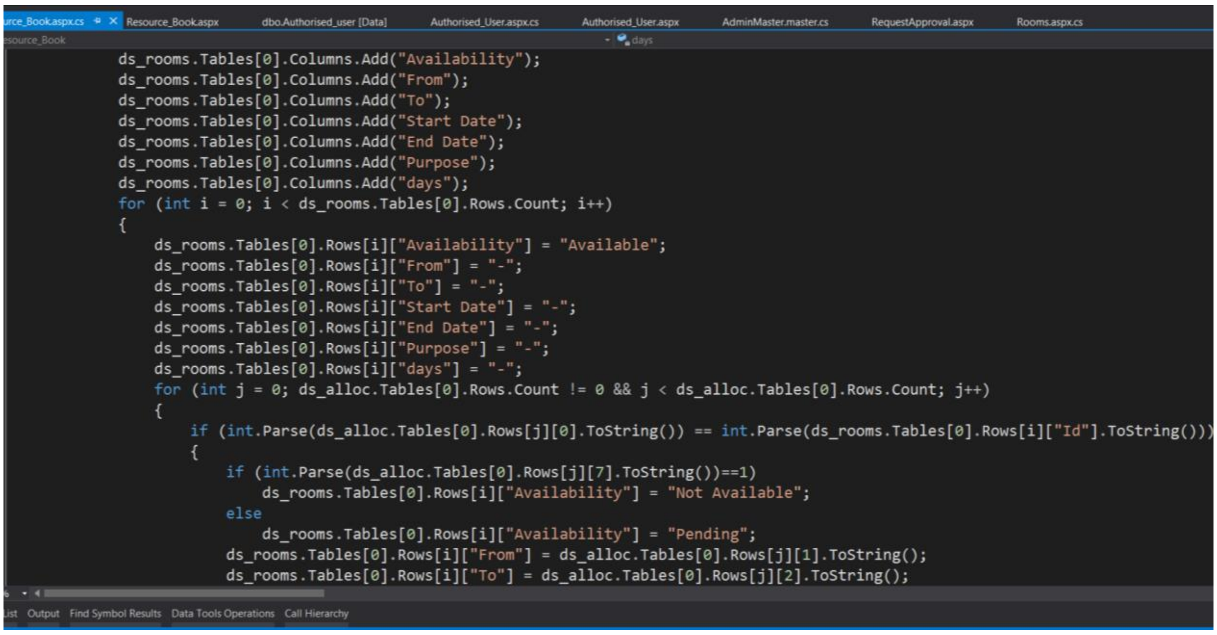
```
SqlDataAdapter da_alloc, da_rooms;
DataSet ds_alloc, ds_rooms;
string query="";
//temporary code to find the pattern
string[] arr = finddays(Convert.ToDateTime(txtsdate.Text), Convert.ToDateTime(txtedate.Text)).Split(',');
string dayquery="and(";
for(int i=0;i<arr.Length;i++)
{
    if (i != arr.Length - 1)
    {
        dayquery = dayquery + " days like'%" + arr[i] + "%' or";
    }
    else
    {
        dayquery = dayquery + " days like'%" + arr[i] + "%')";
    }

}
```

Figure 1 | Snippet Resource Book (days)

➢ Resource Availability:

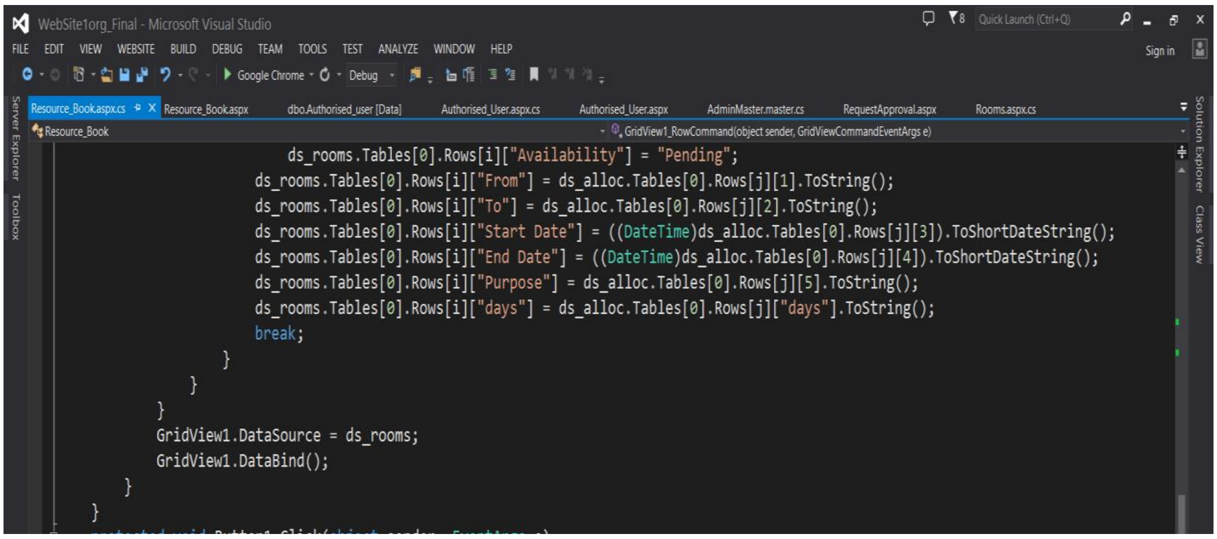
Here I have created the efficient query for retrieving the requested resource from our database table. After retrieving the data and taking the input from the user I had to display the output in terms of whether the resource is available or not. If the resource was not available then I had to mention the reasons for that which includes displaying the details for the resource i.e. Start to End Date, Start to End time, Purpose, Contact Person, etc. Here in checking the availability also, it was very much important to go through comparing the 'days' along with the dates and times. Displaying the data makes it easier for the user to contact the Contact person who has made the request prior to him and discuss and sort out the things to avoid any further clashes amongst their works.

Code Snippets:



```
ds_rooms.Tables[0].Columns.Add("Availability");
ds_rooms.Tables[0].Columns.Add("From");
ds_rooms.Tables[0].Columns.Add("To");
ds_rooms.Tables[0].Columns.Add("Start Date");
ds_rooms.Tables[0].Columns.Add("End Date");
ds_rooms.Tables[0].Columns.Add("Purpose");
ds_rooms.Tables[0].Columns.Add("days");
for (int i = 0; i < ds_rooms.Tables[0].Rows.Count; i++)
{
    ds_rooms.Tables[0].Rows[i]["Availability"] = "Available";
    ds_rooms.Tables[0].Rows[i]["From"] = "-";
    ds_rooms.Tables[0].Rows[i]["To"] = "-";
    ds_rooms.Tables[0].Rows[i]["Start Date"] = "-";
    ds_rooms.Tables[0].Rows[i]["End Date"] = "-";
    ds_rooms.Tables[0].Rows[i]["Purpose"] = "-";
    ds_rooms.Tables[0].Rows[i]["days"] = "-";
    for (int j = 0; ds_alloc.Tables[0].Rows.Count != 0 && j < ds_alloc.Tables[0].Rows.Count; j++)
    {
        if (int.Parse(ds_alloc.Tables[0].Rows[j][0].ToString()) == int.Parse(ds_rooms.Tables[0].Rows[i]["Id"].ToString()))
        {
            if (int.Parse(ds_alloc.Tables[0].Rows[j][7].ToString())==1)
                ds_rooms.Tables[0].Rows[i]["Availability"] = "Not Available";
            else
                ds_rooms.Tables[0].Rows[i]["Availability"] = "Pending";
            ds_rooms.Tables[0].Rows[i]["From"] = ds_alloc.Tables[0].Rows[j][1].ToString();
            ds_rooms.Tables[0].Rows[i]["To"] = ds_alloc.Tables[0].Rows[j][2].ToString();
```

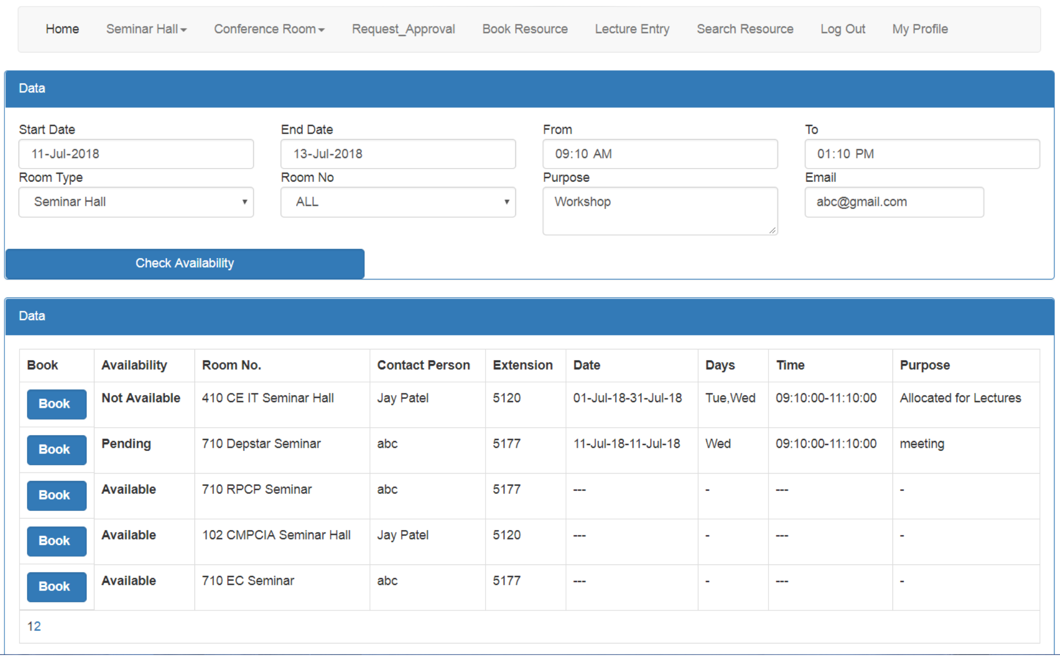
Figure 2 | Snippet Resource Book (Availability)



```
                ds_rooms.Tables[0].Rows[i]["Availability"] = "Pending";
            ds_rooms.Tables[0].Rows[i]["From"] = ds_alloc.Tables[0].Rows[j][1].ToString();
            ds_rooms.Tables[0].Rows[i]["To"] = ds_alloc.Tables[0].Rows[j][2].ToString();
            ds_rooms.Tables[0].Rows[i]["Start Date"] = ((DateTime)ds_alloc.Tables[0].Rows[j][3]).ToShortDateString();
            ds_rooms.Tables[0].Rows[i]["End Date"] = ((DateTime)ds_alloc.Tables[0].Rows[j][4]).ToShortDateString();
            ds_rooms.Tables[0].Rows[i]["Purpose"] = ds_alloc.Tables[0].Rows[j][5].ToString();
            ds_rooms.Tables[0].Rows[i]["days"] = ds_alloc.Tables[0].Rows[j]["days"].ToString();
            break;
        }
    }
}
GridView1.DataSource = ds_rooms;
GridView1.DataBind();
```
```
}
```

Figure 3 | Snippet Resource Book (Availability)

Snap shot:



Start Date	End Date	From	To
11-Jul-2018	13-Jul-2018	09:10 AM	01:10 PM
Room Type	Room No	Purpose	Email
Seminar Hall	ALL	Workshop	abc@gmail.com

Book	Availability	Room No.	Contact Person	Extension	Date	Days	Time	Purpose
Book	Not Available	410 CE IT Seminar Hall	Jay Patel	5120	01-Jul-18-31-Jul-18	Tue,Wed	09:10:00-11:10:00	Allocated for Lectures
Book	Pending	710 Depstar Seminar	abc	5177	11-Jul-18-11-Jul-18	Wed	09:10:00-11:10:00	meeting
Book	Available	710 RPCP Seminar	abc	5177	---	-	---	-
Book	Available	102 CMPCIA Seminar Hall	Jay Patel	5120	---	-	---	-
Book	Available	710 EC Seminar	abc	5177	---	-	---	-

