

Source Code and Explanation:-

CEPOperations.java:-

```
package com.CourseEndProject1;

import java.io.*;

import java.util.*;

public class CEPOperations {

    Scanner sc = new Scanner(System.in);

    public void Retrieving() {

        File folder = new File("/Users/sabareeswaranchandrasekar/Desktop/CEPFolder");
        if (!folder.exists()) {
            folder.mkdirs();
        }

        File[] files = folder.listFiles();

        if (files.length == 0) {

            System.out.println("No Files Found....");

        }

        else {

            List<String> fileList = new ArrayList<>();

            // Loop through each file in the folder and add its name to the list
            for (File file : files) {
                if (file.isFile()) {
                    fileList.add(file.getName());
                }
            }

            int n = fileList.size();
            for (int i = 0; i < n - 1; i++) {
                for (int j = i + 1; j < n; j++) {
                    if (fileList.get(i).compareTo(fileList.get(j)) > 0) {
```

```

        String temp = fileList.get(i);
        fileList.set(i, fileList.get(j));
        fileList.set(j, temp);
    }
}
}

```

```

// Print the list of file names
for (String fileName : fileList) {
    System.out.println(fileName);
}

```

```

}

```

```

}

```

```

public void Add() {

```

```

    File folder = new File("/Users/sabareeswaranchandrasekar/Desktop/
CEPFolder");
    if (!folder.exists()) {
        folder.mkdirs();
    }

```

```

    File[] files = folder.listFiles();

```

```

    boolean checker;

```

```

    String nameoffile="";

```

```

    do {

```

```

        System.out.println("Enter the name of the file to add: ");

```

```

        nameoffile = sc.next();

```

```

        checker = false;

```

```

        for (File file : files) {
            if (nameoffile.equalsIgnoreCase(file.getName())) {
                System.out.println("File name already exists....");
                checker = true;
                break;
            }
        }
    }
}

```

```
    }  
}
```

```
}while(checker);
```

```
System.out.println("Enter the content of the file: ");
```

```
sc.nextLine();
```

```
String contentoffile = sc.nextLine();
```

```
File file = new File(folder, nameoffile);
```

```
try {
```

```
    FileWriter writer = new FileWriter(file);
```

```
    writer.write(contentoffile);
```

```
    writer.close();
```

```
    System.out.println("File created as " + file.getName());
```

```
} catch (IOException e) {
```

```
    System.out.println("An error occurred.");
```

```
    e.printStackTrace();
```

```
}
```

```
}
```

```
public void delete() {
```

```
    File folder = new File("/Users/sabareeswaranchandrasekar/Desktop/  
CEPFolder");
```

```
    if (!folder.exists()) {
```

```
        folder.mkdirs();
```

```
    }
```

```
File[] files = folder.listFiles();
```

```
boolean checker;
```

```
String nameoffile="";
```

```
do {
```

```
    System.out.println("Enter the name of the file to delete: ");
```

```
    nameoffile = sc.next();
```

```
checker = true;
```

```
for (File file : files) {  
    if (nameoffile.equalsIgnoreCase(file.getName())) {  
        file.delete();  
        System.out.println("File deleted...");  
        checker = false;  
        break;  
    }  
}
```

```
if(checker==true) {
```

```
    System.out.println("File name doesn't exists...");
```

```
}
```

```
}while(checker);
```

```
}
```

```
public void search() {
```

```
    File folder = new File("/Users/sabareeswaranchandrasekar/Desktop/  
CEPFolder");
```

```
    if (!folder.exists()) {  
        folder.mkdirs();  
    }
```

```
    File[] files = folder.listFiles();
```

```
    boolean checker;
```

```
    String nameoffile="";
```

```
    do {
```

```
        System.out.println("Enter the name of the file to search: ");
```

```
        nameoffile = sc.next();
```

```
        checker = true;
```

```

        for (File file : files) {
            if (nameoffile.equalsIgnoreCase(file.getName())) {
                System.out.println("File content:");
                try (BufferedReader reader = new BufferedReader(new
FileReader(file))) {
                    String line;
                    while ((line = reader.readLine()) != null) {
                        System.out.println(line);
                    }
                } catch (IOException e) {
                    System.err.println("Error reading file: " + e.getMessage());
                }
                checker = false;
                break;
            }
        }

        if(checker==true) {

            System.out.println("File name doesn't exists...");

        }

        }while(checker);

    }

}

