SOURCE CODE EXPLANATION

Phase1_Func

```
package com.project;
import java.util.*;

public class Phase1_Func {

    public static void main(String[] args) {

        Phase1 ph = new Phase1();
        // TODO Auto-generated method stub

        Scanner sc = new Scanner(System.in);

        System.out.println("Welcome to LockedMe.com Application! - By
NECIKA P");

        System.out.println(" ");




        int s=0;

        do {

            while (true) {

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

          System.out.println("Enter your choice:");

          //try {

            s = sc.nextInt();

            break;

          //}
          /*catch (InputMismatchException e) {

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

            sc.next();
          } */
        }


            switch(s) {

            case 1:

                ph.Retrieving();

                break;

            case 2:

                int t=1;
                String ch;

                do {

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

                 ch = sc.next();

                    switch(ch) {

                    case "a":

                        ph.Add();

                        break;

                    case "b":

                        ph.delete();

                        break;

                    case "c":

                        ph.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("Thank you for using the app.");

                System.exit(0);


            default:

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

                break;


            }


        }while(true);

    }

}
```

Phase1

```
package com.project;
import java.io.*;
import java.util.*;


public class Phase1 {

        Scanner sc = new Scanner(System.in);

        public void Retrieving() {

                  File fr = new File("/Users/necikap/Desktop/Project");
              if (!fr.exists()) {
                fr.mkdirs();
              }

              File[] files = fr.listFiles();

              if(files.length==0) {

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

              }

              else {

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

                  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);
                          }
                      }
                  }


                  for (String fileName : fileList) {
                    System.out.println(fileName);
                  }
```

```
    }

}

public void Add() {

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

    File[] files = folder.listFiles();

    boolean checker;

    String noff="";

    do {

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

    noff = sc.next();


    checker = false;


    for (File file : files) {
      if (noff.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, noff);
    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/necikap/Desktop/Project");
    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/necikap/Desktop/Project");
          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);


      }

}
```

EXPLANATION

Welcome to LockedMe.com Application! - By NECIKA P

Type 1 to Retrieve
Type 2 to do Operations
Type 3 to Exit
Enter your choice:

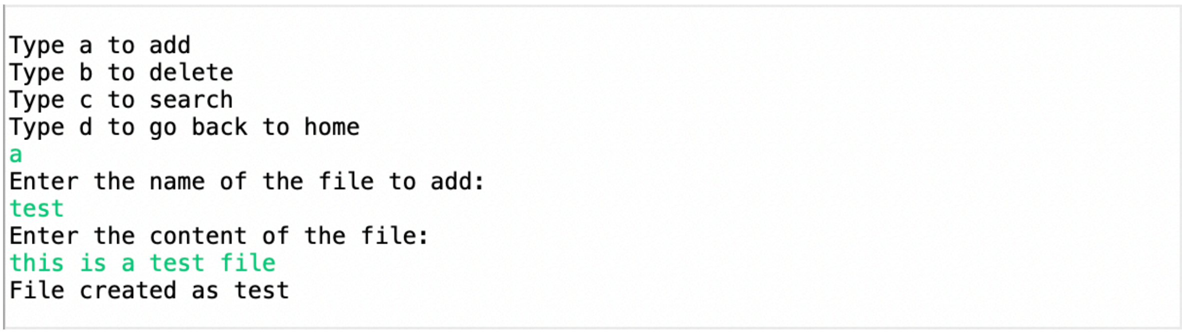
This is the interface, where user can interact with the application. Here the application gives user 3 choices which includes - to retrieve, to do operations , to exit.
Based on the choice of the user, the working will proceed.

