**A**

**PROJECT REPORT**

**ON**

“GTBPI NOTIFICATION ZONE”

**Submitted in partial fulfillment of the requirements for the award of 3 years**

**“Diploma in Computer Science and Engineering”**

**Submitted by: - Under Guidance of: -**

**Megha Attri S. Pardeep Singh**

**Roll No: 1535051223 HOD (Computer Department) GTBPI, Vasant Vihar**

****

Guru Tegh Bahadur Polytechnic Institute

POORVI MARG, VASANT VIHAR, NEW DELHI-110058

(AFFLIATED TO BOARD OF TECHNICAL EDUCATION, PITAMPURA)

& (APPROVED BY AICTE)

**(BATCH: 2015-2018)**

**DECLARATION**

**I hereby certify that the work carried out in this project is an authentic record of my own. I hereby submit this project, as approved by my supervisor/ project guide. I have read and understood the rules related to the project, inventions and innovations and agree to be bound by them. I also declare that this work has not been submitted earlier for the award of any degree or diploma to the best of my knowledge and belief.**

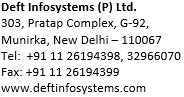
**Megha Attri**

(Diploma in Computer Science & Engineering)

**VI Semester**

**Roll No: 1535051223**

Date: - 10/05/2018GTBPI, Vasant Vihar, New Delhi-58

****

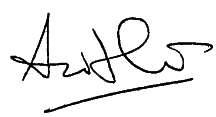
**TO WHOM IT MAY CONCERN**

This is to certify that the project entitled **“GTBPI Notification Zone”** done by **“Ms. Megha Attri”**,D/O- **Mr. Deepak Attri,** pursuing Diploma in Computer Engineering from G.T.B Polytechnic (BTE Delhi), Poorvi Marg, Vasant Vihar, **Roll No “1535051223”** batch (2015-2018) is an authentic work carried out by her at **“Deft Infosystems (P) Ltd.”** under guidance of **“Mr. Bhupender Sharma”** in the organization itself.

Her training activity includes, familiarization to all the software tools and techniques used in the development of Web Applications using ASP.net and C#. She had been majorly involved in development of various small Web and Windows based applications using .NET and C#.

The student has successfully completed two months (From 1st March, 2018 to 1st May, 2018) long training at this Organization/Company. During the period of her training program with us she had been found punctual, hardworking and inquisitive. We wish her every success in her life and career.

**Deft Infosystems Private Limited.**

****

**------------------------------------------------**

**Signature of Authorized Officer**

****

**ACKNOWLEDGMENT**

I would like to thank all my respected teachers (**S. Pardeep Singh, Ms. Ritu Aggrawal, Sh. Satyapal Singh**) for encouraging me and giving me an opportunity to work in the Industry on the project “GTBPI Notification Zone”. This project “GTBPI Notification Zone” gave me an opportunity and pleasure to explore new and challenging field and let me achieve maximum out of my project.

I would also like to say thanks to my project partner **Yash TIwari** (**Roll No: 1535051063**) and our project supervisor at “**Deft Infosystems (P) Ltd.**” **Mr. Bhupender Sharma** who helped me a lot in the whole process of development.

I also wish my gratitude to our honorable Principal, GTBPI “**S. Kulbeer Singh Wazir**” and TPO (Training and placement officer) who helped me at every point and guided me with the right direction to step forward in my career to grab all opportunities during the course of my study in G.T.B Polytechnic from 2015-18 and to under this training.

Signature of the Student Signature

Megha Attri Head of Department

(Roll No. 1535051223) Computer Sc. Engineering

**TABLE OF CONTENTS**

1. **Introduction**

1.1) Purpose 7

1.2) Scope 7

1.3) Terms, Definitions, Acronyms and Abbreviations 8

1.4) Overview of Document 8

2. **General Description**

2.1) Product Perspective

2.1.1) Product Functions 9

2.1.2) Hardware Interface 10

2.1.3) Software Interface 10

2.1.4) Communication Interface 10

3. **System Architectural Design**

3.1) High Level Design Overview 11

3.2) Functional Requirements

3.2.1) Functional Requirements of Admin 12

3.2.2) Functional Requirements of Student 13

3.2.3) Functional Requirements of Faculty 14

3.3) Modules and their Use cases

3.3.1) Use case of Sign Up 15

3.3.2) Use case of Login 16

3.3.3) Use case of Logout 17

3.3.4) Use case of Profile Updation 18

3.3.5) Use case of Query 19

3.3.6) Use case of Post 20

3.4) Structure Approach

3.4.1) Data Flow Diagram 21-27

4. **Data Design**

4.1) Database Description 28-32

4.2) E-R Diagram 33

5. **User Interface Design**

5.1) Description of the user interface

5.1.1) Registration Module 34

5.1.2) Login Module 34

5.1.3) Post 35

5.1.4) Update Module 37

5.1.5) Profile Updation Module 37

5.1.6) Query Module 38

6. **Screen Shots of the Project**

6.1) Login Page 39

6.2) Register-As Page 39

6.3) Register-As Student Page 40

6.4) Register-As Faculty Page 41

6.5) Home Page for Student/Faculty 42

6.6) Post Upload Page for Student/Admin/Faculty 43

6.7) Notification Page for Admin/Student/Faculty 44

6.8) Student Profile Page 44

6.9) Faculty Profile Page 45

6.10) Forgot Password Page 45

6.11) Query Page 46

7. **Code**

7.1) Login Module 47

7.2) Notification Module 49

7.3) Query Module 58

7.4) Signup Module 63

7.5) Home Module 66

7.6) Student Post Module 75

7.7) Student Profile Module 78

8. **Testing Issues** 80

9. **References** 81

1. **Introduction**

**1.1) Purpose**

The purpose of this document is to describe the software requirement in developing the web application “**GTBPI Notification Zone**” which will reinforce and maintain details about: –

* Communication held between faculty members and students
* This web application will be designed for GTBPI members in order to provide them facility of notifying students about class ongoing and solving their related queries.

**1.2) Scope**

“**GTBPI Notification Zone**” is a web-based service which intends to connect students and faculty of GTBPI. It will also help:

* In constructing connection beyond the boundaries of classroom between the users.
* In maintaining an online schedule.
* Users can update themselves with the recent and important information.

The scope of the project can be listed as under:-

* This site aims to provide a special functionality to students to be informed about the followings:

1. Handouts Updations
2. Notes Availability
3. E-books
4. Schedule Updations
5. Assignments
6. Query related to above

* Teachers will be able to add or update different notices to inform the students about the activities going on.
* Administrator will be responsible to update overall database of the website.

Three types of users will be using this website .User can be **Student, Faculty and Administrator.** All the users will have different access and rights granted by this web application

**1.3) Definitions, Acronyms and Abbreviations**

**Title**: **GTBPI Notification Zone**

* Admin- Administrator
* DFD- Data Flow Diagram
* IIS- Internet Information Services
* Asp.Net- Active Server Page Networking
* SQL- Structured Query Language
* E-R Diagram- Entity Relationship Diagram

**1.4) Overview of Document**

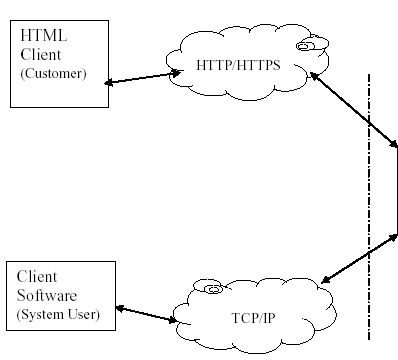
The **rest of the SDS** consists of the following parts: -

* System Architecture Design includes High Level Design, Structure Chart & DFD.
* Data Design includes Database Description.
* Detailed Description of Components.
* User Interface Designs.
* Testing Strategies.

1. **General Description**

**2.1) Product Perspective**

* This website is mainly aimed towards students and faculty of GTBPI.
* This website will be user friendly, and reliable



IIS

SQL Server

**Client-Side Application Server Database Server**

**2.1.1) Product Functions**

**““GTBPI Notification Zone””** will allow:

1. Login to the system through the first page of the application.
2. OTP will be sent to the concerned person about the confirmation of the creation of the account.
3. Users can have their profile.
4. Students can check the post and replies of query updated by faculty.
5. Teachers can broadcast the information through BNZ.
6. Students can query about the teacher’s post.
7. Sign Out from the system.

**2.1.2) Hardware Interface**

Server Side:

* **RAM:** 1GB
* **HDD:** 20 GB or more (Free space excluding data size)
* **Processor:** 1-2 GHz (P4) or onwards

Client Side:

* **RAM:** 512 MB
* **HDD:** 10 GB or more (Free space excluding data size)
* **Processor:** 450 MHz (P2)

**2.1.3) Software Interface**

Server Side:

* **OS:** Windows Server 2007 or onwards
* **Web Server:** IIS 5.0 or onwards with ASP.Net framework

Client Side:

* **OS:** Any OS
* **Browser:** Any browser supporting JavaScript, CSS, xml.

**2.1.4) Communication Interface**

* Client on Internet will be using HTTP/HTTPS protocol.

