

Technical Documentation

Faculty Information System

Made by Group 4

Udbhav Chugh	:	170101081
Aman Mishra	:	170101005
Chirag Gupta	:	170101019
Mani Mannampalli	:	170101036
Pranshu Srivas	:	170101048
Rashi Singh	:	170101052

Introduction:

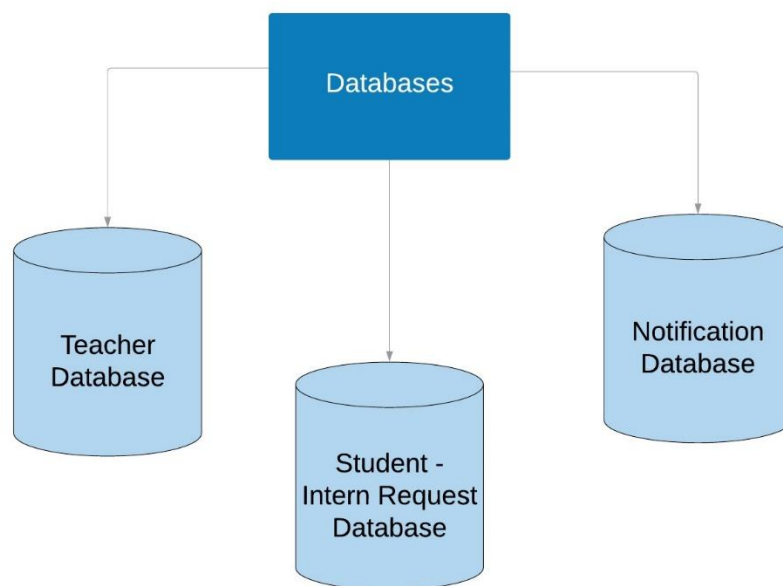
This software is a basic faculty information system with multiple functions. It gives the user an interface to find the required information about faculty members of IIT Guwahati via a search utility. Furthermore, the professors can register and login to edit and view their personal information page. A notifications section for each department is another auxiliary feature of this software. So, this interface is a complete-package of all faculty information systems that is more than anyone could ask for.

