|  |  |
| --- | --- |
|  | |
|  | |
|  | |
|  |

|  |  |
| --- | --- |
| FOCP II | |
| Project Report | |
| **Department of Computer Science and Engineering**  **The NorthCap University, Gurugram** | |
|  |

**FOCP MAJOR PROJECT**

**PRODUCTIVITY HOURS**

**CALCULATOR APP**

BY –

DRIZZLE AHUJA ( 22CSU299 )

YASH SHARMA ( 22CSU295 )

ADITYA MALHOTRA ( 22CSU297 )

EVA SOLANKI ( 22CSU356 )

SATYA MISHRA ( 22CSU475 )

**A REPORT**

**ON**

**PRODUCTIVITY CALCULATOR APP**

BY –

DRIZZLE

YASH

ADITYA

EVA

SATYA

**SUBMITTED TO**

**MS. JYOTI YADAV**

**MAY 2023**

**COPYRIGHT NOTICE**

**@ NCU 2023**

**ALL RIGHTS RESERVED .**

No parts of this report may be reproduced in any form or any means without permission in writing from the publisher.

**ACKNOWLEDGEMENT**

We would like to present our gratitude for the kind support and help of many individuals and Sources. we would like to extend our sincere thanks to all of them.

We are highly indebted to our FOCP professor Ms. Jyoti Yadav for encouraging us to take this project.

We would also like to express our gratitude towards our parents and friends for their kind cooperation and encouragement which helped us in completion of this project

**INTRODUCTION**

This is a java application to display how many productive hours we have in a day where we can also add , update and delete the day to day Routine tasks in a list.

This app would be helpful for a user to manage his / her time properly and segregate his/ her tasks in categories ie Productive and Non Productive.

**TABLE OF CONTENT**

|  |  |  |
| --- | --- | --- |
| **S NO.** | **TOPIC** | **PAGE NO.** |
| 1. | DESCRIPTION | 8 |
| 2. | ALGORITHM | 9 |
| 3. | CLASS DIAGRAM | 11 |
| 4. | BLOCK DIAGRAM | 12 |
| 5. | CODE | 14 |
| 6. | OUTPUT | 20 |
| 7. | FUTURE SCOPE | 23 |
| 8. | BIBLIOGRAPHY | 24 |

**DESCRIPTION**

A java application to display how many productive hours we have in a day where we can also add, update and delete the routine-tasks in the list.

**Definition of Done:**

1. The system should ask user to enter user name and password. On successful login, System asks to create a task List for the logged user.
2. The system should ask user to enter an option from the selected list of operations(Add, Update, Delete)
3. The user should enter the routine tasks, category(Productive/Non-Productive) and the hours the task takes to get accomplished. Upon segregation of essential but non-productive tasks, the user needs to calculate the number of productive hours he is left with.
4. The system should perform the desired operations on the Productivity\_Calculator.txt file for the logged-in user (Add, Delete, Update, display).
5. The project directory structure should explain about all subdirectories and what kind of files to be kept into particular sub directory.
6. All configuration data should be kept in properties files. It should maintainable based on different categories.

**ALGORITHM**

1. Ask the user to enter their username and password.
2. Verify the username and password by checking them against a list of registered users.
3. If the login is successful, ask the user to create a task list for the logged-in user.
4. Present the user with a menu of operations (Add, Update, Delete, Display, Exit).
5. Based on the user's choice, perform the following actions: a. Add:
   1. Ask the user to enter the routine task, category (Productive/Non-Productive), and the hours the task takes to get accomplished.
   2. Create a new Task object with the entered details and add it to the task list. b. Update:
   3. Ask the user to select a task from the task list to update.
   4. Prompt the user to enter the new information for the task.
   5. Update the selected task with the new information. c. Delete:
   6. Ask the user to select a task from the task list to delete.
   7. Remove the selected task from the task list. d. Display:
   8. Show the user the list of tasks and their details. e. Exit:
   9. End the program.
6. Calculate the total hours of non-productive tasks and subtract them from the total hours in a day (e.g., 24 hours). Display the remaining productive hours to the user.
7. Save the task list to a file (e.g., "Productivity\_Calculator.txt") for the logged-in user.
8. Use a project directory structure that explains all subdirectories and what kind of files to be kept in particular subdirectories.
9. Store all configuration data in properties files, which should be maintainable based on different categories.

