**Java Code For-- Virtual Key for Reposiotries ---**

**Main:-**

package virtualKey;

public class Main {

public static void main(String[] args) {

WelcomeScreen welcome = new WelcomeScreen();

welcome.Intro();

welcome.MainMenu();

}

}

**Welcome Screen:--**

package virtualKey;

import java.util.Scanner;

public class WelcomeScreen {

public void Intro() {

String AppName = "Virtual Key For Repositories";

String DeveloperName = "Rav kashyap";

System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

System.out.println(" !Welcome to the " + AppName + " applicaiton! \n ");

System.out.println(" Developer: " + DeveloperName + " ");

System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

}

public void MainMenu() {

System.out.println("\nMain Menu");

System.out.println("1.Show files\n2.File Options Menu\n3.Exit\n");

boolean running = true;

Scanner option = new Scanner(System.in);

do {

try {

int input = option.nextInt();

switch (input) {

case 1:

this.ShowFiles();

this.MainMenu();

break;

case 2: // pending

FileOptionsMenu FileMenu = new FileOptionsMenu();

FileMenu.show();

break;

case 3:

System.out.println("Quitting the application....");

System.out.println("Are you sure? Y/N");

Scanner sure = new Scanner(System.in);

String s = sure.nextLine();

if (s.equals("y") || s.equals("Y")) {

System.out.println("Applicaiton terminated");

running = false;

System.exit(0);

} else {

MainMenu();

}

default:

System.out.println("Invalid option,please enter invalid option");

break;

}

} catch (Exception e) {

System.out.println(e.getClass().getName() + ": Please enter a valid option");

MainMenu();

}

} while (running == true);

}

private void ShowFiles() {

Directory obj = new Directory(); // Retrieve files from directory

obj.getFiles();

}

}

**File Options Menu:--**

package virtualKey;

import java.io.File;

import java.io.IOException;

import java.nio.file.FileSystems;

import java.nio.file.Path;

import java.util.ArrayList;

import java.util.InputMismatchException;

import java.util.Scanner;

public class FileOptionsMenu {

private Directory dir = new Directory();

public void MenuHandler() {

System.out.println("""

1. Add a File

2. Delete a file

3. Search for a file

4. Return to Main Menu""");

boolean running = true;

Scanner option = new Scanner(System.in);

do {

try {

int input = option.nextInt();

switch (input) {

case 1:

this.AddFile();

this.show();

break;

case 2:

this.DeleteFile();

this.show();

break;

case 3:

this.SearchFile();

this.show();

break;

case 4:

WelcomeScreen obj = new WelcomeScreen();

obj.MainMenu();

break;

default:

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

break;

}

} catch (InputMismatchException e) {

System.out.println(e + ": Please select a valid option");

this.MenuHandler();

}

}

while (running);

}

public void show() {

System.out.println("\nFile Options Menu");

MenuHandler();

}

private String getInputSting() {

Scanner in = new Scanner(System.in);

return (in.nextLine());

}

private void AddFile() {

System.out.println("Enter file name:");

String filename = this.getInputSting();

System.out.println("Adding the file:" + filename);

try {

Path path = FileSystems.getDefault().getPath(Directory.pathName + filename).toAbsolutePath();

File file = new File(dir.getPathName() + filename);

if (file.createNewFile()) {

System.out.println("File added: " + file.getName());

dir.ListFiles().add(file);

} else {

System.out.println("File already Exists");

}

} catch (IOException e) {

System.out.println(e);

}

}

private void DeleteFile() {

Directory obj = new Directory();

obj.getFiles();

System.out.println("Enter file name<filename.extension> to delete:");

String filename = this.getInputSting();

System.out.println("Deleting the file: " + filename + "\nAre you sure? (Y/N)");

String sure = this.getInputSting();

if(sure.equals("y") || sure.equals("Y")) { //asks sure?

Path path = FileSystems.getDefault().getPath(Directory.pathName + filename).toAbsolutePath();

File file = path.toFile();

if (file.delete()) {

System.out.println("Deleted file: " + file.getName());

dir.ListFiles().remove(file);

} else {

System.out.println("Failed, file not found");

}

}

else

{

return;

}

}

private void SearchFile() {

boolean found = false;

System.out.println("Enter file name:");

String fileName = this.getInputSting();

System.out.println("Searching for file: " + fileName);

ArrayList<File> files = dir.ListFiles();

for (File file : files) {

if (file.getName().equals(fileName)) {

System.out.println("Found " + fileName);

found = true;

}

}

if (!found) {

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

}

}

}

**Direcctory:--**

package virtualKey;

import java.io.File;

import java.nio.file.FileSystems;

import java.nio.file.Path;

import java.util.ArrayList;

import java.util.Collections;

public class Directory {

public static final String pathName = "C:\\Users\\kashy\\Desktop\\AddedFiles/";

private ArrayList<File> files = new ArrayList<>();

Path path = FileSystems.getDefault().getPath(pathName).toAbsolutePath();

File Dfiles = path.toFile();

public String getPathName() {

return pathName;

}

public ArrayList<File> ListFiles() { //sort files

File[] directoryFiles = Dfiles.listFiles();

files.clear();

for(File directoryFile : directoryFiles) {

if(directoryFile.isFile()) {

files.add(directoryFile);

}

}

Collections.sort(files);

return files;

}

public ArrayList<File> getFiles() {

ListFiles();

if (files.isEmpty()){

System.out.println("No files exist");}

else {

System.out.println("Existing files: ");

for (File file : ListFiles()) {

System.out.println(file.getName());

}

}

return files;

}

}