Contents

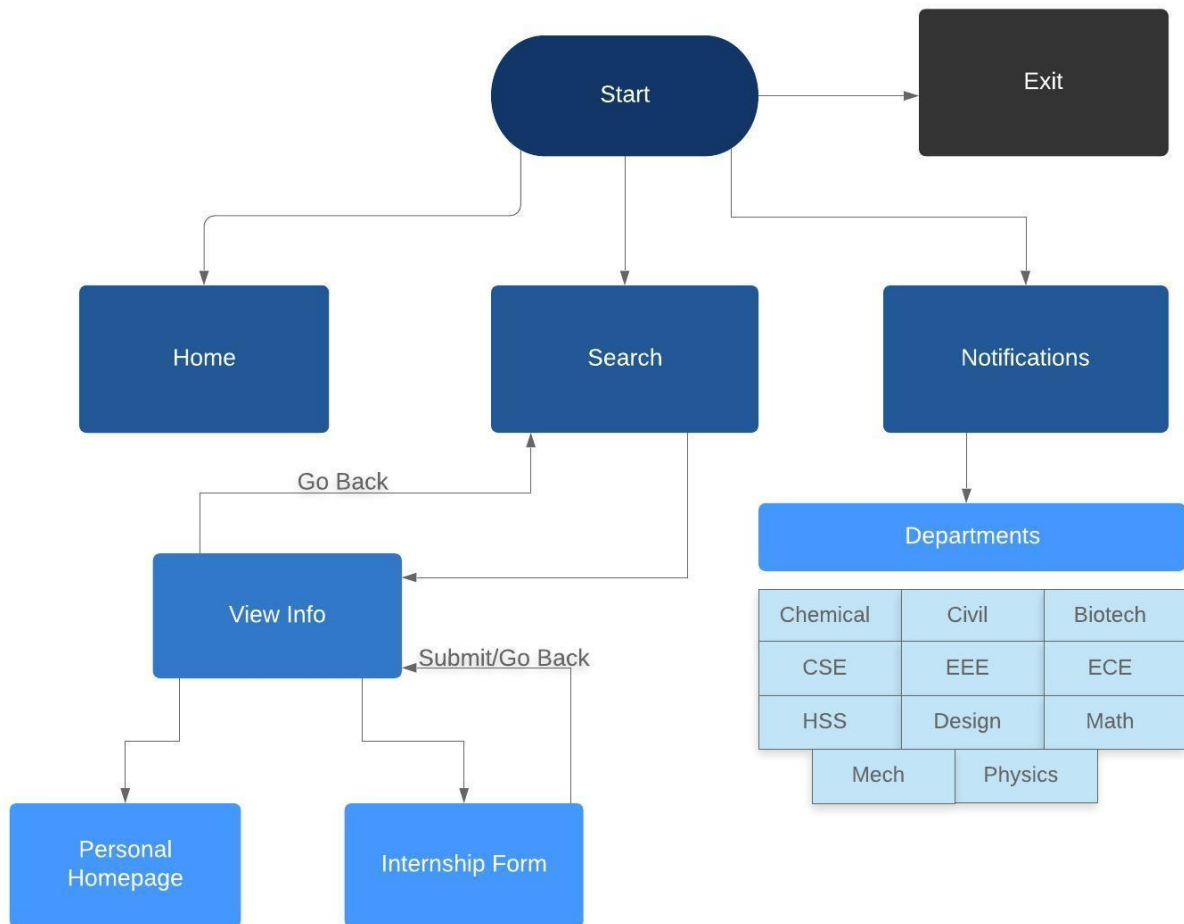
- ✚ Form_Main
- ✚ Login_User_Control
- ✚ Teacher_Homepage
- ✚ Register_User_Control
- ✚ Search_User_Control
- ✚ Notice_User_Control
- ✚ Notification_User_Control
- ✚ Notice_List_User_Control
- ✚ View_Teacher_User_Control
- ✚ Interested_Students_User_Control
- ✚ Form_Internship
- ✚ Change_Password_User_Control

- ✚ Edit_Teacher_User_Control
- ✚ Forget_User_Control
- ✚ Form_Notification
- ✚ Admin_User_Control
- ✚ Form_Encryption

