**RIPHAH INTERNATIONAL UNIVERSITY, ISLAMABAD**

****

**Lab Task 10**

**Bachelors of Software Engineering – 5th Semester**

**Subject: Web Engineering**

**Submitted to: Ms Shazwa tun naeem**

**Submitted by: Manahil Habib – 47876**

**Date of Submission: 4-November-2024**

import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

// Model class representing a Task

class Task {

    private int id;

    private String title;

    private String description;

    private boolean isCompleted;

    // Constructor to initialize task attributes

    public Task(int id, String title, String description) {

        this.id = id;

        this.title = title;

        this.description = description;

        this.isCompleted = false;

    }

    // Getter for task ID

    public int getId() {

        return id;

    }

    // Getter for task title

    public String getTitle() {

        return title;

    }

    // Getter for task description

    public String getDescription() {

        return description;

    }

    // Getter to check if the task is completed

    public boolean isCompleted() {

        return isCompleted;

    }

    // Method to mark the task as completed

    public void markAsCompleted() {

        this.isCompleted = true;

    }

    // Override toString to display task information

    @Override

    public String toString() {

        return "ID: " + id + ", Title: " + title + ", Description: " + description + ", Completed: " + (isCompleted ? "Yes" : "No");

    }

}

// View class responsible for displaying task information

class TaskView {

    // Display a list of tasks

    public void displayTasks(List<Task> tasks) {

        for (Task task : tasks) {

            System.out.println(task);

        }

    }

    // Prompt user for adding a new task

    public void promptAddTask() {

        System.out.println("\n--- Add a New Task ---");

    }

    // Prompt user for marking a task as completed

    public void promptMarkTaskCompleted() {

        System.out.println("\n--- Mark Task as Completed ---");

    }

}

// Controller class to manage task data and interactions

class TaskController {

    private List<Task> tasks = new ArrayList<>();

    private TaskView view;

    // Constructor to initialize view

    public TaskController(TaskView view) {

        this.view = view;

    }

    // Add a new task to the list

    public void addTask(int id, String title, String description) {

        Task task = new Task(id, title, description);

        tasks.add(task);

        System.out.println("Task added successfully!");

    }

    // Display all tasks using the view

    public void displayTasks() {

        view.displayTasks(tasks);

    }

    // Mark a task as completed based on task ID

    public void markTaskAsCompleted(int id) {

        for (Task task : tasks) {

            if (task.getId() == id) {

                task.markAsCompleted();

                System.out.println("Task marked as completed!");

                return;

            }

        }

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

    }

}

// Main application class to run the console-based application

public class Main {

    public static void main(String[] args) {

        // Initialize view and controller objects

        TaskView view = new TaskView();

        TaskController controller = new TaskController(view);

        Scanner scanner = new Scanner(System.in);

        System.out.println("Task Management Application");

        // Main application loop for user interaction

        while (true) {

            clearScreen(); // Clear console screen for a fresh view

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

            System.out.println("       Task Management Application  ");

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

            System.out.println("1. Add a New Task");

            System.out.println("2. View All Tasks");

            System.out.println("3. Mark Task as Completed");

            System.out.println("4. Exit");

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

            System.out.print("Choose an option (1-4): ");

            int choice = scanner.nextInt();

            scanner.nextLine();

            if (choice == 1) {

                view.promptAddTask(); // Prompt for task addition

                System.out.print("Enter Task ID: ");

                int id = scanner.nextInt();

                scanner.nextLine();

                System.out.print("Enter Task Title: ");

                String title = scanner.nextLine();

                System.out.print("Enter Task Description: ");

                String description = scanner.nextLine();

                controller.addTask(id, title, description); // Add new task

                pauseForUser(scanner);

            } else if (choice == 2) {

                System.out.println("\n--- All Tasks ---");

                controller.displayTasks(); // Display all tasks

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

                pauseForUser(scanner);

            } else if (choice == 3) {

                view.promptMarkTaskCompleted();

                System.out.print("Enter Task ID: ");

                int taskId = scanner.nextInt();

                controller.markTaskAsCompleted(taskId); // Mark task as completed

                pauseForUser(scanner);

            } else if (choice == 4) {

                System.out.println("Exiting the application. Goodbye!");

                scanner.close();

                break;

            } else {

                System.out.println("Invalid choice. Please try again.");

                pauseForUser(scanner);

            }

        }

    }

    // Method to clear the console screen

    public static void clearScreen() {

        try {

            if (System.getProperty("os.name").contains("Windows")) {

                new ProcessBuilder("cmd", "/c", "cls").inheritIO().start().waitFor();

            } else {

                System.out.print("\033[H\033[2J");

                System.out.flush();

            }

        } catch (Exception e) {

            System.out.println("Error clearing screen!");

        }

    }

    // Method to pause for user input before returning to the menu

    private static void pauseForUser(Scanner scanner) {

        System.out.println("Press Enter to return to menu...");

        scanner.nextLine();

    }

}