Enter your choice:
2

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

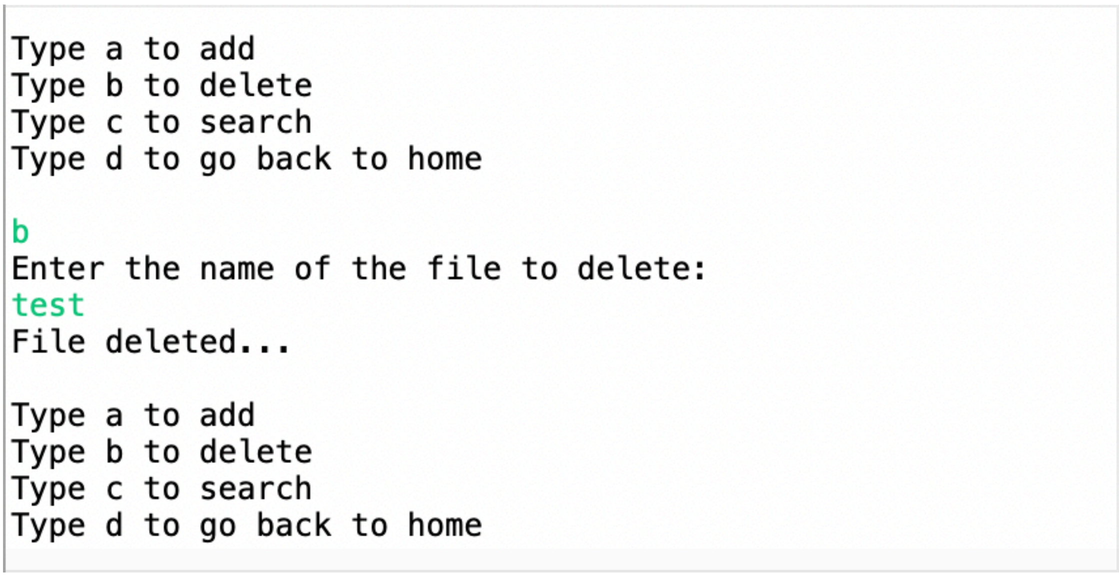
If the user choice is 2. The application will display certain option like - to add, delete, search, to go back home.
Based on the choice the application will proceed.

If it is a, then the application will add a file which the user will enter.



```
Type a to add
Type b to delete
Type c to search
Type d to go back to home
a
Enter the name of the file to add:
test
Enter the content of the file:
this is a test file
File created as test
```

If it is b, then the application will delete the file which the user wants

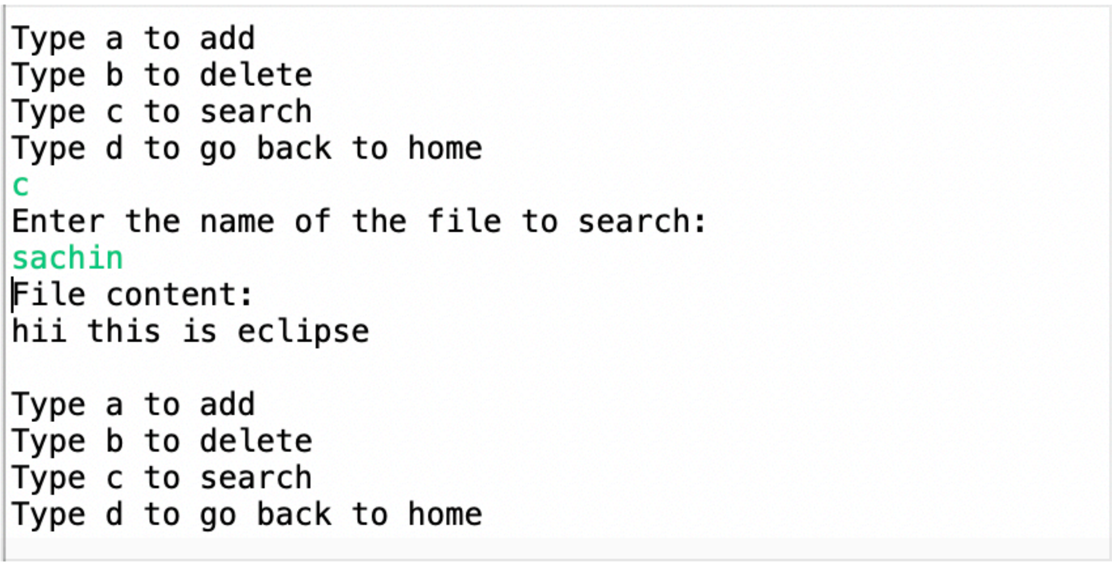


```
Type a to add
Type b to delete
Type c to search
Type d to go back to home

b
Enter the name of the file to delete:
test
File deleted...

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

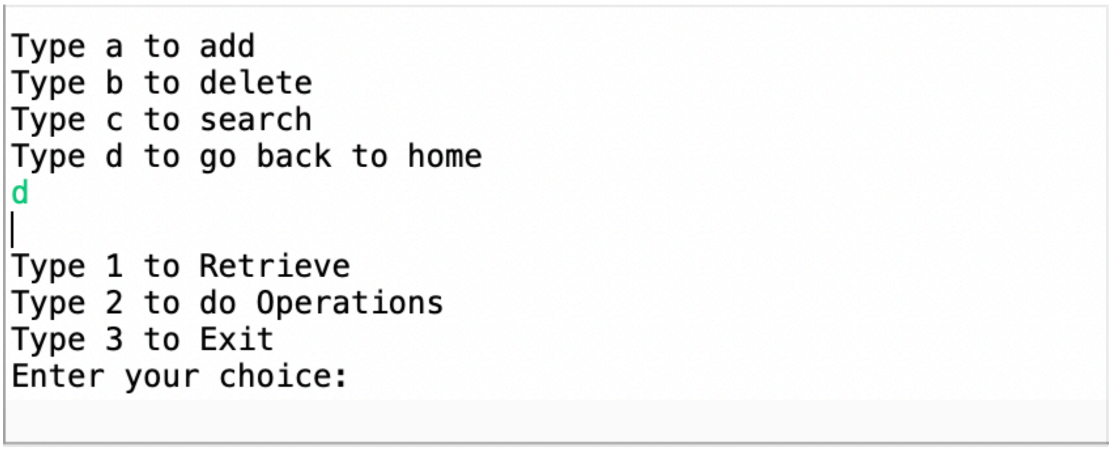
If it is c, then the user will search for the file the user wants