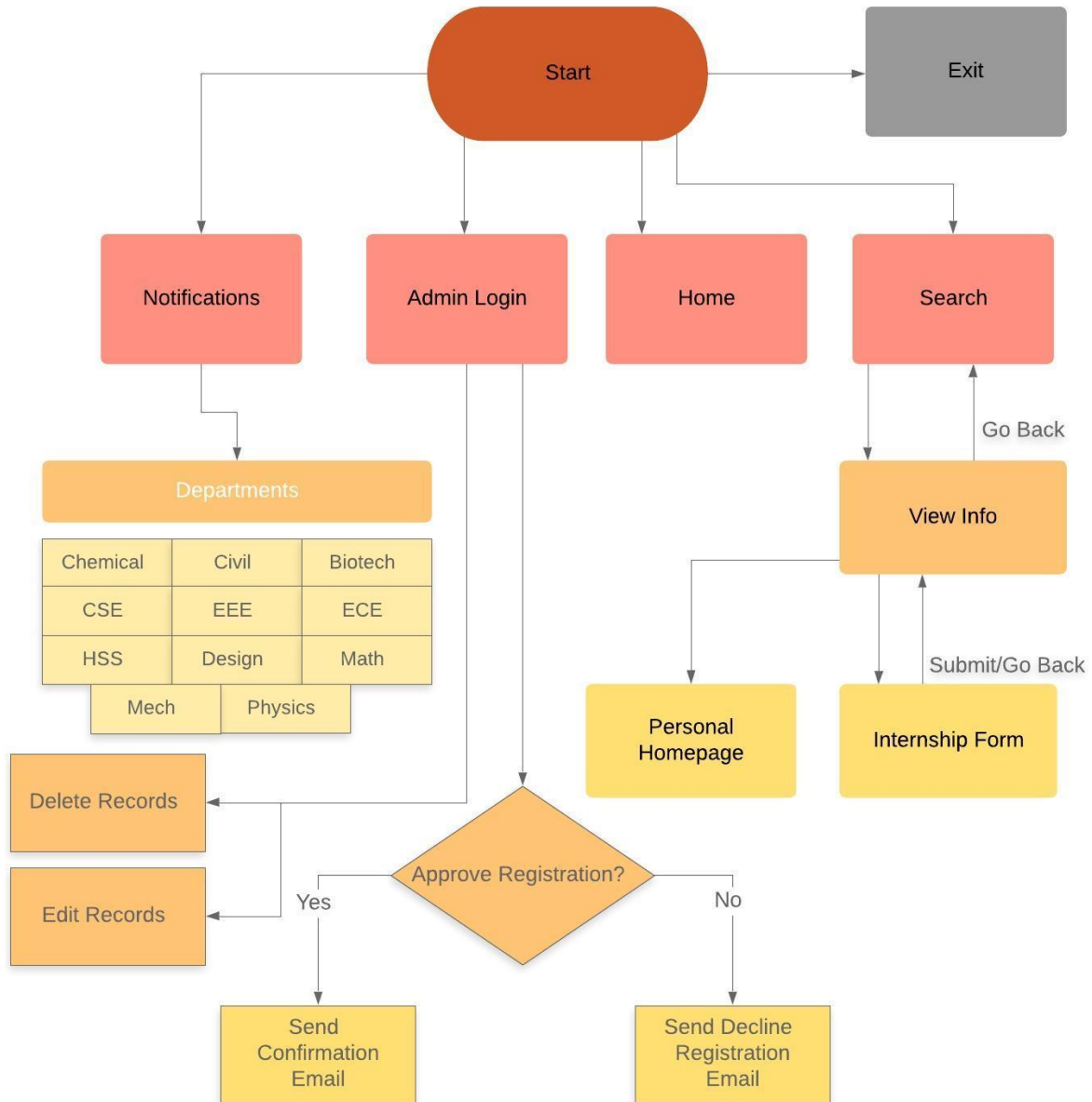
Database Flowchart



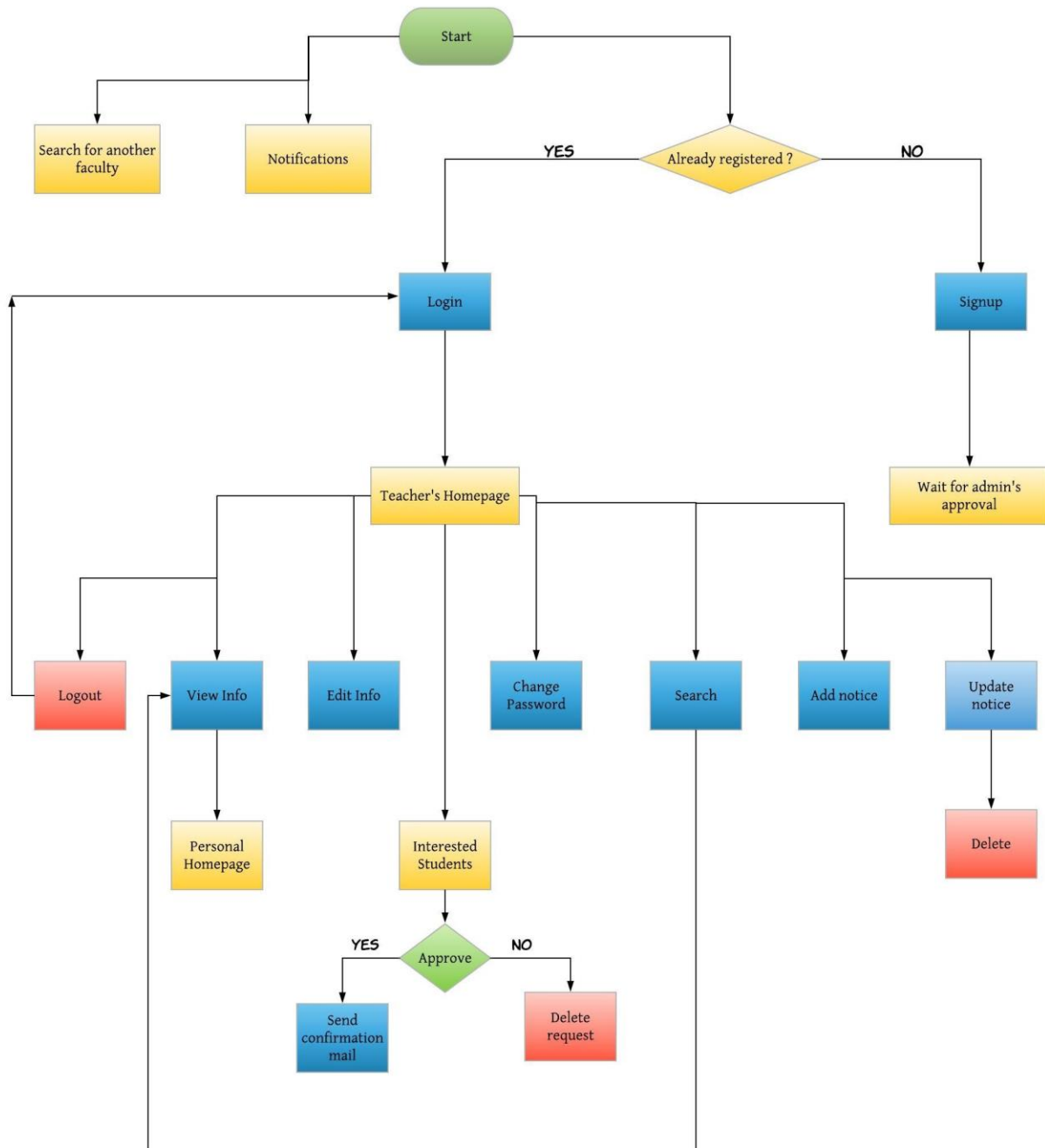
Student Flowchart



Admin Flowchart



Teacher Flowchart



Main Form (Form_Main) functionalities:

Widgets:

- **Panel:**

- **Output_Panel:** The main panel to display the different utilities from the left-hand side buttons.

- **Buttons:**

- **Btn_Teacher_Login**(View Login form) :

- Links to the class "*Login_User_Control*" to login to the personal homepage.

- **Btn_Signup**(View Sign-Up Form):

- Links to the class "*Register_User_Control*" to sign up to the faculty database.

- **Btn_Search**(View Search Form):

- Links to the class "*Search_User_Control*" to search in the faculty database using multiple filters.

- **Btn_Notification** (View Notifications dept-wise):

- Links to the class "*Notification_User_Control*" to view any notifications issued by faculty members of different departments.

- **Btn_HomePge** (View Homepage):

- Links to the class "*HomePage_User_Control*", which is visible when the form loads.

- **Btn_Admin:**

- Leads to the administrator's page to approve new registration requests.

- **Button1** (Exit Button):

- Closes the main form.

- **Functions:**

- **Form1_Load(Form_Main):**

- Runs on loading Form_Main. It makes the Output_Panel invisible during loading.

- **IsAlpha:**

- Returns true if a string contains only alphabets.

[Back to Contents](#)

Login_User_Control functionalities:

Widgets:

- **Panel:**

- **Forget_Panel:**

- The panel is initially hidden. It shows the Forget_User_Control which is used to change the password in case a user forgets his/her password.

- **Buttons:**

- **Btn_Submit :**

- Searches in the database to verify whether the entered information is present as a record and directs the user to *"Teacher_Homepage"*.