```

CEPMain.java:-

```

package com.CourseEndProject1;

import java.util.*;

public class CEPMain {

    public static void main(String[] args) {

        CEPOperations cepo = new CEPOperations();
        // TODO Auto-generated method stub
    }
}

```

```
Scanner sc = new Scanner(System.in);
```

```
System.out.println("Welcome to the LockedMe.com Application!");
```

```
System.out.println("-----");
```

```
System.out.println("Developed By Sabareeswaran Chandrasekar.");
```

```
System.out.println("-----");
```

```
int choice=0;
```

```
do {
```

```
    while (true) {
```

```
        System.out.println("\nEnter 1 to Retrieve\nEnter 2 to do  
Operations\nEnter 3 to Exit");
```

```
        System.out.println("Pick ur choice:");
```

```
        try {
```

```
            choice = sc.nextInt();
```

```
            break;
```

```
        }
```

```
        catch (InputMismatchException e) {
```

```
            System.out.println("Invalid input....! ");
```

```
            sc.next();
```

```
        }
```

```
    }
```

```
    switch(choice) {
```

```
        case 1:
```

```
            cepo.Retrieving();
```

```
            break;
```

case 2:

```
int t=1;
```

```
do {
```

```
    System.out.println("\nEnter a to add\nEnter b to  
delete\nEnter c to search\nEnter d to go back to home");
```

```
    char ch = sc.next().charAt(0);
```

```
    switch(ch) {
```

```
        case 'a':
```

```
            cepo.Add();
```

```
            break;
```

```
        case 'b':
```

```
            cepo.delete();
```

```
            break;
```

```
        case 'c':
```

```
            cepo.search();
```

```
            break;
```

```
        case 'd':
```

```
            t=0;
```

```
            break;
```

```
        default:
```

```
            System.out.println("Invalid Input....!. ");
```

```
            break;
```

```
    }
```

```
    }while(t!=0);
```

```
    break;
```

```
case 3:
```

```
    System.out.println("-----");
```

```
    System.out.println("Thanks for using the app.");
```

```
    System.exit(0);
```

```
default:
```

```
    System.out.println("Invalid Input....!. ");
```

```
    break;
```

```
}
```

```
}while(true);
```

```
}
```

```
}
```

