

# **Organize - IT!**

AUTHOR :

Aagaman Sharma Pokharel (078BC001)

Ashim Nepal(078BCT020)

Version 0.1

Thu Jan 19 2023

# License

Copyright (C) Organize-IT! - All Rights Reserved.

THE CONTENTS OF THIS PROJECT ARE PROPRIETARY AND CONFIDENTIAL. UNAUTHORIZED COPYING, TRANSFERRING OR REPRODUCTION OF THE CONTENTS OF THIS PROJECT, VIA ANY MEDIUM IS STRICTLY PROHIBITED.

The receipt or possession of the source code and/or any parts thereof does not convey or imply any right to use them for any purpose other than the purpose for which they were provided to you.

The software is provided "AS IS", without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose and non-infringement. In no event shall the authors or copyright holders be liable for any claim, damages or other liability, whether in an action of contract, tort or otherwise, arising from, out of or in connection with the software or the use or other dealings in the software.

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

# **Organize-IT-Facial\_recognition\_attendance**

This is the Facial Recognition domain covered Organize-It! group of companies.

## **Team members**

Aagaman Sharma Pokharel (PUL078BCT001)

Ashim Nepal (PUL078BCT020)

## **Installation**

Just copy the debug folder inside the x64 folder and paste in a location of your device.

## **Screenshots**

## **Running the program**

Run `gtk_c_gui_experimental.exe` present inside `x64/Debug` folder after copying the debug folder in your device.

# Module Index

## Modules

Here is a list of all modules:

attendance_window_public.c .....	7
ErrorMessages.c .....	11
gtk_c_gui_main.c .....	12
gtk_dashboard_gui.c .....	15
Login_normal.c .....	18
loginwithgoogle.c .....	19
post__basic_file_to_database.c .....	20
SIGNUP_NORMAL.c .....	22

# Data Structure Index

## Data Structures

Here are the data structures with brief descriptions:

<b>MemoryStruct</b>	.....23
<b>StudentData</b>	.....24

# File Index

## File List

Here is a list of all documented files with brief descriptions:

D:/Organize-IT!/gtk_c_gui/attendance_window_public.c .....	25
D:/Organize-IT!/gtk_c_gui/attendance_window_public.h .....	28
D:/Organize-IT!/gtk_c_gui/ErrorMessage.c .....	29
D:/Organize-IT!/gtk_c_gui/ErrorMessage.h .....	30
D:/Organize-IT!/gtk_c_gui/gtk_c_gui.h .....	31
D:/Organize-IT!/gtk_c_gui/gtk_c_gui_main.c .....	32
D:/Organize-IT!/gtk_c_gui/gtk_dashboard_gui.c .....	34
D:/Organize-IT!/gtk_c_gui/gtk_dashboard_gui.h .....	37
D:/Organize-IT!/gtk_c_gui/Img_to_base64.c .....	38
D:/Organize-IT!/gtk_c_gui/img_to_base64.h .....	39
D:/Organize-IT!/gtk_c_gui/Login_normal.c .....	40
D:/Organize-IT!/gtk_c_gui/LOGIN_NORMAL.h .....	41
D:/Organize-IT!/gtk_c_gui/loginwithgoogle.c .....	42
D:/Organize-IT!/gtk_c_gui/loginwithgoogle.h .....	44
D:/Organize-IT!/gtk_c_gui/post__basic_file_to_database.c .....	45
D:/Organize-IT!/gtk_c_gui/post_basic_file_to_database.h .....	47
D:/Organize-IT!/gtk_c_gui/resource.h .....	48
D:/Organize-IT!/gtk_c_gui/SIGNUP_Normal.h .....	49

# Module Documentation

## attendance\_window\_public.c

### Data Structures

#### struct StudentDataEnumerations

- enum { COL\_NAME = 0, COL\_ROLL\_NO, COL\_TOGGLE1, COL\_TOGGLE2, COL\_DAYS\_PRESENT, COL\_CLASS\_DAYS, NUM\_COLS = 6 }

### Functions

- int **change\_uri\_to\_localpath\_for\_present\_photo** (gchar \*file\_uri)  
*Changes uri of present photo to local path.*
- int **change\_uri\_to\_localpath\_for\_reference\_photo** (gchar \*file\_uri)  
*Changes uri of reference photo to local path.*
- void **facetoken\_present\_image** ()  
*Sends request to the api to send back the facetoken for the present image.*
- void **facetoken\_ref\_image** ()  
*Sends request to the api to send back the facetoken for the reference image.*
- float **attendance** ()  
*Sends request to the api to compare the photos and send back the confidence rate which determines by how much the faces match with each other.*
- void **remove\_present\_photo** ()  
*Removes present photo and allows enter to enter a new present photo where present photo is the photo of the student at that instant.*
- void **remove\_refererence\_photo** ()
- void **remove\_both\_photos** ()  
*Removes both present and reference photos so that user can begin fresh.*
- void **on\_response\_for\_present\_photo** (GtkNativeDialog \*native, int response)  
*Takes present photo as input an call float **attendance()** to carry out attendance.*
- void **on\_response\_for\_reference\_photo** (GtkNativeDialog \*native, int response)  
*Takes reference photo as input.*
- void **on\_response\_for\_reference\_photo\_all\_at\_once** (GtkNativeDialog \*native, int response)  
*Takes all reference photos at once.*
- int **compare\_students** (const void \*a, const void \*b)  
*Sorts names in response to qsort.*

- void **action\_clbk\_for\_reference\_photo\_all\_at\_once** (GSimpleAction \*simple\_action, G\_GNUC\_UNUSED GVariant \*parameter, gpointer \*data)  
*Opens file chooser dialog for taking reference photos all at once.*
- char \* **sort\_names\_and\_send\_to\_db** (char \*filepath\_for\_txt\_containing\_name, char \*email, char \*classname, int number\_of\_students)  
*Sort the names from the txt file entered by user and send to the database.*
- void **activate\_attendance\_window\_for\_taking\_attendance\_demo** (char \*email)  
*Activates demo classroom when user clicks demo classroom button.*
- void **update\_db\_with\_updated\_data** ()  
*Updates the database when the user clicks save button.*
- void **increase\_class\_day\_by\_1** ()  
*Increases class days by 1 where class days is the number of days the class was conducted.*
- void **decrease\_class\_day\_by\_1** ()  
*decreases class days by 1*
- int \* **make\_all\_values\_minus\_one** ()  
*Makes all the values of the values[i] equal to -1.*
- void **activate\_attendance\_window\_for\_taking\_attendance** (char \*email, char \*\*unique\_id, char \*classname, int number\_of\_classroom\_created\_by\_user)  
*Activates the specific classroom as the resposne provided by the user.*
- void **make\_every\_one\_of\_uri\_null** ()  
*makes every one of the uri null*

## Variables

- GtkWidget \* **tree\_view**
- GFile \* **file\_for\_present\_photo**
- GFile \* **file\_for\_reference\_photo**
- char \* **uri\_for\_present\_photo**
- char \* **uri\_for\_reference\_photo**
- gchar \* **uris\_for\_reference\_photos** [100]
- char \* **unique\_id\_for\_classroom**
- char \* **email\_to\_pass\_around**
- int **number\_of\_students**
- GtkWidget \* **window5**
- char \* **face\_token1** = (char\*)"init"
- char \* **face\_token2** = (char\*)"init"

---

## Detailed Description

This is the window where the teacher will actually take attendance

---



## Function Documentation

**void activate\_attendance\_window\_for\_taking\_attendance (char \* email, char \*\* unique\_id, char \* classname, int number\_of\_classroom\_created\_by\_user)**

Activates the specific classroom as the response provided by the user.

