FILE EXPLORER

TEAM MEMBERS:

- Aditya Vadrevu
- Malleswari Tippireddy
- Yaswanth Sen Gunti
- Revanth Jeldi

INTRODUCTION:

As a part of coursework for Operating System Design, we have developed two applications for Project no. 1. One application runs completely on console, while the other is built using GUI. Both the applications are helpful in exploring the files and folders in a computer, hence, we have named our project as **File Explorer**. Our project aims to simplify the tasks of viewing, creating, and duplicating(copy-paste) files and directories. By using this application a user can view and create files and directories with just one click. It eliminates the need to get into multiple subdirectories to create new files.

BACKGROUND:

For a user to view or create a file in a computer, they need to enter multiple subdirectories before they could find it.

IMPLEMENTATION AND EXPERIMENTAL ENVIRONMENT:

File Explorer aims at making the user experience as effortless as possible. All that a user needs to do is enter the file name with its path, and click a button to create it.

Library and Programming Language used:

To build this project we have used GTK, an open-source cross-platform widget toolkit for creating graphical user interfaces using C programming language.

Operating system to test the project:

The application runs on Mac and LINUX operating systems, since it was built on Mac OS. To run the application on a machine, a user is expected to install the GTK library before they execute the code.

Commands to install GTK Library on MacOS:

A User needs to run the following commands to install GTK Library on Mac OS.

```
brew install gtk+3
export PKG_CONFIG_PATH=/usr/local/Cellar/cairo/1.12.16/lib/pkgconfig/
export PKG_CONFIG_PATH=/usr/X11/lib/pkgconfig
```

Test Cases:

A User could run various test scenarios in the app to verify whether they are attaining the expected results.

In the GUI Application,

Scenario-1: User wants to view a File or Folder:

By clicking on the “View Directories and Files”, a user can view all the files, directories and sub-directories of their machine.

Scenario-2: User wants to create a directory:

By clicking on the “Create New Directory”, a user can create a new directory.

Scenario-3: User wants to create a file:

By clicking on the “Create New File”, a user can create a new file.

In the Console based application,

After a user runs the console application, they get to choose from 4 available options,

```
 enter 1 to move into a sub-directory
 enter 2 to copy-paste a file to new file
 enter 3 to create a new directory
 enter 4 to create a new file
```

Scenario-1: User wants move into a sub-directory:

By entering ”1”, and entering the subdirectory name, a user can move into a subdirectory to view the files present.

Scenario-2: User wants copy-paste(duplicate) a file:

By entering ”2”, and entering the names of the file to be copied from and file to be copied to, a user can copy the contents of a file into another.

Scenario-3: User wants to create a file:

By entering ”3”, and entering the subdirectory name, a user can create a new subdirectory.

Scenario-4: User wants to create a file:

By entering ”4”, and entering the subdirectory name, a user can create a new file.

Commands to run the console application:

Since the application is built using C programming language, it first needs to be compiled before it can be executed.

```
Complilation:
gcc fileexplorerproj.c -o output
Execution:
./output
```

Commands to run the GUI application:

Since the application is built using C programming language, it first needs to be compiled before it can be executed.

```
Complilation:
gcc `pkg-config --cflags gtk+-3.0` -o fxoutput
fileexplorergui.c `pkg-config --libs gtk+-3.0`
Execution:
./fxoutput
```

CODE:

Following is the code implementation of Console Application:

DISPLAY SUBDIRECTORIES:

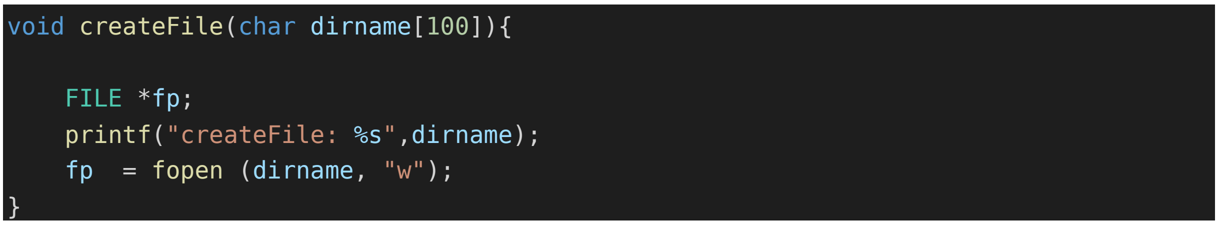
In this we are going to create a child and we are trying to display the subdirectories which are present in present directory.

```
int subdirectoriesdisplay(char *p)
{
    printf("\n\nSub-directories List: \n\n");
    pid_t pid1=0;
    pid1=fork();
    if(pid1==0){
        char *str[]={"ls",p,NULL};
        int res=execvp("ls",str);
        return res;
    }else{
        wait(NULL);
    }
 return 0;
}
```

CREATING NEW DIRECTORY:

```
void createDir(char dirname[100]){
    int check;

    printf("in create dir");
    printf("dirname in cr dir: %s",dirname);
    check = mkdir(dirname,0777);
    printf("\ncheck: %d \n",check);
    if (!check)
        printf("Directory created\n");
    else {
        printf("Unable to create directory\n");
        exit(1);
    }
```

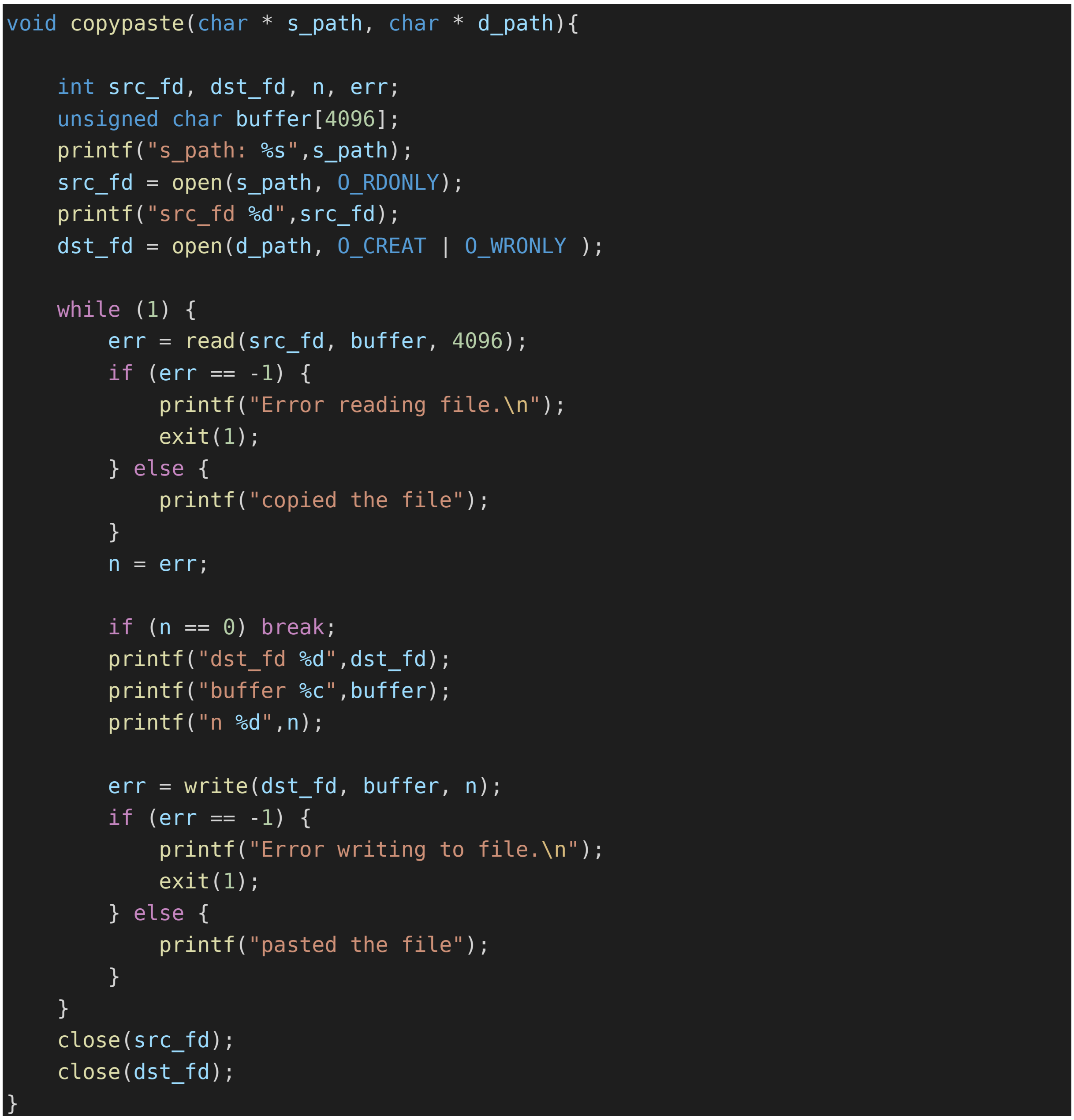
CREATING A NEW FILE:



```
void createFile(char dirname[100]){

    FILE *fp;
    printf("createFile: %s",dirname);
    fp  = fopen (dirname, "w");
}
```