12

Figure 4 | Resource Book

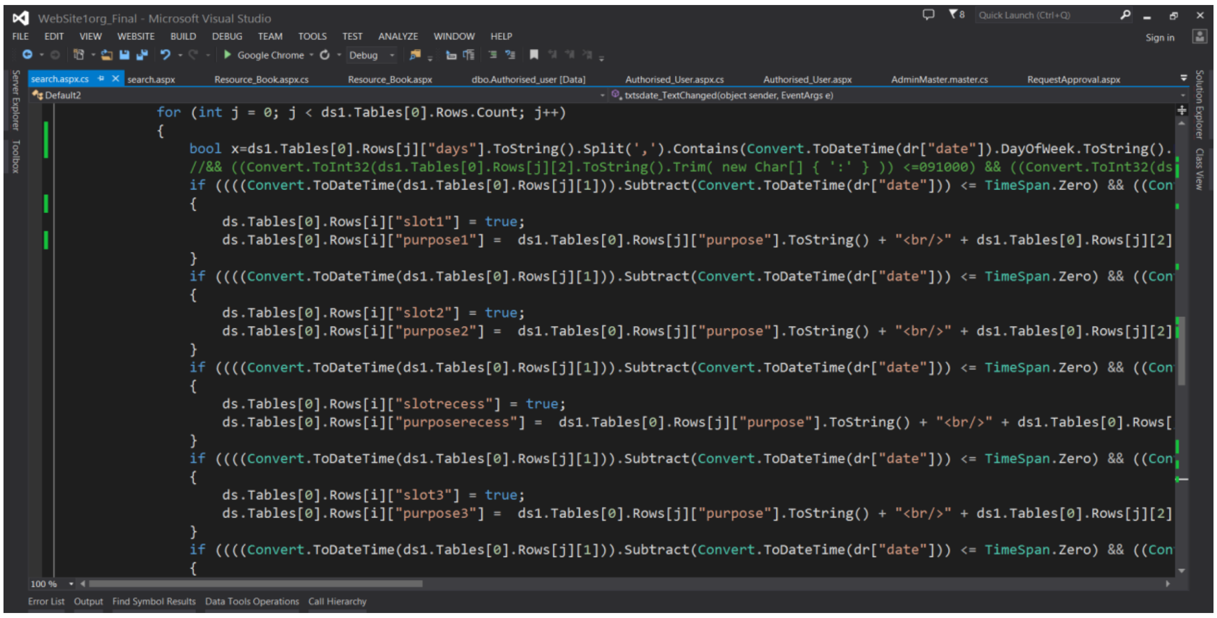
➢ Resource Search:

Whenever the user wants to check the overall occupancy of the particular resource in an entire range of dates and all the time slots he is supposed to look up in this web form. This web form asks the user for the resource name and the range of the dates for which the user wants to check the occupancy. Then in the output the user is served with the occupancy details of all the time slots in that range mentioned along with the details like purpose and time duration etc.

Here the entire range and each and every slot of the day is to be compared with the allocated entries in the database table. So I had to design the efficient query for retrieving the data. After retrieving the data, I was supposed to use the appropriate Data control element for comparing it to the allocated entries.

So here the comparison part is considered to be the life of this web form. Thus, simply comparison itself was not enough, but efficient way of comparing was very much the thing. So here I have used the idea that from which particular index to start till which index from the details of the allocated entries. Here finding this range of index was very much important because by doing so we can avoid the unwanted iterations which don’t even have to be performed because there is no allocation in that particular range of index. After finding these indices comes the part of checking the particular days in that range of allocated dates. Finding these days in that range was the trickiest part. So I used the same thing with dates i.e. finding the index by taking the difference among present day and the day of the allocated date and then incrementing the index in somewhat different manner because as the same schedule is followed up the next week. So if one entire week from range is completed with then the task becomes easier for that range too as it has just to followed up the very similar manner.

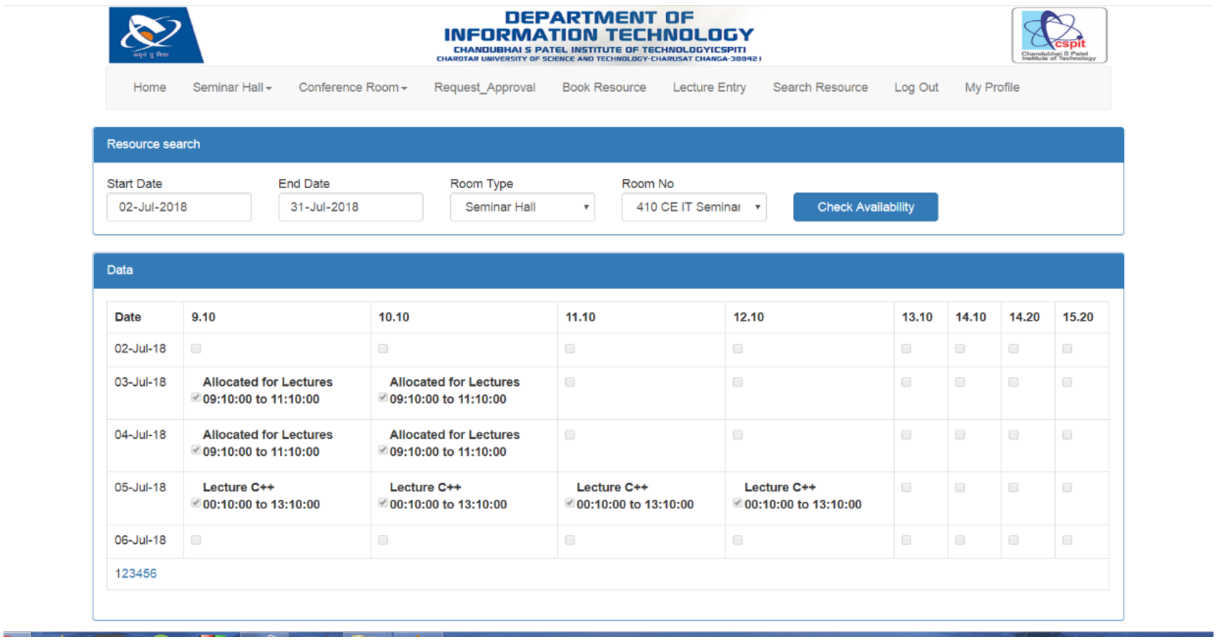
Code Snippets:



```
            for (int j = 0; j < ds1.Tables[0].Rows.Count; j++)
            {
                bool x=ds1.Tables[0].Rows[j]["days"].ToString().Split(',').Contains(Convert.ToDateTime(dr["date"]).DayOfWeek.ToString().
                //&& ((Convert.ToInt32(ds1.Tables[0].Rows[j][2].ToString().Trim( new Char[] { ':' } )) <=091000) && ((Convert.ToInt32(ds
                if ((((Convert.ToDateTime(ds1.Tables[0].Rows[j][1])).Subtract(Convert.ToDateTime(dr["date"])) <= TimeSpan.Zero) && ((Con
                {
                    ds.Tables[0].Rows[i]["slot1"] = true;
                    ds.Tables[0].Rows[i]["purpose1"] =  ds1.Tables[0].Rows[j]["purpose"].ToString() + "<br/>" + ds1.Tables[0].Rows[j][2]
                }
                if ((((Convert.ToDateTime(ds1.Tables[0].Rows[j][1])).Subtract(Convert.ToDateTime(dr["date"])) <= TimeSpan.Zero) && ((Con
                {
                    ds.Tables[0].Rows[i]["slot2"] = true;
                    ds.Tables[0].Rows[i]["purpose2"] =  ds1.Tables[0].Rows[j]["purpose"].ToString() + "<br/>" + ds1.Tables[0].Rows[j][2]
                }
                if ((((Convert.ToDateTime(ds1.Tables[0].Rows[j][1])).Subtract(Convert.ToDateTime(dr["date"])) <= TimeSpan.Zero) && ((Con
                {
                    ds.Tables[0].Rows[i]["slotrecess"] = true;
                    ds.Tables[0].Rows[i]["purposerecess"] =  ds1.Tables[0].Rows[j]["purpose"].ToString() + "<br/>" + ds1.Tables[0].Rows[
                }
                if ((((Convert.ToDateTime(ds1.Tables[0].Rows[j][1])).Subtract(Convert.ToDateTime(dr["date"])) <= TimeSpan.Zero) && ((Con
                {
                    ds.Tables[0].Rows[i]["slot3"] = true;
                    ds.Tables[0].Rows[i]["purpose3"] =  ds1.Tables[0].Rows[j]["purpose"].ToString() + "<br/>" + ds1.Tables[0].Rows[j][2]
                }
                if ((((Convert.ToDateTime(ds1.Tables[0].Rows[j][1])).Subtract(Convert.ToDateTime(dr["date"])) <= TimeSpan.Zero) && ((Con
                {
```

Figure 5 | Snippet Resource Search

Snap shot:



DEPARTMENT OF INFORMATION TECHNOLOGY
CHANDUBHAI S PATEL INSTITUTE OF TECHNOLOGY(CSPIT)
CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY-CHARUSAT CHANGA-388421
Home
Seminar Hall
Conference Room
Request_Approval
Book Resource
Lecture Entry
Search Resource
Log Out
My Profile
Check Availability

Resource search

Start Date	End Date	Room Type	Room No
02-Jul-2018	31-Jul-2018	Seminar Hall	410 CE IT Seminar

Data

Date	9.10	10.10	11.10	12.10	13.10	14.10	14.20	15.20
02-Jul-18	☐	☐	☐	☐	☐	☐	☐	☐
03-Jul-18	Allocated for Lectures ✓ 09:10:00 to 11:10:00	Allocated for Lectures ✓ 09:10:00 to 11:10:00	☐	☐	☐	☐	☐	☐
04-Jul-18	Allocated for Lectures ✓ 09:10:00 to 11:10:00	Allocated for Lectures ✓ 09:10:00 to 11:10:00	☐	☐	☐	☐	☐	☐
05-Jul-18	Lecture C++ ✓ 00:10:00 to 13:10:00	Lecture C++ ✓ 00:10:00 to 13:10:00	Lecture C++ ✓ 00:10:00 to 13:10:00	Lecture C++ ✓ 00:10:00 to 13:10:00	☐	☐	☐	☐
06-Jul-18	☐	☐	☐	☐	☐	☐	☐	☐

123456

Figure 6 | Resource Search

6.ADMIN USER

➢ Authentication and authorization:

There are certain authorised user. So they are supposed to Login and perform their activities. So we have to add the Login web form and add the authentications for logging in the authorised user. Now, after they are logged in they can perform their specific task like approving the requests, classroom/resource entry etc. But here the authorised user is authorised only for certain resource. So whenever the user logins he just should be able to outperform tasks on the resource for which he has the authorization. Thus authorization for this was also added.

Snap shot:

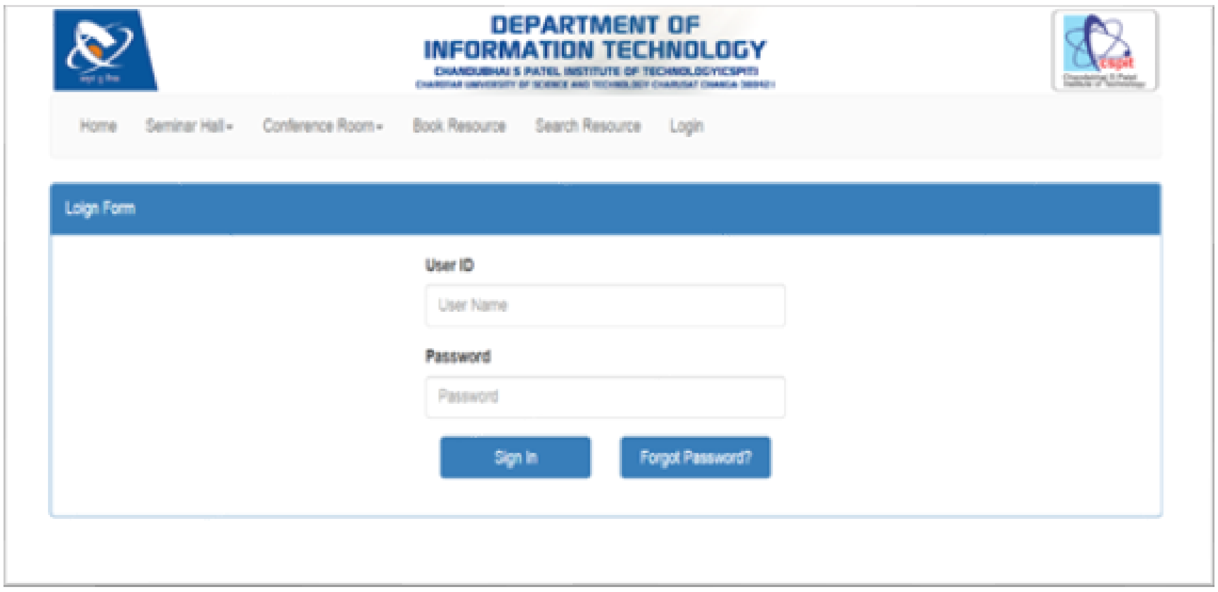


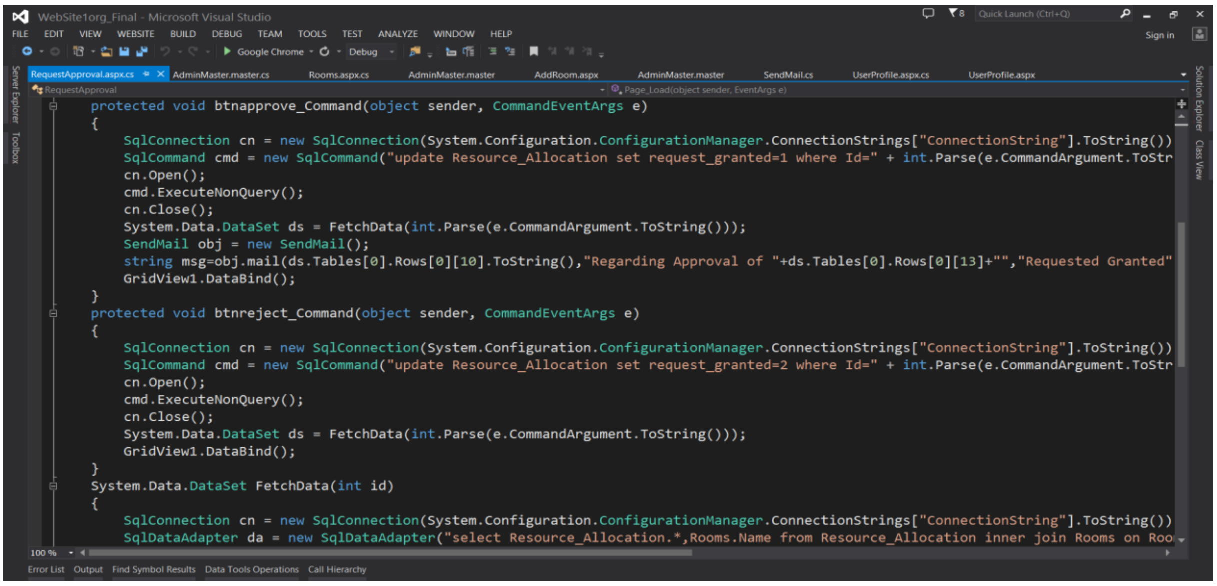
Figure 7 | Login Page

➢ Request approval:

All the request that are made and comes under that authorised user are supposed to be approved. Now after the user logins and selects Request Approval he would be asked to select the particular resource. So the user would be provided with all the requests that are pending to be approved. After going through the details and checking the importance amongst the common entries i.e. for same resource in the same period he will approve the pending request and change the status to granted or not.

Here as soon as the authorised user approves the request the former user who made the request receives the confirmation mail. This confirmation mail includes the details for registering the resource for the given dates. So the former user will come to know as soon as he is granted the permission which is very much helpful.

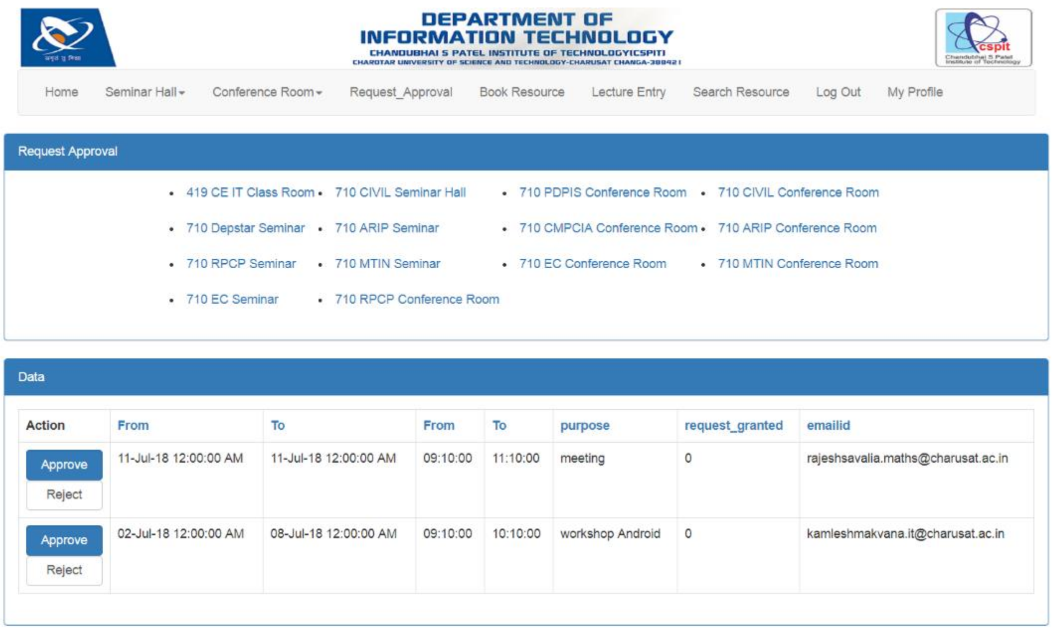
Code Snippets:



```
protected void btnapprove_Command(object sender, CommandEventArgs e)
{
    SqlConnection cn = new SqlConnection(System.Configuration.ConfigurationManager.ConnectionStrings["ConnectionString"].ToString())
    SqlCommand cmd = new SqlCommand("update Resource_Allocation set request_granted=1 where Id=" + int.Parse(e.CommandArgument.ToStr
    cn.Open();
    cmd.ExecuteNonQuery();
    cn.Close();
    System.Data.DataSet ds = FetchData(int.Parse(e.CommandArgument.ToString()));
    SendMail obj = new SendMail();
    string msg=obj.mail(ds.Tables[0].Rows[0][10].ToString(),"Regarding Approval of "+ds.Tables[0].Rows[0][13]+"","Requested Granted"
    GridView1.DataBind();
}
protected void btnreject_Command(object sender, CommandEventArgs e)
{
    SqlConnection cn = new SqlConnection(System.Configuration.ConfigurationManager.ConnectionStrings["ConnectionString"].ToString())
    SqlCommand cmd = new SqlCommand("update Resource_Allocation set request_granted=2 where Id=" + int.Parse(e.CommandArgument.ToStr
    cn.Open();
    cmd.ExecuteNonQuery();
    cn.Close();
    System.Data.DataSet ds = FetchData(int.Parse(e.CommandArgument.ToString()));
    GridView1.DataBind();
}
System.Data.DataSet FetchData(int id)
{
    SqlConnection cn = new SqlConnection(System.Configuration.ConfigurationManager.ConnectionStrings["ConnectionString"].ToString())
    SqlDataAdapter da = new SqlDataAdapter("select Resource_Allocation.*,Rooms.Name from Resource_Allocation inner join Rooms on Roo
```

Figure 8 | Snippet Request Approval

Snap shot:



Request Approval

- 419 CE IT Class Room
- 710 CIVIL Seminar Hall
- 710 PDPIS Conference Room
- 710 CIVIL Conference Room
- 710 Depstar Seminar
- 710 ARIP Seminar
- 710 CMPCIA Conference Room
- 710 ARIP Conference Room
- 710 RPCP Seminar
- 710 MTIN Seminar
- 710 EC Conference Room
- 710 MTIN Conference Room
- 710 EC Seminar
- 710 RPCP Conference Room

Data

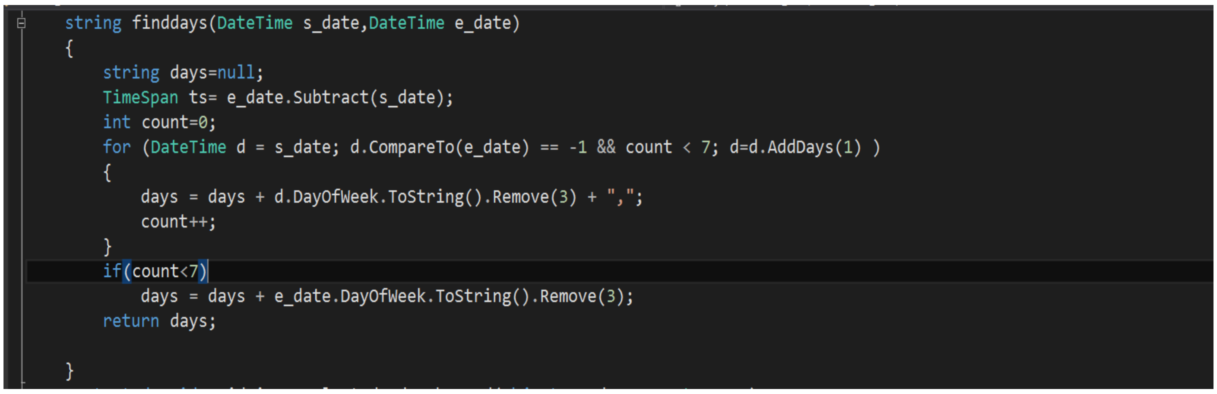
Action	From	To	From	To	purpose	request_granted	emailid
Approve Reject	11-Jul-18 12:00:00 AM	11-Jul-18 12:00:00 AM	09:10:00	11:10:00	meeting	0	rajeshsavalia.maths@charusat.ac.in
Approve Reject	02-Jul-18 12:00:00 AM	08-Jul-18 12:00:00 AM	09:10:00	10:10:00	workshop Android	0	kamleshmakvana.it@charusat.ac.in

Figure 9 | Request Approval

➢ Classroom/Lecture Entry:

The regular routine classrooms are also to be entered and processed if required. So, I have created another web form for the very same purpose. These are the certain tasks that are to be performed by some of the authorised users only. So, only a person logged in can access this web form. The main idea for classroom entry is for a particular range of date the classroom slots are registered on particular days. So, here the user is provided with the controls to select day for the entry.

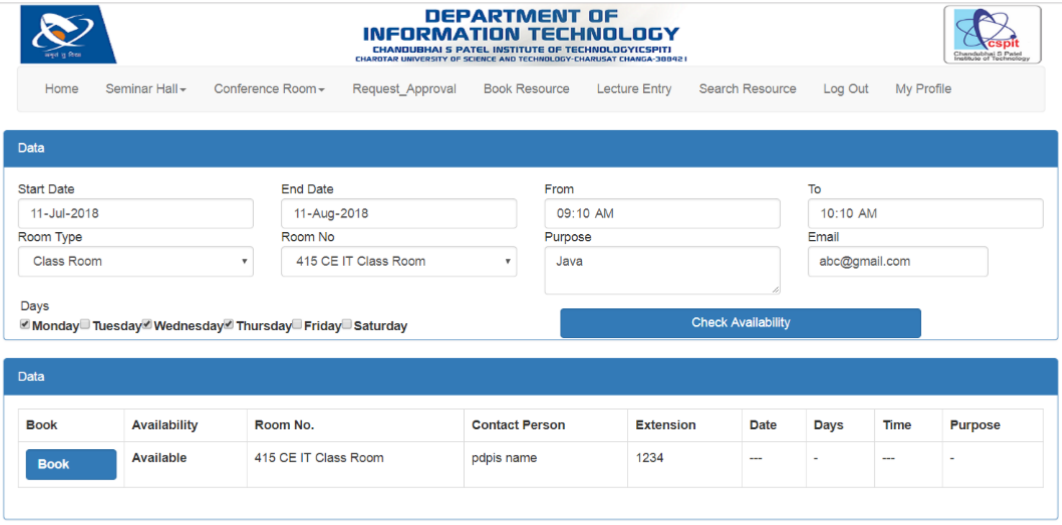
Code Snippets:



```
string finddays(DateTime s_date,DateTime e_date)
{
    string days=null;
    TimeSpan ts= e_date.Subtract(s_date);
    int count=0;
    for (DateTime d = s_date; d.CompareTo(e_date) == -1 && count < 7; d=d.AddDays(1) )
    {
        days = days + d.DayOfWeek.ToString().Remove(3) + ",";
        count++;
    }
    if(count<7)
        days = days + e_date.DayOfWeek.ToString().Remove(3);
    return days;

}
```

Figure 10 | Snippet Resource Book(days)

Snap shot:



Data

Start Date: 11-Jul-2018	End Date: 11-Aug-2018	From: 09:10 AM	To: 10:10 AM
Room Type: Class Room	Room No: 415 CE IT Class Room	Purpose: Java	Email: abc@gmail.com