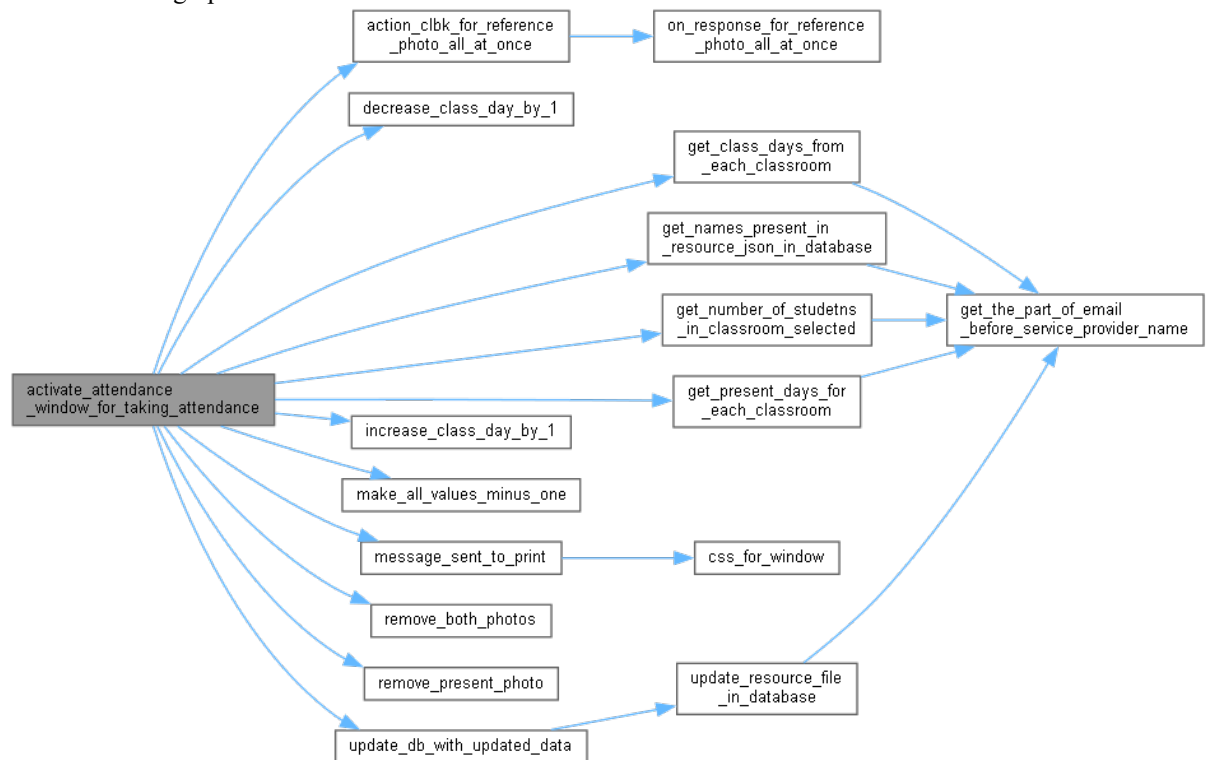
---

\*\*\* Menu Bar

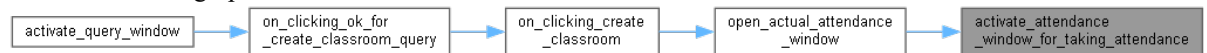
\*\*\* Network\_Menu

\*\*\* Create Connect and Disconnect Actions

Here is the call graph for this function:



Here is the caller graph for this function:



**void activate\_attendance\_window\_for\_taking\_attendance\_demo (char \* email)**

Activates demo classroom when user clicks demo classroom button.

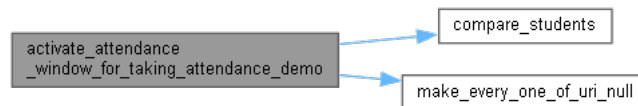
---

\*\*\* Menu Bar

\*\*\* Network\_Menu

\*\*\* Create Connect and Disconnect Actions

Here is the call graph for this function:



Here is the caller graph for this function:



**int change\_uri\_to\_localpath\_for\_present\_photo (gchar \* *file\_uri*)**

Changes uri of present photo to local path.

Here is the caller graph for this function:



**int change\_uri\_to\_localpath\_for\_reference\_photo (gchar \* *file\_uri*)**

Changes uri of reference photo to local path.

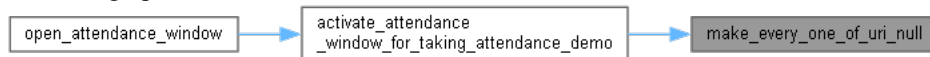
Here is the caller graph for this function:



**void make\_every\_one\_of\_uri\_null ()**

makes every one of the uri null

Here is the caller graph for this function:

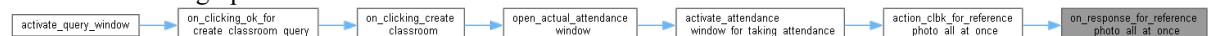


**void on\_response\_for\_reference\_photo\_all\_at\_once (GtkNativeDialog \* *native*, int *response*)**

Takes all reference photos at once.