- In case of no matching records, an *"incorrect password"* message box is displayed.

- **Forget_Password:**

- It makes the Forget_Panel visible so that the user can change his password.

- **TextBox:**

- ✖ **Username:**

- Space allocated to enter the username.

- ✖ **Password:**

- Space allocated to enter the password. Password is encrypted

Teacher_Homepage functionalities:

Widgets:

- **Panel:**

- **Panel1:**

- The main panel to display the different utilities from the left-hand side buttons.

- **Buttons:**

- **Btn_View**(View Homepage form) :

- Links to the class "*View_Teacher_User_Control*" to view the personal homepage.

- **Btn_Edit**(Edit Homepage Form):

- Links to the class "*Edit_Teacher_User_Control*" to edit the details of the personal homepage.

- **Btn_Notice:**

- Links to the class "*Notice_User_Control*" to add notices, which would be visible in the notifications of the respective department.

- **Button1** (Search Button):

- Links to the class "*Search_User_Control*" to search in the faculty database using multiple filters.

- **Btn_HomePge** (View Homepage):

- Links to the class "*HomePage_User_Control*", which is visible when the form loads.

- **Change_Password:**

- Links to the class "*Change_Password_User_Control*", so that the user can change his/her password.

- **Btn_List** (Update Notices):

Links to the class "*Notice_List_User_Control*", so that the user can delete the obsolete notices.

→ **Button2** (Interested Students):

Links to the class "*Interested_Students_User_Control*", so that the user can view the list of students who are interested in internship opportunities under him/her.

→ **Btn_Logout**:

Asks for confirmation before closing the form.

[Back to Contents](#)

Register_User_Control functionalities:

Widgets:

- **Buttons:**

→ **Btn_Create**:

Creates an entry in the database according to the entered information (like Full Name, Username, Password, Courses, Research Tag, etc.). Furthermore, it checks if any discrepancies are present in the entered data.

It also verifies whether the entered username is already present in the database. On successful sign-up, "*registration successful*" msgbox appears on the screen.

- **Labels and TextBox:**

Information is entered according to the respective label names

- ❖ Name
- ❖ Create Username
- ❖ Password
- ❖ Room Number
- ❖ Contact Number
- ❖ Email Address
- ❖ Answer to the security Question

- **ComboBox:**

- **Register_Department:**

- User can choose any option from the given options

- ❖ CSE
 - ❖ Mathematics
 - ❖ ECE
 - ❖ EEE
 - ❖ Mechanical
 - ❖ Chemical
 - ❖ Biotechnology
 - ❖ Design

- **Register_Question:**

- User can choose any option from the given options

- ✖ What was your favorite place to visit as a child?
 - ✖ In what city were you born?
 - ✖ What is the name of your first grade teacher?
 - ✖ What is the name of your pet?

- **Register_ECaptcha:**

- It provides a firewall to prevent bots from accessing the program.

[Back to Contents](#)

Search_User_Control functionalities:

Widgets:

- **Panel:**

- **Search_Panel:**

- Links to the class “*View_Teacher_User_Control*” once the user clicks on any one of the search results, to view the personal homepage of the chosen faculty member.

- **ListBox:**

- **ListBox1:**

- This displays the name of the faculty members in the search results.

- **Labels For TextBox:**

- **Label1 (Name):**

- The user can enter the name of the faculty member that he wants to search about. It is *Case-insensitive*.

- **Label3 (Course):**

- The user can enter the name of the course that he wants to search the faculty member of. It is *Case-insensitive*.

- **Labels For ComboBox:**

- **Label2 (Department):**

- The user can select the department that he wants to search the faculty members of. This includes:

- ❖ CSE
 - ❖ Mathematics
 - ❖ ECE
 - ❖ EEE
 - ❖ Mechanical
 - ❖ Chemical
 - ❖ Biotechnology
 - ❖ Design

- **Label4 (Research Interest):** The user can select the research tag(broad research category) from the following options;

- ❖ Artificial Intelligence
 - ❖ Machine Learning
 - ❖ Computer Vision
 - ❖ Speech Processing
 - ❖ Networking
 - ❖ Operating Systems
 - ❖ DS and Algorithms
 - ❖ Electronic Circuits

- **Buttons:**

- **Search_Click(View-Login form) :**

This compares the entered information with the database and displays the matching entries in the ListBox1.