Days
✓ Monday ☐ Tuesday ✓ Wednesday ✓ Thursday ☐ Friday ☐ Saturday

Data

Book	Availability	Room No.	Contact Person	Extension	Date	Days	Time	Purpose
Book	Available	415 CE IT Class Room	pdpis name	1234	---	-	---	-

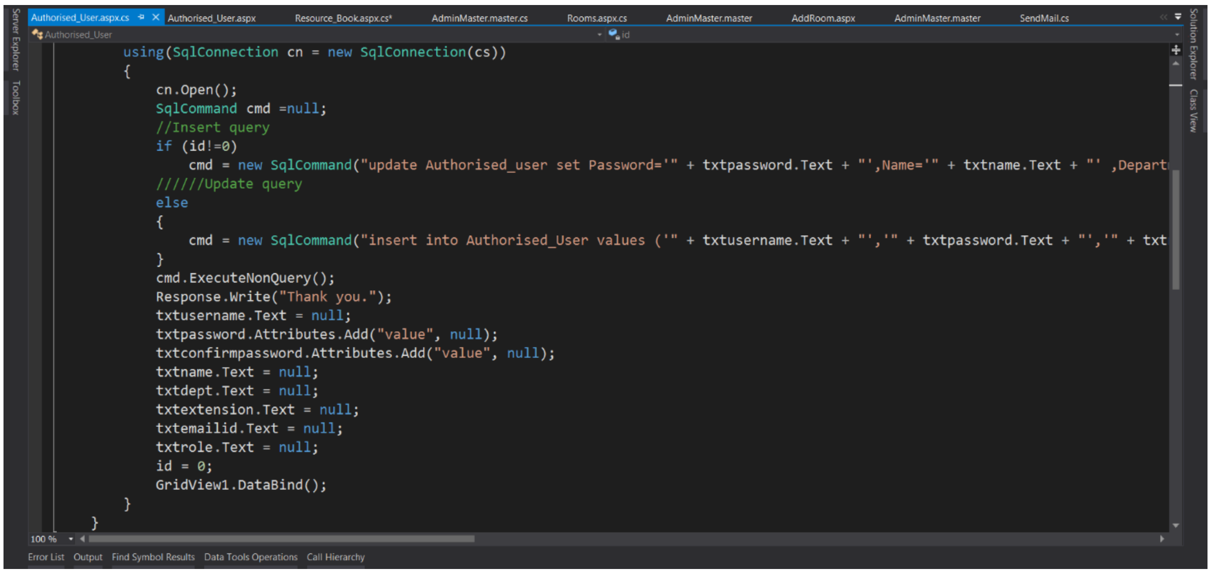
Figure 11 | Lecture Entry

7.MASTER ADMIN USER

➢ Authorised user:

For adding the new authorised user the master user just has to login with his credentials and then by simply clicking Add User he can add the authorised user to the database by adding proper details. Simply he can also update the existing details there too. He can also manage the roles to the user here itself only.

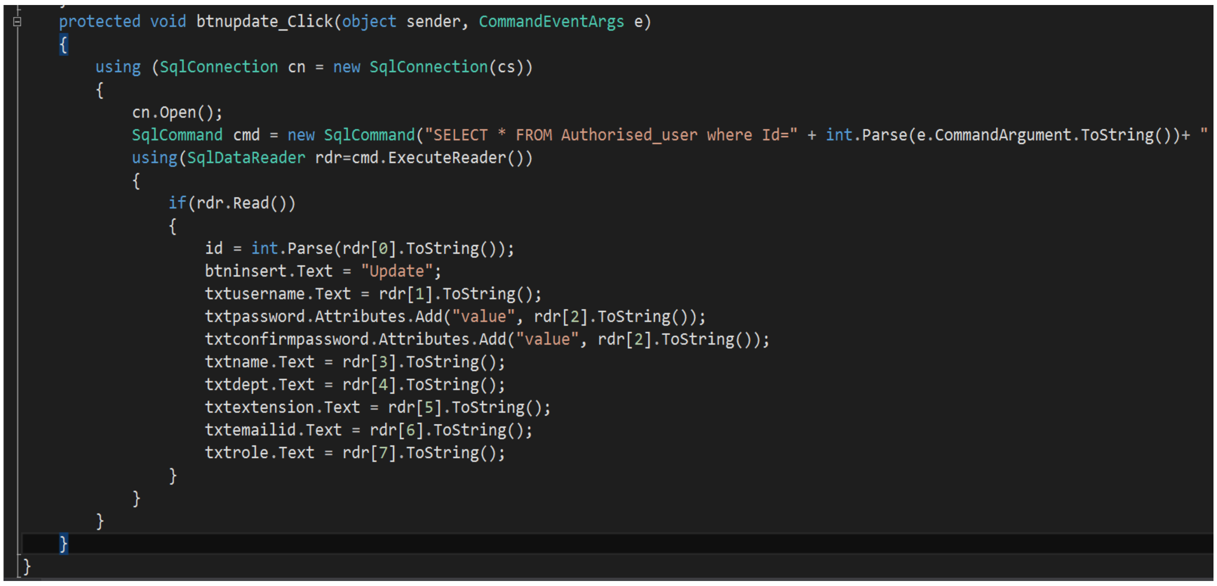
Code Snippet:



```
        using(SqlConnection cn = new SqlConnection(cs))
        {
            cn.Open();
            SqlCommand cmd =null;
            //Insert query
            if (id!=0)
                cmd = new SqlCommand("update Authorised_user set Password='" + txtpassword.Text + "',Name='" + txtname.Text + "' ,Depart
            //////Update query
            else
            {
                cmd = new SqlCommand("insert into Authorised_User values ('" + txtusername.Text + "','" + txtpassword.Text + "','" + txt
            }
            cmd.ExecuteNonQuery();
            Response.Write("Thank you.");
            txtusername.Text = null;
            txtpassword.Attributes.Add("value", null);
            txtconfirmpassword.Attributes.Add("value", null);
            txtname.Text = null;
            txtdept.Text = null;
            txtextension.Text = null;
            txtemailid.Text = null;
            txtrole.Text = null;
            id = 0;
            GridView1.DataBind();
        }
    }
```

Figure 12 | Snippet Authoorised_User

This snippet here includes the btnsubmit method. This method does the functionality of adding and updating the existing user.

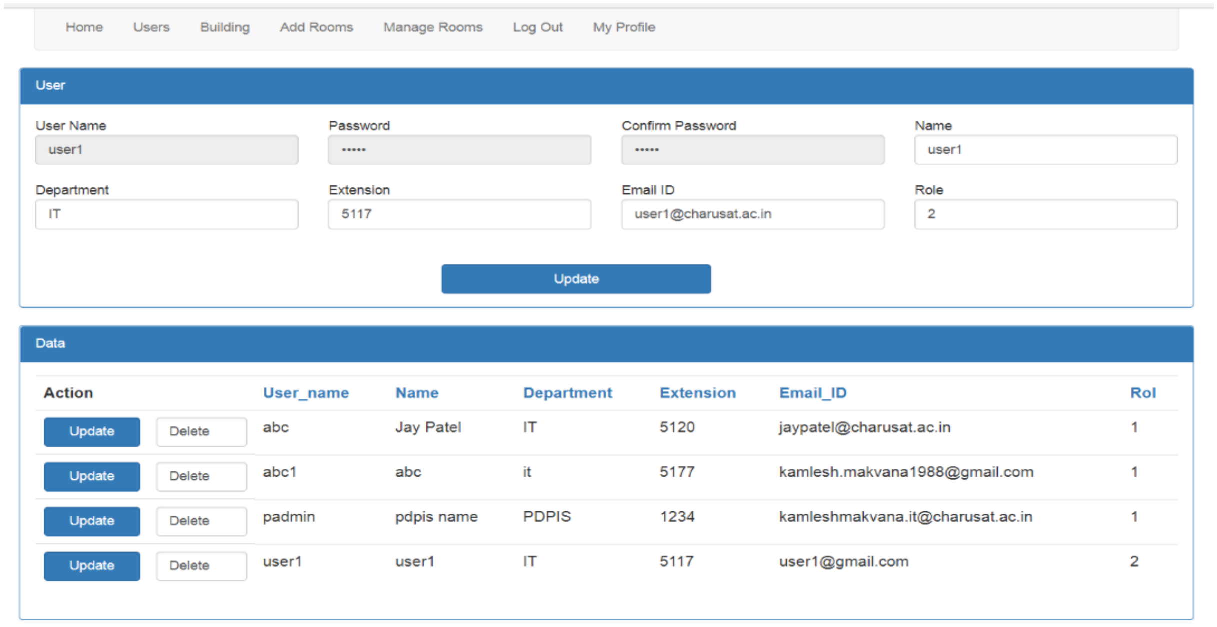


```
    protected void btnupdate_Click(object sender, CommandEventArgs e)
    {
        using (SqlConnection cn = new SqlConnection(cs))
        {
            cn.Open();
            SqlCommand cmd = new SqlCommand("SELECT * FROM Authorised_user where Id=" + int.Parse(e.CommandArgument.ToString())+ "
            using(SqlDataReader rdr=cmd.ExecuteReader())
            {
                if(rdr.Read())
                {
                    id = int.Parse(rdr[0].ToString());
                    btninsert.Text = "Update";
                    txtusername.Text = rdr[1].ToString();
                    txtpassword.Attributes.Add("value", rdr[2].ToString());
                    txtconfirmpassword.Attributes.Add("value", rdr[2].ToString());
                    txtname.Text = rdr[3].ToString();
                    txtdept.Text = rdr[4].ToString();
                    txtextension.Text = rdr[5].ToString();
                    txtemailid.Text = rdr[6].ToString();
                    txtrole.Text = rdr[7].ToString();
                }
            }
        }
    }
}
```

Figure 13 | Snippet Authorised_User(btnupdate)

This snippet here has the btnupdate_Click method this method simply is used to fetch the data from the user selected row and then display the very same data in our form, in order to edit.

Snap shot:

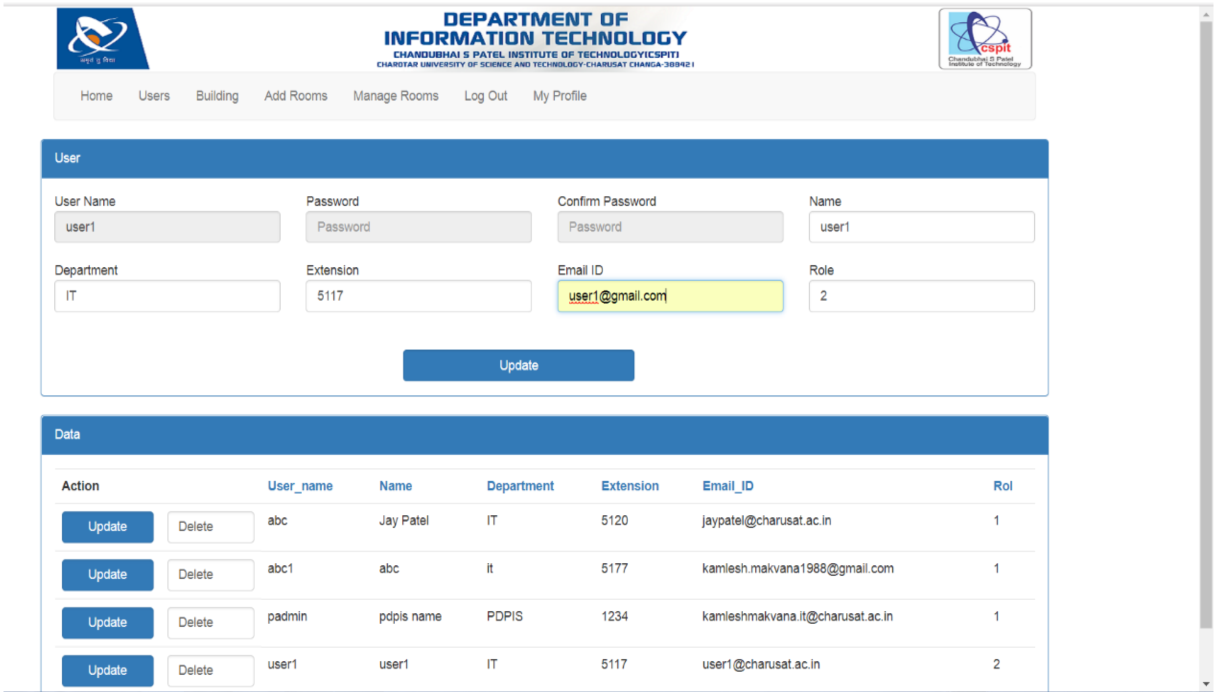


User Name	Password	Confirm Password	Name
user1	•••••	•••••	user1
Department	**Extension**	**Email ID**	**Role**
IT	5117	user1@charusat.ac.in	2