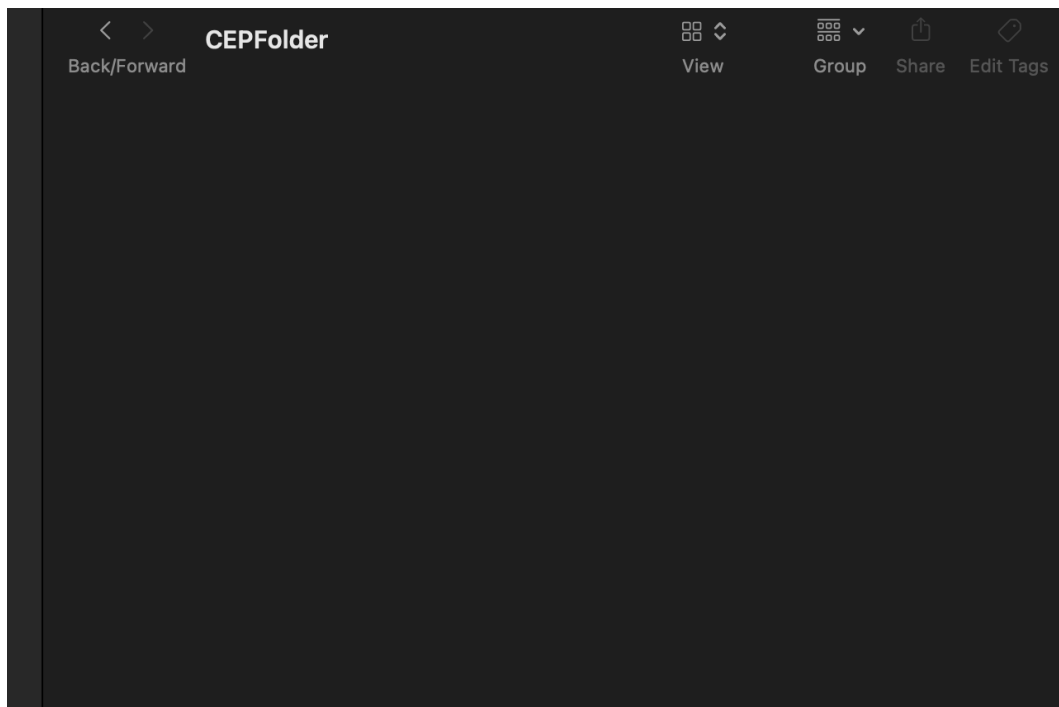

Explanation:-

Welcome to the LockedMe.com Application!

Developed By Sabareeswaran Chandrasekar.

Enter 1 to Retrieve
Enter 2 to do Operations
Enter 3 to Exit
Pick ur choice:

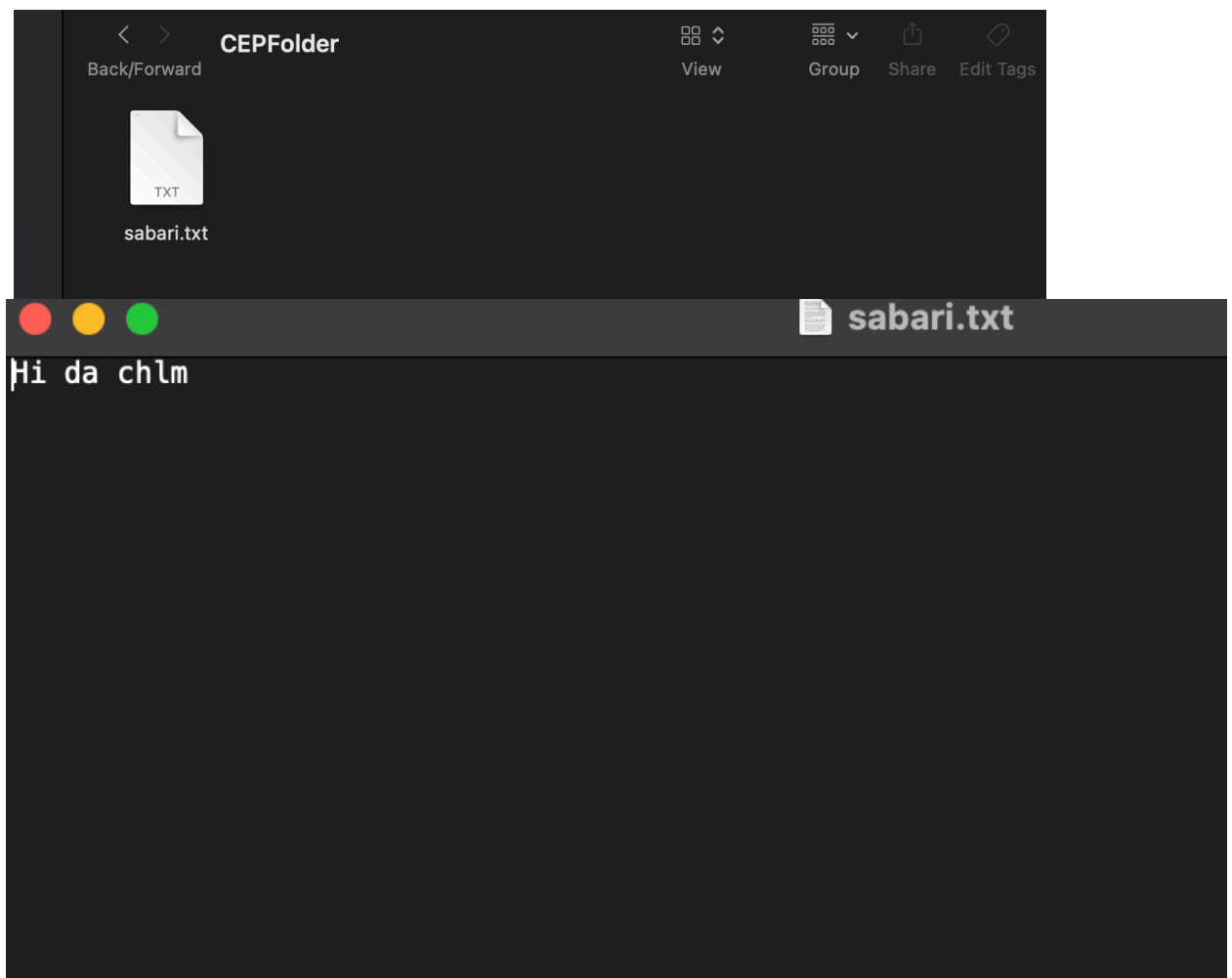
-> The above prompt is a textual user interface for an application called "LockedMe.com" developed by Sabareeswaran Chandrasekar. It presents the user with three options: to retrieve information (option 1), to perform operations (option 2), or to exit the application (option 3). The user is asked to enter a number corresponding to their desired choice.



```
Enter 1 to Retrieve
Enter 2 to do Operations
Enter 3 to Exit
Pick ur choice:
2
```

