

PROJECT 1

VIRTUAL KEY FOR REPOSITORIES

CODING

1 creating the project in eclipse

- open eclipse
- go to file -> new -> project -> java project -> next.
- type in any project name and click on "finish."
- select your project and go to file -> new -> class.
- enter **lockedme** in any class name, check the checkbox "public static void main(string[] args)", and click on "finish."

2 writing a program in java for the entry point of the application

```
import java.io.file;
import java.io.ioexception;
import java.util.arrays;
import java.util.scanner;
import java.util.regex.matcher;
import java.util.regex.pattern;
public class lockedme {
    static string directory;
    file f_name;
    public lockedme() {
        directory = system.getproperty("user.dir");
        f_name = new file(directory+"files");
        if (!f_name.exists())
            f_name.mkdirs();
    }
}
```

3 writing a program in java to display menu options available for the user

```
void displaymainmenu() {
    system.out.println(menu);
    try{
        system.out.print("choose the correct options from the given menu:");
        scanner sc = new scanner(system.in);
        int option = sc.nextInt();
        switch (option){
            case 1 : {
                displayfiles();
                displaymainmenu();
            }
            case 2 : {
                displaysubmenu();
            }
            case 3 : {
                system.out.println("thank you for your valuable time");
                system.exit(0);
            }
            default: displaymainmenu();
        }
    }
    catch (exception e){
        system.out.println("kindly choose the correct option like 1, 2 or 3");
        displaymainmenu();
    }
}
```

SHRADDHA CHOUHAN

FULL STACK DEVELOPER
COMPANY LOCKER PVT LTD.

PROJECT 1

VIRTUAL KEY FOR REPOSITORIES

CODING

```
void displaysubmenu() {
    system.out.println(submenu);
    try{
        scanner sc = new scanner(system.in);
        char[] input = sc.nextline().toLowerCase().trim().toCharArray();
        char option = input[0];
        switch (option){
            case 'a' : {
                system.out.print("enter the file name which you want to add : ");
                string filename = sc.next().trim();
                addfile(filename);
                break;
            }
            case 'b' : {
                system.out.print("enter the file name which you want to delete : ");
                string filename = sc.next().trim();
                deletefile(filename);
                break;
            }
            case 'c' : {
                system.out.print("enter the file name which you want to search : ");
                string filename = sc.next().trim();
                searchfile(filename);
                break;
            }
            case 'd' : {
                system.out.println("moving back to the previuos menu");
                displaymainmenu();
                break;
            }
            default : system.out.println("kindly enter the correct option a, b, c or d ");
        }
        displaysubmenu();
    }
    catch (exception e){
        system.out.println("kindly enter the correct option a, b, c or d ");
        displaysubmenu();
    }
}

public static void main(string[] args) {
    system.out.println(welcome_screen);
    lockedme menu = new lockedme();
    menu.displaymainmenu();
}
```

PROJECT 1

VIRTUAL KEY FOR REPOSITORIES

CODING

4 Writing a program in Java to handle Menu options selected by user

```
void DisplayFiles() {
    if (f_name.list().length==0)
        System.out.println("The folder is empty , no files found");
    else {
        String[] list = f_name.list();
        System.out.println("The files in " + f_name + " are :");
        Arrays.sort(list);
        for (String str:list) {
            System.out.println(str);
        }
    }
}
```

5 Writing a program in Java to perform the File operations as specified by user (FileOperations.java)

```
void AddFile(String filename) throws IOException {
    File filepath = new File(f_name + "/" + filename);
    String[] list = f_name.list();
    for (String file: list) {
        if (filename.equalsIgnoreCase(file)) {
            System.out.println("File " + filename + " already exists with similar name at " + f_name);
            return;
        }
    }
    filepath.createNewFile();
    System.out.println("File " + filename + " added to " + f_name);
}

void DeleteFile(String filename) throws IOException{
    Pattern p=Pattern.compile("a-zA-Z0-9");
    File filepath = new File(f_name + "/" + filename);
    Matcher matcher= p.matcher(filename);
    String[] list = f_name.list();
    for (String file: list) {
        if (filename.equals(file) && filepath.delete()) {
            System.out.println("File " + filename + " deleted from " + f_name);
            return;
        }
    }
    System.out.println("Delete Operation failed. FILE NOT FOUND");
}

void SearchFile(String filename) {
    File f=new File("D:\\ECLIPSE WORKSPACE\\LockedMe.com\\files");
    Pattern p=Pattern.compile("a-zA-Z0-9");
    File filenames[]=f.listFiles();
    Matcher matcher= p.matcher(filename);
    int flag=0;
    for(File ff: filenames) {
        if(ff.getName().equals(filename)) {
            flag=1;
            break;
        }
        else {
            flag=0;
        }
    }
    if(flag==1) {
        System.out.println("File is available, Status -> FILE FOUND : File " + filename + " exists at " + f_name);
    }
    else {
        System.out.println("File is not available, Status -> FILE NOT FOUND");
    }
}
```

6 Pushing the code to GitHub repository

- Open your command prompt and navigate to the folder where you have created your files.

cd <folder path>

- Initialize repository using the following command:

git init

- Add all the files to your git repository using the following command:

git add .

- Commit the changes using the following command:

git commit . -m <commit message>

- Push the files to the folder you initially created using the following command:

git push -u origin master