Action		User_name	Name	Department	Extension	Email_ID	Rol
Update	Delete	abc	Jay Patel	IT	5120	jaypatel@charusat.ac.in	1
Update	Delete	abc1	abc	it	5177	kamlesh.makvana1988@gmail.com	1
Update	Delete	padmin	pdpis name	PDPIS	1234	kamleshmakvana.it@charusat.ac.in	1
Update	Delete	user1	user1	IT	5117	user1@gmail.com	2

Figure 14 | Authorised User

For updating the data, you just simply have to select the row of the user which you want to modify the data


DEPARTMENT OF
INFORMATION TECHNOLOGY
CHANDUBHAI S PATEL INSTITUTE OF TECHNOLOGY(CSPIT)
CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY-CHARUSAT CHANGA-388421
cspit
Home
Users
Building
Add Rooms
Manage Rooms
Log Out
My Profile
User
Update
Data

User Name	Password	Confirm Password	Name
user1	Password	Password	user1
Department	**Extension**	**Email ID**	**Role**
IT	5117	user1@gmail.com	2

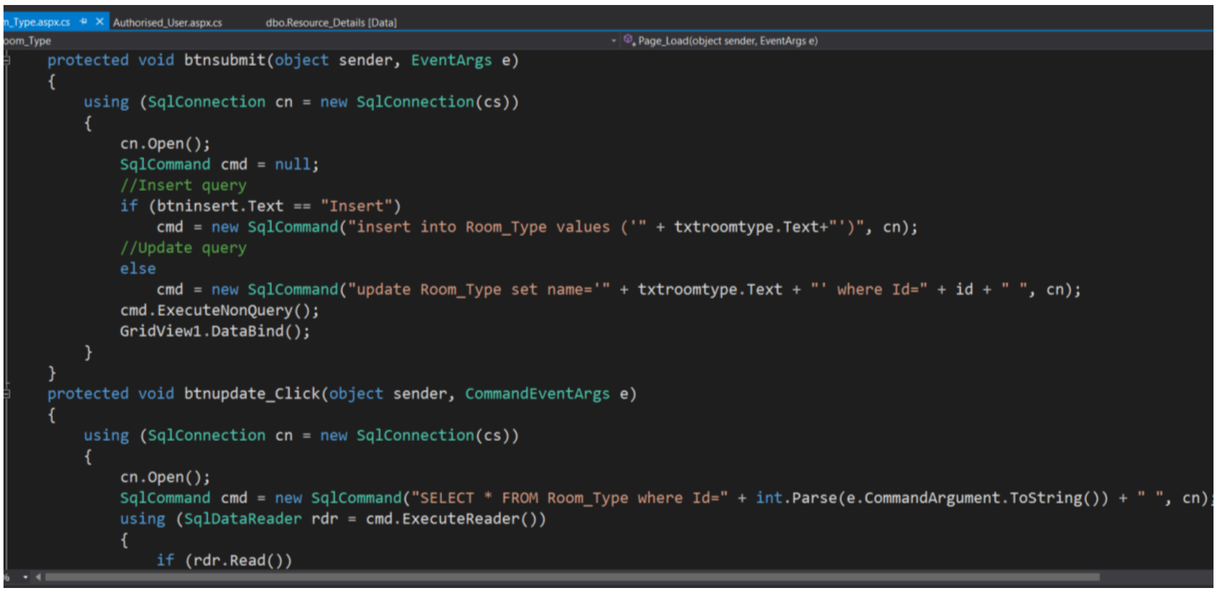
Action		User_name	Name	Department	Extension	Email_ID	Rol
Update	Delete	abc	Jay Patel	IT	5120	jaypatel@charusat.ac.in	1
Update	Delete	abc1	abc	it	5177	kamlesh.makvana1988@gmail.com	1
Update	Delete	padmin	pdpis name	PDPIS	1234	kamleshmakvana.it@charusat.ac.in	1
Update	Delete	user1	user1	IT	5117	user1@charusat.ac.in	2

Figure 15 | Authorised User(Updated)

➢ Room Type:

There may arise a situation when we want to add room type. So this task can be performed using this web form available here.

Code Snippet:

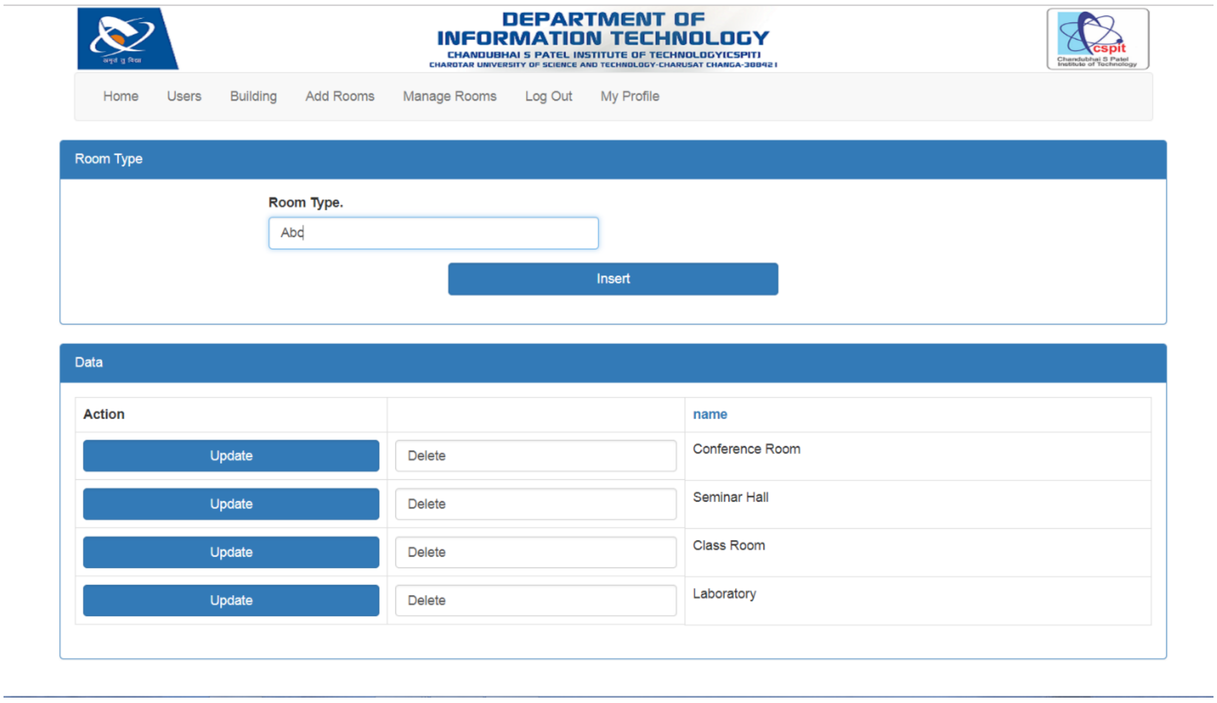


n_Type.aspx.cs
Authorised_User.aspx.cs
dbo.Resource_Details [Data]
oom_Type
Page_Load(object sender, EventArgs e)

```
protected void btnsubmit(object sender, EventArgs e)
{
    using (SqlConnection cn = new SqlConnection(cs))
    {
        cn.Open();
        SqlCommand cmd = null;
        //Insert query
        if (btninsert.Text == "Insert")
            cmd = new SqlCommand("insert into Room_Type values ('" + txtroomtype.Text+"')", cn);
        //Update query
        else
            cmd = new SqlCommand("update Room_Type set name='" + txtroomtype.Text + "' where Id=" + id + " ", cn);
        cmd.ExecuteNonQuery();
        GridView1.DataBind();
    }
}
protected void btnupdate_Click(object sender, CommandEventArgs e)
{
    using (SqlConnection cn = new SqlConnection(cs))
    {
        cn.Open();
        SqlCommand cmd = new SqlCommand("SELECT * FROM Room_Type where Id=" + int.Parse(e.CommandArgument.ToString()) + " ", cn)
        using (SqlDataReader rdr = cmd.ExecuteReader())
        {
            if (rdr.Read())
```

Figure 16 | Snippet Room_Type

Snap shot:



DEPARTMENT OF
INFORMATION TECHNOLOGY
CHANDUBHAI S PATEL INSTITUTE OF TECHNOLOGY(CSPIT)
CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY-CHARUSAT CHANGA-388421
Home
Users
Building
Add Rooms
Manage Rooms
Log Out
My Profile
Room Type

Room Type.

Abc

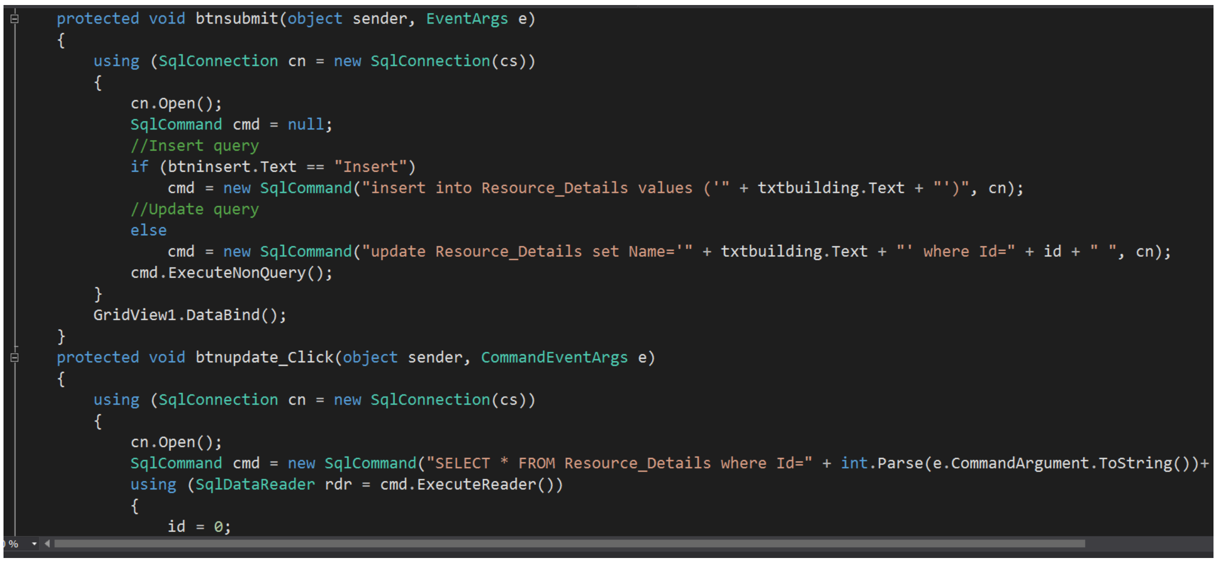
Action		name
Update	Delete	Conference Room
Update	Delete	Seminar Hall
Update	Delete	Class Room
Update	Delete	Laboratory

Figure 17 | Room_Type

➢ Building:

Whenever we have a new building with us and we need to add it to the records then this web form can be helpful.