*open\_file(filename); /*

Here is the caller graph for this function:



## ErrorMessages.c

### Functions

- void **passwords\_do\_not\_match** ()  
*Displays passwords do not match prompt.*
- void **signup\_successful** ()  
*Displays signup\_successful message.*
- void **invalid\_email\_id** ()  
*Displays invalid email id prompt.*
- void **error\_message\_sent\_by\_api** (char \*message)  
*Displays any message sent by api.*
- void **message\_sent\_to\_print** (char \*message)  
*Displays any message sent to print.*

---

### Detailed Description

This file contains features that allows the new user to sign up

## gtk\_c\_gui\_main.c

### Functions

- **int css\_for\_window** (GtkWidget \*window)  
*sets appearance properties for the text labels present in titlebar and sets appearance properties for window*
- **int css\_for\_buttons** (GtkWidget \*button)  
*sets appearance properties for the text labels present in buttons*
- **void change\_focus\_func** (GtkWidget \*Widget1, GtkWidget \*Widget2)
- **char \* return\_email** ()
- **char \* return\_password** ()
- **void invalid\_input** ()  
*displays invalid input message when the user enters any invalid data*
- **int css\_for\_label** (GtkWidget \*label)
- **void clear\_entry** ()  
*clears entry of the initial window once the user is logged-in*
- **void clear\_entry\_for\_signupWindow** ()  
*clears entry of the sign-up window once a new user is signed-up*
- **char \* return\_name** ()
- **char \* return\_text** ()
- **int return\_valueresult** ()
- **void under\_development** ()
- **int check\_email\_validity** ()  
*checks for the validity of email using "abstract" api*
- **void call\_to\_signup\_normal** ()
- **void signup\_normal\_window** ()  
*creates a signup query window where user is prompted to enter name, email, password, and confirm the password*
- **int main** (int argc, char \*\*argv)

### Variables

- **gchar \* email**
- **gchar \* password**
- **gchar \* confirmPassword**
- **gchar \* name**
- **GtkWidget \* window**
- **GtkWidget \* entry\_1**
- **GtkWidget \* entry\_2**
- **GtkWidget \* entry\_0**
- **GtkWidget \* entry**
- **GtkWidget \* entry2**
- **GtkWidget \* entry\_3**
- **GtkApplication \* app**

- `char * text`
- `int valueresult`

---

## Detailed Description

This is the initial window where users will be able to login and signup. Main file of the project.

---

## Function Documentation

### **int check\_email\_validity ()**

checks for the validity of email using "abstract" api

Here is the caller graph for this function:



### **void clear\_entry ()**

clears entry of the initial window once the user is logged-in

Here is the caller graph for this function:



### **void clear\_entry\_for\_signupWindow ()**

clears entry of the sign-up window once a new user is signed-up

Here is the caller graph for this function:



### **int css\_for\_buttons (GtkWidget \* button)**

sets apperance properties for the text labels present in buttons

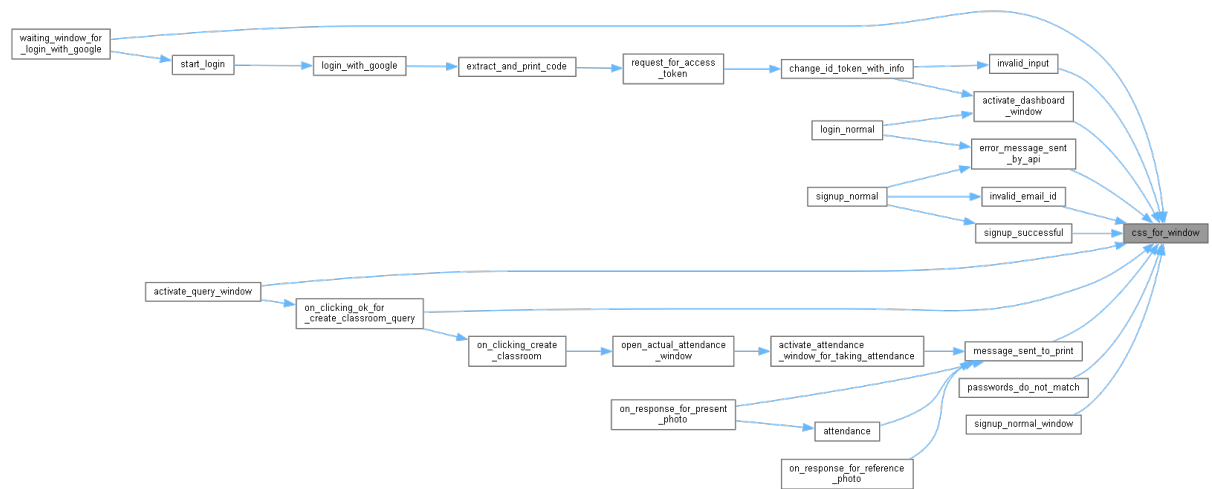
Here is the caller graph for this function:



### **int css\_for\_window (GtkWidget \* window)**

sets appearance properties for the text labels present in titlebar and sets appearance properties for window

Here is the caller graph for this function:



### void invalid\_input ()

displays invalid input message when the user enters any invalid data

Here is the call graph for this function:



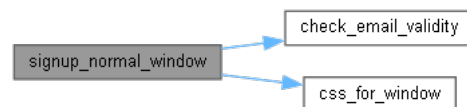
Here is the caller graph for this function:



### void signup\_normal\_window ()

creates a signup query window where user is prompted to enter name, email, password, and confirm the password

Here is the call graph for this function:



## gtk\_dashboard\_gui.c

### Functions

- void **activate\_dashboard\_window** (char \*email\_sent\_by\_main\_file)  
*activates dashboard for the user from where user can enter different windows*
- void **activate\_profile\_window** ()  
*activates the profile window <<<<<—under construction—>>>>>*
- void **open\_attendance\_window** ()  
*Opens demo attendance window.*
- void **activate\_query\_window** ()  
*Activates query window for taking details of the classroom to be created.*
- void **on\_response\_for\_txt\_file** (GtkNativeDialog \*native, int response)  
*This function is executed when dialog box is opened for taking the txt file input. This function takes the uri of text file from user and passes it to another function to turn it into local path.*
- void **open\_parent3** ()  
*opens parent window3 when child is closed*
- void **open\_parent2** ()  
*opens parent window2 when window3 is closed*
- void **open\_parent1** ()  
*opens parent window when window2 is closed*
- void **on\_clicking\_ok\_for\_create\_classroom\_query** ()  
*Describes the behavior of program on clicking create classroom query.*
- void **save\_basic\_details** ()  
*saves basic details of the user to realtime database provided by firebase. Basic details include: user email, no of classrooms created, current class status, unique\_id of classroom if any classroom is created.*
- void **on\_clicking\_create\_classroom** ()  
*Creates new buttons for new classroom when user creates new classroom. Also keeps track of the no of classroom created by user and creates the classroom whenever the user logs into the account.*
- void **open\_actual\_attendance\_window** (GtkWidget \*button)  
*opens actual attendance window for taking attendance*
- int **change\_uri\_to\_localpath\_for\_txt\_file** (gchar \*file\_uri)
- void **add\_metadata\_to\_main\_json** (json\_t \*root)  
*This function is created to create a "Json Payload" which will be delivered to firebase realtime database.*

- **int convert\_to\_int ()**  
*converts no of students to integer as it is initially entered as char\**
- **void activate\_print\_window ()**