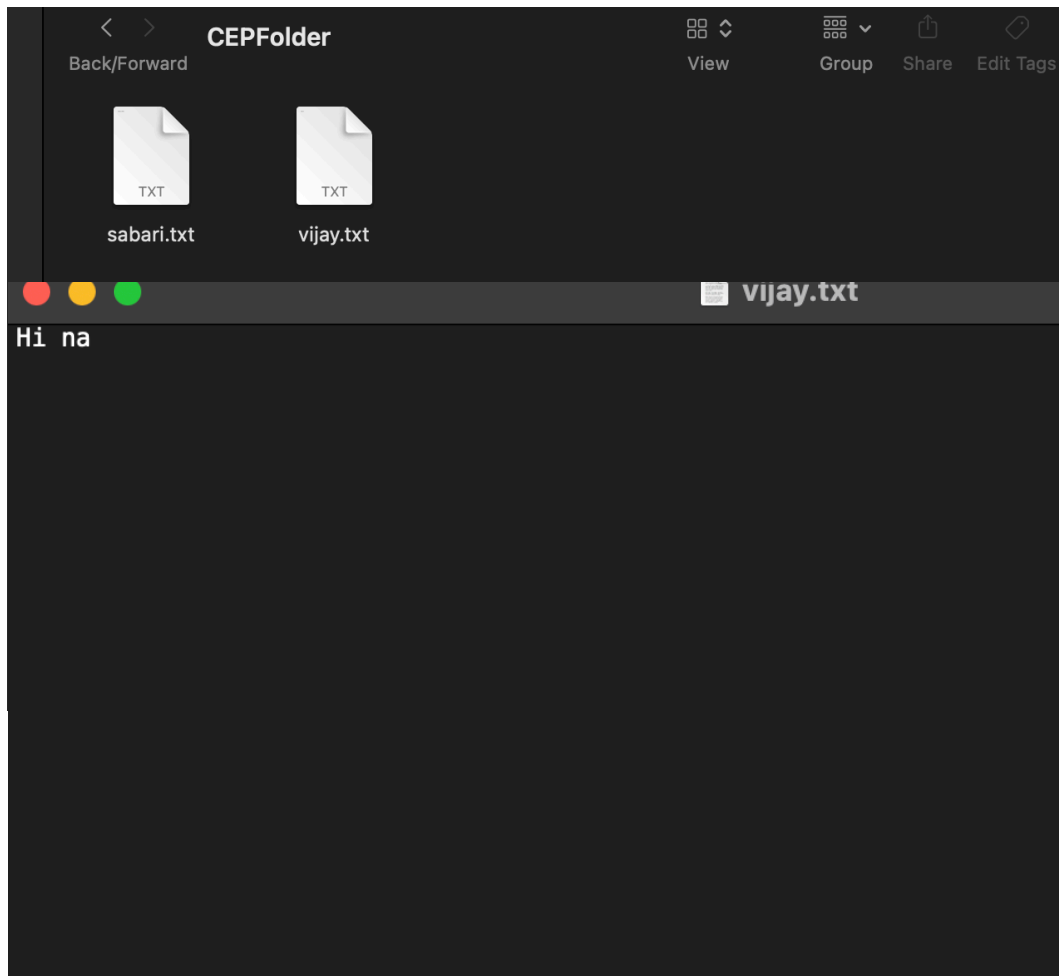
```
Enter a to add
Enter b to delete
Enter c to search
Enter d to go back to home
a
Enter the name of the file to add:
sabari.txt
Enter the content of the file:
Hi da chl̃m
File created as sabari.txt
```

