Marathwada Mitra mandal’s

COLLEGE OF ENGINEERING

Karvenagar, Pune

An Autonomous Institute

Presentation

On

PROJECT TOPIC Name

DESIGN AND MANAGEMENT OF DEPARTMENTAL CALENDAR....

GROUP ID == 11

DEPARTMENTAL CALENDAR

Sr.No	Prn No	Name of students
1)	B24IT1056	ATHARVA PARDESHI
2)	B24IT1005	KARAN BHAPKAR
3)	B24IT1020	OMKAR MARKAD
4)	B24IT1002	DEEPAK ILLPATE

Outline:

A. Introduction

In today's fast-paced world, managing time effectively is crucial. A calendar serves as an essential tool for organizing our schedules, tracking events, and celebrating special occasions. While traditional calendars provide a simple view of days, months, and years, incorporating events such as festivals can enhance their utility and make them more meaningful.
This project focuses on creating a calendar program in C that not only displays the days of a specific month and year but also highlights important festivals and events. By combining basic calendar functionalities with the ability to showcase cultural and national celebrations, this program aims to offer a more interactive and informative experience.
The calendar program is designed to:

- Prompt users for a specific month and year.
- Calculate the starting day of that month.
- Adjust for leap years to ensure accurate day counts.
- Display the calendar layout with the days of the month aligned correctly.
- Include a list of predefined festivals, indicating their occurrence within the calendar.

Through this implementation, users will gain a better appreciation of how programming can intersect with everyday life, enabling them to plan and celebrate important dates more effectively. The program serves as a foundation that can be further expanded with features such as user-defined events, reminders, or even integration with digital calendars.

As we delve into the details of the program, we will explore the logic behind calendar calculations, the handling of user input, and the incorporation of events, all while emphasizing the importance of clear and maintainable code.

B.RESEARCH

PRIMARY RESEARCH: Primary research refers to the direct collection of data from relevant stakeholders to understand their needs and preferences. This research will help gather firsthand insights that will influence the design and functionality of the departmental calendar system.

1. Surveys

o A survey was conducted among departmental employees to understand their expectations from the calendar system. The survey contained questions about:

Preferred calendar features (e.g., event reminders, color coding, recurring events).

Preferred integration with other software (e.g., email, project management tools).

Frequency of calendar usage (daily, weekly, monthly).

Pain points with existing calendar tools (e.g., Google Calendar, Outlook).

o Key Findings:

80% of respondents prefer automated reminders for upcoming events and meetings.

65% prefer a color-coded system to differentiate between events.

Integration with email and project management tools (like Asana and Jira) was cited as essential by 75% of the participants.

40% of users experience difficulties in managing overlapping meetings or events

C. Analysis

The system meets the needs of the department and its users. This analysis breaks down the main challenges and
requirements, categorizing them into functional, technical, and user-related concerns.
1. Functional Challenges
These challenges pertain to the core features and capabilities of the departmental calendar system:
• Event Management:
o Users need to easily create, update, and delete events. The calendar must support various types of events such as meetings, deadlines, and reminders.
o Recurring Events: A significant challenge arises with setting up recurring events with different frequencies (e.g., weekly, monthly, yearly). This requires a sophisticated system to handle complex recurrence rules (e.g., "every first Monday of the month").
o Overlapping Events: Users often experience difficulty in managing overlapping meetings or events. The system should visually indicate and manage conflicts when scheduling, ensuring users are alerted or guided to resolve the issue.
• Collaboration and Sharing:
o A major requirement is the ability to share calendars among multiple users within a department or between departments. This includes permissions for who can view, edit, and schedule events.
o Different roles (e.g., managers, team members, administrative staff) may need different levels of access to the calendar, making user access control crucial.
• Integration with Other Tools:
o Many departments rely on other tools like email, project management software (e.g., Asana, Jira), and communication platforms (e.g., Microsoft Teams, Slack). The calendar system must integrate seamlessly with these tools to ensure smooth workflow, data synchronization, and notifications.
Automated Reminders: Integration with email and messaging platforms to send automated reminders or notifications for events, ensuring that users are timely informed.
•mation

D.Ideate

1. Core Features and Functional Ideas
A.le Calendar Views
• Multiple View Options:
o Users should be able to toggle between different views (day, week, month, list, agenda) to fit their needs.
o Year View: A broad overview of the entire year for long-term planning.
7
o Task and Event Integration: Allow users to view tasks and deadlines alongside calendar events for better project management.
• Color-Coded Events and Categories:
o Allow users to categorize events (e.g., meetings, deadlines, holidays, personal tasks) and assign each category a specific color. This would make it easier to distinguish between types of activities at a glance.
o
Intelligent Event Management
• Smart Scheduling:
o An AI-driven feature that suggests optimal meeting times based on the schedules of participants, avoiding overlaps and minimizing idle time.
o Conflict Resolution: Automatically identify and resolve scheduling conflicts, notifying users of available slots or suggesting rescheduling options.
• Recurring Events:
o Allow users to create complex recurring events (e.g., every second Thursday, or quarterly review meetings) with options for exceptions (e.g., skip a specific date).
o Advanced Reminders: Set reminders at various intervals (e.g., 1 day before, 1 hour before, 5 minutes before) to ensure no event is missed.
Frontend Technologies