## Variables

- **GtkWidget \* window2**
- **GtkWidget \* window3**
- **GtkWidget \* window4**
- **GtkWidget \* window6**
- **GtkWidget \* grid\_for\_attendance\_window**
- **gchar \* classname = (gchar\*)""**
- **GtkWidget \* entry**
- **GtkWidget \* entry2**
- **char \* no\_of\_students =(char\*)""**
- **const char displayNameForProfile [50]**
- **int initial\_no\_of\_classes = 0**
- **int class\_status = 0**
- **int number\_of\_students**
- **int number\_of\_classroom\_created\_by\_user = 0**
- **int depth\_of\_new\_classroom = 4**
- **char namep [50]**
- **FILE \* file\_for\_txt\_file**
- **gchar \* uri\_for\_txt\_file = (gchar\*)""**
- **GtkWidget \* toggle**
- **char user\_email [50]**
- **char \*\* return\_response**
- **char \*\* unique\_id**
- **char email\_to\_be\_sent\_back\_to\_db [50]**

---

## Detailed Description

This is the dashboard window where users will be able to access different functionalities of the program.

---

## Function Documentation

**void activate\_dashboard\_window (char \* *email\_sent\_by\_main\_file*)**

activates dashboard for the user from where user can enter different windows

---

\*\*\*

\*\*\* Menu Bar

\*\*\* Network\_Menu

\*\*\* Server\_Menu

\*\*\*\*\*  
\*\*\*\*\*

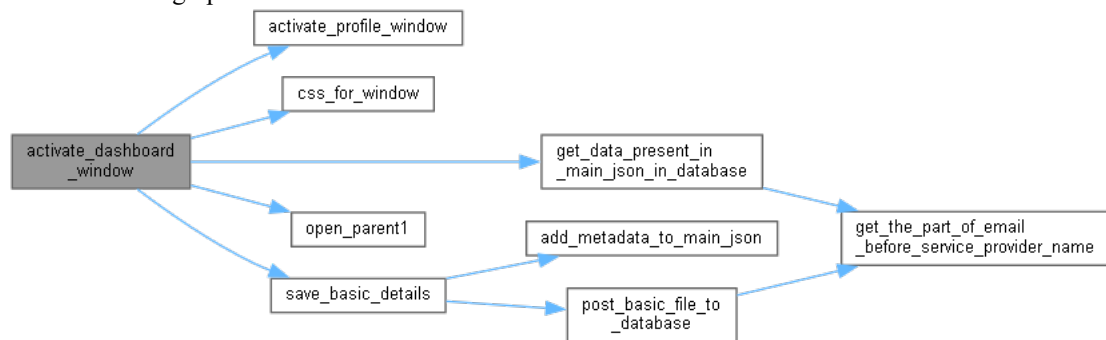


\*\*\*\*\*  
\*\*\*\*\*

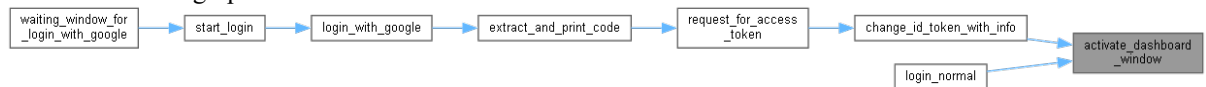
\*\*\* Create Connect and Disconnect Actions  
\*\*\* Add them to the ActionMap  
\*\*\* Connect them to the activate Signal  
\*\*\* Create the Connect Item  
\*\*\* Create the Disconnect Item  
\*\*\* Create the Disconnect Item

\*\*\* Clean  
\*\*\* Clean

Here is the call graph for this function:



Here is the caller graph for this function:

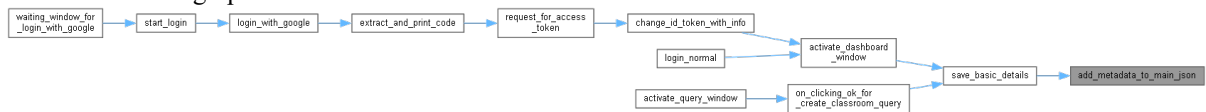


**void add\_metadata\_to\_main\_json (json\_t \* root)**

This function is created to create a "Json Payload" which will be delivered to firebase realtime database.

This function is create to convert uri extracted by g\_file\_get\_uri() to local path. Uri looks like this : file:///C:/Users/user/folder/json\_draft.txt . Local path looks like this : C:\Users\user\folder\json\_draft.txt.

Here is the caller graph for this function:



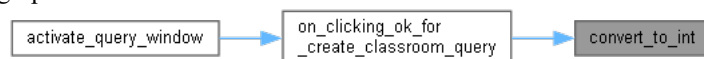
**int convert\_to\_int ()**

converts no of students to integer as it is initially entered as char\*

### Parameters

<i>number_of_students</i>	-> saves the number of students in a classroom created by user in integer format
---------------------------	--

Here is the caller graph for this function:



## Login\_normal.c

### Functions

- `int login_normal ()`  
*logs in the user that is present in the database only if the password matches*
- `char * return_displayName ()`  
*returns display name to the function trying to access the display name*

### Variables

- `gchar * name`
- 

### Detailed Description

This file contains features that allows users to login using email and password

# loginwithgoogle.c

## Macros

- `#define BUFSIZE 1024`

## Functions

- `int login_with_google ()`  
*Initiates logging in process using google and tries to copy the url to clipboard.*
- `gboolean start_login ()`  
*starts login\_with\_google with a certain amount of timeout so that the infinite loop doesnot freeze the program from executing*
- `int extract_and_print_code ()`  
*extracts and prints code, print code statement only useful for devs to check whether the code is working properly or not*
- `int request_for_access_token (char *code)`  
*Requeeqsts rest api endpoint for access token.*
- `int change_id_token_with_info (char *idtoken)`  
*Exchanges id\_token with user info.*
- `void waiting_window_for_login_with_google ()`  
*Initiates waiting window for login\_with\_google.*
- `void change_value_for_choice ()`  
*Changes choice variable to zero which is used to control the infinite loop.*
- `void print_code (char *code)`

## Variables

- `int choice`
- `int j = 1`
- `LPTSTR lptstr = (LPTSTR)"sample"`
- `GtkWidget * window2`

---

## Detailed Description

This file contains features that allows users to login via the google account already logged in, in their browser



## **Variables**

- char `json_file_name` [20]

---

## **Detailed Description**

This file contains all the functions that will be posting or fetching data to and fromt the database

## SIGNUP\_NORMAL.c

### Functions

- `int signup_normal()`  
*Only signs up the new user if the email is valid.*

---

### Detailed Description

This file contains features that allows the new user to sign up

# Data Structure Documentation

## MemoryStruct Struct Reference

### Data Fields

- `char * memory`
- `size_t size`

---

The documentation for this struct was generated from the following file:

- `D:/Organize-IT!/gtk_c_gui/post__basic_file_to_database.c`

## StudentData Struct Reference

### Data Fields

- char **name** [100]
- int **roll\_no**
- gboolean **toggle1**
- gboolean **toggle2**
- int **days\_present**
- int **class\_days**
- GtkTreeIter \* **iter**

---

The documentation for this struct was generated from the following file:

- D:/Organize-IT!/gtk\_c\_gui/**attendance\_window\_public.c**

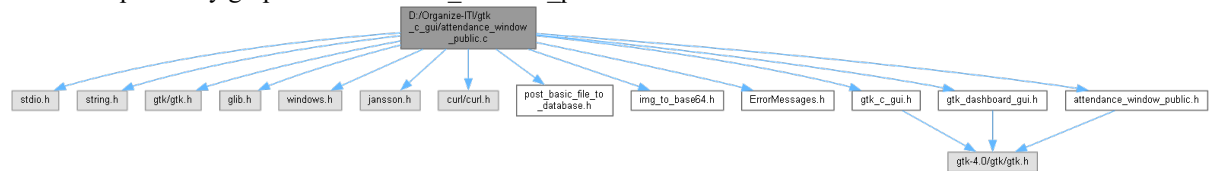


# File Documentation

## D:/Organize-IT!/gtk\_c\_gui/attendance\_window\_public.c File Reference

```
#include <stdio.h>
#include <string.h>
#include <gtk/gtk.h>
#include <glib.h>
#include <windows.h>
#include <jansson.h>
#include <curl/curl.h>
#include "post_basic_file_to_database.h"
#include "img_to_base64.h"
#include "ErrorMessages.h"
#include "gtk_c_gui.h"
#include "gtk_dashboard_gui.h"
#include "attendance_window_public.h"
```

Include dependency graph for attendance\_window\_public.c:



## Data Structures

### struct StudentDataEnumerations

- enum { COL\_NAME = 0, COL\_ROLL\_NO, COL\_TOGGLE1, COL\_TOGGLE2, COL\_DAYS\_PRESENT, COL\_CLASS\_DAYS, NUM\_COLS = 6 }

## Functions

- int **change\_uri\_to\_localpath\_for\_present\_photo** (gchar \*file\_uri)  
*Changes uri of present photo to local path.*
- int **change\_uri\_to\_localpath\_for\_reference\_photo** (gchar \*file\_uri)  
*Changes uri of reference photo to local path.*
- void **facetoken\_present\_image** ()  
*Sends request to the api to send back the facetoken for the present image.*
- void **facetoken\_ref\_image** ()  
*Sends request to the api to send back the facetoken for the reference image.*
- float **attendance** ()  
*Sends request to the api to compare the photos and send back the confidence rate which determines by how much the faces match with each other.*
- void **remove\_present\_photo** ()

*Removes present photo and allows enter to enter a new present photo where present photo is the photo of the student at that instant.*

- void **remove\_refererence\_photo** ()
- void **remove\_both\_photos** ()  
*Removes both present and reference photos so that user can begin fresh.*
- void **on\_response\_for\_present\_photo** (GtkNativeDialog \*native, int response)  
*Takes present photo as input an call float **attendance()** to carry out attendance.*
- void **on\_response\_for\_reference\_photo** (GtkNativeDialog \*native, int response)  
*Takes reference photo as input.*
- void **on\_response\_for\_reference\_photo\_all\_at\_once** (GtkNativeDialog \*native, int response)  
*Takes all reference photos at once.*
- int **compare\_students** (const void \*a, const void \*b)  
*Sorts names in response to qsort.*
- void **action\_clbk\_for\_reference\_photo\_all\_at\_once** (GSimpleAction \*simple\_action, G\_GNUC\_UNUSED GVariant \*parameter, gpointer \*data)  
*Opens file chooser dialog for taking reference photos all at once.*
- char \* **sort\_names\_and\_send\_to\_db** (char \*filepath\_for\_txt\_containing\_name, char \*email, char \*classname, int number\_of\_students)  
*Sort the names from the txt file entered by user and send to the database.*
- void **activate\_attendance\_window\_for\_taking\_attendance\_demo** (char \*email)  
*Activates demo classroom when user clicks demo classroom button.*
- void **update\_db\_with\_updated\_data** ()  
*Updates the database when the user clicks save button.*
- void **increase\_class\_day\_by\_1** ()  
*Increases class days by 1 where class days is the number of days the class was conducted.*
- void **decrease\_class\_day\_by\_1** ()  
*decreases class days by 1*
- int \* **make\_all\_values\_minus\_one** ()  
*Makes all the values of the values[i] equal to -1.*
- void **activate\_attendance\_window\_for\_taking\_attendance** (char \*email, char \*\*unique\_id, char \*classname, int number\_of\_classroom\_created\_by\_user)  
*Activates the specific classroom as the resposne provided by the user.*
- void **make\_every\_one\_of\_uri\_null** ()  
*makes every one of the uri null*

## Variables

- GtkWidget \* **tree\_view**
  - GFile \* **file\_for\_present\_photo**
  - GFile \* **file\_for\_reference\_photo**
  - char \* **uri\_for\_present\_photo**
  - char \* **uri\_for\_reference\_photo**
  - gchar \* **uris\_for\_reference\_photos** [100]
  - char \* **unique\_id\_for\_classroom**
  - char \* **email\_to\_pass\_around**
  - int **number\_of\_students**
  - GtkWidget \* **window5**
  - char \* **face\_token1** = (char\*)"init"
  - char \* **face\_token2** = (char\*)"init"
- 

## Detailed Description

### Author

Organize-IT!

### Date

2023

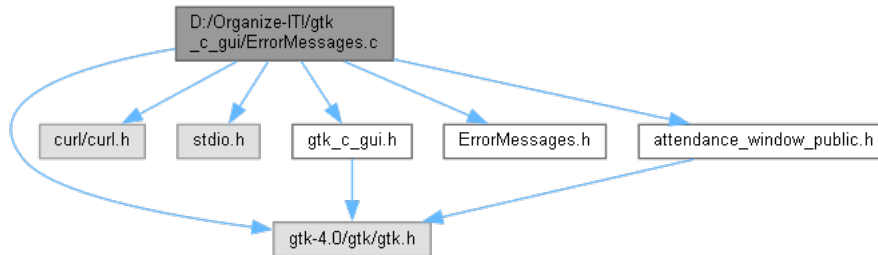
## attendance\_window\_public.h

```
1 #ifndef ATTENDANCE_WINDOW_PUBLIC
2 #define ATTENDANCE_WINDOW_PUBLIC
3 #include<gtk-4.0/gtk/gtk.h>
4
5 void activate_attendance_window_for_taking_attendance(char* email, char** unique_id,
char* classname,int number_of_classroom_created_by_user);
6 static void toggle_toggled(GtkWidget* widget, gpointer data);
7 static void action_clbk_for_reference_photo(GSimpleAction* simple_action,
G_GNUC_UNUSED GVariant* parameter, gpointer* data);
8 static void action_clbk_for_present_photo(GSimpleAction* simple_action, G_GNUC_UNUSED
GVariant* parameter, gpointer* data);
9 int compare_students(const void* a, const void* b);
10 void on_response_for_present_photo(GtkNativeDialog* native, int response);
11 char* sort_names_and_send_to_db(char* filepath_for_txt_containing_name, char* email,
char* classname, int number_of_students);
12
13 void activate_attendance_window_for_taking_attendance_demo(char* email);
14
15 GtkWidget* window5;
16 #endif
```

