**Source Code for Phase 1 Project**

import java.io.File;

import java.util.Arrays;

import java.util.Scanner;

import java.io.IOException;

public class Phase\_Proj {

static String Folder;

File fold\_nm;

public Phase\_Proj() {

Folder = System.getProperty("user.dir");

fold\_nm = new File(Folder+"/files");

if (!fold\_nm.exists())

fold\_nm.mkdirs();

System.out.println("Folder:"+ fold\_nm.getAbsolutePath());

}

private static final String Proj\_Info =

"\n SimpliLearn Phase1 Assessment Project by Jagrut Goenka";

private static final String Main\_Menu =

"\nMAIN MENU - Select any of the following: \n"+

"1 : Sort files in ascending order\n"+

"2 : Adding file, Deleting file or Searching file in folder\n"+

"3 : Quit Program";

private static final String Sub\_Menu =

" \nSelect any of the following: \n"+

" 1 : Add a file in folder\n"+

" 2 : Delete a file from the folder \n"+

" 3 : Search a file in the folder\n"+

" 4 : Return back to the previous menu ";

void Init\_Menu () {

System.out.println(Main\_Menu);

try{

Scanner scanner = new Scanner(System.in);

int option = scanner.nextInt();

switch (option){

case 1 : {

Asc\_Operat();

Init\_Menu ();

}

case 2 : {

Inner\_Menu();

}

case 3 : {

System.out.println("\n END ");

System.exit(0);

}

default: Init\_Menu ();

}

scanner.close();

}

catch (Exception e){

System.out.println("Enter a valid choice from 1.2.3");

Init\_Menu ();

}

}

void Inner\_Menu() {

System.out.println(Sub\_Menu);

try{

Scanner scanner = new Scanner(System.in);

char[] input = scanner.nextLine().toLowerCase().trim().toCharArray();

char option = input[0];

switch (option){

case '1' : {

System.out.print("1 : Add a file in folder : \t ");

String filename = scanner.next().trim().toLowerCase();

Add\_Operat(filename);

break;

}

case '2' : {

System.out.print("2 : Delete a file from the folder : \t");

String filename = scanner.next().trim();

Delete\_Operat(filename);

break;

}

case '3' : {

System.out.print("3 : Search a file in the folder \t");

String filename = scanner.next().trim();

Search\_Operat(filename);

break;

}

case '4' : {

System.out.println("4 : Return back to the previous menu ");

Init\_Menu ();

break;

}

default : System.out.println("Please enter a, b, c or d");

}

Inner\_Menu();

scanner.close();

}

catch (Exception e){

System.out.println("Enter a valid choice from 1.2.3.4");

Inner\_Menu();

}

}

void Asc\_Operat() {

if (fold\_nm.list().length==0)

System.out.println("The folder is empty");

else {

String[] list = fold\_nm.list();

System.out.println("The files in "+ fold\_nm +" are :");

Arrays.sort(list);

for (String str:list) {

System.out.println(str);

}

}

}

void Delete\_Operat(String filename) {

File filepath = new File(fold\_nm +"/"+filename);

String[] list = fold\_nm.list();

for (String file: list) {

if (filename.equals(file) && filepath.delete()) {

System.out.println("File " + filename + " deleted from " + fold\_nm);

return;

}

}

System.out.println("Delete Operation failed. FILE NOT FOUND");

}

void Search\_Operat(String filename) {

String[] list = fold\_nm.list();

for (String file: list) {

if (filename.equals(file)) {

System.out.println("File " + filename + " is located at " + fold\_nm);

return;

}

}

System.out.println("File not found ");

}

void Add\_Operat(String filename) throws IOException {

File filepath = new File(fold\_nm +"/"+filename);

String[] list = fold\_nm.list();

for (String file: list) {

if (filename.equalsIgnoreCase(file)) {

System.out.println("File " + filename + " already exists at " + fold\_nm);

return;

}

}

filepath.createNewFile();

System.out.println("File "+filename+" added to "+ fold\_nm);

}

public static void main(String[] args) {

System.out.println(Proj\_Info );

Phase\_Proj menu = new Phase\_Proj();

menu.Init\_Menu ();

}

}