All kinds of exception handling for wrong inputs is also done here.

In case of multiple filters, it checks different cases for different types of filters being empty and applies the effects of the remaining filters only.

- **Back_Button:**

Makes the search_panel invisible and brings the user back to the seach_form.

- **Functions:**

- **IsAlpha:**

This function from form1 is used here during the exception handling as it returns true if a string contains only alphabets.

[Back to Contents](#)

Notice_User_Control functionalities:

Widgets:

- **TextBox:**

- **Txt_Notice:**

The user can type in the notification here that he wants to display in the main department-wise bifurcated notification area.

- **Buttons:**

- **Btn_Notice :**

Adds the new notice to the notice database and refreshes the Txt_Notice TextBox so that the user can add the next notice.

[Back to Contents](#)

Notification_User_Control functionalities:

Widgets:

- **Buttons:**

Each department is allocated a button for notifications of their respective faculty members.

- ✕ CSE
- ✕ Mathematics
- ✕ ECE
- ✕ EEE
- ✕ Mechanical
- ✕ Chemical
- ✕ Biotechnology
- ✕ Design

On clicking any one of these buttons, the user is directed to “*Form_Notification*” where the notifications are dynamically placed on a new window.

[Back to Contents](#)

Notice_List_User_Control functionalities:

Widgets:

- **TextBox:**

→ Txt1(x):

This denotes the dynamically allocated TextBoxes, where x is concatenated to the name of the TextBox from the number of entry in the notice database that is read at the runtime.

- **Buttons:**

→ Disapp(x) (Delete button) :

This denotes the dynamically allocated Delete Buttons, where x is concatenated to the name of the Button from the number of entry in the notice database that is read at the runtime.

- **Functions:**

- **Notice_List_User_Control_Load:**

This function adds the username, date and time of every entry of the notice to the start of each TextBox dynamically allocated.

- **OutTextAdder:**

It adds the notice text in the respective entry of notice database.

It also sets the properties of the dynamically allocated TextBoxes and the Delete Buttons, like the Font Style, height, width to name a few.

- **UpdateFunc2:**

It deletes the notice entry from the database once the user clicks on the delete button for the respective entry.

It also shows the messageBox *"Notice Removed"*, once the removal is complete.

[Back to Contents](#)

View_Teacher_User_Control Functionalities:

Widgets:

- **Buttons:**

- **View_CV:**

Directs to the Webpage/CV link provided in the database. In case of no entry, messageBox "CV not added to Faculty" is displayed on the screen.

- **Apply:**

This hides the current form and directs the user to Form_Internship so that the user can apply for internship under the professor of his/her choice.

- **Labels and TextBox:**

Information is entered according to the respective label names

- ❖ Name
- ❖ Department
- ❖ Research Tag
- ❖ Research Interests
- ❖ Current Courses Code
- ❖ Current Course
- ❖ DOB
- ❖ Room Number
- ❖ Contact Number
- ❖ Email Address

- **PictureBox:**

→ **ImagePictureBox2:**

This displays the profile picture of the selected faculty member.

- **Functions:**

→ ***View_Teacher_User_Control_Load:***

This displays the information of the selected faculty member from the faculty database in the respective textBoxes against their labels.

[Back to Contents](#)

Interested_Students_User_Control Functionalities:

Widgets:

- **TextBox:**

→ **Txt1(x):**

This denotes the dynamically allocated TextBoxes, where x is concatenated to the name of the TextBox from the number of entry in the StudentAppTable database that is read at the runtime.