1. **System Architectural Design**

**3.1) High-level Design Overview**

SQL Server

Web Server

(IIS with ASP.NET run time)

INTERNET

User

1-TIER

3-TIER

2-TIER

**3.2) Functional Requirements**

**3.2.1) Functional Requirements of Admin**

**Admin**

**3.2.2) Functional Requirements of Student**

**Student**

**3.2.3) Functional Requirements of Faculty**

**Faculty**

**3.3) Modules and their Use cases**

**3.3.1) Use Case of Sign Up**

|  |  |
| --- | --- |
| Use Case No | 1 |
| Use Case Name |  |
| Actors: | Students and faculty. |
| Descriptions | This module helps the faculty and  students to create new accounts. |
| Input | Name, smart card ID, course, password, department, semester. |
| Normal Course Events | 1. Users enter their smart card id. 2. Users enter their passwords. 3. User enters other required Information. 4. Users click submit button. 5. System connects to database. 6. A message appears which shows the   Account created applied. |
| Alternative Courses | 1. Same smart card id exists in database 2. Error message appears. 3. Continue with step 1 in the normal course   events. |
| 1. An error may occur during the database   operation.   1. System shows error messages. |
| Output | New user account will be created. |

**Table 1: Use Case of Sign Up Function**

**3.3.2) Use Case of Login**

|  |  |
| --- | --- |
| Use Case No | 2 |
| Use Case Name | Login |
| Actors: | Students, Admin and Faculty. |
| Descriptions | This module helps the admin, faculty and  student to login. |
| Pre-conditions | The user must be a member of the system. |
| Input | Smart Card Id and password. |
| Normal Course Events | 1. Users enter their smart card id. 2. Users enter their password. 3. Users click login button. 4. System connects to database. 5. Homepage displayed. |
| Alternative Courses | 1. User enters incorrect id and/or   password.   1. Error message appears. 2. Continue with step 1 in the normal course   events.   1. An error may occur during the database   operation.   1. System shows error messages. |
| Output | User homepage will be displayed. |

**Table 2: Use Case of Login Function**

**3.3.3) Use Case of Logout**

|  |  |
| --- | --- |
| Use Case No. | 3 |
| Use Case Name | Logout |
| Actors: | Students, Admin and Faculty. |
| Descriptions | This module helps the administrator, faculty and  student to logout. |
| Pre-conditions | The user must be logged in to the website. |
| Normal Course Events | 1. Users click to Logout button. 2. DB connection terminated. 3. The website users Logout successfully. 4. The website will be directed to Login page. |
| Alternative Courses | In case, user accidentally closes the tab then the user will be automatically logout from her/his account but it is recommended to Logout from the account using “Logout” option for security purposes. |
| Output | Website login page will be displayed. |

**Table 3: Use Case of Logout Function**

**3.3.4) Use Case of Profile Updation**

|  |  |
| --- | --- |
| Use Case No | 4 |
| Use Case Name | Profile Updation |
| Actors: | Students, Admin and Faculty. |
| Descriptions | This module helps the faculty, admin and student to  update the information about themselves. |
| Pre-conditions | The user must be logged on to the system. |
| Input | Information to be modified. |
| Normal Course Events | 1. The user must be logged on to the system   which is defined in use case 2.   1. User may go to profile view option. 2. The system retrieves user information from   DB and shows the information on new page.   1. The user can update his/her profile in desired   information field.   1. The user clicks on Update Button. 2. The user profile is updated. |
| Alternative Courses | 1. The system cannot access the DB. 2. The system redirects to the homepage. 3. The system puts a message on the top of the window about the problem. 4. Continue with step 2 in the normal course   events. |
| Output | User Profile Information will be updated. |

**Table 4: Use Case of Profile Updation Function**

**3.3.5) Use Case of Query**

|  |  |
| --- | --- |
| Use Case No | 5 |
| Use Case Name | Ask query |
| Actors: | Students and Faculty |
| Descriptions | This module helps the student to ask query to their faculty. |
| Pre-conditions | The user must be logged on to the system. |
| Input | Query to be asked |
| Normal Course Events | 1. The student must select the post for which   they want to query.  2. The user clicks the Query option.  3. The user types the text in the box provided.  4. The user clicks the Submit Button.  5. The query will be posted. |
| Alternative Courses | 1. The user should enter the query within the   character limit.   1. The user clicks the Submit Button without   entering any query in the textbox provided.   1. Error message appears. 2. Continue with step 3 in the normal course   events. |
| Output | query will be added on the post. |

**Table 5: Use Case of Query Function**

**3.3.6) Use Case of Post**

|  |  |
| --- | --- |
| Use Case No | 6 |
| Use Case Name | Add and delete posts. |
| Actors | Admin, Student and Faculty |
| Descriptions | This module helps the admin, student and faculty to  add or delete the posts. |
| Pre-conditions | The Administrator, student and faculty should have  appropriate rights to add and delete post. |
| Input | Information about the notices. |
| Normal Course Events | 1. The user clicks on POST Link 2. The user enters the information about notices   in the textbox provided and image/files if needed in image box.   1. The user clicks the Submit Button. 2. The notices will be posted with the entered   information.   1. If any inappropriate post has been posted by the user, then the   facility to delete the post can be achieved by Delete Button. |
| Alternative Courses | 1. If the user clicks on Submit button without any information in   notice textbox or in select any file then an error message will appear. |
| Output | Notice will be added or deleted. |

**Table 6: Use Case of Post Function**

**3.4) Structure Approach**

**3.4.1) Data flow diagram**

* DFD is a graphical representation of a system that shows data flows to, from and within the system.
* These are used to depict specific data flows from both physical & logical view point.
* The DFD's are divided into different levels starting from 0th level until we get the final description of system.

**0-Level DFD**

VIEW POST

STUDENT QUERY

LOGIN

PROFILE VIEW

VIEW QUERY/POST

UPDATE DATABASE

SIGN UP

REPLY TO QUERY

UPLOAD POST

LOGIN

SIGN UP

UPLOAD POST

ASK QUERY

VIEW POST

STUDENT QUERY

REPLY TO QUERY

REPLY TO QUERY

LOGIN

UPLOAD/VIEW POST

**1-Level DFD for Student**

UPDATE DETAILS

STORED DETAILS

UPDATE PROFILE

PERSONNELDETAILS

PERSONNELDETAILS

FACULTY TABLE

FACULTY TABLE

VIEW PROFILE

ACCOUNT CREATED

ID AND PASSWORD

VALID/NOT

REPLY TO QUERY

**FACULTY**

ID AND PASSWORD

VALID/NOT VALID

LOGIN DETAILS

VIEW QUERY

DELETE POST

UPLOAD POST

VIEW POST

LOGIN TABLE

REPLY DETAILS

NEW DETAILS

EXISTING DETAILS

DELETE DETAILS

QUERY DETAILS

QUERY TABLE

POST TABLE

**1-Level DFD for Faculty**

UPDATE DETAILS

STORED DETAILS

UPDATE PROFILE

PERSONNELDETAILS

PERSONNELDETAILS

FACULTY TABLE

FACULTY TABLE

VIEW PROFILE

ACCOUNT CREATED

ID AND PASSWORD

VALID/NOT

REPLY TO QUERY

**STUDENT**

ID AND PASSWORD

VALID/NOT VALID

LOGIN DETAILS

VIEW QUERY

DELETE POST

UPLOAD POST

VIEW POST

LOGIN TABLE

REPLY DETAILS

NEW DETAILS

EXISTING DETAILS

DELETE DETAILS

QUERY DETAILS

QUERY TABLE

POST TABLE

**2-Level DFD faculty query**

REPLY DETAILS

DELETE QUERY DETAILS

REPLY OF QUERY

FACULTY

DELETE QUERY

PASSWORD AND ID

VALID/ NOT VALID

QUERY TABLE

SELECT

VIEW QUERY

VALID/ NOT VALID

PASSWORD AND ID

Login table

DETAILS

**2-Level DFD for student query**

ADD DETAILS

DELETE QUERY DETAILS

REPLY OF QUERY

FACULTY

DELETE QUERY

PASSWORD AND ID

VALID/ NOT VALID

QUERY TABLE

SELECT

VIEW QUERY

VALID/ NOT VALID

PASSWORD AND ID

Login table

DETAILS

**2-Level DFD for faculty post**

POST DETAILS

ADD NEW POST

FACULTY

DELETE DETAILS

DELETE OLD POST

PASSWORD AND ID

VALID/ NOT VALID

SELECT

PASSWORD AND ID

VIEW POST

POST TABLE

VALID/ NOT VALID

LOGIN TABLE

POST DETAILS

**2-Level DFD for Student post**

POST DETAILS

ADD NEW POST

STUDENT

DELETE DETAILS

DELETE OLD POST

PASSWORD AND ID

VALID/ NOT VALID

SELECT

PASSWORD AND ID

VIEW POST

POST TABLE

VALID/ NOT VALID

LOGIN TABLE

POST DETAILS

1. **Data design**

**4.1) Database description**

* In this we include, maintain & format Databases and its tables.
* The tables corresponding to each of the entity, holding the information about them are designed.
* The tables have the fields, their description, and their data type as well as integrity constraints.

|  |  |  |  |
| --- | --- | --- | --- |
| **LOGIN TABLE** | | | |
| **User\_id** | Varchar(20) | Not null | **Smart card id of the user** |
| **Password** | Varchar (8) | Not null | **Password of user** |

|  |  |  |  |
| --- | --- | --- | --- |
| **ADMIN TABLE** | | | |
| **Field** | **Type** | **Constraints** | **Description** |
| **Smart\_card\_id** | Varchar(20) | Primary key | **Smart card id of the Admin** |
| **Password** | Varchar(8) | Not null | **Password of Admin** |
| **Sec\_ques1** | Varchar(20) | Not null | **Security Question** |
| **Sec\_ans2** | **Varchar(20)** | **Not Null** | **Security answer** |
| **Sec\_ques1** | Varchar(20) | Not null | **Security Question** |
| **Sec\_ans2** | Varchar(20) | Not null | **Security answer** |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **QUERY TABLE** | | | | | | | | |
| **Field** | | | **Type** | | **Constraints** | | **Description** | |
| **Query\_id** | | | Numeric(20) | | Primary key | | **query id** | |
| **Login\_id** | | | Varchar(20) | | Foreign Key | | **Sender’s user id** | |
| **Content** | | | Varchar(MAX) | | Not null | | **Query content** | |
| **Date\_query** | | | DateTime | | Not Null | | **Date and time of query** | |
| **STUDENT TABLE** | | | | | | |
| **Field** | **Type** | | **Constraints** | | **Description** | |
| **Smart\_card\_id** | Varchar(20) | | Primary key | | **Smart card id of the Student** | |
| **Name** | Varchar(20) | | Not Null | | **name of the Student** | |
| **Course** | Varchar(20) | | Not Null | | **Course of the Student** | |
| **Sem** | Numeric(02) | | Not Null | | **Semester of students** | |
| **Password** | Varchar(8) | | Not null | | **Password of student account** | |
| **Sec\_ques1** | Varchar(20) | | Not null | | **Question for password recovery** | |
| **Sec\_ques2** | Varchar(20) | | Not null | | **Question for password recovery** | |
| **Sec\_ans1** | Varchar(20) | | Not null | | **Answer for security question** | |
| **Sec\_ans2** | Varchar(20) | | Not null | | **Answer for security question** | |