**Class diagram**

**A picture containing text, businesscard, screenshot

Description automatically generated**

**BLOCK DIAGRAM**

**A screenshot of a computer screen

Description automatically generated with low confidence**

**A screenshot of a computer screen

Description automatically generated with low confidence**

**CODE**

import java.io.\*;

import java.util.\*;

public class Task {

    public static void main(String args[]) {

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

        ArrayList<Integer> durations = new ArrayList<>();

        Scanner scanner = new Scanner(System.in);

        boolean loggedIn = false;

        String username = "";

        String password = "";

        System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_     PRODUCTIVITY CALCULATOR APP     \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n\n");

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

        System.out.println("  || THIS IS A PLACE WHERE YOU CAN KEEP TRACK ON YOUR DAILY ROUTINE TASK  ||\n ");

        System.out.println("  \n\*\*  IF YOU ARE A NEW USER ENTER YOUR NEW USERNAME AND PASSWORD ");

        System.out.println("  \n\*\*  IF YOU ARE AN EXISTING USER ENTER YOUR EXISTING USERNAME AND PASSWORD\n ");

        while (true) {

            if (!loggedIn) {

                System.out.print("   Enter username : ");

                String enteredUsername = scanner.nextLine();

                System.out.print("   Enter password : ");

                String enteredPassword = scanner.nextLine();

                // create file with username and password

                File inputFile = new File(enteredUsername + "\_Productivity\_Calculator.txt");

                if (!inputFile.exists()) {

                    try {

                        FileWriter fw = new FileWriter(inputFile);

                        PrintWriter pw = new PrintWriter(fw);

                        pw.println("USERNAME = " + enteredUsername);

                        pw.println("PASSWORD = " + enteredPassword);

                        System.out.println(" \n\* NEW ACCOUNT CREATED!! \*");

                        pw.close();

                    } catch (IOException e) {

                        e.printStackTrace();

                    }

                }

                // check if entered username and password match stored values

                try {

                    Scanner fileScanner = new Scanner(inputFile);

                    String storedUsername = fileScanner.nextLine().substring(11); // remove "USERNAME = " from the stored value

                    String storedPassword = fileScanner.nextLine().substring(11); // remove "PASSWORD = " from the stored value

                    if (enteredUsername.equals(storedUsername) && enteredPassword.equals(storedPassword)) {

                        System.out.println("\n\n \_\_\_\_\_\_\_ WELCOME!! LOGGED IN SUCCESSFULLY AS " + enteredUsername + " \_\_\_\_\_\_\n\n");

                        loggedIn = true;

                        username = enteredUsername;

                        password = enteredPassword;

                    } else {

                        System.out.println("Incorrect username or password, please try again.\n");

                    }

                } catch (IOException e) {

                    System.out.println("Incorrect username or password, please try again.\n");

                }

            }

            if (loggedIn) {

                System.out.println("  BEGIN BY ADDING YOUR FIRST TASK ");

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

                String task = scanner.nextLine();

                System.out.println("Enter task duration in minutes:");

                int duration = scanner.nextInt();

                scanner.nextLine();

                tasks.add(task);

                durations.add(duration);

                // write task to file

                try {

                    FileWriter fw = new FileWriter(username + "\_Productivity\_Calculator.txt", true);

                    PrintWriter pw = new PrintWriter(fw);

                    pw.println(task + " ( " + duration + " ) \n");

                    pw.close();

                } catch (IOException e) {

                    e.printStackTrace();

                }

                System.out.println("\nNOW YOU HAVE ADDED YOUR TASK AND NOW YOU CAN PERFORM OTHER OPERATIONS");

                int choice = 0;

                while (choice != 5) {

                    System.out.println("\nTHESE ARE VARIOUS OPTIONS YOU CAN PERFORM ");

                    System.out.println("\nEnter\n 1 : To Add Task\n 2 : To Update Task\n 3 : To Delete Task\n 4 : To Display Tasks\n 5 : EXIT HERE\n\n");

                    choice = scanner.nextInt();

                    scanner.nextLine();