```
Type a to add
Type b to delete
Type c to search
Type d to go back to home
c
Enter the name of the file to search:
sachin
File content:
hii this is eclipse

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

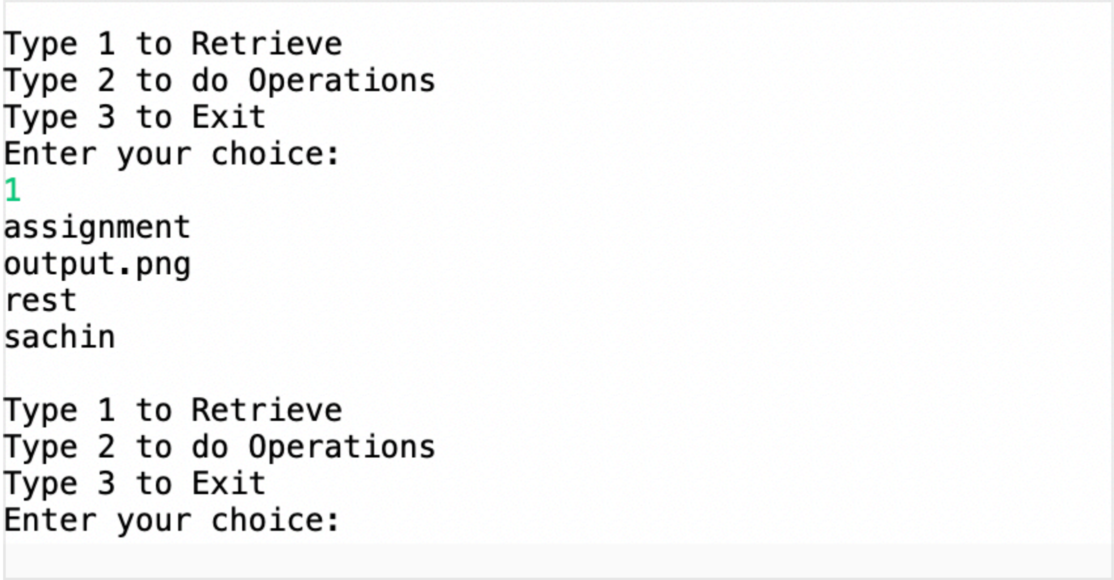
If it is d, then the application will return the user to the main menu