|  |  |  |  |
| --- | --- | --- | --- |
| **FACULTY TABLE** | | | |
| **Field** | **Type** | **Constraints** | **Description** |
| **Smart\_card\_id** | Varchar(20) | Primary key | **Smart card id of the Staff member** |
| **Name** | Varchar(20) | Not Null | **First name of the Staff member** |
| **Password** | Varchar(8) | Not null | **password** |
| **Dept** | Varchar(20) | Not Null | **Department** |
| **Sec\_ques1** | Varchar(20) | Not Null | **Question for password recovery** |
| **Sec\_ques2** | Varchar(20) | Not Null | **Question for password recovery** |
| **Sec\_ans1** | Varchar(20) | Not Null | **answer for password recovery** |
| **Sec\_ans2** | Varchar(20) | Not Null | **answer for password recovery** |

|  |  |  |  |
| --- | --- | --- | --- |
| **POST TABLE** | | | |
| **Field** | **Type** | **Constraints** | **Description** |
| **post\_id** | Numeric(20) | Primary key | **Id of the post** |
| **post\_topic** | Varchar(50) | Not Null | **Topic of the post** |
| **post\_content** | Varchar(MAX) | Null | **Text content of post** |
| **Date\_Post** | DateTime | Not Null | **Date of post** |
| **Login\_id** | Varchar(20) | Foreign Key | **Id of the person who is posting info** |
| **Course** | **Varchar(5)** | **Not Null** | **Post Corse** |
| **Sub** | **Varchar(20)** | **Not Null** | **Subject of post** |
| **Sem** | **Int** | **Not null** | **Semester to which post is intended to be** |
| **Post\_upload** | **Varchar(max)** | **null** | **Images/pdf/doc files** |

**4.2) E-R DIAGRAM**

post

M

Profile update

M

View/ update

View/update

Profile update

1 M 1 1

M

1 1

faculty

student

1 1

Login to system

Can have

1

login

**M**

query

1. **User Interface Design**

**5.1) Description of the user interface**

**5.1.1) Registration Module**

**Description of Component**

This module allows the new user to become the member of the site.

**Interface Description**

**Inputs: -** All user details

**Outputs: -**Message of successful registration and redirected to login page.

**Processing Details**

* Display the form containing fields of user details to be filled by user in order to get registered.
* Check whether the information submitted is proper or not and all fields are properly filled or not user’s smart card id is unique or not.

**5.1.2) Login Module**

**Description of component**

In order to make the system safe from unauthorized access it is protected with user name and password, so that only valid and legitimate users can access the system.

**Interface Description**

**Inputs: -** Smart Card ID and Password

**Outputs: -** The Home page will open up.

**Processing Details**

* Display the form with smart card id and password field.
* Mark all the fields as compulsory.
* Check whether the user’s smart card id is valid or not, if yes check whether password matches or not.
* Open the main form after clicking Login button after validation checking.

**5.1.3) POST**

**Description of components**

This module is used to view, add and delete post according to user authorization.

* **ADD POST**

**Description**

Admin, faculty member and students can add the post.

**Interface Description**

**Input: -** Post topic, Post detail, date of issue, concerned course with semester, subject and pdf file/images(optional)

**Output: -** New post

**Processing Details**

Admin, Faculty members and student can click post option to add any new post

* **VIEW NOTICES**

**Description**

Admin, staff member and student can view the notice.

**Interface Description**

**Input: -** All post will be displayed at the home page.

**Output: -** View notice

**Processing Details**

All user’s after login can view all the posts at their home page.

* **DELETE NOTICES**

**Description**

* Faculty member and students can delete the post that are uploaded by them only.
* Admin can delete any post.

**Interface Description**

**Input: -** Notice ID

**Output: -** Noticeis deleted.

**Processing Details**

Admin, faculty members and students can click on notice subject to open the notice and then use delete option to delete any existing notice.

**5.1.4) Update module**

**Description of components**

This module is used to update the database by Admin.

**Interface Description**

**Input: -** 1. Smart card id of the user whose account is to be

delete**.**

2**.**Post id of the post to be deleted.

3**.** Query id of the query to be deleted.

**Output: -** Delete the required details from the database and

display a message for delete successfully.

**Processing Details**

First admin enters the desired information which is to be modified. And then submission of changes is done.

**5.1.5) Profile Updation module**

**Description of components**

This module is used to update the profile information of the user.

**Interface Description**

**Input: -** User can change any field information.

**Output: -** Updated information.

**Processing detail**

* User enters the required new information
* Verification is done on input data
* If it is invalid, an error message is displayed
* Otherwise users profile information is updated.

**5.1.6) Query module**

**Description of components**

This module is used to submit query on any post.

**Interface Description**

**Input: -** User enter query to be asked.

**Output: -** Message of query submission.

**Processing detail**

* User enters the query they want to ask in the area provided.
* Verification is done whether query is written before clicking on submit button.
* If not then, user is required to enter query.
* Message of successful submission is displayed.

1. **Screen Shots of the Project**

**6.1) Login Page**



**6.2) Register-As page**



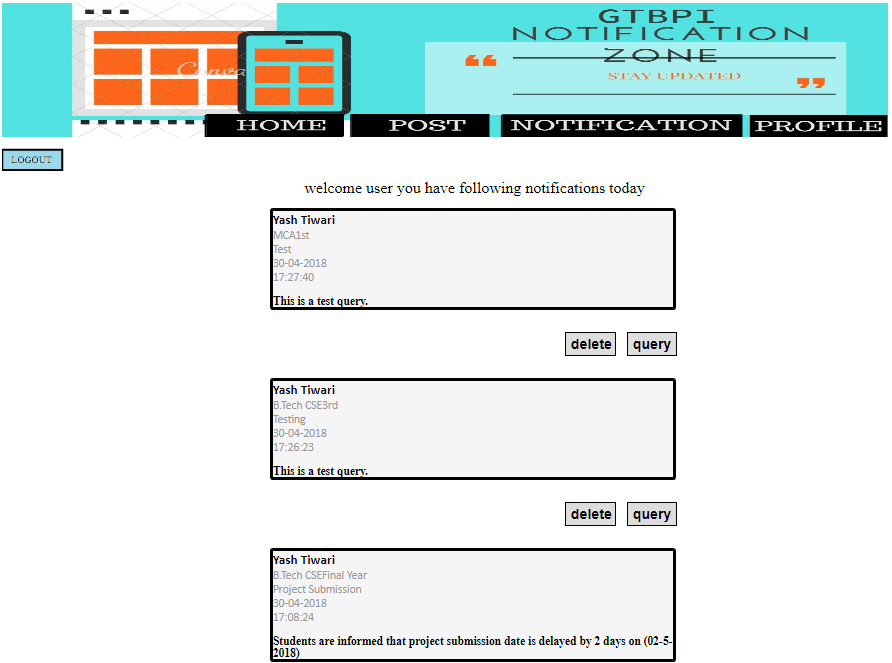
**6.3) Register as student page**



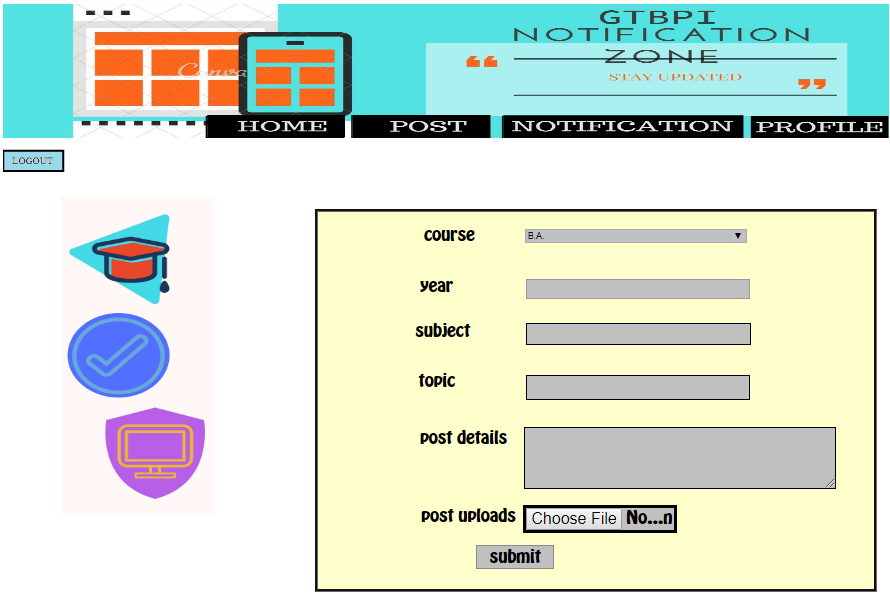
**6.4) Register as faculty page**



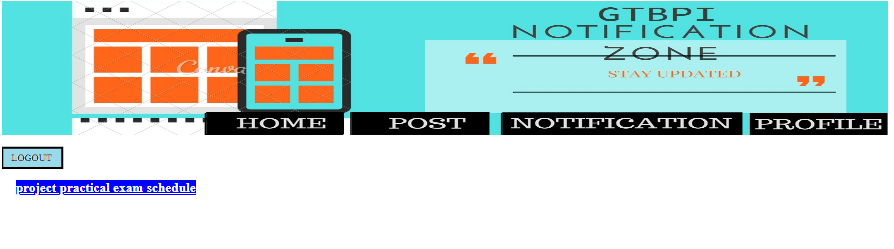
**6.5) Home page for student/faculty**



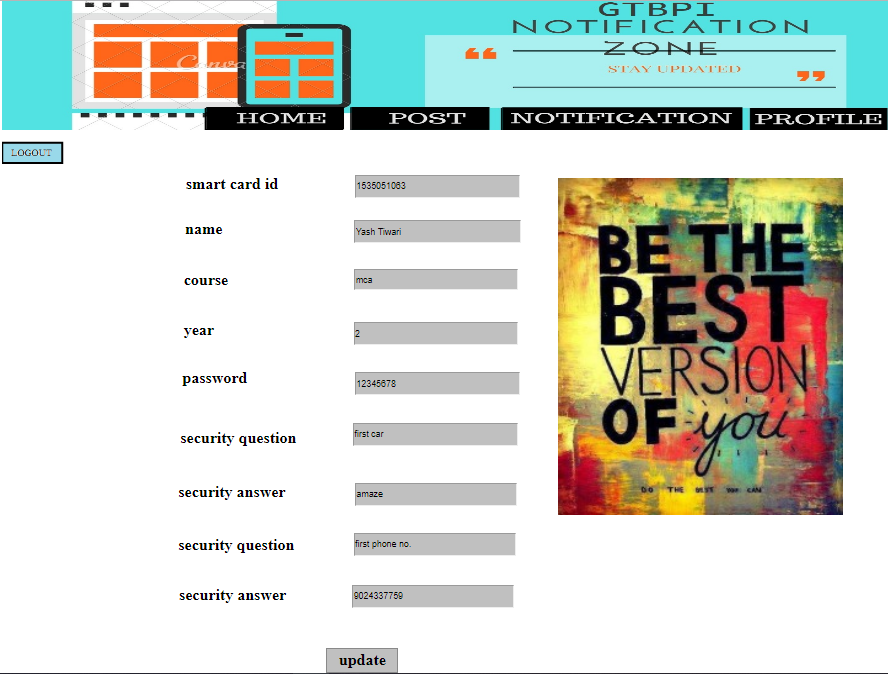
**6.6) Post upload page for Student/Admin/Faculty**



