**Implement OOPS using JAVA with Data Structures and Beyond**

**Assignment**

**Problem Statement:** As a Full Stack Developer, complete the features of the application by planning the development in terms of sprints and then push the source code to the GitHub repository. As this is a prototyped application, the user interaction will be via a command line.

Background of the problem statement:

**Coding section:**

import java.io.File;

import java.util.Scanner;

public class FileProject {

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

    static String path = "C:\\Phase 1 Project\\phase1\\";

    public static void main(String[] args) {

        int choice;

        boolean whilLoopExit = false;

        while (!whilLoopExit) {

            try {

                welcomeScreen();

                System.out.println("Enter an option from above: ");

                choice = Integer.parseInt(sc.nextLine());

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

                System.out.println(">> Selected option: " + choice);

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

                switch (choice) {

                    case 1:

                        displayFiles();

                        break;

                    case 2:

                        createFile();

                        break;

                    case 3:

                        deleteFile();

                        break;

                    case 4:

                        searchFile();

                        break;

                    case 5:

                        whilLoopExit = true;

                        break;

                }

            } catch (Exception e) {

                System.out.println("Invalid option selected. Enter a valid option.");

            }

        }

    }

    // Function to display welcome screen with 5 option

    public static void welcomeScreen() {

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

        System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*Lockers Pvt. Ltd.\*\*\*\*\*\*\*\*\*\*\*\*");

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

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

        System.out.println("\*\*\*\*\*\*\*\*\*Developed by Ashwani Kumar\*\*\*\*\*\*\*");

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

        System.out.println("          1) Display all files");

        System.out.println("          2) Add a file");

        System.out.println("          3) Delete a file");

        System.out.println("          4) Search a file");

        System.out.println("          5) Exit");

    }

    // Fuction to display all files in root directory.

    public static void displayFiles() {

        File folder = new File(path);

        File listOfFiles[] = folder.listFiles();

        System.out.println("List of files and folders:");

        for (File fileName : listOfFiles) {

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

        }

    }

    // Function to create a file.

    public static void createFile() {

        try {

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

            String fileName = sc.next();

            String finalFile = path + fileName;

            File f = new File(finalFile);

            boolean res = f.createNewFile();

            if (res == false) {

                System.out.println("File is not created.");

            } else {

                System.out.println("File is created.");

            }

        } catch (Exception e) {

            System.out.println("File is not created.");

        }

    }

    // Function to delete a file

    public static void deleteFile() {

        try {

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

            String fileName = sc.next();

            String finalFile = path + fileName;

            File f = new File(finalFile);

            boolean res = f.delete();

            if (res == false) {

                System.out.println("There was an error in deleting the file.");

            } else {

                System.out.println("File is deleted.");

            }

        } catch (Exception e) {

            System.out.println("There was an error in deleting the file.");

        }

    }

    // Function to search for a file

    public static void searchFile() {

        try {

            File folder = new File(path);

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

            File fileToSearch = new File(folder + "//" + sc.nextLine());

            if (fileToSearch.exists()) {

                System.out.println("File exist.");

            } else {

                System.out.println("File does not exist.");

            }

        } catch (Exception e) {

            System.out.println("There was an error searching for the file.");

        }

    }

}