COPYING FILES FROM SRC TO DEST:

In this we are going to copy files from one location to another location. The location can be between directories or in directory. We need to use system calls like open, read, write, close for this task.



```
void copypaste(char * s_path, char * d_path){

    int src_fd, dst_fd, n, err;
    unsigned char buffer[4096];
    printf("s_path: %s",s_path);
    src_fd = open(s_path, O_RDONLY);
    printf("src_fd %d",src_fd);
    dst_fd = open(d_path, O_CREAT | O_WRONLY );

    while (1) {
        err = read(src_fd, buffer, 4096);
        if (err == -1) {
            printf("Error reading file.\n");
            exit(1);
        } else {
            printf("copied the file");
        }
        n = err;

        if (n == 0) break;
        printf("dst_fd %d",dst_fd);
        printf("buffer %c",buffer);
        printf("n %d",n);

        err = write(dst_fd, buffer, n);
        if (err == -1) {
            printf("Error writing to file.\n");
            exit(1);
        } else {
            printf("pasted the file");
        }
    }
    close(src_fd);
    close(dst_fd);
}
```

Following is the code for GUI Application:

CREATING A NEW FILE:

```
static void create_directory (GtkWidget *widget, gint response_id, gpointer data)
{
    GtkEntry* entry = data;
    const gchar *entrydata;
    char *basepath = g_get_home_dir();
    entrydata = gtk_entry_get_text(entry);
    printf("\nentrydata: %s", entrydata);
    printf("\ng_get_home_dir(): %s",g_get_home_dir());
    strcat(basepath,"/");
    strcat(basepath,entrydata);
    int check = mkdir(basepath,0777);
    printf("\ncheck: %d",check);
    if(check>=0){
        printf("\nnew Direcory %s created!\n",entrydata);
    } else {
        printf("\nfailed to create a new directory\n");
    }
    gtk_widget_destroy (widget); // This will close the dialog
}
```

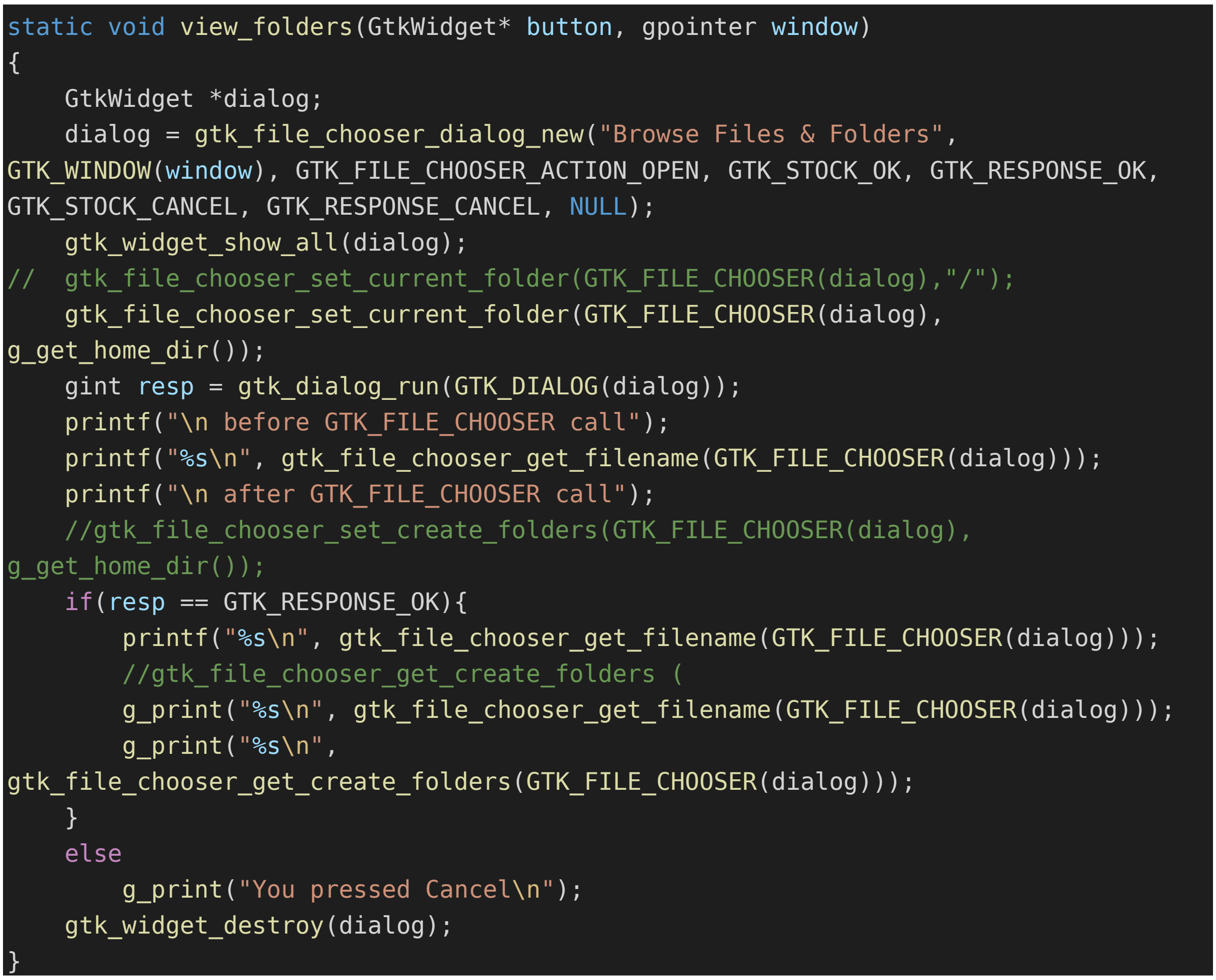
```
static void call_create_directory (GtkWidget *widget, gpointer data)
{
    GtkWidget *window = data;
    GtkWidget *dialog;
    GtkWidget *content_area;
    GtkWidget *grid;
    GtkWidget *label;
    GtkWidget *button;
    static GtkEntry *textbox;

    dialog = gtk_dialog_new_with_buttons ("Create a new Directory", window,
GTK_DIALOG_MODAL, GTK_STOCK_OK, GTK_RESPONSE_OK, NULL);
    content_area = gtk_dialog_get_content_area (GTK_DIALOG (dialog));
    grid = gtk_grid_new();
    gtk_container_add (GTK_CONTAINER (content_area), grid);

    label = gtk_label_new("Directory path: ");
    gtk_grid_attach(GTK_GRID(grid), label, 0, 0, 1, 1);
    textbox = gtk_entry_new();
    gtk_entry_set_text(textbox, "");
    gtk_grid_attach(GTK_GRID(grid), textbox, 1, 0, 1, 1);

    gtk_widget_show_all (dialog);
    g_signal_connect (GTK_DIALOG (dialog), "response", G_CALLBACK
(create_directory), textbox);
}
```