-> The user has chosen option 2 from the main menu, which allows them to perform operations. They have then selected option 'a' to add a new file. They are prompted to enter the name of the file they want to add ('sabari.txt' in this case), followed by the content they want to add to the file ('Hi da chl̃m' in this case). The application confirms that the file has been created successfully with the given name, and the content has been added to it.



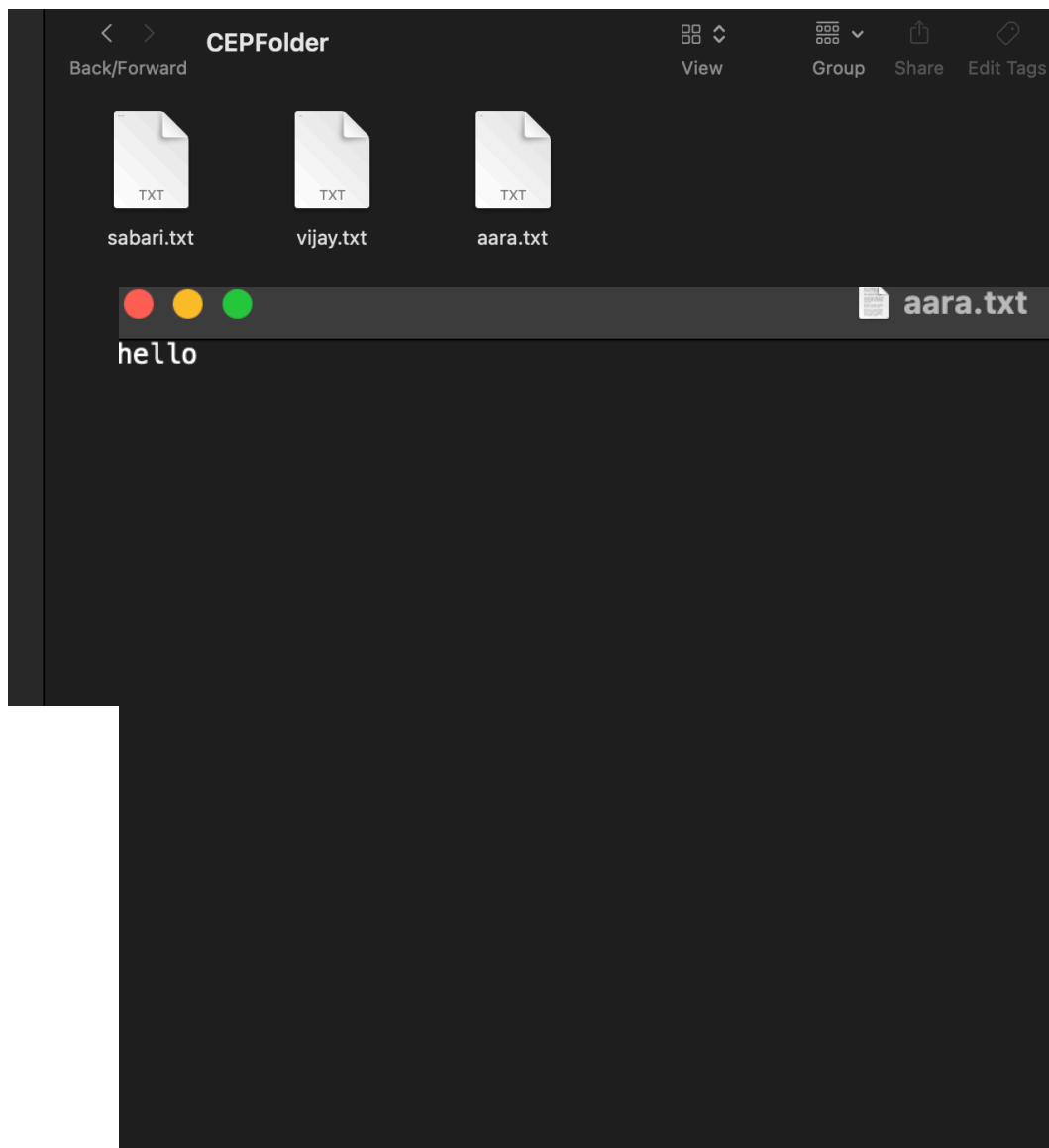
```
Enter a to add
Enter b to delete
Enter c to search
Enter d to go back to home
a
Enter the name of the file to add:
sabari.txt
File name already exists....
Enter the name of the file to add:
vijay.txt
Enter the content of the file:
Hi na
File created as vijay.txt
```