## D:/Organize-IT!/gtk\_c\_gui/ErrorMessage.c File Reference

```
#include <gtk-4.0/gtk/gtk.h>
#include <curl/curl.h>
#include <stdio.h>
#include "gtk_c_gui.h"
#include "ErrorMessage.h"
#include "attendance_window_public.h"
```

Include dependency graph for ErrorMessage.c:



## Functions

- void **passwords\_do\_not\_match** ()  
*Displays passwords do not match prompt.*
- void **signup\_successful** ()  
*Displays signup\_successful message.*
- void **invalid\_email\_id** ()  
*Displays invalid email id prompt.*
- void **error\_message\_sent\_by\_api** (char \*message)  
*Displays any message sent by api.*
- void **message\_sent\_to\_print** (char \*message)  
*Displays any message sent to print.*

---

## Detailed Description

### Author

Organize-IT!

### Date

2023

## ErrorMessages.h

```
1 #ifndef ERROR_MESSAGES
2 #define ERROR_MESSAGES
3 void passwords_do_not_match();
4 void signup_successful();
5 void invalid_email_id();
6 void error_message_sent_by_api(char* message);
7 void message_sent_to_print(char* message);
8 #endif
```

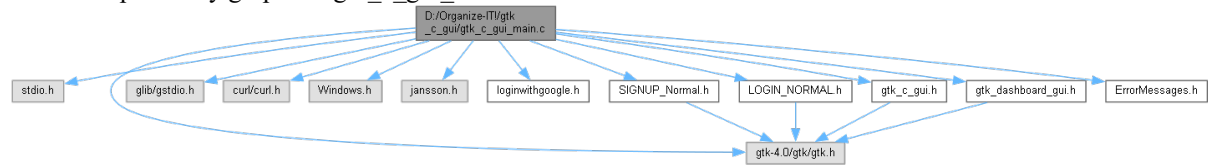
## gtk\_c\_gui.h

```
1 #ifndef GTK_C_GUI
2 #define GTK_C_GUI
3 #include<gtk-4.0/gtk/gtk.h>
4
5 //function declarations
6 static void activate(GtkApplication* app, gpointer user_data);
7 int css_for_window(GtkWidget* window);
8 int css_for_buttons(GtkWidget* button);
9 static void activate_entry_for_password();
10 static void activate_entry_for_email();
11 void change_focus_func(GtkWidget* Widget1, GtkWidget* Widget2);
12 char* return_email();
13 char* return_password();
14 void invalid_input();
15 char* return_password();
16 char* return_email();
17 int css_for_label(GtkWidget* label);
18 static void activate_entry_for_name();
19 static void activate_entry_for_password_from_signup();
20 void clear_entry();
21 static void activate_entry_for_email_from_signup();
22 void clear entry for signupWindow();
23 char* return_name();
24 char* return_text();
25 int return_valueresult();
26
27
28 //global variable and not to be changed manually
29 gchar* email;
30 gchar* password;
31 GtkWidget* window;
32 GtkApplication* app;
33
34 #endif
```

## D:/Organize-IT!/gtk\_c\_gui/gtk\_c\_gui\_main.c File Reference

```
#include <stdio.h>
#include <gtk-4.0/gtk/gtk.h>
#include <glib/gstdio.h>
#include <curl/curl.h>
#include <Windows.h>
#include <jansson.h>
#include "loginwithgoogle.h"
#include "SIGNUP_Normal.h"
#include "LOGIN_NORMAL.h"
#include "gtk_c_gui.h"
#include "gtk_dashboard_gui.h"
#include "ErrorMessages.h"
```

Include dependency graph for gtk\_c\_gui\_main.c:



## Functions

- **int `css_for_window`** (GtkWidget \*window)  
*sets appearance properties for the text labels present in titlebar and sets appearance properties for window*
- **int `css_for_buttons`** (GtkWidget \*button)  
*sets appearance properties for the text labels present in buttons*
- **void `change_focus_func`** (GtkWidget \*Widget1, GtkWidget \*Widget2)
- **char \* `return_email`** ()
- **char \* `return_password`** ()
- **void `invalid_input`** ()  
*displays invalid input message when the user enters any invalid data*
- **int `css_for_label`** (GtkWidget \*label)
- **void `clear_entry`** ()  
*clears entry of the initial window once the user is logged-in*
- **void `clear_entry_for_signupWindow`** ()  
*clears entry of the sign-up window once a new user is signed-up*
- **char \* `return_name`** ()
- **char \* `return_text`** ()
- **int `return_valueresult`** ()
- **void `under_development`** ()
- **int `check_email_validity`** ()  
*checks for the validity of email using "abstract" api*
- **void `call_to_signup_normal`** ()



- void **signup\_normal\_window** ()  
*creates a signup query window where user is prompted to enter name, email, password, and confirm the password*
- int **main** (int argc, char \*\*argv)

## Variables

- gchar \* **email**
- gchar \* **password**
- gchar \* **confirmPassword**
- gchar \* **name**
- GtkWidget \* **window**
- GtkWidget \* **entry\_1**
- GtkWidget \* **entry\_2**
- GtkWidget \* **entry\_0**
- GtkWidget \* **entry**
- GtkWidget \* **entry2**
- GtkWidget \* **entry\_3**
- GtkApplication \* **app**
- char \* **text**
- int **valueresult**

---

## Detailed Description

### Author

Organize-IT!