CREATING A NEW FILE:

```
static void create_file (GtkWidget *widget, gint response_id, gpointer data)
{
    FILE *fp;
    GtkEntry* entry = data;
    const gchar *entrydata;
    char *basepath = g_get_home_dir();
    entrydata = gtk_entry_get_text(entry);
    printf("\nentry() : %s", gtk_entry_get_text(entry));
    printf("\nentrydata: %s", entrydata);
    printf("\nentry: %s",entry);
    printf("\ng_get_home_dir(): %s",g_get_home_dir());
    strcat(basepath,"/");
    strcat(basepath,entrydata);
    printf("\ncreateFile: %s",basepath);
    fp  = fopen (basepath, "w");

    gtk_widget_destroy (widget); // This will close the dialog
}
```

```
static void call_create_file (GtkWidget *widget, gpointer data)
{
    GtkWidget *window = data;
    GtkWidget *dialog;
    GtkWidget *content_area;
    GtkWidget *grid;
    GtkWidget *label;
    GtkWidget *button;
    static GtkEntry *textbox;

    dialog = gtk_dialog_new_with_buttons ("Create a new File", window,
GTK_DIALOG_MODAL, GTK_STOCK_OK, GTK_RESPONSE_OK, NULL);
    content_area = gtk_dialog_get_content_area (GTK_DIALOG (dialog));
    grid = gtk_grid_new();
    gtk_container_add (GTK_CONTAINER (content_area), grid);

    label = gtk_label_new("File path: ");
    gtk_grid_attach(GTK_GRID(grid), label, 0, 0, 1, 1);
    textbox = gtk_entry_new();
    gtk_entry_set_placeholder_text(textbox, "enter filename");
    gtk_entry_set_text(textbox, "");
    gtk_grid_attach(GTK_GRID(grid), textbox, 1, 0, 1, 1);

    gtk_widget_show_all (dialog);
    g_signal_connect (GTK_DIALOG (dialog), "response", G_CALLBACK (create_file),
textbox);
}
```

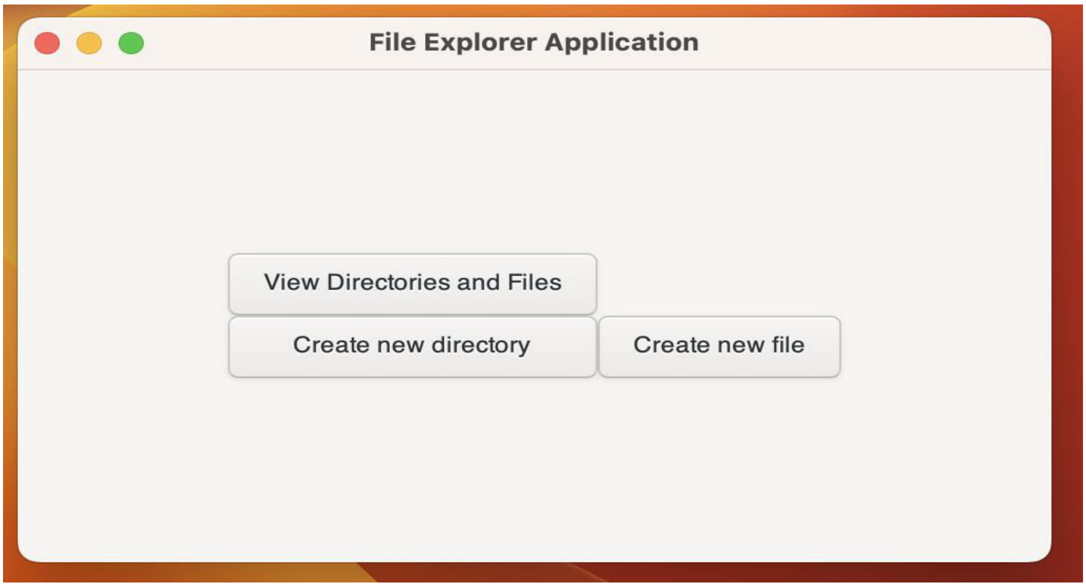
VIEW FILES AND DIRECTORIES:



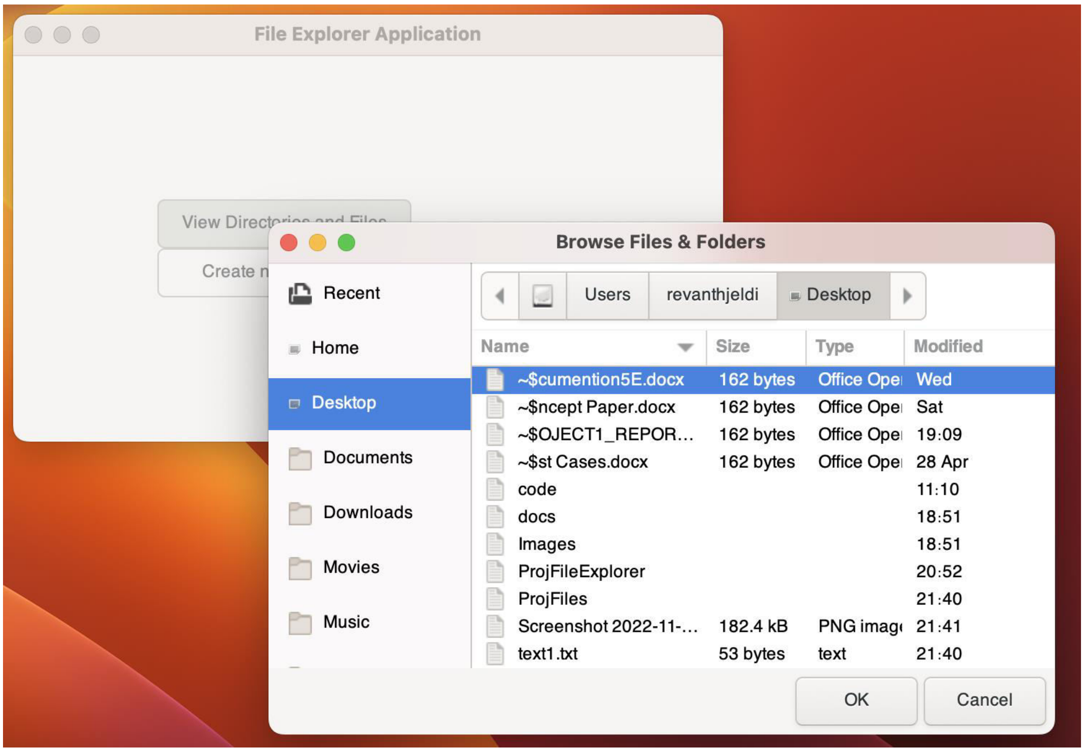
```
static void view_folders(GtkWidget* button, gpointer window)
{
    GtkWidget *dialog;
    dialog = gtk_file_chooser_dialog_new("Browse Files & Folders",
GTK_WINDOW(window), GTK_FILE_CHOOSER_ACTION_OPEN, GTK_STOCK_OK, GTK_RESPONSE_OK,
GTK_STOCK_CANCEL, GTK_RESPONSE_CANCEL, NULL);
    gtk_widget_show_all(dialog);
//  gtk_file_chooser_set_current_folder(GTK_FILE_CHOOSER(dialog),"/");
    gtk_file_chooser_set_current_folder(GTK_FILE_CHOOSER(dialog),
g_get_home_dir());
    gint resp = gtk_dialog_run(GTK_DIALOG(dialog));
    printf("\n before GTK_FILE_CHOOSER call");
    printf("%s\n", gtk_file_chooser_get_filename(GTK_FILE_CHOOSER(dialog)));
    printf("\n after GTK_FILE_CHOOSER call");
    //gtk_file_chooser_set_create_folders(GTK_FILE_CHOOSER(dialog),
g_get_home_dir());
    if(resp == GTK_RESPONSE_OK){
        printf("%s\n", gtk_file_chooser_get_filename(GTK_FILE_CHOOSER(dialog)));
        //gtk_file_chooser_get_create_folders (
        g_print("%s\n", gtk_file_chooser_get_filename(GTK_FILE_CHOOSER(dialog)));
        g_print("%s\n",
gtk_file_chooser_get_create_folders(GTK_FILE_CHOOSER(dialog)));
    }
    else
        g_print("You pressed Cancel\n");
    gtk_widget_destroy(dialog);
}
```

EXPERIMENTAL RESULTS:

When a user runs the application, the following output screens are displayed to them.