-> The user has chosen option 'a' to add a new file. They are prompted to enter the name of the file they want to add ('sabari.txt' in this case), but the application informs them that a file with that name already exists. So, the user is prompted again to enter a different file name ('vijay.txt' in this case). After entering the content they want to add to the file ('Hi na' in this case), the application confirms that the file has been created successfully with the given name, and the content has been added to it.



```
Enter a to add
Enter b to delete
Enter c to search
Enter d to go back to home
a
Enter the name of the file to add:
aara.txt
Enter the content of the file:
hello
File created as aara.txt
```

-> The user has chosen option 'a' to add a new file. They are prompted to enter the name of the file they want to add ('aara.txt' in this case), followed by the content they want to add to the file ('hello' in this case). The application confirms that the file has been created successfully with the given name, and the content has been added to it.



```
Enter a to add
Enter b to delete
Enter c to search
Enter d to go back to home
d
```

```
Enter 1 to Retrieve
Enter 2 to do Operations
Enter 3 to Exit
Pick ur choice:
1
.DS_Store
aara.txt
sabari.txt
vijay.txt
```

-> The user has chosen option 'd' to go back to the main menu. They are presented with the main menu again, and they have chosen option '1' to retrieve information. The application then displays a list of files as ascending order in the current directory.

```
Enter 1 to Retrieve
Enter 2 to do Operations
Enter 3 to Exit
Pick ur choice:
2
```

```
Enter a to add
Enter b to delete
Enter c to search
Enter d to go back to home
c
Enter the name of the file to search:
syed.txt
File name doesn't exists...
Enter the name of the file to search:
vijay.txt
File content:
Hi na
```