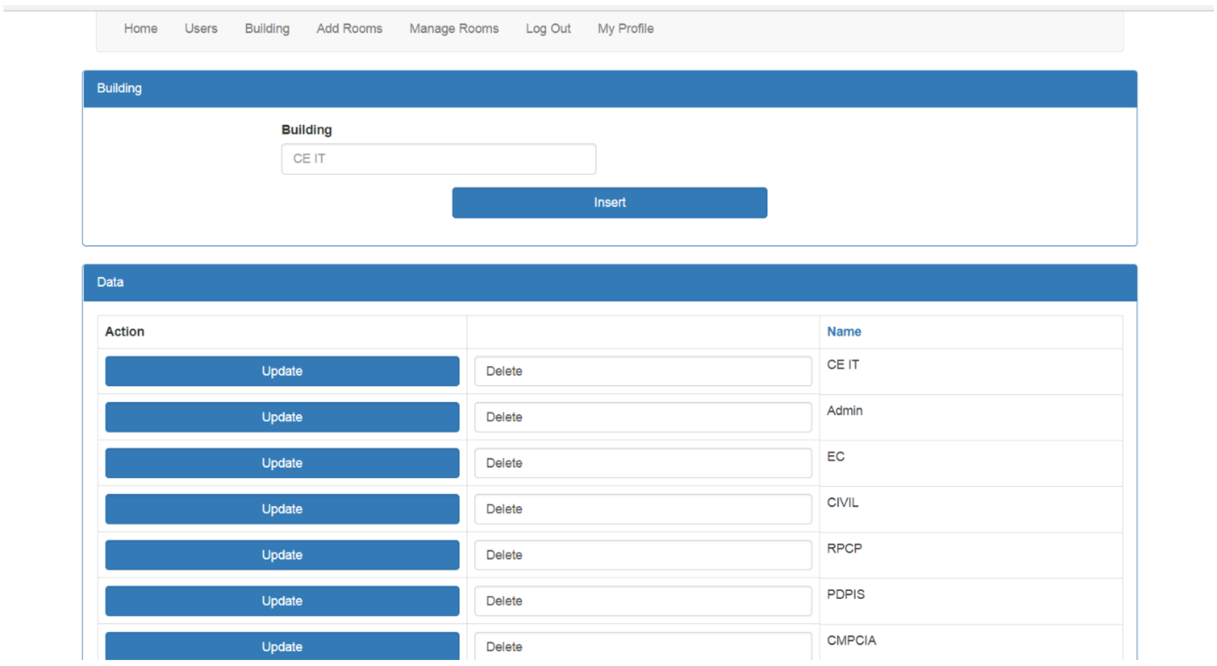
Code Snippet:



```
protected void btnsubmit(object sender, EventArgs e)
{
    using (SqlConnection cn = new SqlConnection(cs))
    {
        cn.Open();
        SqlCommand cmd = null;
        //Insert query
        if (btninsert.Text == "Insert")
            cmd = new SqlCommand("insert into Resource_Details values ('" + txtbuilding.Text + "')", cn);
        //Update query
        else
            cmd = new SqlCommand("update Resource_Details set Name='" + txtbuilding.Text + "' where Id=" + id + " ", cn);
        cmd.ExecuteNonQuery();
    }
    GridView1.DataBind();
}
protected void btnupdate_Click(object sender, CommandEventArgs e)
{
    using (SqlConnection cn = new SqlConnection(cs))
    {
        cn.Open();
        SqlCommand cmd = new SqlCommand("SELECT * FROM Resource_Details where Id=" + int.Parse(e.CommandArgument.ToString())+
        using (SqlDataReader rdr = cmd.ExecuteReader())
        {
            id = 0;
```

Figure 18 | Snippet Building

Snap shot:



Building: CE IT

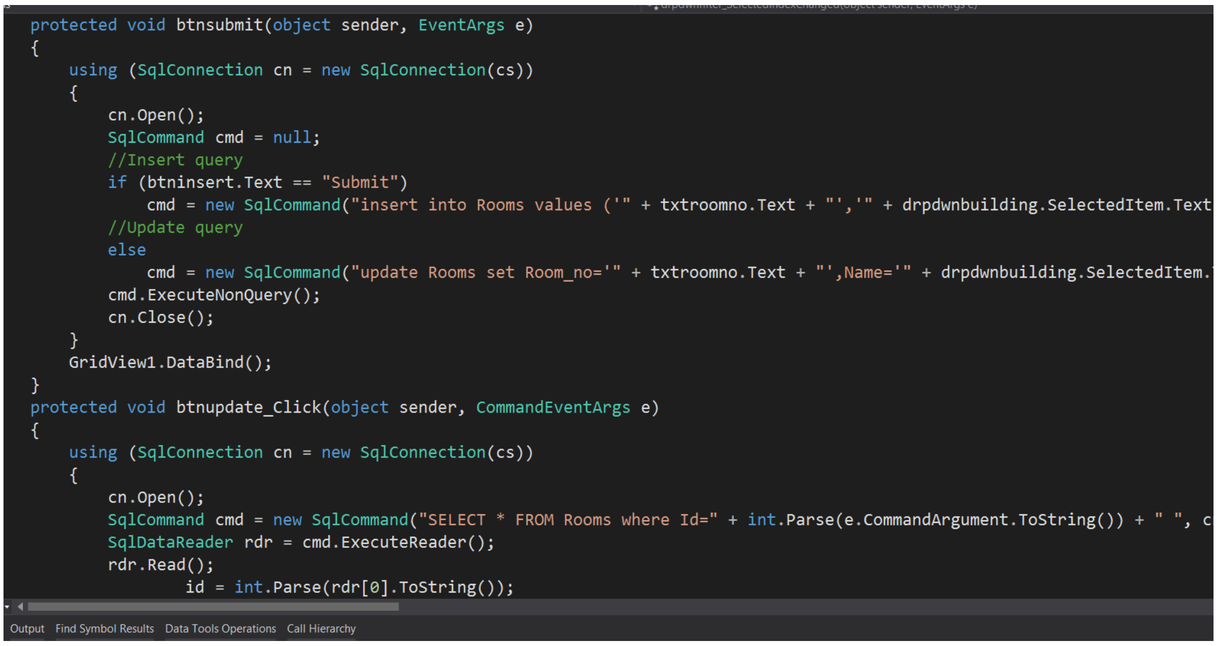
Action		Name
Update	Delete	CE IT
Update	Delete	Admin
Update	Delete	EC
Update	Delete	CIVIL
Update	Delete	RPCP
Update	Delete	PDPIS
Update	Delete	CMPCIA

Figure 19 | Building

➢ Rooms:

Whenever new rooms are to be inserted to records then this web form comes in handy.

Code Snippet:

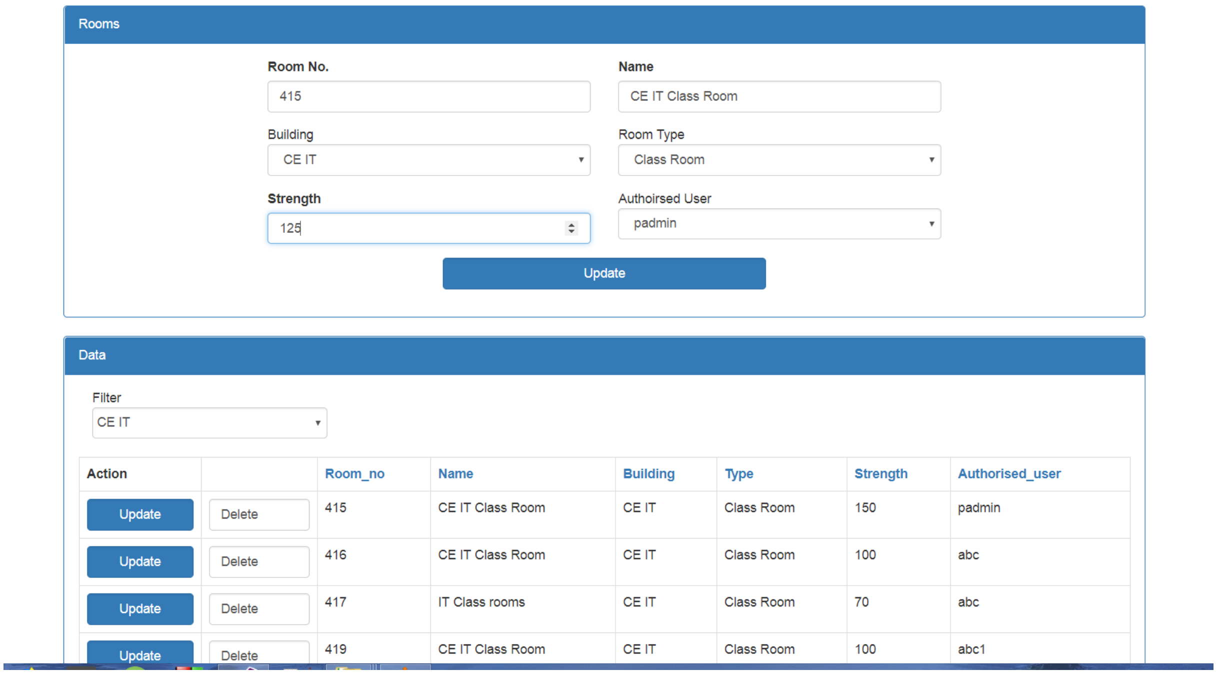


```
protected void btnsubmit(object sender, EventArgs e)
{
    using (SqlConnection cn = new SqlConnection(cs))
    {
        cn.Open();
        SqlCommand cmd = null;
        //Insert query
        if (btninsert.Text == "Submit")
            cmd = new SqlCommand("insert into Rooms values ('" + txtroomno.Text + "','" + drpdwnbuilding.SelectedItem.Text
        //Update query
        else
            cmd = new SqlCommand("update Rooms set Room_no='" + txtroomno.Text + "',Name='" + drpdwnbuilding.SelectedItem.
        cmd.ExecuteNonQuery();
        cn.Close();
    }
    GridView1.DataBind();
}
protected void btnupdate_Click(object sender, CommandEventArgs e)
{
    using (SqlConnection cn = new SqlConnection(cs))
    {
        cn.Open();
        SqlCommand cmd = new SqlCommand("SELECT * FROM Rooms where Id=" + int.Parse(e.CommandArgument.ToString()) + " ", c
        SqlDataReader rdr = cmd.ExecuteReader();
        rdr.Read();
                id = int.Parse(rdr[0].ToString());
```

Output Find Symbol Results Data Tools Operations Call Hierarchy

Figure 20 | Snippet Rooms

Snap shot:



Rooms

Room No.	415
Name	CE IT Class Room
Building	CE IT
Room Type	Class Room
Strength	125
Authoirsed User	padmin

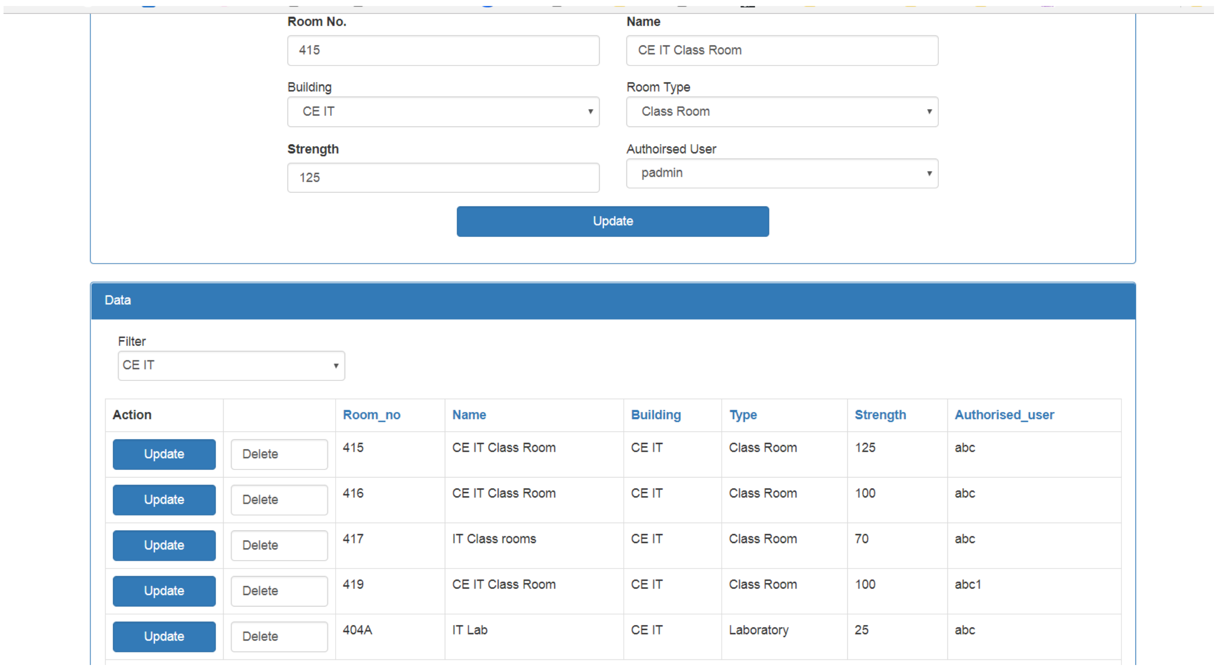
Data

Filter: CE IT

Action		Room_no	Name	Building	Type	Strength	Authorised_user
Update	Delete	415	CE IT Class Room	CE IT	Class Room	150	padmin
Update	Delete	416	CE IT Class Room	CE IT	Class Room	100	abc
Update	Delete	417	IT Class rooms	CE IT	Class Room	70	abc
Update	Delete	419	CE IT Class Room	CE IT	Class Room	100	abc1

Figure 21 | Rooms

Below is the figure after the data being updated.