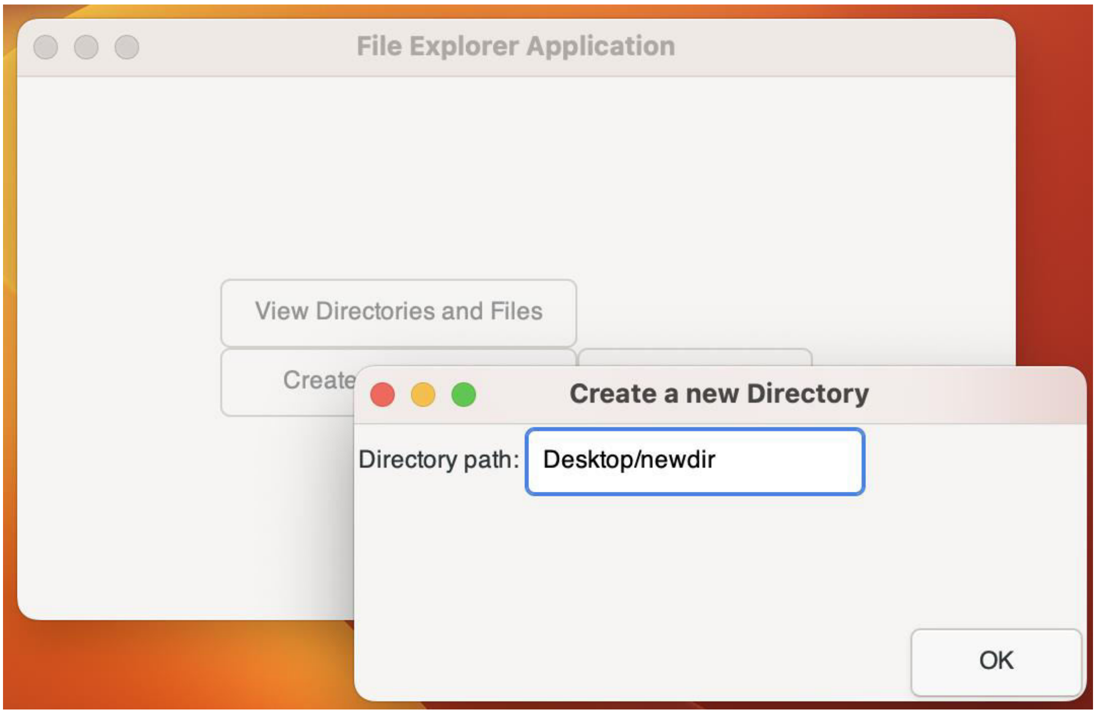


When they click on View Directories and files, a window pops up showing all the files and folder present in the computer.



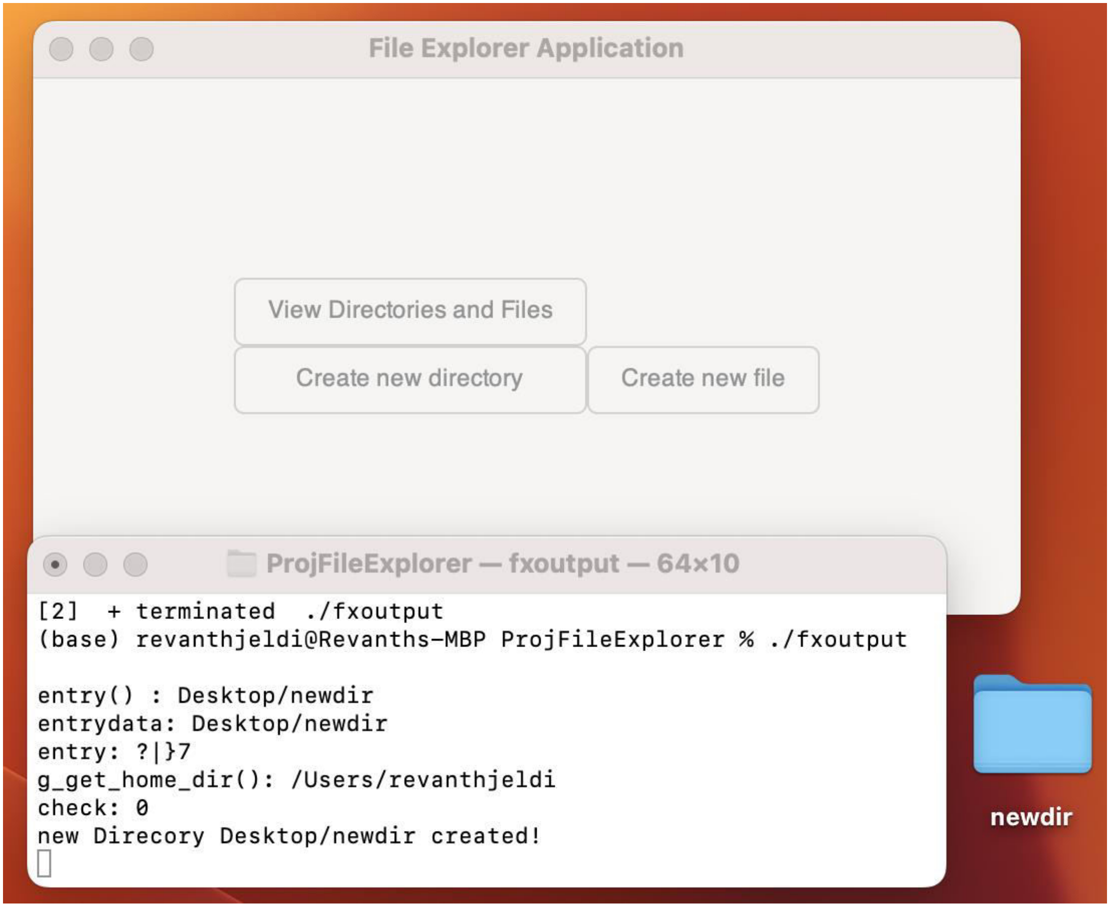
Name	Size	Type	Modified
~$cumention5E.docx	162 bytes	Office Ope	Wed
~$ncept Paper.docx	162 bytes	Office Ope	Sat
~$OJECT1_REPOR…	162 bytes	Office Ope	19:09
~$st Cases.docx	162 bytes	Office Ope	28 Apr
code			11:10
docs			18:51
Images			18:51
ProjFileExplorer			20:52
ProjFiles			21:40
Screenshot 2022-11-…	182.4 kB	PNG imag	21:41
text1.txt	53 bytes	text	21:40

When they click on Create new directory, a text window pops up and the user has to enter the directory they want to create.