                                    if (choice == 1) {

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

                    task = scanner.nextLine();

                    System.out.println("Enter task duration in minutes:");

                    duration = scanner.nextInt();

                    scanner.nextLine();

                    tasks.add(task);

                    durations.add(duration);

                    // write task to file

                    try {

                        FileWriter fw = new FileWriter(username + "\_Productivity\_Calculator.txt", true);

                        PrintWriter pw = new PrintWriter(fw);

                        pw.println(task + " ( " + duration + " ) \n");

                        pw.close();

                    } catch (IOException e) {

                        e.printStackTrace();

                    }

                    System.out.println("TASK ADDED SUCCESSFULLY");

                } else if (choice == 2) {

                    System.out.println("Enter task name to update:");

                    String oldTask = scanner.nextLine();

                    int index = tasks.indexOf(oldTask);

                    if (index == -1) {

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

                    } else {

                        System.out.println("Enter new task name:");

                        task = scanner.nextLine();

                        System.out.println("Enter new task duration in minutes:");

                        duration = scanner.nextInt();

                        scanner.nextLine();

                        tasks.set(index, task);

                        durations.set(index, duration);

                        // update task in file

                        try {

                            FileWriter fw = new FileWriter(username + "\_Productivity\_Calculator.txt");

                            PrintWriter pw = new PrintWriter(fw);

                            for (int i = 0; i < tasks.size(); i++) {

                                pw.println(tasks.get(i) + " ( " + durations.get(i) + " ) \n");

                            }

                            pw.close();

                        } catch (IOException e) {

                            e.printStackTrace();

                        }

                        System.out.println("TASK UPDATED SUCCESSFULLY");

                    }

                } else if (choice == 3) {

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

                    String taskToDelete = scanner.nextLine();

                    int index = tasks.indexOf(taskToDelete);

                    if (index == -1) {

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

                    } else {

                        tasks.remove(index);

                        durations.remove(index);

                        // update file

                        try {

                            FileWriter fw = new FileWriter(username + "\_Productivity\_Calculator.txt");

                            PrintWriter pw = new PrintWriter(fw);

                            for (int i = 0; i < tasks.size(); i++) {

                                pw.println(tasks.get(i) + " ( " + durations.get(i) + " ) \n");

                            }

                            pw.close();

                        } catch (IOException e) {

                            e.printStackTrace();

                        }

                        System.out.println("TASK DELETED SUCCESSFULLY");

                    }

                } else if (choice == 4) {

                    System.out.println("YOUR CURRENT TASKS ARE: ");

                    for (int i = 0; i < tasks.size(); i++) {

                        System.out.println(tasks.get(i) + " ( " + durations.get(i) + " minutes )");

                    }

             } else if (choice == 5) {

    System.out.println("EXITING APPLICATION");

    int totalDuration = 0;

    for (int d : durations) {

        totalDuration += d;

    }

    int productiveHours = totalDuration / 60;

    int productiveMinutes = totalDuration % 60;

    System.out.println("Total productive time: " + productiveHours + " hours " + productiveMinutes + " minutes");

    // write productive hours to file

    try {

        FileWriter fw = new FileWriter(username + "\_Productivity\_Calculator.txt", true);

        PrintWriter pw = new PrintWriter(fw);

        pw.println("Total productive time: " + productiveHours + " hours " + productiveMinutes + " minutes");

        pw.println("TOTAL PRODUCTIVE TIME = " + totalDuration + " minutes");

        pw.close();

    } catch (IOException e) {

        e.printStackTrace();

    }

    System.out.println("Thank you for using the productivity calculator!");

    System.exit(0);

} else {

    System.out.println("Invalid input, please try again.");