- **Buttons:**

- **Disapp(x)** (Disapprove button) :

- This denotes the dynamically allocated Disapprove Buttons, where x is concatenated to the name of the Button from the number of entry in the Interested Student database that is read at the runtime.

- **App(x)** (Approve button) :

- This denotes the dynamically allocated Approve Buttons, where x is concatenated to the name of the Button from the number of entry in the Interested Student database that is read at the runtime.

- **Functions:**

- **Interested_Students_User_Control_Load:**

- This function stores the name, email, college, etc. of every entry of the interested student to respectively named strings.

- **OutTextAdder:**

- It adds the text stored in the strings from the respective entry of Interested student database to respective textBoxes txt1(x).

- It also sets the properties of the dynamically allocated TextBoxes and the Disapp and App Buttons, like the Font Style, height, width to name a few.

- **UpdateFunc2:**

- It deletes the Interested Student entry from the database once the user clicks on the Disapprove button for the respective entry.

- It also shows the messageBox "*Student Removed*", once the removal is complete.

- **UpdateFunc:**

- It sends an Email to the Email ID of the applicant provided in the database when the user clicks on the Approve button for the respective entry.

It also shows the messageBox *“Mail Sent”*, once the removal is complete.

[Back to Contents](#)

Form_Internship functionalities:

Widgets:

- **Buttons:**

- **Btn_Submit:**

Creates an entry in the Student database based on the entered information (like Full Name, Email, Degree, Branch, link to CV, etc.). Furthermore, it checks if any discrepancies are present in the entered data.

On a successful entry, a *“registration successful”* message box appears on the screen.

- **Btn_Exit:**

Closes the Internship Form and opens up the Main Form

- **Labels and TextBox:**

Information is entered corresponding to the respective label names

- ❖ Name
- ❖ Email
- ❖ College Name
- ❖ Degree Pursuing
- ❖ Branch
- ❖ Current Year
- ❖ Resume Link
- ❖ Additional Information

- **PictureBox:**

- **PictureBox1:**

Displays Institute Logo at Top left corner

[Back to Contents](#)

Change_Password_User_Control functionalities:

Widgets:

- **Buttons:**

- **Btn_Submit :**

Initially, it checks if entered input is valid. It then verifies whether entered Password is the current one. If so, it displays the New_Password and Renew_Password Textboxes. If not, an “incorrect password” msgbox pops on the screen.

- **Btn_Update:**

Initially, it checks if entered input is valid. It then confirms if the two textboxes containing the new password match. If so, it updates the info in the Teacher database. Else, a “passwords do not match” error message pops on screen.

- **TextBox:**

- **Current_Password:**

Space allocated to enter the current password.

- **New_Password:**

Space allocated to enter the new password. It is initially hidden.

- **Renew_Password:**

Space allocated to re-enter the new password. It is initially hidden.

[Back to Contents](#)

Edit_Teacher_User_Control functionalities:

Widgets:

- **Buttons:**

- **Btn_Edit_Submit:**

Creates an entry in the Teacher database based on the entered information (like Full Name, Research Interests, Current Courses, Contact Number, etc.). Furthermore, it checks if any discrepancies are present in the entered data.

→ **Browse:**

Enables User to select an image from his local machine. The image is then dragged to the ImagePictureBox and the image file name is stored in the database. If this operation fails, a “Please Drag and Drop image” message is displayed on screen.

- **Labels and TextBox:**

Information is entered according to the respective label names

- ❖ Name
- ❖ Research Interests
- ❖ Current Courses Code
- ❖ Current Course
- ❖ Date Of Birth (DOB)
- ❖ Room Number
- ❖ Contact Number
- ❖ Email Address
- ❖ Add Webpage / CV Link

- **ComboBox:**

→ **Edit_Department:**

User can choose any option from the given options

- ❖ CSE
- ❖ Mathematics
- ❖ ECE
- ❖ EEE
- ❖ Mechanical
- ❖ Chemical
- ❖ Biotechnology
- ❖ Design