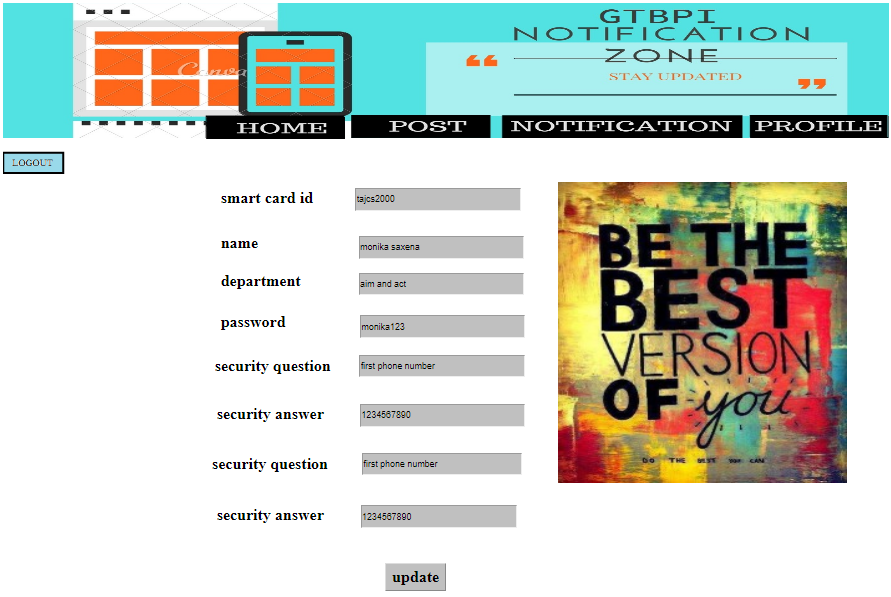
**6.7) Notification page for Admin/Student/Faculty**



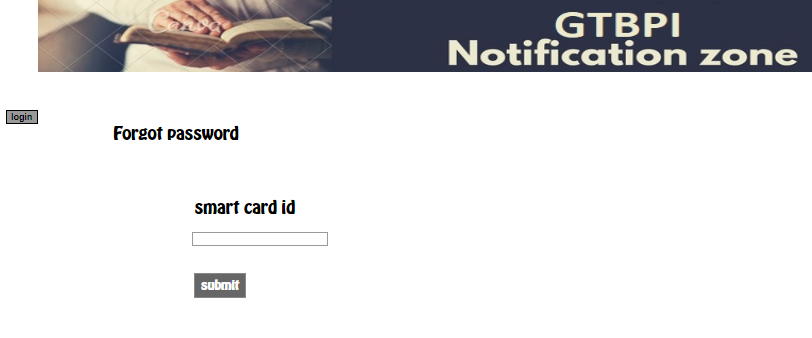
**6.8) Student profile page**

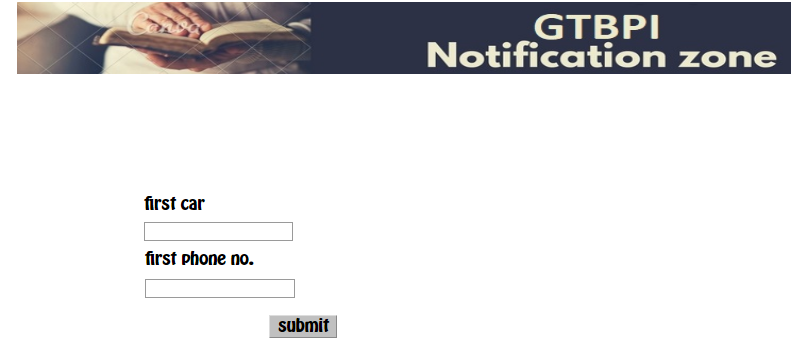


**6.9) Faculty profile page**

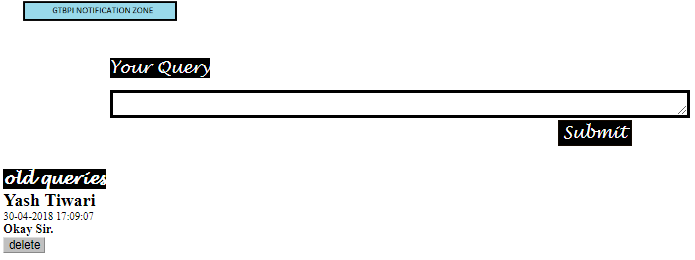


**6.10) Forgot password page**





**6.11) Query page**



1. **Code**

**7.1) Login Module**

using System;

using System.Data.SqlClient;

using System.Web.UI;

namespace GTBPINotificationZone

{

public partial class login : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

}

SqlConnection con = new SqlConnection("Server=.\\SQLEXPRESS; AttachDbFilename=|DataDirectory|\\bnz.mdf;Integrated Security = True ; User Instance = True; trusted\_Connection = yes");

protected void ImageButton1\_Click(object sender, ImageClickEventArgs e)

{

Session["s1"] = "1";

con.Open();

SqlCommand cmds = new SqlCommand("select count(\*) from login where smart\_card\_id='" + loginid.Text + "' and password='" + password.Text + "'", con); //for counting number of user with given id nd password

SqlDataReader drs;

drs = cmds.ExecuteReader();

string s = "0";

while (drs.Read())

{

s = drs[0].ToString();// count value

}

con.Close();

if (s != "1") //if not valid user

{

Label1.Text = "enter correct smart card id if exiting user otherwise register yourself with the link below";

}

else //for valid uder

{

con.Open();

SqlCommand cmd = new SqlCommand("select \* from login where smart\_card\_id='" + loginid.Text + "' and password='" + password.Text + "'", con);

SqlDataReader dr;

dr = cmd.ExecuteReader();

string who = "";

string id = "";

while (dr.Read())

{

who = dr[2].ToString();//either teacher or student

id = dr[0].ToString();

}

Session["login\_as"] = who;

con.Close();

if (who == "student")

{

if (s == "1")

{

Session["id"] = id;

Response.Redirect("studhome.aspx");

}

else

{

Label1.Text = "invalid id or password";

//Response.Redirect("login.aspx");

}

}

if (who == "faculty")

{

if (s == "1")

{

Session["id"] = id;

Response.Redirect("teachhome.aspx");

}

else

{

Label1.Text = "invalid id or password";

//Response.Redirect("login.aspx");

}

}

}

}

}

}

**7.2) Notification Module**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Net;

namespace GTBPINotificationZone

{

public partial class notify : System.Web.UI.Page

{

SqlConnection con = new SqlConnection("server=.\\SQLEXPRESS;AttachDbFilename=|DataDirectory|\\bnz.mdf;Integrated Security=True;MultipleActiveResultSets=True;User Instance=True;trusted\_connection=yes ");

protected void Page\_Load(object sender, EventArgs e)

{

con.Open();

string s = Session["postid1"].ToString();

string s1, s3, s4, id1;

string[] d\_time;

string[] ext;

string s2 = "0";

SqlCommand cmd1, cmd2;

SqlDataReader dr1, dr2;

SqlCommand cmd = new SqlCommand("select \* from post where post\_id='"+s+"' ", con);

SqlDataReader dr = cmd.ExecuteReader();

while (dr.Read())

{

id1 = dr[0].ToString();

s1 = dr[4].ToString();

//outer panel of home page.

Panel outer = new Panel();

outer.BorderStyle = BorderStyle.Groove;

outer.Style["border-radius"] = "5px";

outer.BorderColor = Color.Black;

outer.Style["width"] = "600px";

outer.BackColor = Color.WhiteSmoke;

Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp"));

Panel1.Controls.Add(outer);

//inner subpanel1

Panel inner1 = new Panel();

inner1.BackColor = Color.WhiteSmoke;

//inner subpanel2

Panel inner2 = new Panel();

inner2.BackColor = Color.WhiteSmoke;

//controls to be uploaded.

Label name = new Label(); //label for name of the user who posted the info.

Label course = new Label(); //course for which the info is intended.

