**TREUE TECHNOLIGIES**

**INTERNSHIP PROGRAME**

**TASK - 6**

**TASK REMAINDER :**

**Develop a task reminder application that helps users stay organized and manage their tasks effectively. The system should allow users to create tasks, set due dates, and receive reminders or notifications for upcoming tasks. Implement features like task categorization, priority levels, and task status tracking. Users should be able to view their tasks in a list or calendar view and mark tasks as completed.**

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.Date;

import java.util.List;

import java.util.Scanner;

import java.util.Timer;

import java.util.TimerTask;

class Task {

private String name;

private String category;

private String priority;

private Date dueDate;

private boolean completed;

public Task(String name, String category, String priority, Date dueDate) {

this.name = name;

this.category = category;

this.priority = priority;

this.dueDate = dueDate;

this.completed = false;

}

public String getName() {

return name;

}

public String getCategory() {

return category;

}

public String getPriority() {

return priority;

}

public Date getDueDate() {

return dueDate;

}

public boolean isCompleted() {

return completed;

}

public void markAsCompleted() {

completed = true;

}

}

class TaskReminder {

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

private Timer timer = new Timer();

public void addTask(Task task) {

tasks.add(task);

}

public void displayTasks() {

SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");

for (Task task : tasks) {

System.out.println("Task: " + task.getName());

System.out.println("Category: " + task.getCategory());

System.out.println("Priority: " + task.getPriority());

System.out.println("Due Date: " + dateFormat.format(task.getDueDate()));

System.out.println("Status: " + (task.isCompleted() ? "Completed" : "Not Completed"));

System.out.println();

}

}

public void markTaskAsCompleted(String taskName) {

for (Task task : tasks) {

if (task.getName().equalsIgnoreCase(taskName)) {

task.markAsCompleted();

return;

}

}

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

}

public void setReminderForTasks() {

for (Task task : tasks) {

if (!task.isCompleted() && task.getDueDate().after(new Date())) {

timer.schedule(new ReminderTask(task), task.getDueDate());

}

}

}

private class ReminderTask extends TimerTask {

private Task task;

public ReminderTask(Task task) {

this.task = task;

}

@Override

public void run() {

System.out.println("Reminder: Task '" + task.getName() + "' is due today!");

}

}

}

public class TaskReminderApp {

public static void main(String[] args) throws ParseException {

TaskReminder taskReminder = new TaskReminder();

Scanner scanner = new Scanner(System.in);

while (true) {

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

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

System.out.println("2. Display Tasks");

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

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

System.out.print("Select an option: ");

int choice = scanner.nextInt();

scanner.nextLine(); // Consume the newline character

switch (choice) {

case 1:

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

String name = scanner.nextLine();

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

String category = scanner.nextLine();

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

String priority = scanner.nextLine();

System.out.print("Enter due date (yyyy-MM-dd): ");

String dueDateStr = scanner.nextLine();

SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");

Date dueDate = dateFormat.parse(dueDateStr);

Task newTask = new Task(name, category, priority, dueDate);

taskReminder.addTask(newTask);

taskReminder.setReminderForTasks();

break;

case 2:

System.out.println("Tasks:");

taskReminder.displayTasks();

break;

case 3:

System.out.print("Enter the name of the task to mark as completed: ");

String taskName = scanner.nextLine();

taskReminder.markTaskAsCompleted(taskName);

break;

case 4:

scanner.close();

System.exit(0);

default:

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

break;

}

}

}

}