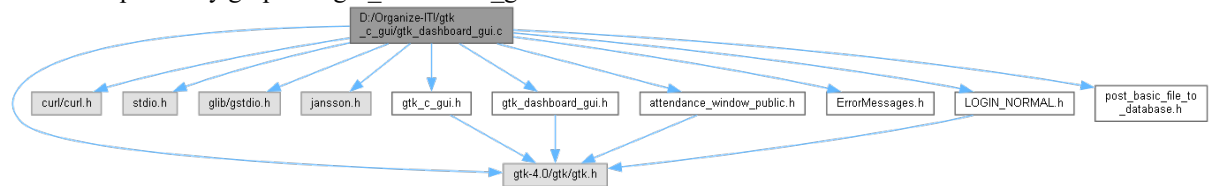
### Date

2023

## D:/Organize-IT!/gtk\_c\_gui/gtk\_dashboard\_gui.c File Reference

```
#include <gtk-4.0/gtk/gtk.h>
#include <curl/curl.h>
#include <stdio.h>
#include <glib/gstdio.h>
#include <jansson.h>
#include "gtk_c_gui.h"
#include "gtk_dashboard_gui.h"
#include "attendance_window_public.h"
#include "ErrorMessages.h"
#include "LOGIN_NORMAL.h"
#include "post_basic_file_to_database.h"
```

Include dependency graph for `gtk_dashboard_gui.c`:



## Functions

- **void activate\_dashboard\_window (char \*email\_sent\_by\_main\_file)**  
*activates dashboard for the user from where user can enter different windows*
- **void activate\_profile\_window ()**  
*activates the profile window <<<<<—under construction—>>>>>*
- **void open\_attendance\_window ()**  
*Opens demo attendance window.*
- **void activate\_query\_window ()**  
*Activates query window for taking details of the classroom to be created.*
- **void on\_response\_for\_txt\_file (GtkNativeDialog \*native, int response)**  
*This function is executed when dialog box is opened for taking the txt file input. This function takes the uri of text file from user and passes it to another function to turn it into local path.*
- **void open\_parent3 ()**  
*opens parent window3 when child is closed*
- **void open\_parent2 ()**  
*opens parent window2 when window3 is closed*
- **void open\_parent1 ()**  
*opens parent window when window2 is closed*
- **void on\_clicking\_ok\_for\_create\_classroom\_query ()**  
*Describes the behavior of program on clicking create classroom query.*

- **void save\_basic\_details ()**  
*saves basic details of the user to realtime database provided by firebase. Basic details include: user email, no of classrooms created, current class status, unique\_id of classroom if any classroom is created.*
- **void on\_clicking\_create\_classroom ()**  
*Creates new buttons for new classroom when user creates new classroom. Also keeps track of the no of classroom created by user and creates the classroom whenever the user logs into the account.*
- **void open\_actual\_attendance\_window (GtkWidget \*button)**  
*opens actual attendance window for taking attendance*
- **int change\_uri\_to\_localpath\_for\_txt\_file (gchar \*file\_uri)**
- **void add\_metadata\_to\_main\_json (json\_t \*root)**  
*This function is created to create a "Json Payload" which will be delivered to firebase realtime database.*
- **int convert\_to\_int ()**  
*converts no of students to integer as it is initially entered as char\**
- **void activate\_print\_window ()**

## Variables

- GtkWidget \* **window2**
- GtkWidget \* **window3**
- GtkWidget \* **window4**
- GtkWidget \* **window6**
- GtkWidget \* **grid\_for\_attendance\_window**
- gchar \* **classname** = (gchar\*)""
- GtkWidget \* **entry**
- GtkWidget \* **entry2**
- char \* **no\_of\_students** =(char\*)""
- const char **displayNameForProfile** [50]
- int **initial\_no\_of\_classes** = 0
- int **class\_status** = 0
- int **number\_of\_students**
- int **number\_of\_classroom\_created\_by\_user** = 0
- int **depth\_of\_new\_classroom** = 4
- char **namep** [50]
- FILE \* **file\_for\_txt\_file**
- gchar \* **uri\_for\_txt\_file** = (gchar\*)""
- GtkWidget \* **toggle**
- char **user\_email** [50]
- char \*\* **return\_response**
- char \*\* **unique\_id**
- char **email\_to\_be\_sent\_back\_to\_db** [50]

---

## Detailed Description

**Author**

Organize-IT!

**Date**

2023

## gtk\_dashboard\_gui.h

```
1 #ifndef gtk_dashBoard_gui
2 #define gtk_dashBoard_gui
3 #include<gtk-4.0/gtk/gtk.h>
4
5 void activate dashboard window(char*);
6
7
8
9 #endif
```

```
#include <openssl/bio.h>
#include <openssl/evp.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <windows.h>
#include "img_to_base64.h"
Include dependency graph for Img_to_base64.c:
```



- ### Detailed Description

**Author**

## Organize-IT!

**Date**

2023

## Function Documentation

```
int convert_back_to_jpg (char * base64)
```

Converts base64 format back to image.

```
@defgroup Img_to_base64 Img_to_base64.c This file contains function that converts  
image to base64 format and base4 format back to jpg, will be used for saving images to  
database <<<<<<<<<<<<<<<<<<<-----(under  
construction)----->>>>>>>>>>>>>
```

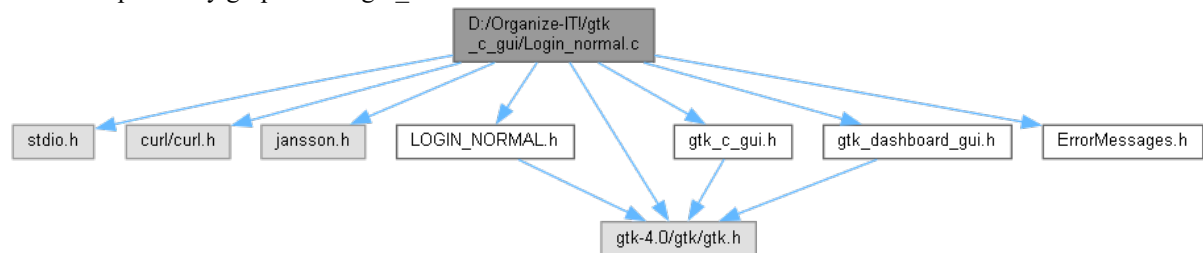
## img\_to\_base64.h

```
1 #ifndef ENCODE_TO_BASE_64_DECODE_FROM_BASE_64
2 #define ENCODE_TO_BASE_64_DECODE_FROM_BASE_64
3
4 int convert_back_to_jpg(char* base64);
5 char* encode_to_base_64(char* uri, char*);
6
7 #endif
```

## D:/Organize-IT!/gtk\_c\_gui/Login\_normal.c File Reference

```
#include <stdio.h>
#include <curl/curl.h>
#include <jansson.h>
#include "LOGIN_NORMAL.h"
#include "gtk-4.0/gtk/gtk.h"
#include "gtk_c_gui.h"
#include "gtk_dashboard_gui.h"
#include "ErrorMessages.h"
```

Include dependency graph for Login\_normal.c:



## Functions

- **int login\_normal ()**  
*logs in the user that is present in the database only if the password matches*
- **char \* return\_displayName ()**  
*returns display name to the function trying to access the display name*

## Variables

- **gchar \* name**

---

## Detailed Description

### Author

Organize-IT!