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

Type 1 to Retrieve
Type 2 to do Operations
Type 3 to Exit
Enter your choice:
```

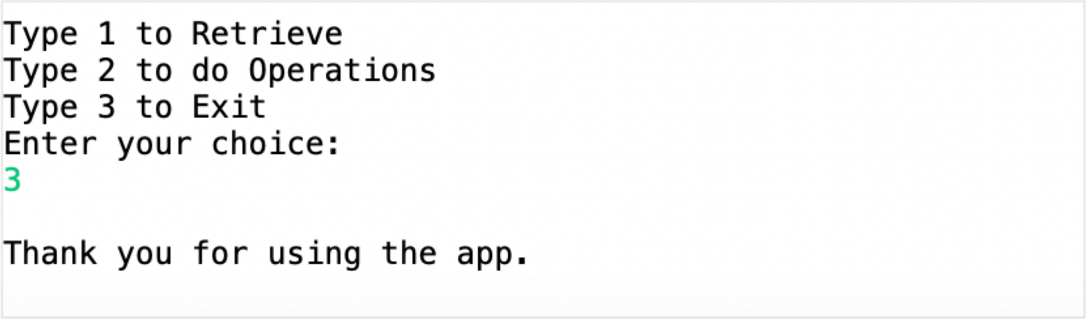
If the user types 1, then the application will retrieve all the files present in the folder



```
Type 1 to Retrieve
Type 2 to do Operations
Type 3 to Exit
Enter your choice:
1