Label sem = new Label(); //semester of the course

Label sub = new Label(); //subject of which info is posted

Label date1 = new Label(); //date at which info is posted

Label time1 = new Label(); //time at which info is posted

Label content = new Label(); //content of the post

//fatching name of the user who posted the info

name.Font.Size = 15;

name.Font.Bold = true;

name.Font.Name = "Calibri";

name.Style["TextAlign"] = "top\_left";

cmd1 = new SqlCommand("select login\_as from login where smart\_card\_id='" + s1 + "'", con);

dr1 = cmd1.ExecuteReader();

while (dr1.Read())

{

s2 = dr1[0].ToString();

}

if (s2 == "0")

{

name.Text = "admin";

}

if (s2 == "student")

{

cmd2 = new SqlCommand("select name from student where smart\_card\_id='" + s1 + "'", con);

dr2 = cmd2.ExecuteReader();

while (dr2.Read())

{

name.Text = dr2[0].ToString();

}

}

if (s2 == "faculty")

{

cmd2 = new SqlCommand("select name from faculty where smart\_card\_id='" + s1 + "'", con);

dr2 = cmd2.ExecuteReader();

while (dr2.Read())

{

name.Text = dr2[0].ToString();

}

}

outer.Controls.Add(name);

outer.Controls.Add(inner1);

outer.Controls.Add(inner2);

//fatching course,semester,subject,date,topic of the post.

course.Font.Size = 13;

course.Font.Name = "Calibri";

course.Text = dr[2].ToString();

course.Style["color"] = "#999999";

sem.Font.Size = 13;

sem.Style["color"] = "#999999";

sem.Font.Name = "Calibri";

sem.Text = dr[8].ToString();

sub.Font.Size = 13;

sub.Font.Name = "Calibri";

sub.Text = dr[7].ToString();

sub.Style["color"] = "#999999";

d\_time = dr[3].ToString().Split(' ');

date1.Font.Size = 13;

date1.Font.Name = "Calibri";

date1.Text = d\_time[0];

date1.Style["color"] = "#999999";

time1.Font.Size = 13;

time1.Font.Name = "Calibri";

time1.Text = d\_time[1];

time1.Style["color"] = "#999999";

inner1.Controls.Add(course);

inner1.Controls.Add(sem);

inner1.Controls.Add(new LiteralControl("<br/>"));

inner1.Controls.Add(sub);

inner1.Controls.Add(new LiteralControl("<br/>"));

inner1.Controls.Add(date1);

inner1.Controls.Add(new LiteralControl("<br/>"));

inner1.Controls.Add(time1);

inner1.Controls.Add(new LiteralControl("<br/>"));

//content and uploads of post.

s3 = dr[5].ToString();

s4 = dr[6].ToString();

if ((s3 == "") && (s4 != "null")) //when file is uploaded

{

ext = s4.Split('.');

if ((ext[1] == "jpg") || (ext[1] == "jpeg") || (ext[1] == "png")) //download button for images

{

System.Web.UI.WebControls.Image uploads = new System.Web.UI.WebControls.Image(); //images if any

uploads.Style["width"] = "275px";

uploads.Style["height"] = "183px";

uploads.ImageUrl = "~\\uploads\\" + s4;

inner2.Controls.Add(uploads);

inner2.Controls.Add(new LiteralControl("<br/>"));

//download button

Button download1 = new Button();

download1.Text = "Download Image";

download1.ID = id1 + " c";

download1.Font.Size = 14;

download1.BorderStyle = BorderStyle.Groove;

download1.Style["width"] = "150px";

download1.Style["height"] = "36px";

download1.Click += new EventHandler(this.download1\_Click);

inner2.Controls.Add(download1);

inner2.Controls.Add(new LiteralControl("<br/>"));

}

else

if ((ext[1] == "pdf") || (ext[1] == "docx") || (ext[1] == "pptx")) //download button for text files

{

//download button

inner2.Controls.Add(new LiteralControl("<br/>"));

LinkButton download = new LinkButton();

download.Text = dr[1].ToString();

download.Font.Size = 14;

download.ID = "Download from here -> "+id1 + " b";

download.Click += new EventHandler(this.download\_Click);

inner2.Controls.Add(download);

inner2.Controls.Add(new LiteralControl("<br/>"));

}

}

else

if ((s3 != "") && (s4 != "null"))// file and content are uploaded

{

content.Text = s3;

content.Font.Size = 13;

content.Font.Bold = true;

content.Style["color"] = "Black";

content.Style["width"] = "520px";

content.Style["height"] = "180px";

inner2.Controls.Add(content);

inner2.Controls.Add(new LiteralControl("<br/>"));

ext = s4.Split('.');

if ((ext[1] == "jpg") || (ext[1] == "jpeg") || (ext[1] == "png")) //download button for images

{

System.Web.UI.WebControls.Image uploads = new System.Web.UI.WebControls.Image(); //images if any

uploads.Style["width"] = "275px";

uploads.Style["height"] = "183px";

uploads.ImageUrl = "~\\uploads\\" + s4;

inner2.Controls.Add(uploads);

inner2.Controls.Add(new LiteralControl("<br/>"));

//download button

Button download1 = new Button();

download1.Text = "Download Image";

download1.ID = id1 + " c";

download1.BorderStyle = BorderStyle.Groove;

download1.Style["width"] = "150px";

download1.Style["height"] = "36px";

download1.Font.Size = 14;

download1.Click += new EventHandler(this.download1\_Click);

inner2.Controls.Add(download1);

inner2.Controls.Add(new LiteralControl("<br/>"));

}

else

if ((ext[1] == "pdf") || (ext[1] == "docx") || (ext[1] == "pptx")) //download button for text files

{

//download button

inner2.Controls.Add(new LiteralControl("<br/>"));

LinkButton download = new LinkButton();

download.Text ="Download from here -> "+ dr[1].ToString();

download.ID = id1 + " b";

download.Font.Size = 14;

download.Style["width"] = "28px";

download.Style["height"] = "15px";

download.Click += new EventHandler(this.download\_Click);

inner2.Controls.Add(download);

inner2.Controls.Add(new LiteralControl("<br/>"));

}

}

else

{

content.Text = s3; //when only text is there

content.Font.Size = 13;

content.Font.Bold = true;

content.Style["color"] = "Black";

content.Style["width"] = "520px";

content.Style["height"] = "180px";

inner2.Controls.Add(content);

}

//button for query on post

Button query = new Button();

query.Style["width"] = "76px";

query.Style["height"] = "36px";

query.Text = "query";

query.Font.Bold = true;

query.BorderColor = Color.Black;

query.Font.Size = 16;

query.ID = id1 + " a";

query.Click += new EventHandler(this.query\_Click);

query.BorderStyle = BorderStyle.Groove;

Panel1.Controls.Add(new LiteralControl("<br/>"));

Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp"));

Panel1.Controls.Add(query);

Panel1.Controls.Add(new LiteralControl("<br/>")); Panel1.Controls.Add(new LiteralControl("<br/>"));

}

con.Close();

}

protected void query\_Click(object sender, EventArgs e)

{

Button btn1 = (Button)sender;

string[] s = btn1.ID.Split(' ');

int pid = Convert.ToInt32(s[0]);

Session["postid"] = s[0];

Response.Redirect("queries.aspx");

}

protected void download\_Click(object sender, EventArgs e)

{

LinkButton lbtn = (LinkButton)sender;

string[] s = lbtn.ID.Split(' ');

int pid = Convert.ToInt32(s[0]);

string s1 = "";

con.Open();

SqlCommand cmd = new SqlCommand("select post\_uploads from post where post\_id='" + pid + "'", con);

SqlDataReader dr = cmd.ExecuteReader();

while (dr.Read())

{

s1 = "~//uploads//" + dr[0].ToString();

}

con.Close();

try

{

string strURL = s1;

WebClient req = new WebClient();

HttpResponse response = HttpContext.Current.Response;

response.Clear();

response.ClearContent();

response.ClearHeaders();

response.Buffer = true;

response.AddHeader("Content-Disposition", "attachment;filename=\"" + Server.MapPath(strURL) + "\"");

byte[] data = req.DownloadData(Server.MapPath(strURL));

response.BinaryWrite(data);

response.End();

}

catch (Exception ex)

{

}

}

protected void download1\_Click(object sender, EventArgs e)

{

Button lbtn = (Button)sender;

string[] s = lbtn.ID.Split(' ');

int pid = Convert.ToInt32(s[0]);

string s1 = "";

con.Open();

SqlCommand cmd = new SqlCommand("select post\_uploads from post where post\_id='" + pid + "'", con);

SqlDataReader dr = cmd.ExecuteReader();

while (dr.Read())

{

s1 = "~//uploads//" + dr[0].ToString();

}

con.Close();

try

{

string strURL = s1;

WebClient req = new WebClient();

HttpResponse response = HttpContext.Current.Response;

response.Clear();

response.ClearContent();

response.ClearHeaders();

response.Buffer = true;

response.AddHeader("Content-Disposition", "attachment;filename=\"" + Server.MapPath(strURL) + "\"");

byte[] data = req.DownloadData(Server.MapPath(strURL));

response.BinaryWrite(data);

response.End();

}

catch (Exception ex)

{

}

}

protected void ImageButton1\_Click(object sender, ImageClickEventArgs e)

{

con.Open();

string s = Session["id"].ToString();

string b = "";

SqlCommand cmd2 = new SqlCommand("select login\_as from login where smart\_card\_id ='" + s + "' ", con);

SqlDataReader dr2;

dr2 = cmd2.ExecuteReader();

while (dr2.Read())

{

b = dr2[0].ToString(); // b=student or teacher

}

con.Close();

if (b == "student")

{

Response.Redirect("studhome.aspx");

}

else

if (b == "faculty")

{

Response.Redirect("teachhome.aspx");

}

else

if (b == "")

{

Response.Redirect("adminhome.aspx");

}

}

}

}

**7.3) Query Module**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Drawing;

using System.Web.UI.WebControls;

using System.Data;

using System.Data.SqlClient;

namespace GTBPINotificationZone

{

public partial class queries : System.Web.UI.Page

{

SqlConnection con = new SqlConnection("server=.\\SQLEXPRESS;AttachDbFilename=|DataDirectory|\\bnz.mdf;Integrated Security=True;MultipleActiveResultSets=True;User Instance=True;trusted\_connection=yes ");

protected void Page\_Load(object sender, EventArgs e)

{

try

{

int i = 1;

string s = "0";

string pid = Session["postid"].ToString();

con.Open();

SqlCommand cmd = new SqlCommand("select count(\*) from query where post\_id='" + pid + "' ", con);

SqlDataReader dr;

dr = cmd.ExecuteReader();

while (dr.Read())

{

s = dr[0].ToString();

}

if (s != "0")

{

Label old = new Label();

old.Text = "old queries";

old.Font.Bold = true;

old.ForeColor = System.Drawing.Color.White;

old.BackColor = Color.Black;

old.Font.Size = 18;

old.Font.Name = " Lucida Handwriting ";

Panel1.Controls.Add(old);//label old placed on panel

}

string a = Session["id"].ToString();

SqlCommand cmd1 = new SqlCommand("select \* from query where post\_id='" + pid + "' order by query\_id DESC", con);

SqlDataReader dr1;

dr1 = cmd1.ExecuteReader();

while ((dr1.Read()) && (i <= 5))