E. Build

These prototypes are typically created using digital tools or even
sketches, and their primary purpose is to validate ideas and test the user experience.
A. Wireframes
• What They Are: Wireframes are basic visual representations of the layout of the calendar system. They
focus on the structure of the user interface (UI), showing where elements like buttons, calendars, and text
will appear.
• Tools to Use:
o Figma, Sketch, Adobe XD for digital wireframes.
o Balsamiq for low-fidelity wireframes.
• Key Elements to Include:
o Calendar Layout: Show day, week, and month views of the calendar.
o Event Management Interface: Space for adding, editing, and viewing events.
9
o Sidebar or Navigation: A panel for switching between calendar views (day/week/month), task lists,
and team calendars.
o Event Popups: Design simple popups for event creation, editing, and viewing.
• Example Scenario:
o A wireframe for the day view could show a large time grid with sections for hours of the day, and
small buttons to create or edit events.
o A week view might show all the days horizontally, with a vertical timeline, and tasks/events listed
vertically in time blocks.
B. Mockups
• What They Are: Mockups are higher-fidelity representations compared to wireframes. They show detailed
designs with colors, typography, icons, and actual UI components.
• Tools to Use:
o Figma, Sketch, Adobe XD for mockups.
o Event Editing: Test how users modify or delete events So Notifications and Reminders: Simulate notifications for events or
conflicts.

F.Test

A. Usability Testing

• Goal: To identify pain points in the user interface (UI) and user experience (UX). This ensures that users can intuitively interact with the system and complete their tasks efficiently.

• Methods:

1. Task-Based Testing: Ask users to perform specific tasks on the calendar, such as scheduling a meeting, setting reminders, and adjusting recurring events. Observe where they encounter difficulties.
2. Think-Aloud Protocol: Ask users to vocalize their thoughts as they use the system. This allows testers to understand the reasoning behind their actions and identify points of confusion.
3. Error Logging: Track where users experience errors (e.g., form submission issues, navigation confusion) and analyze the frequency of these error

B. Performance Testing

11

• Goal: To ensure that the calendar system can handle multiple users simultaneously and perform well under stress.

• Methods:

1. Load Testing: Simulate high numbers of users (e.g., hundreds of users accessing the calendar at the same time) to assess how well the system scales.
2. Stress Testing: Push the system to its limits by increasing load to see where performance breaks down.
3. Response Time Testing: Measure the speed of the calendar's response to common actions, such as loading events, creating events, or changing views

C. Accessibility Testing

• Goal: To ensure that the calendar is accessible to all users, including those with disabilities.

• Methods:

. Screen Reader Testing: Ensure that all text and buttons are readable by screen readers for visually impaired users.

G.Implement

Problem: Managing departmental schedules, meetings, and events can be chaotic without a centralized sysem.
o Difficulty in coordinating schedules.
o Missed events, double-bookings, or poor communication.
o Lack of integration with existing tools.
• Solution: A Departmental Calendar System that helps teams schedule, track, and manage events efficiently with features like:
o Multiple Calendar Views (Day, Week, Month)
o Event Reminders & Notifications
o User Collaboration & Shared Calendar
o Seamless Integration with tools like Google Calendar, Outlook, etc.
: The DT&I Process Overview
The DT&I Process (Design, Testing, and Iterative Design) ensures that the calendar system is user-friendly, functional, and refined over time. It is a continuous cycle of improvement based on testing and feedback.
• Design:
o Create wireframes, mockups, and interactive prototypes to visualize the system's layout and functionality.
o Focus on user experience and usability from the start.
• Testing:
o Conduct usability testing, performance testing, A/B testing, and accessibility testing to ensure the system meets user needs and performs well under real-world conditions.
o Collect feedback through surveys, user interviews, and analytics.
• Iterative Design:
o Implement changes based on feedback and testing results.
 Rapidly prototype, test, and refine the system in cycles.
Keep improving the system by incorporating new features and making adjustments

H. Links

1. https://github.com/Atharva992/College_Codes.git
2. https://atharvapardeshi2006.blogspot.com/2024/11/design-and-management-of-departmental.html
3. https://www.programiz.com/online-compiler/0uyVGO632Hga2

I. Reference

- C Programming Tutorials:
-
- GeeksforGeeks - C Programming Basics
- TutorialsPoint - C Language
- Date and Time Handling in C:
-
- C Library <time.h> Reference
- Programiz - C <time.h> Library
- Calendar-Specific Examples:
-
- GeeksforGeeks - Calendar Program in C
- CodewithC - Simple Calendar Program
- Zeller's Congruence:
-
- Wikipedia - Zeller's Congruence
- Books for Reference
- "The C Programming Language" by Brian W. Kernighan and Dennis M. Ritchie:
-
- This classic book explains the fundamentals of C programming in depth.
- "C: How to Program" by Paul Deitel and Harvey Deitel:
-
- Includes practical examples and projects for understanding C programming concepts.
- "Let Us C" by Yashavant Kanetkar:
-