}

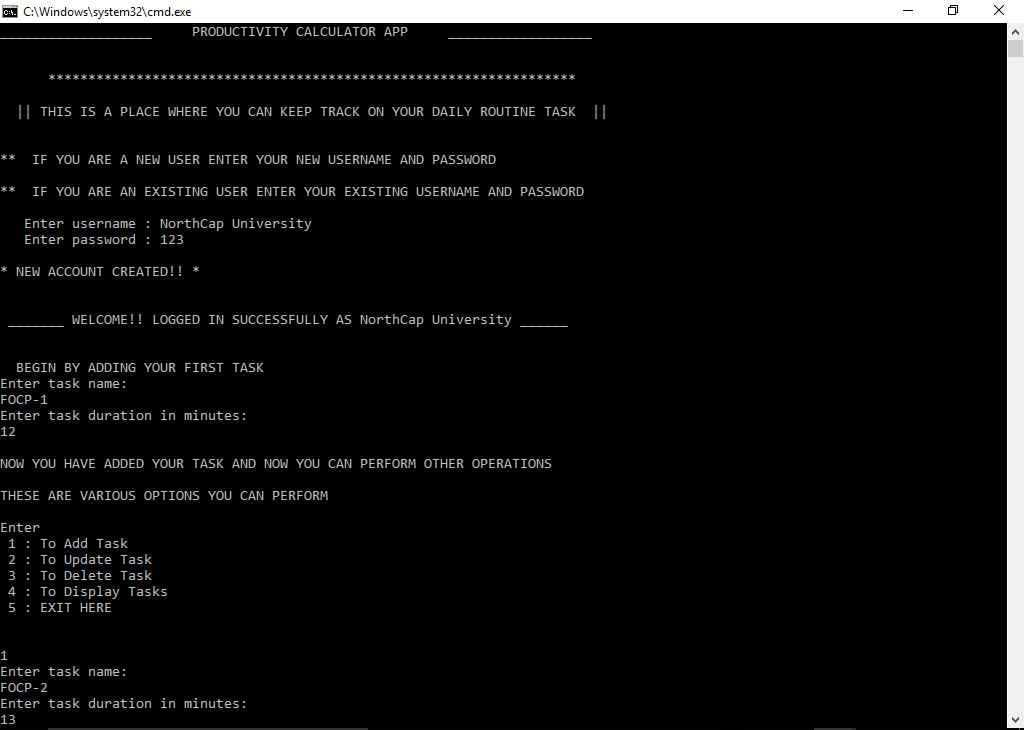
}

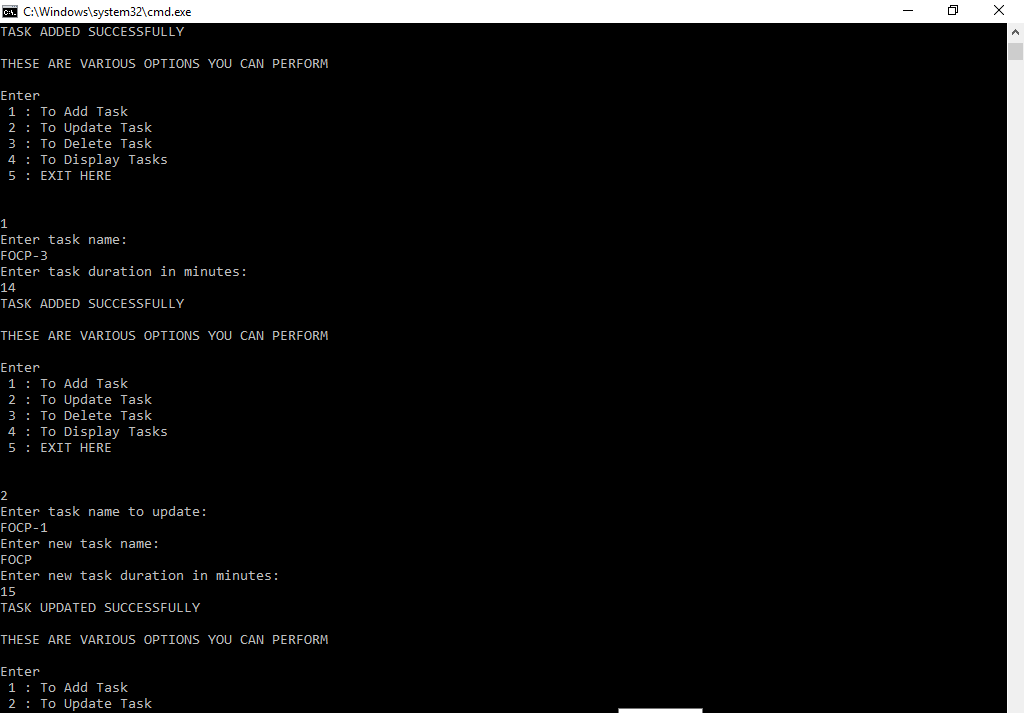
}

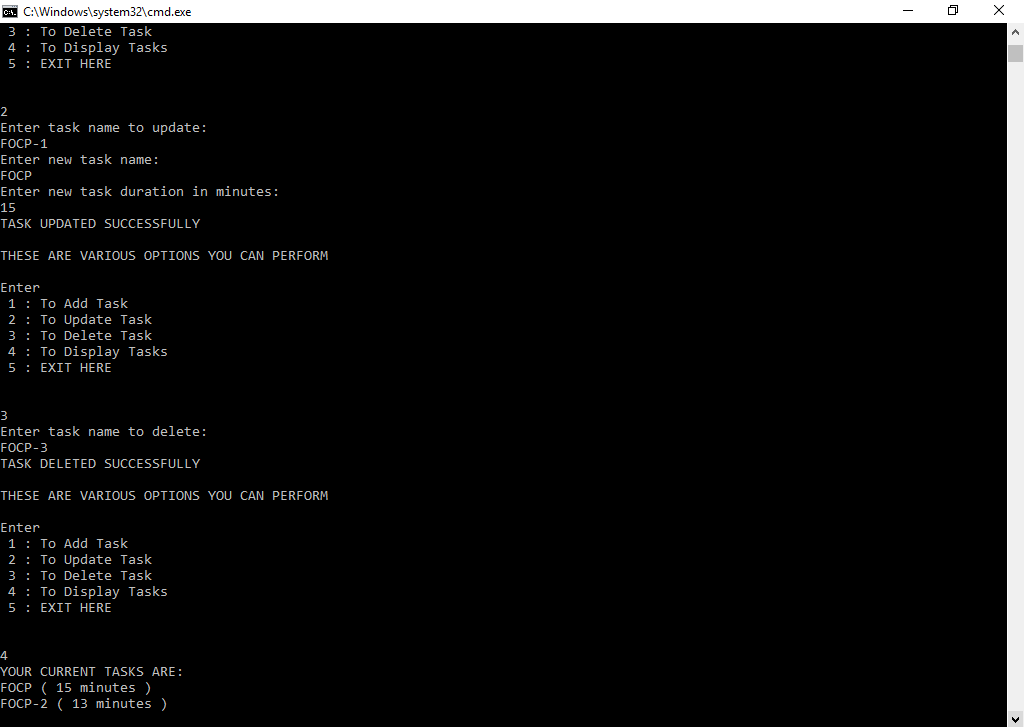
}

}}

**OUTPUT**





**FUTURE SCOPE**

**User authentication**: Currently, the app requires users to enter a username and password to access it, but there is no authentication process to ensure that the correct credentials are entered. Adding an authentication mechanism, such as two-factor authentication or OAuth, would increase the app's security.

**Data persistence**: Currently, the app does not store tasks or their durations once the program is closed. Adding a database or file system to persist data would allow users to close and reopen the app without losing their task list.

**Task categorization**: Users may want to categorize their tasks based on type, priority, or due date. Adding the ability to categorize tasks and sort them based on these criteria would improve the user experience.

**Notifications**: Users may want to receive notifications when a task is due or when a task has been updated or deleted. Adding notification functionality would improve the app's usability.

**Integration with other apps**: Users may want to integrate this appwith other productivity tools they use, such as calendars or to-do lists. Adding integration capabilities would increase the app's versatility and usefulness.

**User interface improvements**: Currently, the app has a simple command-line interface. Adding a graphical user interface (GUI) would make the app more user-friendly and visually appealing

**BIBLIOGRAPHY**

[**https://www.oracle.com/in/java/**](https://www.oracle.com/in/java/)

[**https://www.geeksforgeeks.org/java/**](https://www.geeksforgeeks.org/java/)

[**https://www.javatpoint.com/java-tutorial**](https://www.javatpoint.com/java-tutorial)