{

i++;

string q\_id = "";

q\_id = dr1[0].ToString();

Label name1 = new Label();

Label content = new Label();

Label date = new Label();

name1.Font.Size = 20;

name1.Font.Bold = true;

name1.ForeColor = System.Drawing.Color.Black;

name1.Text = dr1[5].ToString();

content.Text = dr1[3].ToString();

content.Font.Size = 14;

content.Font.Bold = true;

content.Style["width"] = "858px";

content.ForeColor = System.Drawing.Color.Black;

date.Text = dr1[4].ToString();

date.ForeColor = System.Drawing.Color.Black;

Panel1.Controls.Add(new LiteralControl("<br/>"));

Panel1.Controls.Add(name1);

Panel1.Controls.Add(new LiteralControl("<br/>"));

Panel1.Controls.Add(date);

Panel1.Controls.Add(new LiteralControl("<br/>"));

Panel1.Controls.Add(content);

Panel1.Controls.Add(new LiteralControl("<br/>"));

if ((a == dr1[1].ToString()) || (Session["admins"] == "1"))

{

Button del = new Button();

del.ID = q\_id;

del.Click += new EventHandler(this.delete\_query);

del.Text = "delete";

del.BackColor = Color.Silver;

del.ForeColor = System.Drawing.Color.Black;

del.Font.Size = 13;

Panel1.Controls.Add(del);

Panel1.Controls.Add(new LiteralControl("<br/>"));

}

//query button for admin

}

con.Close();

}

catch (Exception ec)

{ }

}

protected void Button1\_Click(object sender, EventArgs e)

{

if (TextBox1.Text != "")

{

con.Open();

string p = Session["postid"].ToString();

string s= Session["id"].ToString();

string b = "";

SqlCommand cmd2 = new SqlCommand("select login\_as from login where smart\_card\_id ='" + s + "' ", con);

SqlDataReader dr2;

dr2 = cmd2.ExecuteReader();

while (dr2.Read())

{

b = dr2[0].ToString(); // b=student or teacher

}

string name = "";

if (b == "student")

{

SqlCommand cmd3 = new SqlCommand(" select name from student where smart\_card\_id='" + s + "'", con);

SqlDataReader dr3;

dr3 = cmd3.ExecuteReader();

while (dr3.Read())

{

name = dr3[0].ToString();

}

Session["name"] = name;//name of the person is logged in

}

if (b == "faculty")

{

SqlCommand cmd3 = new SqlCommand(" select name from faculty where smart\_card\_id='" + s + "'", con);

SqlDataReader dr3;

dr3 = cmd3.ExecuteReader();

while (dr3.Read())

{

name = dr3[0].ToString();

}

Session["name"] = name;//name of the person is logged in

}

if (b == "")

{

Session["name"] = "admin";//name of the person is logged in

}

string a = "";

a = Session["name"].ToString();

try

{

SqlCommand cmd = new SqlCommand("insert into query values('" + s + "','" + Session["postid"].ToString() + "','" + TextBox1.Text + "','" + DateTime.Now.ToString() + "','" + a + "')", con);

cmd.ExecuteNonQuery();

con.Close();

TextBox1.Text = "";

Response.Redirect("queries.aspx");

}

catch (Exception ec) { }

}

else

{

Label10.Text = "enter your query";

}

}

protected void delete\_query(object sender, EventArgs e)

{

Button btn = (Button)sender;

string s = btn.ID;

con.Open();

SqlCommand cmd = new SqlCommand("delete from query where query\_id='" + s + "'", con);

cmd.ExecuteNonQuery();

con.Close();

Response.Redirect("queries.aspx");

}

protected void ImageButton1\_Click(object sender, ImageClickEventArgs e)

{

con.Open();

string s = Session["id"].ToString();

string b = "";

SqlCommand cmd2 = new SqlCommand("select login\_as from login where smart\_card\_id ='" + s + "' ", con);

SqlDataReader dr2;

dr2 = cmd2.ExecuteReader();

while (dr2.Read())

{

b = dr2[0].ToString(); // b=student or teacher

}

con.Close();

if (b == "student")

{

Response.Redirect("studhome.aspx");

}

else

if (b == "faculty")

{

Response.Redirect("teachhome.aspx");

}

else

if(b=="")

{

Response.Redirect("adminhome.aspx");

}

}

}

}

**7.4) Signup Module**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.IO;

using System.Web.UI.WebControls;

using System.Data;

using System.Data.SqlClient;

namespace GTBPINotificationZone

{

public partial class registerstud : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

if (Session["redirect"] == "1")

{

if (!IsPostBack)

{

con.Open();

string com = "select name from course ";

SqlDataAdapter adpt = new SqlDataAdapter(com, con);

DataTable dt = new DataTable();

adpt.Fill(dt);

scourse.DataSource = dt;

scourse.DataBind();

scourse.DataTextField = "name";

scourse.DataBind();

scourse.DataValueField = "--select--";

con.Close();

//for security questions

ssecurity\_ques1.Items.Add("first phone number");

ssecurity\_ques1.Items.Add("first car");

ssecurity\_ques1.Items.Add("favourite teacher");

ssecurity\_ques1.Items.Add("favourite game");

ssecurity\_ques1.Items.Add("favourite colour");

ssecurity\_ques2.Items.Add("first phone number");

ssecurity\_ques2.Items.Add("first car");

ssecurity\_ques2.Items.Add("favourite teacher");

ssecurity\_ques2.Items.Add("favourite game");

ssecurity\_ques2.Items.Add("favourite colour");

}

}

else

{

Response.Redirect("login.aspx");

}

}

SqlConnection con = new SqlConnection("server=.\\SQLEXPRESS;AttachDbFilename=|DataDirectory|\\bnz.mdf;Integrated Security=True;User Instance=True;trusted\_connection=yes");

protected void Button1\_Click(object sender, EventArgs e)

{

string i1= "0";

con.Open();

SqlCommand cmd1 = new SqlCommand("Select count(\*) from student where smart\_card\_id='" + sid.Text + "'", con);

SqlDataReader dr;

dr = cmd1.ExecuteReader();

while (dr.Read())

{

i1 = dr[0].ToString();

}

con.Close();

if (i1!="0")

{

Label20.Text = "please enter valid smart card id and enter your details again";

sid.Text = "";

sname.Text = "";

spassword.Text = "";

sconfirmpassword.Text = "";

ssecurity\_ques1.Text = "";

ssecurity\_answer.Text = "";

ssecurity\_ques2.Text = "";

ssecurity\_answer0.Text = "";

}

else

{

con.Open();

SqlCommand cmd = new SqlCommand("Insert into student values('" + sid.Text + "','" + sname.Text + "','" + scourse.SelectedItem.Text + "','" + syear.Text + "','" + spassword.Text + "','" + ssecurity\_ques1.SelectedItem.Text + "','" + ssecurity\_answer.Text + "','" + ssecurity\_ques2.SelectedItem.Text + "','" + ssecurity\_answer0.Text + "')", con);

cmd.ExecuteNonQuery();

string s = "student";

con.Close();

con.Open();

SqlCommand cmd2 = new SqlCommand("insert into login values('" + sid.Text + "','" + spassword.Text + "','" + s + "')", con);

cmd2.ExecuteNonQuery();

con.Close();

Response.Redirect("login.aspx");

con.Close();

}

}

protected void Button2\_Click(object sender, EventArgs e)

{

sid.Text= "" ;

sname.Text = "" ;

spassword.Text = "";

syear.Text = "";

sconfirmpassword.Text = "";

ssecurity\_answer.Text = "";

ssecurity\_answer0.Text = "";

}

protected void ssecurity\_ques2\_TextChanged(object sender, EventArgs e)

{

}

protected void DropDownList1\_SelectedIndexChanged(object sender, EventArgs e)

{

}

protected void Login\_Click(object sender, EventArgs e)

{

Response.Redirect("login.aspx");

}

}

}

**7.5) Home Module**

using System;

using System.Web;

using System.Web.UI;

using System.Data.SqlClient;

using System.Drawing;

using System.Web.UI.WebControls;

using System.Net;

namespace GTBPINotificationZone

{

public partial class studhome : System.Web.UI.Page

{

SqlConnection con = new SqlConnection("server=.\\SQLEXPRESS;AttachDbFilename=|DataDirectory|\\bnz.mdf;Integrated Security=True;MultipleActiveResultSets=True;User Instance=True;trusted\_connection=yes ");

protected void Page\_Load(object sender, EventArgs e)

{

if (Session["s1"] != "1")

{

Response.Redirect("login.aspx");

}

else

{

con.Open();

string s1, s3, s4, id1;

string[] d\_time;

string[] ext;

string s2 = "0";

string sc = "";

int ss = 0;

string sid = Session["id"].ToString();

SqlCommand scmd = new SqlCommand("select course,year from student where smart\_card\_id='" + sid + "'", con);

SqlDataReader sdr;

sdr = scmd.ExecuteReader();

while (sdr.Read())

{

sc = sdr[0].ToString();

ss = Convert.ToInt32(sdr[1]);

}

SqlCommand cmd1, cmd2;

SqlDataReader dr1, dr2;

SqlCommand cmd = new SqlCommand("select \* from post where course='" + sc + "' and year='" + ss + "' or smart\_card\_id='"+sid+"' order by post\_id DESC ", con);

SqlDataReader dr = cmd.ExecuteReader();

while (dr.Read())

{

id1 = dr[0].ToString();//post id

s1 = dr[4].ToString();//smart card id of the person posted the post

//outer panel of home page.

Panel outer = new Panel();

outer.BorderStyle = BorderStyle.Groove;

outer.Style["border-radius"] = "5px";

outer.BorderColor = Color.Black;

outer.Style["width"] = "600px";

outer.BackColor = Color.WhiteSmoke;

Panel1.Controls.Add(new LiteralControl(" &nbsp"));

Panel1.Controls.Add(new LiteralControl(" &nbsp"));

Panel1.Controls.Add(new LiteralControl(" &nbsp"));

Panel1.Controls.Add(new LiteralControl(" &nbsp"));

Panel1.Controls.Add(new LiteralControl(" &nbsp"));

Panel1.Controls.Add(new LiteralControl(" &nbsp"));

Panel1.Controls.Add(new LiteralControl(" &nbsp"));

Panel1.Controls.Add(new LiteralControl(" &nbsp"));

Panel1.Controls.Add(new LiteralControl(" &nbsp"));

Panel1.Controls.Add(new LiteralControl(" &nbsp"));

Panel1.Controls.Add(outer);

//inner subpanel1

Panel inner1 = new Panel();

inner1.BackColor = Color.WhiteSmoke;

//inner subpanel2

Panel inner2 = new Panel();

inner2.BackColor = Color.WhiteSmoke;

//controls to be uploaded.