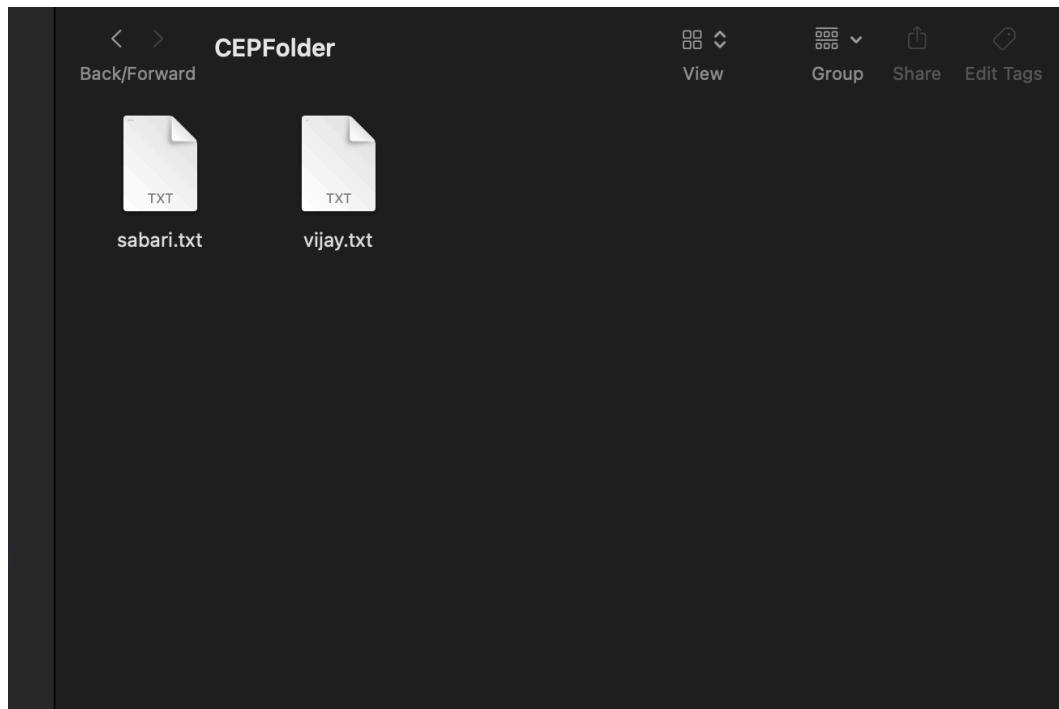
-> The user has chosen option '2' to perform operations. They have then selected option 'c' to search for a file. They are prompted to enter the name of the file they want to search for ('syed.txt' in this case), but the application informs them that the file name doesn't exist. The user is prompted again to enter a different file name ('vijay.txt' in this case). The application confirms that the file exists and displays its content ('Hi na' in this case).

```
Enter a to add
Enter b to delete
Enter c to search
Enter d to go back to home
b
Enter the name of the file to delete:
syed.txt
File name doesn't exists...
Enter the name of the file to delete:
aara.txt
File deleted...
```

```
Enter a to add
Enter b to delete
Enter c to search
Enter d to go back to home
d
```

```
Enter 1 to Retrieve
Enter 2 to do Operations
Enter 3 to Exit
Pick ur choice:
1
.DS_Store
sabari.txt
vijay.txt
```

-> The user has chosen option 'b' to delete a file. They are prompted to enter the name of the file they want to delete ('syed.txt' in this case), but the application informs them that the file name doesn't exist. The user is prompted again to enter a different file name ('aara.txt' in this case), and the application confirms that the file has been deleted successfully. Then the user has chosen option 'd' to go back to the main menu. They are presented with the main menu again, and they have chosen option '1' to retrieve information. The application then displays a list of files in the current directory.



Enter 1 to Retrieve
Enter 2 to do Operations
Enter 3 to Exit
Pick ur choice:

a

Invalid input....!

Enter 1 to Retrieve
Enter 2 to do Operations
Enter 3 to Exit
Pick ur choice:

4

Invalid Input....!.

Enter 1 to Retrieve
Enter 2 to do Operations
Enter 3 to Exit
Pick ur choice:

2

Enter a to add
Enter b to delete
Enter c to search
Enter d to go back to home

e

Invalid Input....!.

```
Enter a to add
Enter b to delete
Enter c to search
Enter d to go back to home
1
Invalid Input....!.
```

```
Enter a to add
Enter b to delete
Enter c to search
Enter d to go back to home
0
Invalid Input....!.
```

```
Enter a to add
Enter b to delete
Enter c to search
Enter d to go back to home
d
```

```
Enter 1 to Retrieve
Enter 2 to do Operations
Enter 3 to Exit
Pick ur choice:
3
```

```
-----
Thanks for using the app.
```

-> The user has interacted with the application by choosing various options. They started by selecting option '1' to retrieve information, then option '2' to perform operations. They made some mistakes by selecting invalid options ('a', '4', 'e', '1', and '0'), but they eventually selected option 'd' to go back to the main menu.

Once back in the main menu, the user selected option '3' to exit the application. The application then displayed a message thanking the user for using it.