Room No.	415
Name	CE IT Class Room
Building	CE IT
Room Type	Class Room
Strength	125
Authoirsed User	padmin

Data

Filter: CE IT

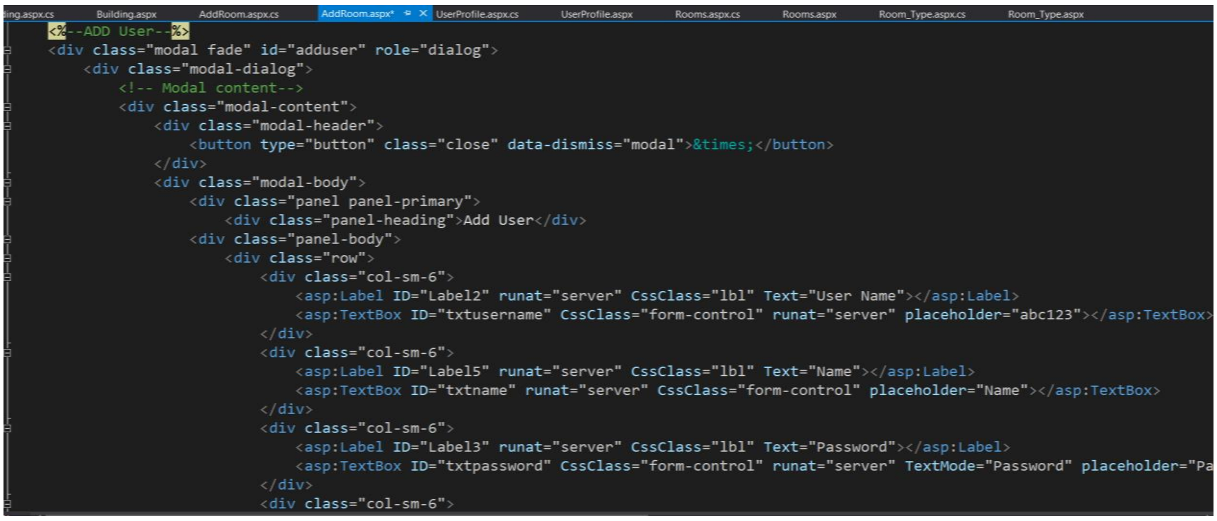
Action		Room_no	Name	Building	Type	Strength	Authorised_user
Update	Delete	415	CE IT Class Room	CE IT	Class Room	125	abc
Update	Delete	416	CE IT Class Room	CE IT	Class Room	100	abc
Update	Delete	417	IT Class rooms	CE IT	Class Room	70	abc
Update	Delete	419	CE IT Class Room	CE IT	Class Room	100	abc1
Update	Delete	404A	IT Lab	CE IT	Laboratory	25	abc

Figure 22 | Rooms(Updated)

➢ AddRooms:

Whenever there are new rooms to be added, this can be done from this web form. In this the master user has to add all the specified and required details of the rooms like Room number, Room type, Building, Strength, Authorised user etc. If the user feels the requirement to add one of these fields like Authorised User, Room Type, Building etc. then they are not supposed to switch to other web form. Because this web form has all that functionalities.

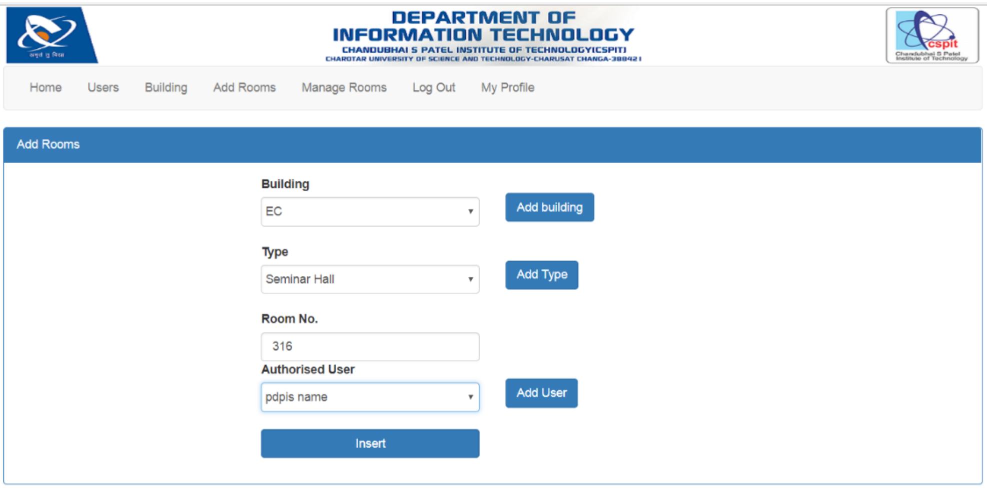
Code Snippet:



```
<%--ADD User--%>
<div class="modal fade" id="adduser" role="dialog">
    <div class="modal-dialog">
        <!-- Modal content-->
        <div class="modal-content">
            <div class="modal-header">
                <button type="button" class="close" data-dismiss="modal">&times;</button>
            </div>
            <div class="modal-body">
                <div class="panel panel-primary">
                    <div class="panel-heading">Add User</div>
                <div class="panel-body">
                    <div class="row">
                        <div class="col-sm-6">
                            <asp:Label ID="Label2" runat="server" CssClass="lbl" Text="User Name"></asp:Label>
                            <asp:TextBox ID="txtusername" CssClass="form-control" runat="server" placeholder="abc123"></asp:TextBox>
                        </div>
                        <div class="col-sm-6">
                            <asp:Label ID="Label5" runat="server" CssClass="lbl" Text="Name"></asp:Label>
                            <asp:TextBox ID="txtname" runat="server" CssClass="form-control" placeholder="Name"></asp:TextBox>
                        </div>
                        <div class="col-sm-6">
                            <asp:Label ID="Label3" runat="server" CssClass="lbl" Text="Password"></asp:Label>
                            <asp:TextBox ID="txtpassword" CssClass="form-control" runat="server" TextMode="Password" placeholder="Pa
                        </div>
                        <div class="col-sm-6">
```

Figure 23 | Snippet AddRooms

Snap shot:


DEPARTMENT OF INFORMATION TECHNOLOGY
CHANDUBHAI S PATEL INSTITUTE OF TECHNOLOGY(CSPIT)
CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY-CHARUSAT CHANGA-388421
Home
Users
Building
Add Rooms
Manage Rooms
Log Out
My Profile
Add Rooms
Add building
Add Type
Add User
Insert

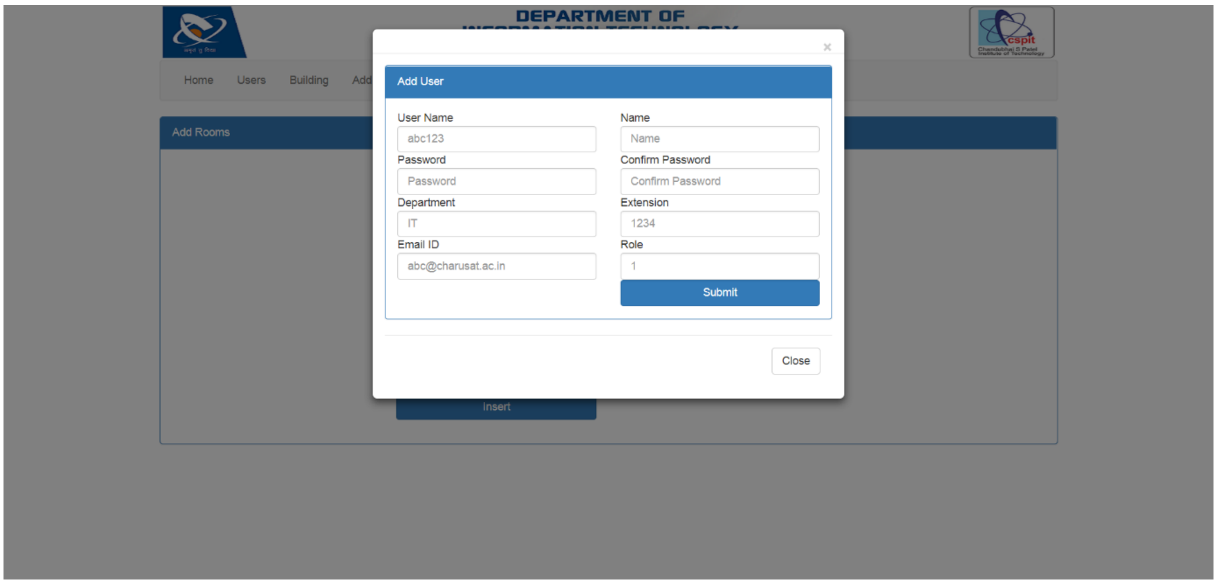
Building: EC

Type: Seminar Hall

Room No.: 316

Authorised User: pdpis name

Figure 24 | AddRooms



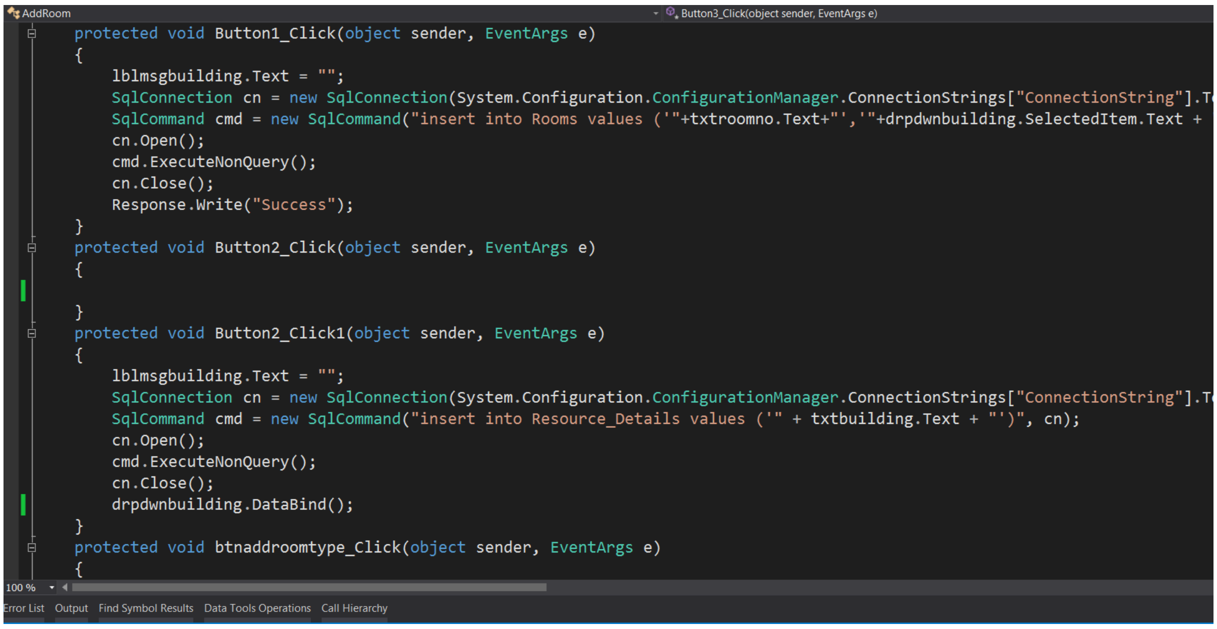
User Name: abc123	Name: Name
Password: Password	Confirm Password: Confirm Password
Department: IT	Extension: 1234
Email ID: abc@charusat.ac.in	Role: 1

Figure 25 | AddRooms(User Pop up)

➢ Forgot Password:

If the authorised user by any chance forges the password, then this web form comes into the role. The user just simply has to enter his username and registered email id. After the verification of the registered username and the email id in database the user gets the message that the password has been mailed to the user. So he can use his email id and fetch the password from there. This thing comes in handy a very few times.