Label name = new Label(); //label for name of the user who posted the info.

Label course = new Label(); //course for which the info is intended.

Label sem = new Label(); //semester of the course

Label sub = new Label(); //subject of which info is posted

Label date1 = new Label(); //date at which info is posted

Label time1 = new Label(); //time at which info is posted

Label content = new Label(); //content of the post

//fatching name of the user who posted the info

name.Font.Size = 15;

name.Font.Bold = true;

name.Font.Name = "Calibri";

name.Style["TextAlign"] = "top\_left";

cmd1 = new SqlCommand("select login\_as from login where smart\_card\_id='" + s1 + "'", con);

dr1 = cmd1.ExecuteReader();

while (dr1.Read())

{

s2 = dr1[0].ToString();

}

if (s2 == "0")

{

name.Text = "admin";

}

if (s2 == "student")

{

cmd2 = new SqlCommand("select name from student where smart\_card\_id='" + s1 + "'", con);

dr2 = cmd2.ExecuteReader();

while (dr2.Read())

{

name.Text = dr2[0].ToString();

}

}

if (s2 == "faculty")

{

cmd2 = new SqlCommand("select name from faculty where smart\_card\_id='" + s1 + "'", con);

dr2 = cmd2.ExecuteReader();

while (dr2.Read())

{

name.Text = dr2[0].ToString();

}

}

outer.Controls.Add(name);

outer.Controls.Add(inner1);

outer.Controls.Add(inner2);

//fatching course,semester,subject,date,topic of the post.

course.Font.Size = 13;

course.Font.Name = "Calibri";

course.Text = dr[2].ToString();

course.Style["color"] = "#999999";

sem.Font.Size = 13;

sem.Style["color"] = "#999999";

sem.Font.Name = "Calibri";

sem.Text = dr[8].ToString();

sub.Font.Size = 13;

sub.Font.Name = "Calibri";

sub.Text = dr[7].ToString();

sub.Style["color"] = "#999999";

d\_time = dr[3].ToString().Split(' ');

date1.Font.Size = 13;

date1.Font.Name = "Calibri";

date1.Text = d\_time[0];

date1.Style["color"] = "#999999";

time1.Font.Size = 13;

time1.Font.Name = "Calibri";

time1.Text = d\_time[1];

time1.Style["color"] = "#999999";

inner1.Controls.Add(course);

inner1.Controls.Add(sem);

inner1.Controls.Add(new LiteralControl("<br/>"));

inner1.Controls.Add(sub);

inner1.Controls.Add(new LiteralControl("<br/>"));

inner1.Controls.Add(date1);

inner1.Controls.Add(new LiteralControl("<br/>"));

inner1.Controls.Add(time1);

inner1.Controls.Add(new LiteralControl("<br/>"));

inner1.Controls.Add(new LiteralControl("<br/>"));

//content and uploads of post.

s3 = dr[5].ToString();

s4 = dr[6].ToString();

if ((s3 == "") && (s4 != "null")) //when file is uploaded

{

ext = s4.Split('.');

if ((ext[1] == "jpg") || (ext[1] == "jpeg") || (ext[1] == "png")) //download button for images

{

System.Web.UI.WebControls.Image uploads = new System.Web.UI.WebControls.Image(); //images if any

uploads.Style["width"] = "275px";

uploads.Style["height"] = "183px";

uploads.ImageUrl = "~\\uploads\\" + s4;

inner2.Controls.Add(uploads);

inner2.Controls.Add(new LiteralControl("<br/>"));

//download button

Button download1 = new Button();

download1.Text = "Download Image";

download1.ID = id1 + " c";

download1.Font.Size = 14;

download1.BorderStyle = BorderStyle.Groove;

download1.Style["width"] = "150px";

download1.Style["height"] = "36px";

download1.Click += new EventHandler(this.download1\_Click);

inner2.Controls.Add(download1);

inner2.Controls.Add(new LiteralControl("<br/>"));

}

else

if ((ext[1] == "pdf") || (ext[1] == "docx") || (ext[1] == "pptx")) //download button for text files

{

//download button

inner2.Controls.Add(new LiteralControl("<br/>"));

LinkButton download = new LinkButton();

download.Text ="Download from here ->"+dr[1].ToString() ;

download.ID = id1 + " b";

download.Font.Size = 14;

download.Click += new EventHandler(this.download\_Click);

inner2.Controls.Add(download);

inner2.Controls.Add(new LiteralControl("<br/>"));

}

}

else

if ((s3 != "") && (s4 != "null"))// file and content are uploaded

{

content.Text = s3;

content.Font.Size = 13;

content.Font.Bold = true;

content.Style["color"] = "Black";

content.Style["width"] = "520px";

content.Style["height"] = "180px";

inner2.Controls.Add(content);

inner2.Controls.Add(new LiteralControl("<br/>"));

ext = s4.Split('.');

if ((ext[1] == "jpg") || (ext[1] == "jpeg") || (ext[1] == "png")) //download button for images

{

System.Web.UI.WebControls.Image uploads = new System.Web.UI.WebControls.Image(); //images if any

uploads.Style["width"] = "275px";

uploads.Style["height"] = "183px";

uploads.ImageUrl = "~\\uploads\\" + s4;

inner2.Controls.Add(uploads);

inner2.Controls.Add(new LiteralControl("<br/>"));

//download button

Button download1 = new Button();

download1.Text = "Download Image";

download1.ID = id1 + " c";

download1.Font.Size = 14;

download1.BorderStyle = BorderStyle.Groove;

download1.Style["width"] = "150px";

download1.Style["height"] = "36px";

download1.Click += new EventHandler(this.download1\_Click);

inner2.Controls.Add(download1);

inner2.Controls.Add(new LiteralControl("<br/>"));

}

else

if ((ext[1] == "pdf") || (ext[1] == "docx") || (ext[1] == "pptx")) //download button for text files

{

//download button

inner2.Controls.Add(new LiteralControl("<br/>"));

LinkButton download = new LinkButton();

download.Text = "Download from here -> "+dr[1].ToString();

download.ID = id1 + " b";

download.Font.Size = 14;

download.Style["width"] = "28px";

download.Style["height"] = "15px";

download.Click += new EventHandler(this.download\_Click);

inner2.Controls.Add(download);

inner2.Controls.Add(new LiteralControl("<br/>"));

}

}

else

{

content.Text = s3; //when only text is there

content.Font.Size = 13;

content.Font.Bold = true;

content.Style["color"] = "Black";

content.Style["width"] = "520px";

content.Style["height"] = "180px";

inner2.Controls.Add(content);

}

if (Session["id"].ToString() == dr[4].ToString()) //login id and post smart card id comparison

{

//button for delete post

Button del = new Button();

del.ID = id1;

del.Click += new EventHandler(this.test1);

del.Style["width"] = "76px";

del.Style["height"] = "36px";

del.Text = "delete";

del.Font.Bold = true;

del.BorderColor = Color.Black;

del.Font.Size = 16;

del.BorderStyle = BorderStyle.Groove;

//button for query on post

Button query = new Button();

query.Style["width"] = "76px";

query.Style["height"] = "36px";

query.Text = "query";

query.ID = id1 + " a";

query.Font.Bold = true;

query.BorderColor = Color.Black;

query.Font.Size = 16;

query.Click += new EventHandler(this.query\_Click);

query.BorderStyle = BorderStyle.Groove;

Panel1.Controls.Add(new LiteralControl("<br/>"));

for (int i = 0; i < 132; i++)

{

Panel1.Controls.Add(new LiteralControl(" &nbsp"));

}

Panel1.Controls.Add(del);

Panel1.Controls.Add(new LiteralControl(" &nbsp")); Panel1.Controls.Add(new LiteralControl(" &nbsp"));

Panel1.Controls.Add(query);

Panel1.Controls.Add(new LiteralControl("<br/>")); Panel1.Controls.Add(new LiteralControl("<br/>"));

}

else

{

//button for query on post

Button query = new Button();

query.Style["width"] = "76px";

query.Style["height"] = "36px";

query.Text = "query";

query.Font.Bold = true;

query.BorderColor = Color.Black;

query.Font.Size = 16;

query.ID = id1 + " a";

query.Click += new EventHandler(this.query\_Click);

query.BorderStyle = BorderStyle.Groove;

for (int i = 0; i < 132; i++)

{

Panel1.Controls.Add(new LiteralControl(" &nbsp"));

}

Panel1.Controls.Add(query);

Panel1.Controls.Add(new LiteralControl("<br/>")); Panel1.Controls.Add(new LiteralControl("<br/>"));

}

}

con.Close();

}

}

protected void test1(object sender, EventArgs e)

{

Button btn = (Button)sender;

string s = btn.ID;

con.Open();

SqlCommand cmd = new SqlCommand("delete from post where post\_id='" + s + "'", con);

cmd.ExecuteNonQuery();

con.Close();

Response.Redirect("studhome.aspx");

}

protected void query\_Click(object sender, EventArgs e)

{

Button btn1 = (Button)sender;

string[] s = btn1.ID.Split(' ');

int pid = Convert.ToInt32(s[0]);

Session["postid"] = s[0];

Response.Redirect("queries.aspx");

}

protected void download\_Click(object sender, EventArgs e)

{

LinkButton lbtn = (LinkButton)sender;

string[] s = lbtn.ID.Split(' ');

int pid = Convert.ToInt32(s[0]);

string s1 = "";

con.Open();

SqlCommand cmd = new SqlCommand("select post\_uploads from post where post\_id='"+pid+"'", con);

SqlDataReader dr = cmd.ExecuteReader();

while (dr.Read())

{

s1 = "~//uploads//" + dr[0].ToString();

}

con.Close();

try

{

string strURL = s1;