assignment
output.png
rest
sachin

Type 1 to Retrieve
Type 2 to do Operations
Type 3 to Exit
Enter your choice:
```

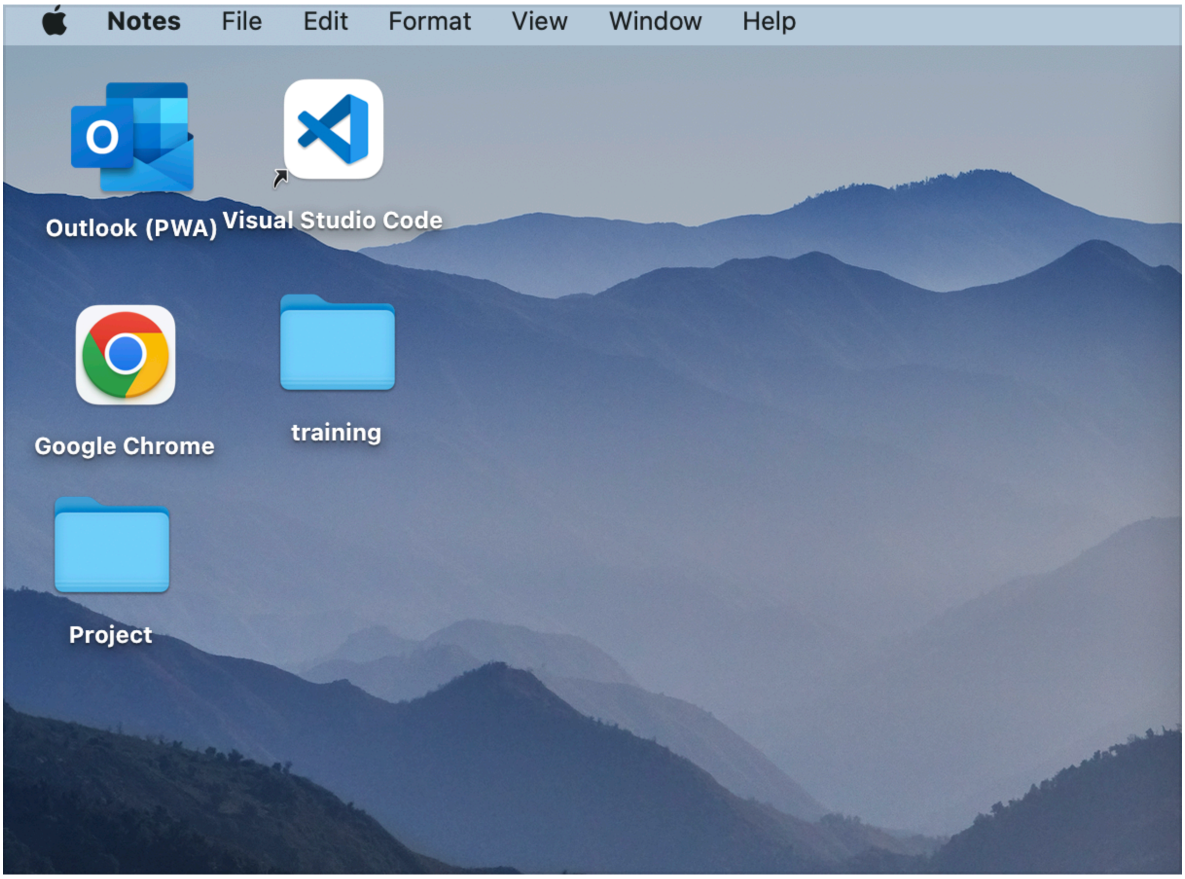
If the user types 3, then the application will exit with a thank you message



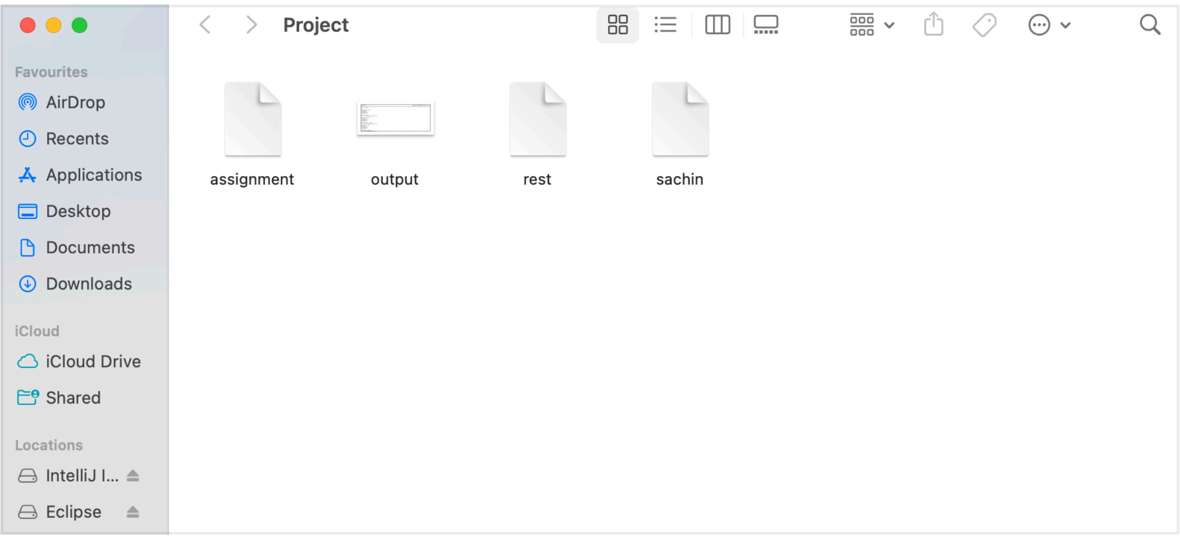
```
Type 1 to Retrieve
Type 2 to do Operations
Type 3 to Exit
Enter your choice:
3