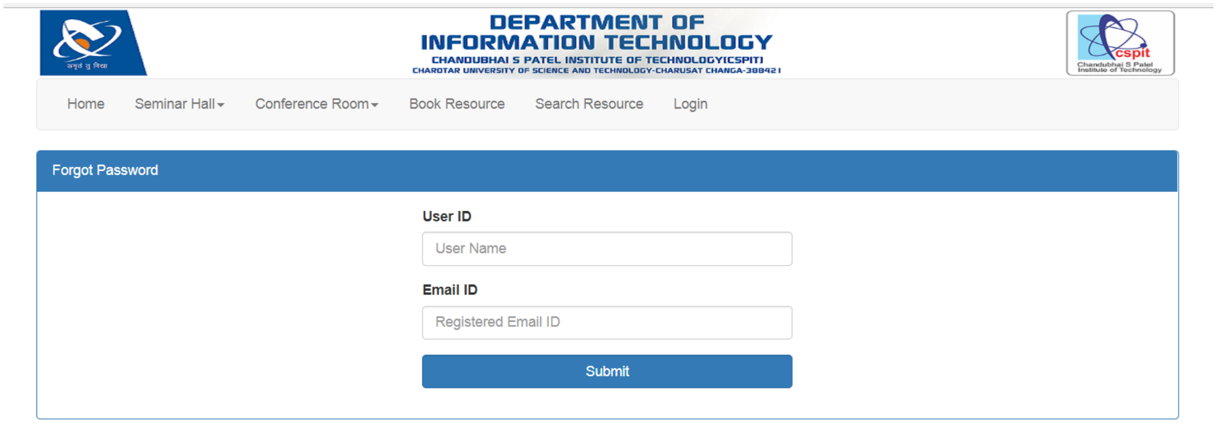
Code Snippet:



```
AddRoom                                                    Button3_Click(object sender, EventArgs e)
    protected void Button1_Click(object sender, EventArgs e)
    {
        lblmsgbuilding.Text = "";
        SqlConnection cn = new SqlConnection(System.Configuration.ConfigurationManager.ConnectionStrings["ConnectionString"].T
        SqlCommand cmd = new SqlCommand("insert into Rooms values ('"+txtroomno.Text+"','"+drpdwnbuilding.SelectedItem.Text +
        cn.Open();
        cmd.ExecuteNonQuery();
        cn.Close();
        Response.Write("Success");
    }
    protected void Button2_Click(object sender, EventArgs e)
    {

    }
    protected void Button2_Click1(object sender, EventArgs e)
    {
        lblmsgbuilding.Text = "";
        SqlConnection cn = new SqlConnection(System.Configuration.ConfigurationManager.ConnectionStrings["ConnectionString"].T
        SqlCommand cmd = new SqlCommand("insert into Resource_Details values ('" + txtbuilding.Text + "')", cn);
        cn.Open();
        cmd.ExecuteNonQuery();
        cn.Close();
        drpdwnbuilding.DataBind();
    }
    protected void btnaddroomtype_Click(object sender, EventArgs e)
    {
```

Figure 26 | Snippet Forgot_Password

Snap shot:


DEPARTMENT OF
INFORMATION TECHNOLOGY
CHANDUBHAI S PATEL INSTITUTE OF TECHNOLOGY(CSPIT)
CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY-CHARUSAT CHANGA-388421
cspit
Home
Seminar Hall
Conference Room
Book Resource
Search Resource
Login

Forgot Password

User ID

User Name

Email ID

Registered Email ID

Submit

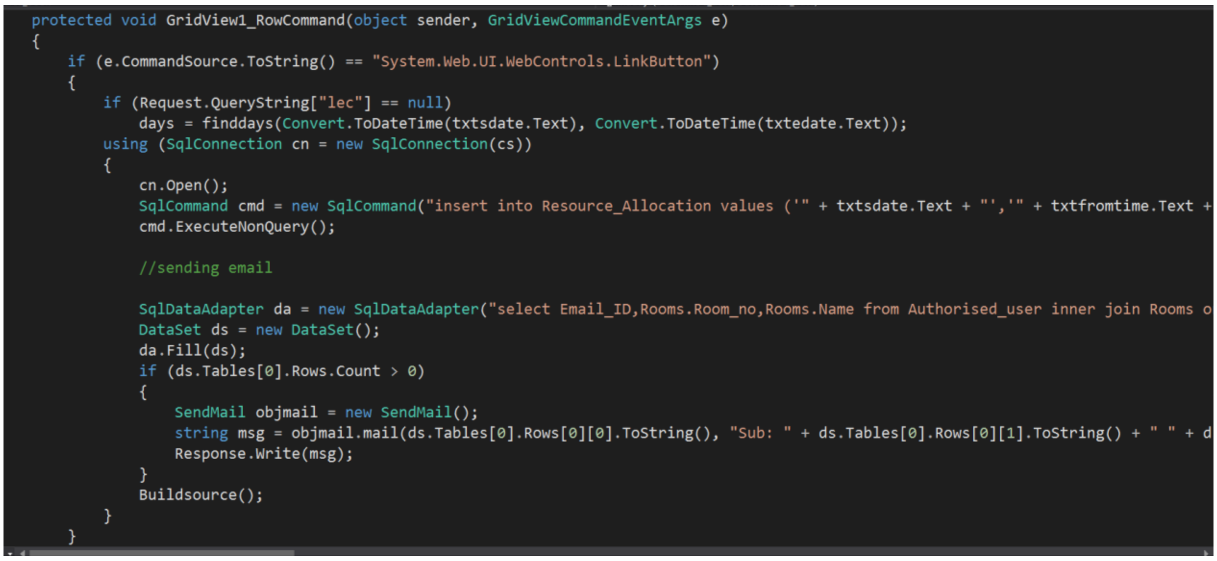
Figure 27 | Forgot_Password

➢ Sending E-mails:

While the process is still carried out manually the user is always supposed to write the mail in order to request for the resource. And also the master user needs to notify if the request was granted or rejected.

So all this can be done through the system generated mail. Which reduces our manual work to zero. In this project the mail service is used for requesting resource, and also to notify the user about the status of his request i.e. whether granted or rejected.

Code Snippet:



```
protected void GridView1_RowCommand(object sender, GridViewCommandEventArgs e)
{
    if (e.CommandSource.ToString() == "System.Web.UI.WebControls.LinkButton")
    {
        if (Request.QueryString["lec"] == null)
            days = finddays(Convert.ToDateTime(txtsdate.Text), Convert.ToDateTime(txtedate.Text));
        using (SqlConnection cn = new SqlConnection(cs))
        {
            cn.Open();
            SqlCommand cmd = new SqlCommand("insert into Resource_Allocation values ('" + txtsdate.Text + "','" + txtfromtime.Text +
            cmd.ExecuteNonQuery();

            //sending email

            SqlDataAdapter da = new SqlDataAdapter("select Email_ID,Rooms.Room_no,Rooms.Name from Authorised_user inner join Rooms o
            DataSet ds = new DataSet();
            da.Fill(ds);
            if (ds.Tables[0].Rows.Count > 0)
            {
                SendMail objmail = new SendMail();
                string msg = objmail.mail(ds.Tables[0].Rows[0][0].ToString(), "Sub: " + ds.Tables[0].Rows[0][1].ToString() + " " + d
                Response.Write(msg);
            }
            Buildsource();
        }
    }
```

Figure 28 | Snippet Sending Mails

8.DESIGN

➢ Database design:

Here we have created the database tables. Various tables for their various intended requirements. The main tables are:

- Resource_Allocation
- Rooms
- Authorised_User
- Room_Type
- Resource Details, etc.

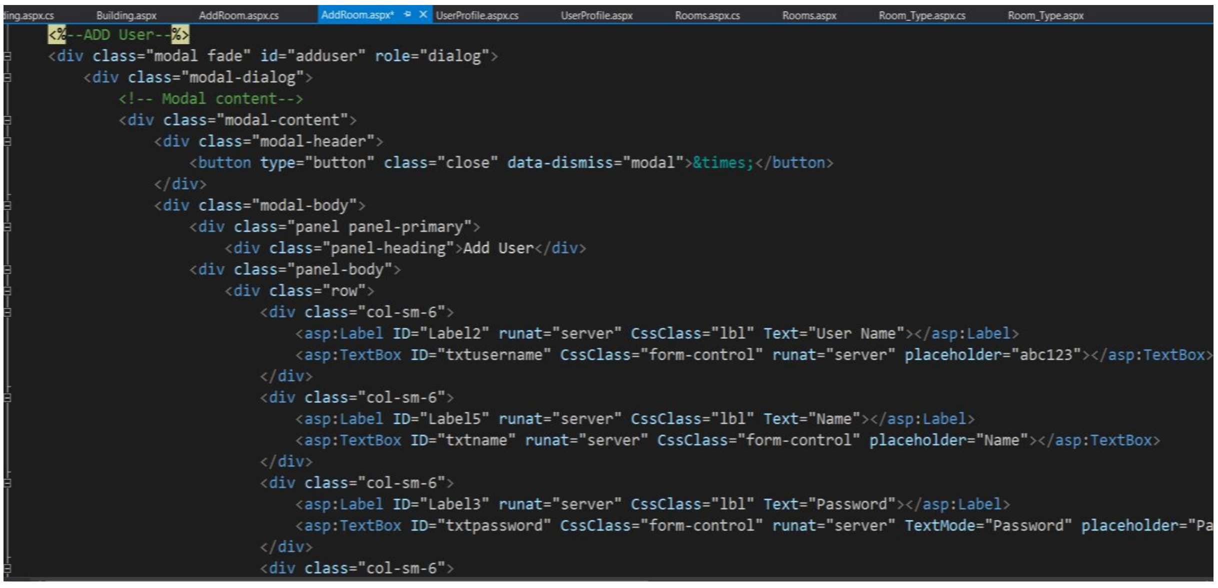
The columns were added, modified, removed etc. after they were created as per the need arises. The foreign key primary key constraints were added to columns as per the requirements.

➢ Page Design:

The page design mainly includes creating the master page and adding CSS properties in various web forms as per the requirements. The master page design includes the header and the navigation bar as per the authorised user. Then it contains the form for user inputs and the Data section for all the data which is collected from the user inputs.

For making this application responsive and also for few designing purposes we have added the Bootstrap into our web forms. Modal pop ups and many other things are added using BootStrap

Code Snippet:



```
<%--ADD User--%>
<div class="modal fade" id="adduser" role="dialog">
    <div class="modal-dialog">
        <!-- Modal content-->
        <div class="modal-content">
            <div class="modal-header">
                <button type="button" class="close" data-dismiss="modal">&times;</button>
            </div>
            <div class="modal-body">
                <div class="panel panel-primary">
                    <div class="panel-heading">Add User</div>
                <div class="panel-body">
                    <div class="row">
                        <div class="col-sm-6">
                            <asp:Label ID="Label2" runat="server" CssClass="lbl" Text="User Name"></asp:Label>
                            <asp:TextBox ID="txtusername" CssClass="form-control" runat="server" placeholder="abc123"></asp:TextBox>
                        </div>
                        <div class="col-sm-6">
                            <asp:Label ID="Label5" runat="server" CssClass="lbl" Text="Name"></asp:Label>
                            <asp:TextBox ID="txtname" runat="server" CssClass="form-control" placeholder="Name"></asp:TextBox>
                        </div>
                        <div class="col-sm-6">
                            <asp:Label ID="Label3" runat="server" CssClass="lbl" Text="Password"></asp:Label>
                            <asp:TextBox ID="txtpassword" CssClass="form-control" runat="server" TextMode="Password" placeholder="Pa
                        </div>
                        <div class="col-sm-6">
```

Figure 29 | BootStrap(Modal Popup User)

9.CONCLUSION

This project meant a new experience for me as I have learned so many new things in the field of Web site designing to meet few of the real time requirements of the project. This summer internship programme provided me with the opportunity to invest my precious time in learning some of the new technologies like ASP.NET, ADO.NET, C#, BootStrap etc. and working with them. I got to experience something new, as all this the things were totally different from whatever I just learned through the books theoretically but I got the first-hand experience to implement all this knowledge I gained in some of the real time project. It also lets us learn some new things which I just can’t learn from books without experiencing them myself while going through the hard times in the project. The internship programme now let me have a bit clear idea of how the things work out in real apart from the theories that we always go through. So, I got to learn many new things and had a great experience creating something totally different from what I was used to.

10.FUTURE WORKS

Almost all the required functionalities are added in this application. Now for making this application more optimized we are planning to add few of the properties and concepts like Stored procedures, caching etc. We may now further focus and make some changes on the frontend too. We were thinking to make some kind of notifications that may display some important things. Or maybe notifications just pops up whenever the user logins.

REFERENCES

1.) C# Kudvenkat
https://www.youtube.com/watch?v=SXmVym6L8dw&list=PLAC325451207E3105

2.) ADO.NET Kudvenkat
https://www.youtube.com/watch?v=aoFDyt8oG0k&list=PL6n9fhu94yhX5dzHunAI2t4kE0kOuv4D7

3.) ASP.NET Kudvenkat
https://www.youtube.com/watch?v=3AYoipyqOkQ&list=PL6n9fhu94yhXQS_p1iHLIftB9Y7Vnxlo

4.) https://getbootstrap.com/

5.) https://www.w3schools.com/bootstrap/default.asp