WebClient req = new WebClient();

HttpResponse response = HttpContext.Current.Response;

response.Clear();

response.ClearContent();

response.ClearHeaders();

response.Buffer = true;

response.AddHeader("Content-Disposition", "attachment;filename=\"" + Server.MapPath(strURL) + "\"");

byte[] data = req.DownloadData(Server.MapPath(strURL));

response.BinaryWrite(data);

response.End();

}

catch (Exception ex)

{

}

}

protected void download1\_Click(object sender, EventArgs e)

{

Button lbtn = (Button)sender;

string[] s = lbtn.ID.Split(' ');

int pid = Convert.ToInt32(s[0]);

string s1 = "";

con.Open();

SqlCommand cmd = new SqlCommand("select post\_uploads from post where post\_id='" + pid + "'", con);

SqlDataReader dr = cmd.ExecuteReader();

while (dr.Read())

{

s1 = "~//uploads//" + dr[0].ToString();

}

con.Close();

try

{

string strURL = s1;

WebClient req = new WebClient();

HttpResponse response = HttpContext.Current.Response;

response.Clear();

response.ClearContent();

response.ClearHeaders();

response.Buffer = true;

response.AddHeader("Content-Disposition", "attachment;filename=\"" + Server.MapPath(strURL) + "\"");

byte[] data = req.DownloadData(Server.MapPath(strURL));

response.BinaryWrite(data);

response.End();

}

catch (Exception ex)

{

}

}

}

}

**7.6) Student Post Module**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data;

using System.Data.SqlClient;

using System.IO;

namespace GTBPINotificationZone

{

public partial class studpost : System.Web.UI.Page

{

SqlConnection con = new SqlConnection("server=.\\SQLEXPRESS;AttachDbFilename=|DataDirectory|\\bnz.mdf;Integrated Security=True;User Instance=True;trusted\_connection=Yes");

protected void Page\_Load(object sender, EventArgs e)

{

if (Session["s1"] != "1")

{

Response.Redirect("login.aspx");

}

else

{

if (!IsPostBack)

{

con.Open();

string com = "select name from course ";

SqlDataAdapter adpt = new SqlDataAdapter(com, con);

DataTable dt = new DataTable();

adpt.Fill(dt);

spcourse.DataSource = dt;

spcourse.DataBind();

spcourse.DataTextField = "name";

spcourse.DataBind();

spcourse.DataValueField = "--select--";

con.Close();

}

con.Open();

int yr = 0;

string s = spcourse.SelectedItem.Text;

SqlCommand cmd1 = new SqlCommand("select year from course where name='" + s + "'", con);

SqlDataReader dr = cmd1.ExecuteReader();

while (dr.Read())

{

yr = Convert.ToInt32(dr[0]);

}

con.Close();

}

}

protected void spsubmit\_Click(object sender, EventArgs e)

{

string id = Session["id"].ToString();

con.Open();

string s = "null";

string ex;

try

{

if ((spfiles.HasFile) && (spdetails.Text == ""))

{

s = Path.GetFileName(spfiles.FileName);

ex = Path.GetExtension(spfiles.FileName).ToLower();

if ((ex == ".jpg") || (ex == ".png") || (ex == ".jpeg") || (ex == ".docx") || (ex == ".pptx"))

{

spfiles.SaveAs(Server.MapPath("~\\uploads\\") + s);

SqlCommand cmd = new SqlCommand("insert into post values('" + sptopic.Text + "','" + spcourse.SelectedItem.Text + "','" + DateTime.Now.ToString() + "','" + id + "','" + spdetails.Text + "','" + s + "','" + spsub.Text + "','" + spyear.Text + "')", con);

cmd.ExecuteNonQuery();

}

else

{

Label1.Text = "enter file of type 'docx','png','jpeg','jpg','pptx'";

}

}

if ((spfiles.HasFile) && (spdetails.Text != ""))

{

s = Path.GetFileName(spfiles.FileName);

ex = Path.GetExtension(spfiles.FileName).ToLower();

if ((ex == ".jpg") || (ex == ".png") || (ex == ".jpeg") || (ex == ".docx") || (ex == ".pptx"))

{

spfiles.SaveAs(Server.MapPath("~\\uploads\\") + s);

SqlCommand cmd = new SqlCommand("insert into post values('" + sptopic.Text + "','" + spcourse.SelectedItem.Text + "','" + DateTime.Now.ToString() + "','" + id + "','" + spdetails.Text + "','" + s + "','" + spsub.Text + "','" + spyear.Text + "')", con);

cmd.ExecuteNonQuery();

}

else

{

Label1.Text = "enter file of type 'docx','png','jpeg','jpg','pptx'";

}

}

if ((!spfiles.HasFile) && (spdetails.Text != ""))

{

SqlCommand cmd = new SqlCommand("insert into post values('" + sptopic.Text + "','" + spcourse.SelectedItem.Text + "','" + DateTime.Now.ToString() + "','" + id + "','" + spdetails.Text + "','" + s + "','" + spsub.Text + "','" + spyear.Text + "')", con);

cmd.ExecuteNonQuery();

}

if ((!spfiles.HasFile) && (spdetails.Text == ""))

{

Label1.Text = "you have to enter the post information";

}

con.Close();

spyear.Text = "";

spsub.Text = "";

sptopic.Text = "";

spdetails.Text = "";

}

catch (Exception ec) { }

}

}

}

**7.7) Student Profile Module**

using System;

using System.Web.UI.WebControls;

using System.Data.SqlClient;

namespace GTBPINotificationZone

{

public partial class studprofile : System.Web.UI.Page

{

SqlConnection con = new SqlConnection("server=.\\SQLEXPRESS;AttachDbFilename=|DataDirectory|\\bnz.mdf;Integrated security=True;user Instance=True;trusted\_connection=yes");

protected void Page\_Load(object sender, EventArgs e)

{

if (Session["s1"] == "0")

{

Response.Redirect("login.aspx");

}

else

{

if (!IsPostBack)

{

ussid.Text = Session["id"].ToString();

con.Open();

SqlCommand cmd = new SqlCommand("select \* from student where smart\_card\_id='" + ussid.Text + "'", con);

SqlDataReader dr;

dr = cmd.ExecuteReader();

while (dr.Read())

{

usname.Text = dr[1].ToString();

uscourse.Text = dr[2].ToString();

usyear.Text = dr[3].ToString();

uspassword.Text = dr[4].ToString();

ussec\_ques1.Text = dr[5].ToString();

ussec\_ans1.Text = dr[6].ToString();

ussec\_ques2.Text = dr[7].ToString();

ussec\_ans2.Text = dr[8].ToString();

}

con.Close();

}

}

}

protected void Button1\_Click(object sender, EventArgs e)

{

con.Open();

SqlCommand cmd = new SqlCommand("update student set smart\_card\_id='" + ussid.Text + "' , name='" + usname.Text + "',course='" + uscourse.Text + "',year='" + usyear.Text + "',password='" + uspassword.Text + "',sec\_ques1='" + ussec\_ques1.Text + "',sec\_ans1='" + ussec\_ans1.Text + "',sec\_ques2='" + ussec\_ques2.Text + "',sec\_ans2='" + ussec\_ans2.Text + "' where smart\_card\_id='" + Session["id"].ToString() + "'", con);

cmd.ExecuteNonQuery();

SqlCommand cmd1 = new SqlCommand("update login set smart\_card\_id='" + ussid.Text + "',password='" + uspassword.Text + "' where smart\_card\_id='"+Session["id"].ToString()+"'", con);

cmd1.ExecuteNonQuery();

SqlCommand cmd2 = new SqlCommand("update post set smart\_card\_id='" + ussid.Text + "' where smart\_card\_id='" + Session["id"].ToString() + "'", con);

cmd2.ExecuteNonQuery();

SqlCommand cmd3 = new SqlCommand("update query set smart\_card\_id='" + ussid.Text + "' where smart\_card\_id='" + Session["id"].ToString() + "'", con);

cmd3.ExecuteNonQuery();

SqlCommand cmd4 = new SqlCommand("select \*from faculty where smart\_card\_id='" + ussid.Text + "'", con);

SqlDataReader dr;

dr = cmd.ExecuteReader();

while (dr.Read())

{

ussid.Text = dr[0].ToString();

usname.Text = dr[1].ToString();

uscourse.Text = dr[2].ToString();

usyear.Text = dr[3].ToString();

uspassword.Text = dr[4].ToString();

ussec\_ques1.Text = dr[5].ToString();

ussec\_ans1.Text = dr[6].ToString();

ussec\_ques2.Text = dr[7].ToString();

ussec\_ans2.Text = dr[8].ToString();

}

con.Close();

}

protected void CustomValidator1\_ServerValidate(object source, ServerValidateEventArgs args)

{

if ((args.Value.Length > 8) && (args.Value.Length < 20))

args.IsValid = true;

else

args.IsValid = false;

}

}

}

1. **Testing Issues**

Software testing can be stated as the process of validating ad verifying that a software program/ application/ product:

1. Meets the requirements that guided its design and development;
2. Works as expected; and
3. Can be implemented with the same characteristics.

Testing is the process of making sure that the program performs the intended tasks.

* Unit Testing: Each component or part of the system is tested individually.
* Module Testing: A collection of dependent components such as an object class, procedures and functions are tested in this testing.
* Integration Testing: In this, many units tested modules are combined into sub systems, which are then tested.
* System Testing: Entire system software is tested. It is a testing of the system against its initial objectives.

1. **References**

[1] Pressman Roger S., Software Engineering “A Practitioner’s Approach”

Fifth Edition, McGraw-Hill Publication, 2000.

[2] Navathe Shamkant B.*,* Fundamentals of Database Systems,

Fifth Edition, Pearson Publication*.*

[3] Evjen Bill, Hanselman Scott, Rader Devin,

Professional ASP.Net 3.5 in C# and VB,

Sp1Edition, Wrox Publication, 2009.

[4] Evjen Bill, Professional C#,

Wrox Publication, 2008.

[5] Bayross Ivan, SQL, PL/SQL,

Third Edition, BPB Publication.

[6] Schildt Herbert, The Complete Reference C# 3.0

Third Edition, Tata McGraw-Hill.

[7] IEEE STD 830-1998, IEEE Recommended Practice for Software Requirement Specifications.