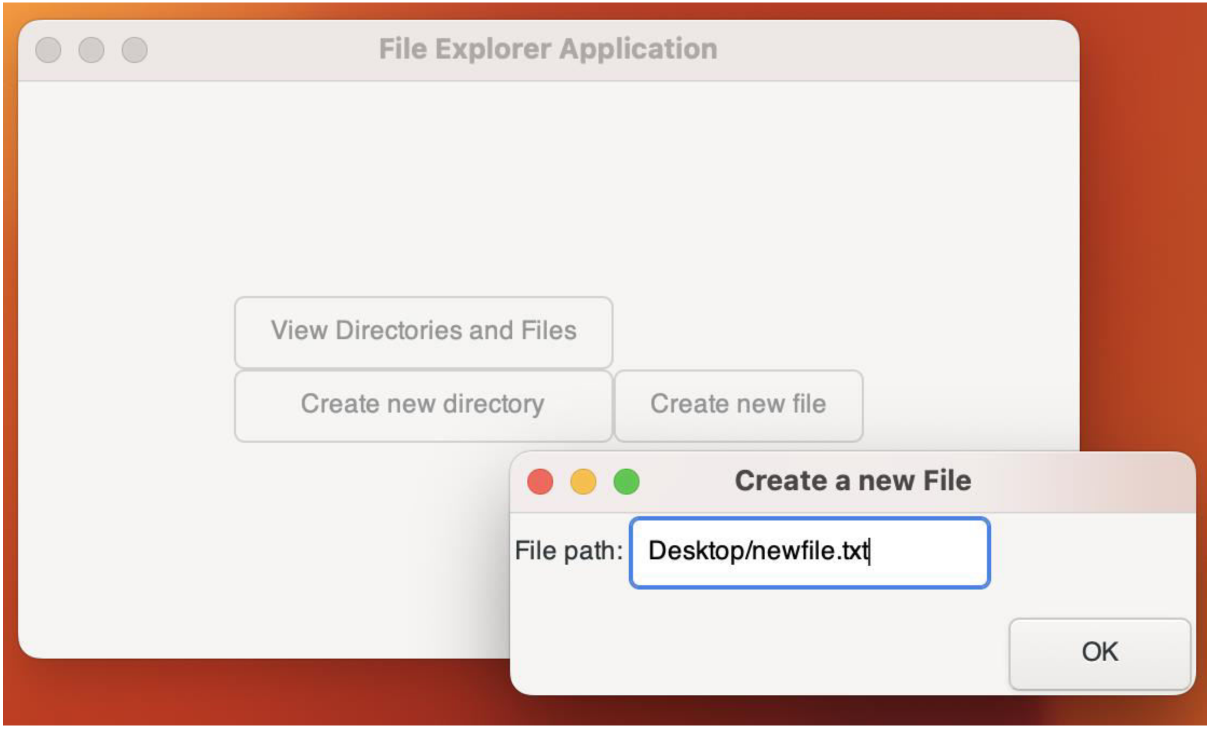
Directory path: Desktop/newdir

And a new directory is created!


File Explorer Application
View Directories and Files
Create new directory
Create new file
ProjFileExplorer — fxoutput — 64×10
newdir

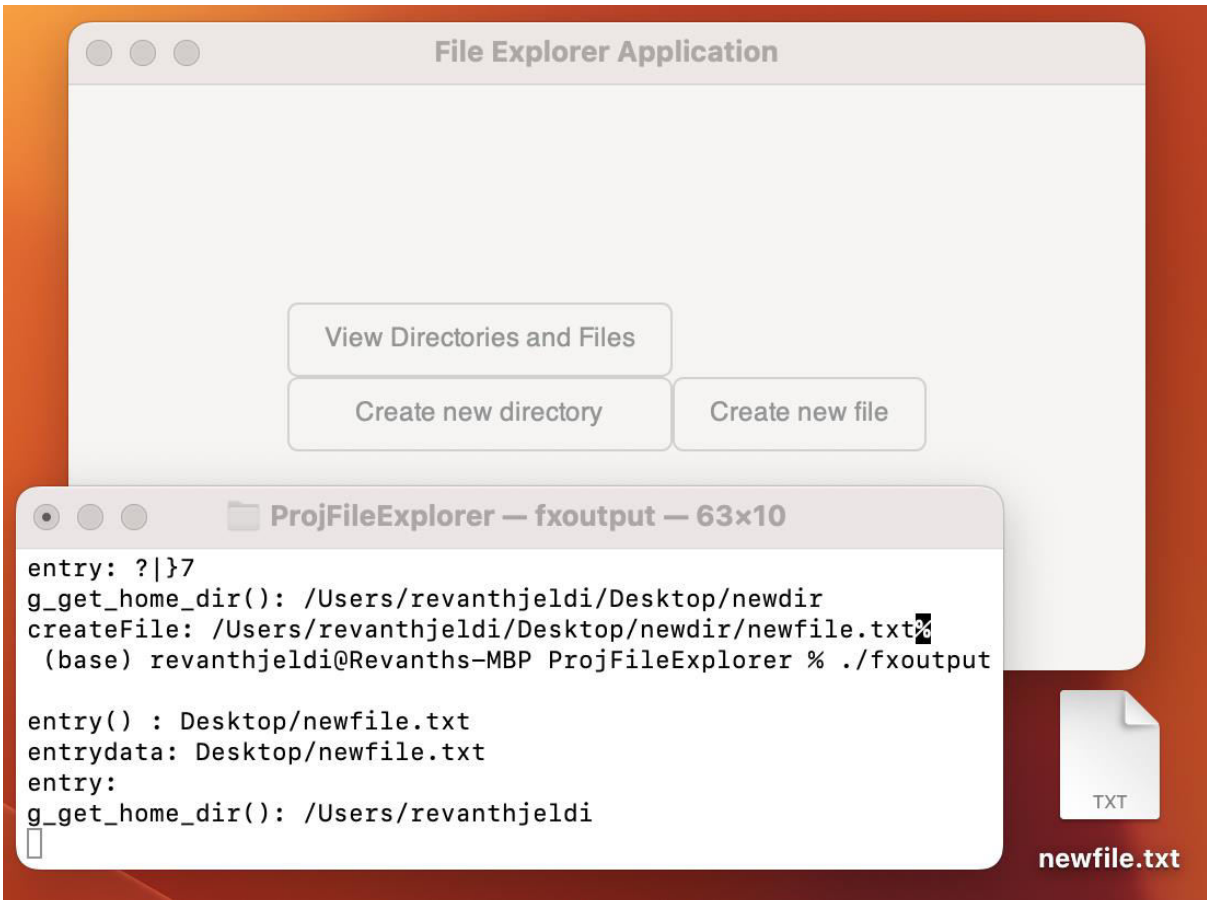
```
[2]  + terminated  ./fxoutput
(base) revanthjeldi@Revanths-MBP ProjFileExplorer % ./fxoutput

entry() : Desktop/newdir
entrydata: Desktop/newdir
entry: ?|}7
g_get_home_dir(): /Users/revanthjeldi
check: 0
new Direcory Desktop/newdir created!
```

When they click on Create new directory, a text window pops up and the user has to enter the directory they want to create.



File path: Desktop/newfile.txt

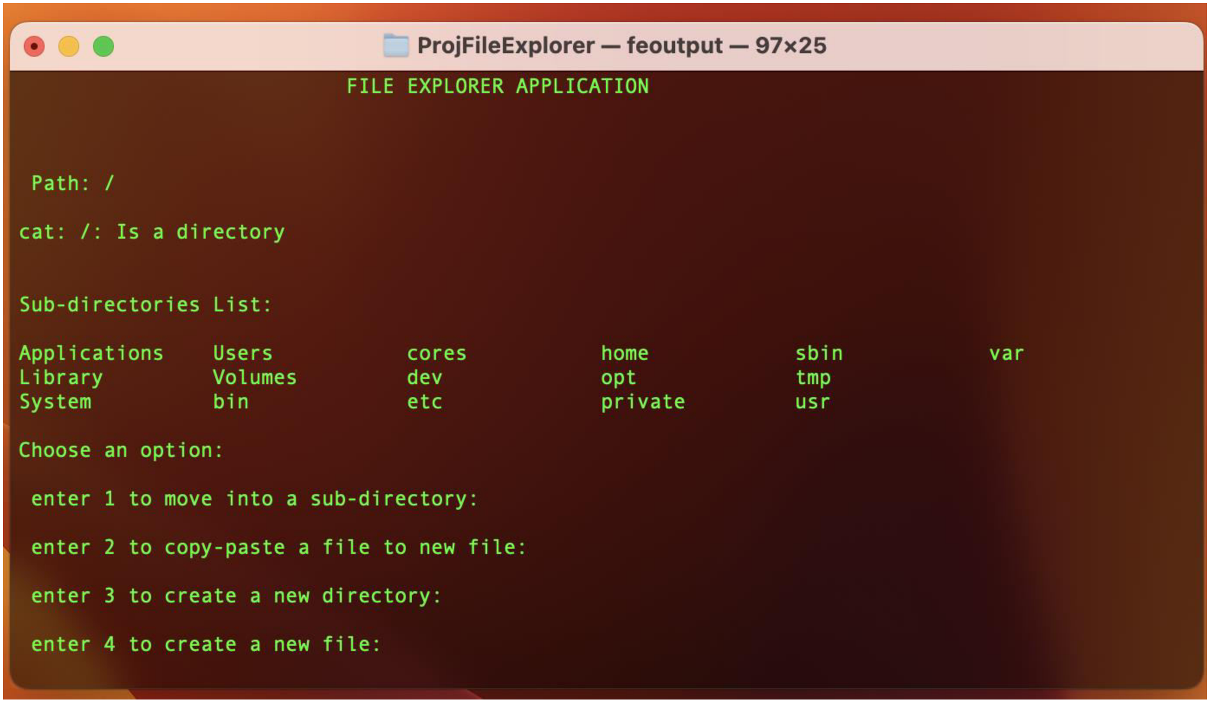
And a new file is created!