Thank you for using the app.
```

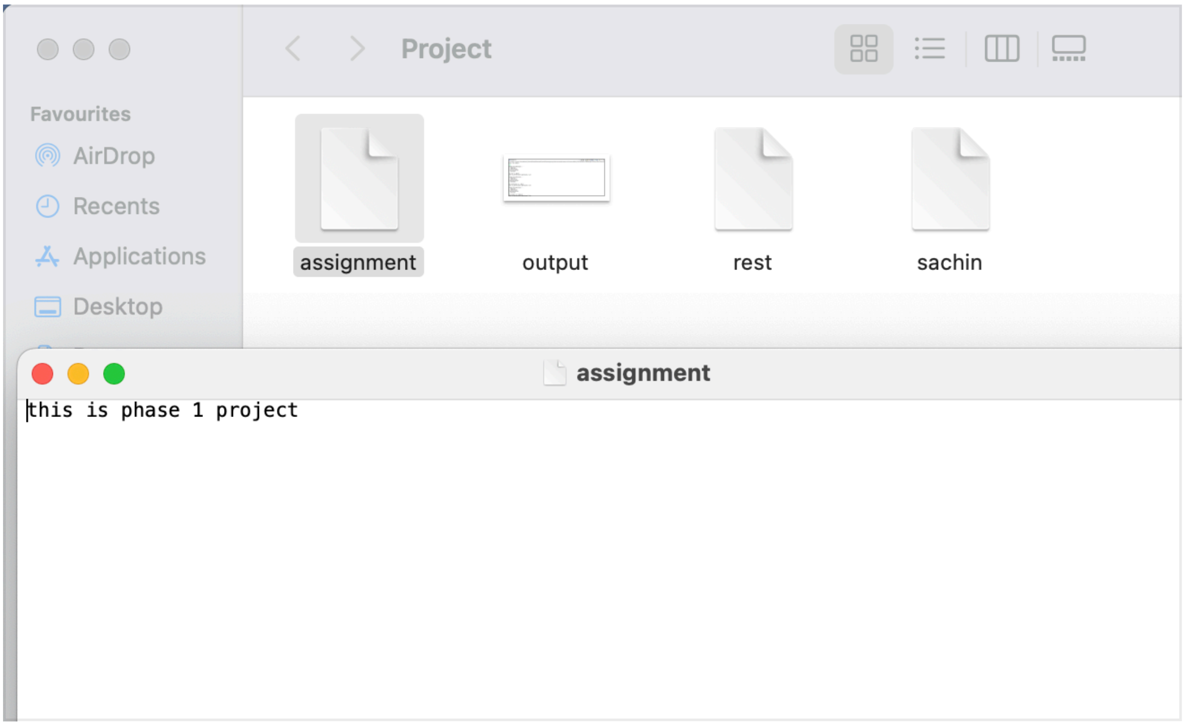
The folder created is



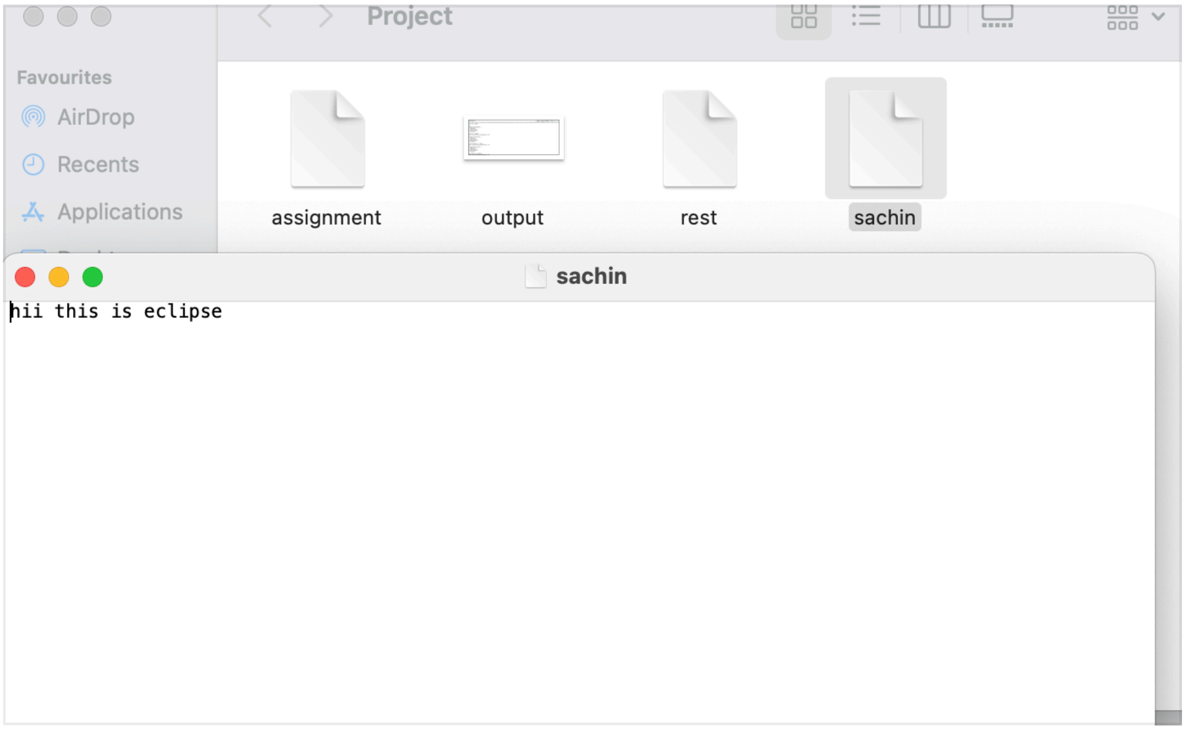
The files inside the folder is:



The content inside the file:



```
this is phase 1 project
```



```
hii this is eclipse
```