### Date

2023



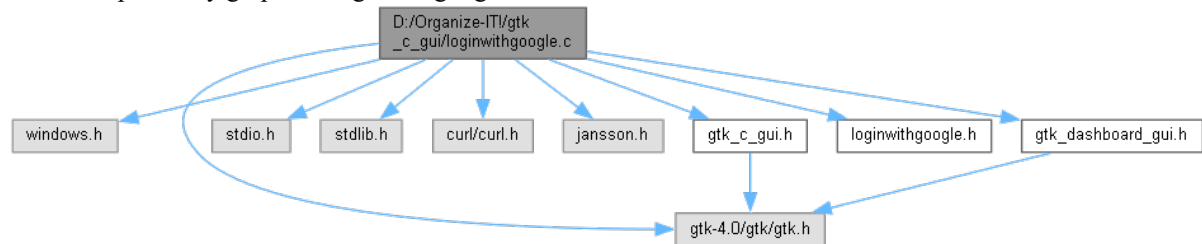
## LOGIN\_NORMAL.h

```
1 #ifndef LOGIN_NORMAL_C
2 #define LOGIN_NORMAL_C
3 #include<gtk-4.0/gtk/gtk.h>
4 int login_normal();
5
6
7 char* return_displayName();
8 gchar* name;
9 #endif
```

## D:/Organize-IT!/gtk\_c\_gui/loginwithgoogle.c File Reference

```
#include <windows.h>
#include <gtk-4.0/gtk/gtk.h>
#include <stdio.h>
#include <stdlib.h>
#include <curl/curl.h>
#include <jansson.h>
#include "gtk_c_gui.h"
#include "loginwithgoogle.h"
#include "gtk_dashboard_gui.h"
```

Include dependency graph for loginwithgoogle.c:



## Macros

- `#define BUFSIZE 1024`

## Functions

- `int login_with_google ()`  
*Initiates logging in process using google and tries to copy the url to clipboard.*
- `gboolean start_login ()`  
*starts login\_with\_google with a certain amount of timeout so that the infinite loop doesnot freeze the program from executing*
- `int extract_and_print_code ()`  
*extracts and prints code, print code statement only useful for devs to check whether the code is working properly or not*
- `int request_for_access_token (char *code)`  
*Requeqsts rest api endpoint for access token.*
- `int change_id_token_with_info (char *idtoken)`  
*Exchanges id\_token with user info.*
- `void waiting_window_for_login_with_google ()`  
*Initiates waiting window for login\_with\_google.*
- `void change_value_for_choice ()`  
*Changes choice variable to zero which is used to control the infinite loop.*
- `void print_code (char *code)`

## Variables

- `int choice`
  - `int j = 1`
  - `LPTSTR lptstr = (LPTSTR)"sample"`
  - `GtkWidget * window2`
- 

## Detailed Description

### Author

Organize-IT!

### Date

2023

## loginwithgoogle.h

```
1 #ifndef login_with_google_app
2 #define login_with_google_app
3
4 int login_with_google(); void waiting_window_for_login_with_google();
5 #endif
```

```
#include <stdio.h>
#include <curl/curl.h>
#include <jansson.h>
#include "gtk_c_gui.h"
#include "post_basic_file_to_database.h"
#include "gtk_dashboard_gui.h"
```

```

graph TD
    A["D:/Organize-IT/IT/gtk_c_gui/post_basic_file_to_database.c"] --> B["stdio.h"]
    A --> C["curl/curl.h"]
    A --> D["jansson.h"]
    A --> E["gtk_c_gui.h"]
    A --> F["post_basic_file_to_database.h"]
    A --> G["gtk_dashboard_gui.h"]
    E --> H["gtk-4.0/gtk/gtk.h"]
    G --> H
  
```

## struct MemoryStructFunctions

- 45

*Fetches names present in resource.json saved in the database.*

- **char \* get\_classname\_only\_from\_resource\_json** (char \*email, char \*unique\_id)  
*Fetches classname only from resource.json saved in firebase realtime database.*
- **int post\_basic\_file\_to\_database** (char \*json\_data, char \*email)  
*Posts basic json payload containing user email, number of classroom created by user, current class status and unique\_id of classrooms if any.*
- **int update\_resource\_file\_in\_database** (char \*json\_data, char \*email, char \*unique\_identifier\_for\_each\_class)  
*Updates the resource.json present in database.*
- **char \* post\_resource\_file\_to\_database** (char \*json\_data, char \*email)  
*Posts resource file to database containing the details of the classroom like students name, number of students.*

## Variables

- **char json\_file\_name** [20]

---

## Detailed Description

### Author

Organize-IT!

### Date

2023

## post\_basic\_file\_to\_database.h

```
1 #ifndef POST_BASIC_FILE_TO_DATABASE
2 #define POST_BASIC_FILE_TO_DATABASE
3
4 int post_basic_file_to_database(char* json_data, char* email);
5 char* post_resource_file_to_database(char* json_data, char* email);
6 int update_resource_file_in_database(char* json_data, char* email, char*
unique_identifier_for_each_class);
7 char** get_data_present_in_main_json_in_database(char *email, int*
number_of_classes_created_by_user, int* class_status);
8 char* get_names_present_in_resource_json_in_database(char* email, char** unique_id,
char* classname, int number of classroom created by user);
9 char* get_classname_only_from_resource_json(char* email, char* unique_id);
10 void delete_data_previously_present_in_main_json_in_database(char* email);
11 int get_number_of_students_in_classroom_selected(char* email, char*
unique_id_for_classroom);
12 int* get_present_days_for_each_classroom(char* email, char* unique_id_for_classroom,
int number_of_students);
13 int get_class_days_from_each_classroom(char* email, char* unique_id_for_classroom);
14
15 #endif
```

## resource.h

```
1 //{NO_DEPENDENCIES}
2 // Microsoft Visual C++ generated include file.
3 // Used by gtk_c_gui_experimental.rc
4
5 // Next default values for new objects
6 //
7 #ifdef APSTUDIO_INVOKED
8 #ifndef APSTUDIO_READONLY_SYMBOLS
9 #define _APS_NEXT_RESOURCE_VALUE        101
10 #define _APS_NEXT_COMMAND_VALUE        40001
11 #define _APS_NEXT_CONTROL_VALUE        1001
12 #define _APS_NEXT_SYMED_VALUE        101
13 #endif
14 #endif
```



## SIGNUP\_Normal.h

```
1 #ifndef SIGNUP_NORMAL_C
2 #define SIGNUP_NORMAL_C
3
4 #include<gtk-4.0/gtk/gtk.h>
5
6 int signup_normal();
7
8 #end
```