File Explorer Application
View Directories and Files
Create new directory
Create new file
ProjFileExplorer — fxoutput — 63×10
TXT
newfile.txt

```
entry: ?|}7
g_get_home_dir(): /Users/revanthjeldi/Desktop/newdir
createFile: /Users/revanthjeldi/Desktop/newdir/newfile.txt%
 (base) revanthjeldi@Revanths-MBP ProjFileExplorer % ./fxoutput

entry() : Desktop/newfile.txt
entrydata: Desktop/newfile.txt
entry:
g_get_home_dir(): /Users/revanthjeldi
```

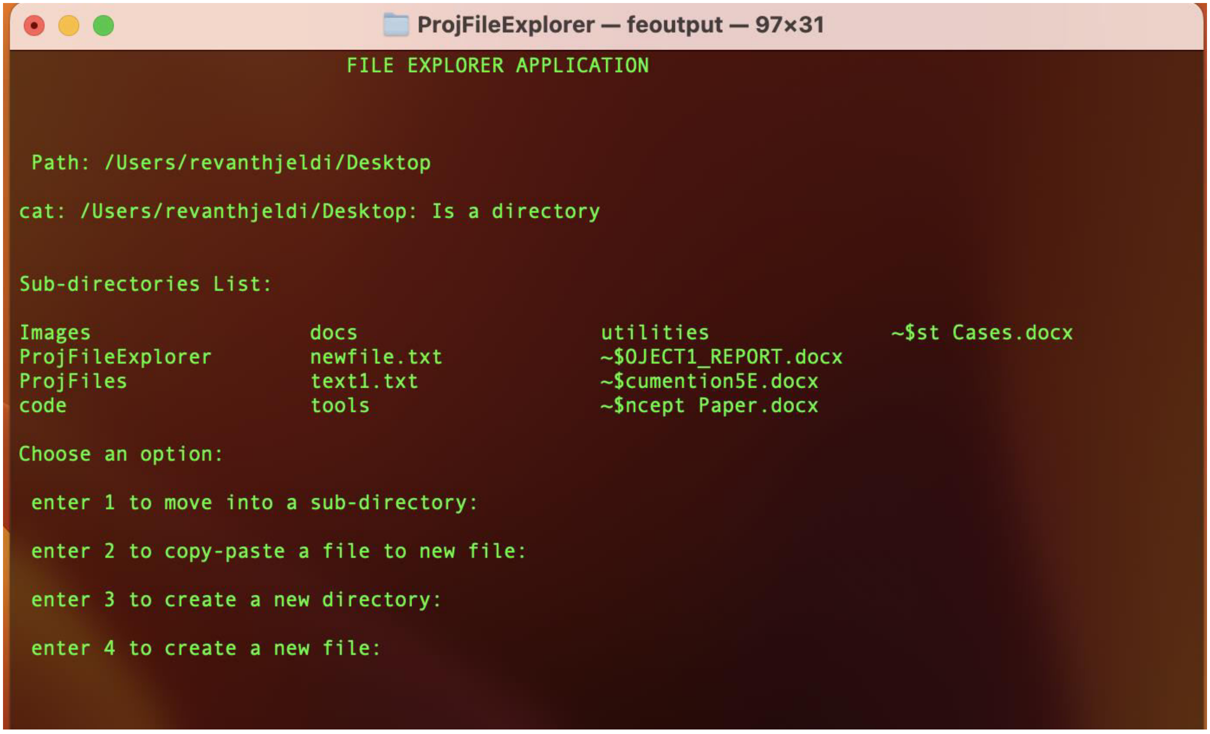
OUTPUT SCREENS FOR CONSOLE APPLICATION:

When a user executes the C program, the following output is displayed:



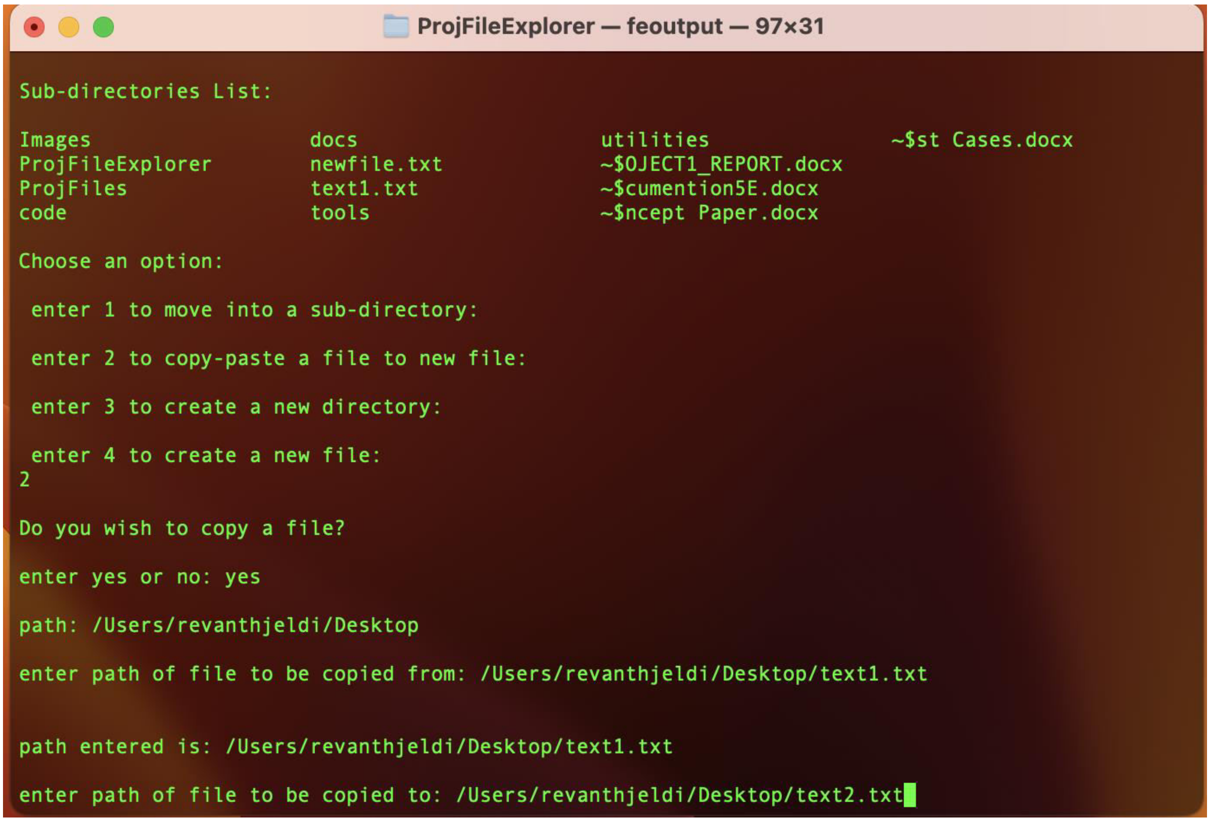
```
                              FILE EXPLORER APPLICATION


 Path: /

cat: /: Is a directory


Sub-directories List:

Applications    Users           cores           home            sbin            var
Library         Volumes         dev             opt             tmp
System          bin             etc             private         usr

Choose an option:

 enter 1 to move into a sub-directory:

 enter 2 to copy-paste a file to new file:

 enter 3 to create a new directory:

 enter 4 to create a new file:
```

Users select option “1” and enters the path of the subdirectory to view the files and folders present in it.