→ **Edit_Research_Tags:**

User can choose any option from the given options

- ❖ Artificial Intelligence
- ❖ Machine Learning
- ❖ Computer Vision
- ❖ Speech Processing
- ❖ Networking
- ❖ Operating Systems
- ❖ DS and Algorithms
- ❖ Electronic Circuits

• **PictureBox:**

→ **ImagePictureBox:**

User can display his profile picture here

[Back to Contents](#)

Forget_User_Control functionalities:

Widgets:

• **Buttons:**

→ **Btn_Submit1 :**

Initially, it checks if entered input is valid. It then verifies whether entered Username is currently present in database. If so, it displays the Forget_Answer Textbox and the Forget_Question Combobox. If not, an “Enter correct username” msgbox pops on the screen.

→ **Btn_Submit2:**

It checks whether the value in Forget_Answer textbox matches the corresponding security question. If so, it displays the Forget_Password textbox. Else, a “Wrong Answer” message box is displayed .

→ Btn_Submit3 :

Initially, it checks if entered input is valid. If so, it updates the Teacher Database with the new value of password.

- **TextBox:**

→ Forget_Username:

Space allocated to enter the current password.

→ Forget_Answer:

Space allocated to enter the new password. It is initially hidden.

→ Forget_Password:

Space allocated to re-enter the new password. It is initially hidden.

- **ComboBox:**

→ Forget_Question:

User can choose any option from the given options

- ✕ What was your favorite place to visit as a child?
- ✕ In what city were you born?
- ✕ What is the name of your first grade teacher?
- ✕ What is the name of your pet?

*It is initially hidden.

[Back to Contents](#)

Form_Notification functionalities:

On loading the form, all the notices of the selected department are displayed in Richtextboxes. Each Notice mentions the Name of the issuer, along with date and time.

Widgets:

- **Buttons:**

- **Btn_Go_Back:**

- Closes the Form_Notification form and opens up the Form_Main.

- **Functions:**

- **outTextAdder:**

- This function is used to dynamically allocate new richtextboxes.

- **TextBox:**

- **Txt1:**

- It is a dynamically allocated richtextbox containing the Notice information.

[Back to Contents](#)

Admin_User_Control functionalities:

Widgets:

- **TextBox:**

- **Admin_Password:**

- Space allocated to enter the administrator's password.

- **Txt1:**

- It is a dynamically allocated richtextbox containing the new user's registration details.

- **Buttons:**

- **Btn_Submit:**

- Initially, it checks if entered input is valid. It then verifies whether entered admin password is the correct password stored in database. If so, it displays all pending registrations in

Richtextboxes, replete with all the new users details. If not, an “Incorrect password” msgbox pops on the screen.

→ **App:**

This is a dynamically allocated button, created when function outTextAdder is executed. On selection, function updateFunc is executed.

→ **Disapp:**

This is a dynamically allocated button, created when function outTextAdder is executed. On selection, function updateFunc2 is executed.

- **Functions:**

→ **outTextAdder:**

This function is used to dynamically allocate new richtextboxes. It creates two new buttons, app and disapp, and also a textbox txt1.

→ **updateFunc:**

This function updates the newly approved entry by changing the “Approved” field in Teacher Database to YES. It then sends a confirmation message via email to the new user. Also, an “Approved” message is displayed on screen.

→ **updateFunc2:**

This function deletes the new entry in Teacher Database. It then sends a message via email to the new user to inform them of the disapproval. Also, a “Disapproved” message is displayed on screen.

[Back to Contents](#)

Form_Encryption functionalities:

Widgets:

- **Functions:**

- **TruncateHash:**

- This function generates the Hash value. It takes key and length as arguments.

- **EncryptPassword:**

- It uses the TruncateHash function to generate the hash value of the password string to generate an encrypted password.

[Back to Contents](#)

Bugs:

- We haven't done proxy settings. You might face problem while sending an email through institute lan.
- Changing images too frequently in Edit_Teacher_User_Control might lead to an error because memory needs time to dispose previously loaded image.

[Back to Contents](#)