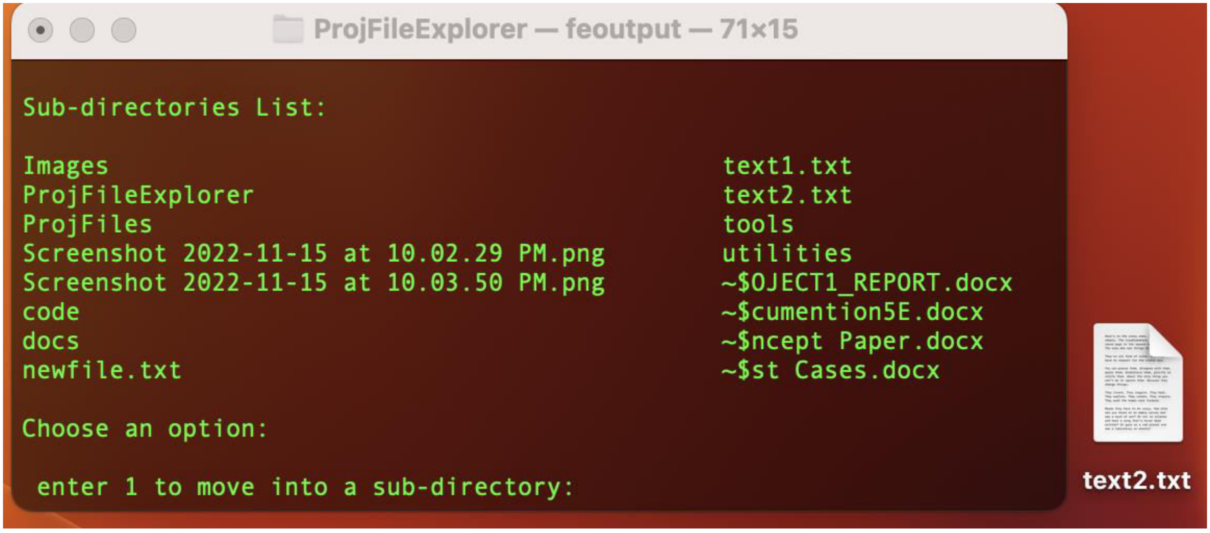
```
                           FILE EXPLORER APPLICATION


 Path: /Users/revanthjeldi/Desktop

cat: /Users/revanthjeldi/Desktop: Is a directory


Sub-directories List:

Images                  docs                    utilities               ~$st Cases.docx
ProjFileExplorer        newfile.txt             ~$OJECT1_REPORT.docx
ProjFiles               text1.txt               ~$cumention5E.docx
code                    tools                   ~$ncept Paper.docx

Choose an option:

 enter 1 to move into a sub-directory:

 enter 2 to copy-paste a file to new file:

 enter 3 to create a new directory:

 enter 4 to create a new file:
```

Users select option “2” and enters the path of the file to be copied from and then enter the path of the file to be copied to.



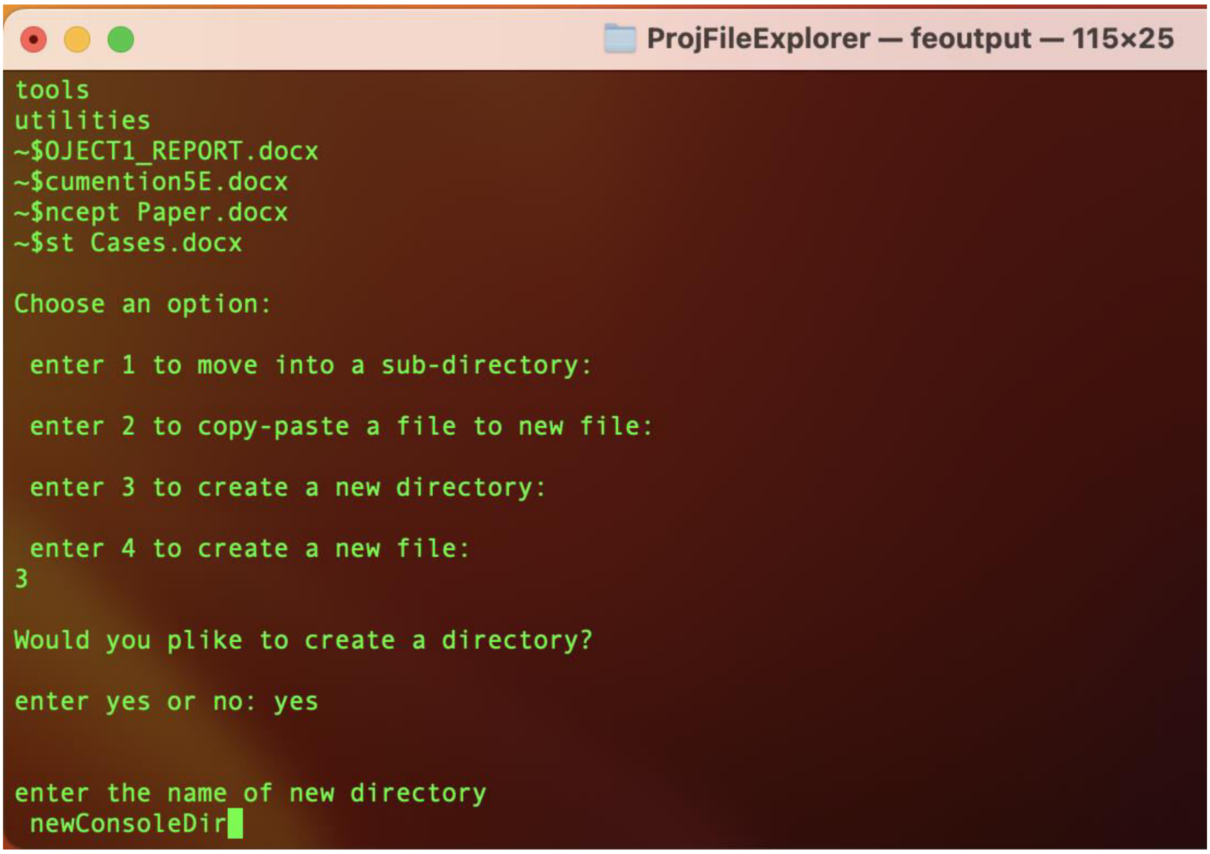
```
Sub-directories List:

Images                  docs                    utilities               ~$st Cases.docx
ProjFileExplorer        newfile.txt             ~$OJECT1_REPORT.docx
ProjFiles               text1.txt               ~$cumention5E.docx
code                    tools                   ~$ncept Paper.docx

Choose an option:

 enter 1 to move into a sub-directory:

 enter 2 to copy-paste a file to new file:

 enter 3 to create a new directory:

 enter 4 to create a new file:
2

Do you wish to copy a file?

enter yes or no: yes

path: /Users/revanthjeldi/Desktop

enter path of file to be copied from: /Users/revanthjeldi/Desktop/text1.txt


path entered is: /Users/revanthjeldi/Desktop/text1.txt

enter path of file to be copied to: /Users/revanthjeldi/Desktop/text2.txt
```

And the content of file-1 are copied to file-2.



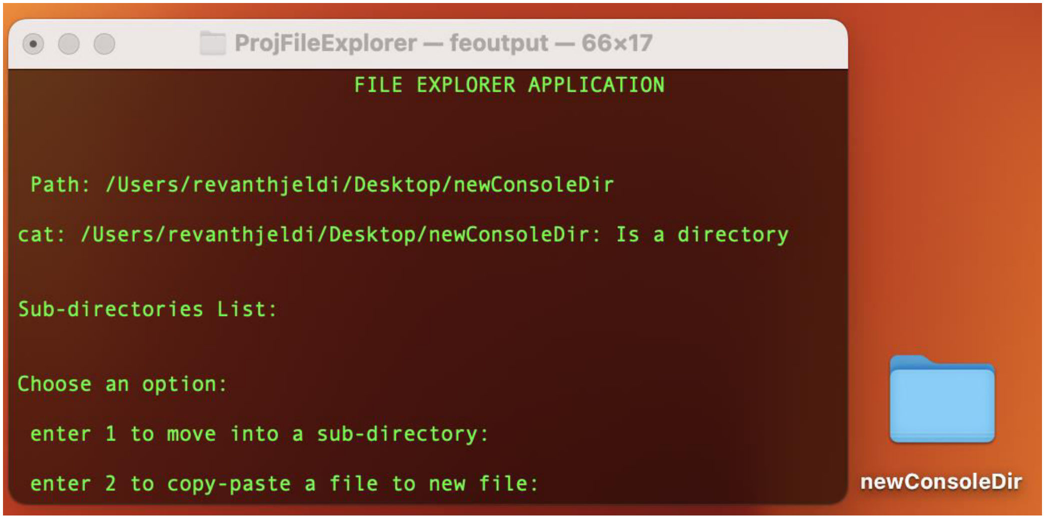
```
Sub-directories List:

Images                                          text1.txt
ProjFileExplorer                                text2.txt
ProjFiles                                       tools
Screenshot 2022-11-15 at 10.02.29 PM.png        utilities
Screenshot 2022-11-15 at 10.03.50 PM.png        ~$OJECT1_REPORT.docx
code                                            ~$cumention5E.docx
docs                                            ~$ncept Paper.docx
newfile.txt                                     ~$st Cases.docx

Choose an option:

 enter 1 to move into a sub-directory:
```

Users select option "3" and enters the name of directory to be created.



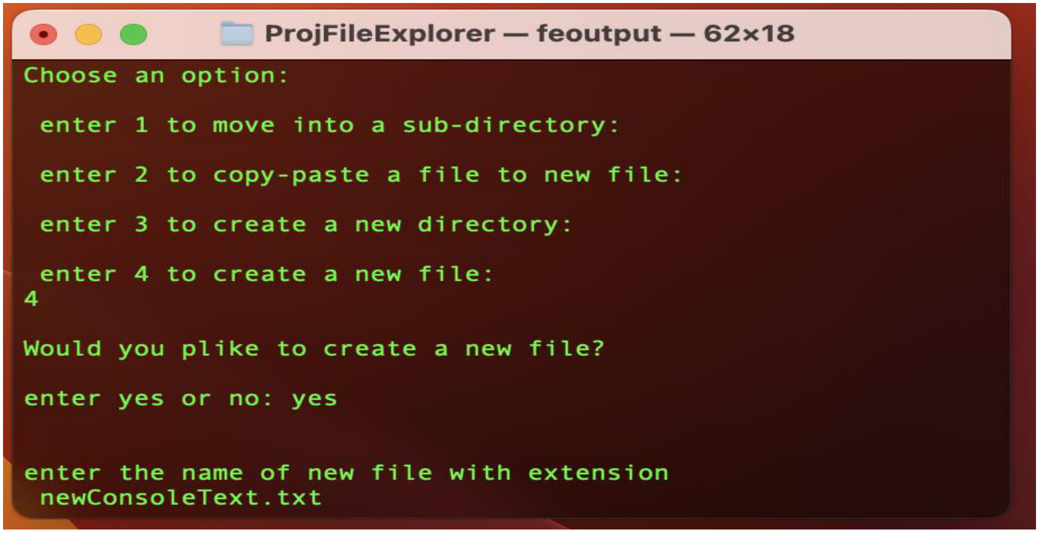
```
tools
utilities
~$OJECT1_REPORT.docx
~$cumention5E.docx
~$ncept Paper.docx
~$st Cases.docx

Choose an option:

 enter 1 to move into a sub-directory:

 enter 2 to copy-paste a file to new file:

 enter 3 to create a new directory:

 enter 4 to create a new file:
3

Would you plike to create a directory?

enter yes or no: yes


enter the name of new directory
 newConsoleDir
```

The directory is created.



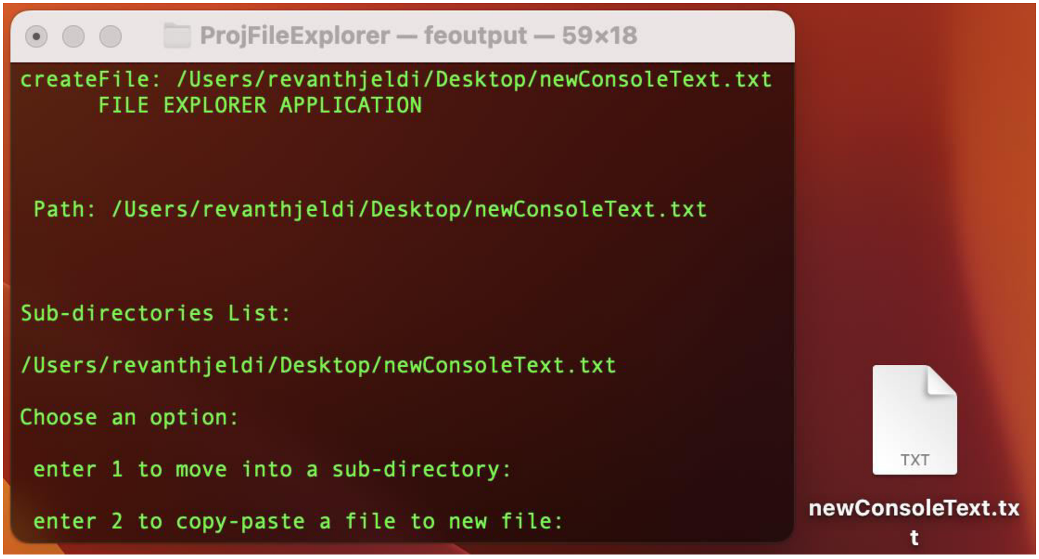
```
                   FILE EXPLORER APPLICATION


 Path: /Users/revanthjeldi/Desktop/newConsoleDir

cat: /Users/revanthjeldi/Desktop/newConsoleDir: Is a directory


Sub-directories List:


Choose an option:

 enter 1 to move into a sub-directory:

 enter 2 to copy-paste a file to new file:
```

Users select option “4” and enters the name of file to be created.



```
Choose an option:

 enter 1 to move into a sub-directory:

 enter 2 to copy-paste a file to new file:

 enter 3 to create a new directory:

 enter 4 to create a new file:
4

Would you plike to create a new file?

enter yes or no: yes


enter the name of new file with extension
 newConsoleText.txt
```

The required file is created.



```
createFile: /Users/revanthjeldi/Desktop/newConsoleText.txt
      FILE EXPLORER APPLICATION



 Path: /Users/revanthjeldi/Desktop/newConsoleText.txt



Sub-directories List:

/Users/revanthjeldi/Desktop/newConsoleText.txt

Choose an option:

 enter 1 to move into a sub-directory:

 enter 2 to copy-paste a file to new file:
```

REFERENCES:

https://www.gtk.org/

https://www.gtk.org/docs/getting-started/hello-world/

https://zetcode.com/gui/gtk2/firstprograms/

https://www.geeksforgeeks.org/how-to-create-gui-in-c-programming-using-gtk-toolkit/