# Front-End UI/UX Mini Project
# Interactive Calendar Application with Event Management

**Submitted By:**


Team Member: Dinesh Babu R S
Roll Number: 2460360
College Email: dinesh.babu@btech.christuniversity.in

Team Member: Aditya V J
Roll Number: 2460311
College Email: aditya.vj@btech.christuniversity.in

Team Member: Jashwanth B S
Roll Number: 2460374
College Email: jashwanth.bs@btech.christuniversity.in


**Course:** UI/UX Design Fundamentals
**Instructor Name:** Mr.Narendra
**Institution:** Christ University
**Date of Submission:** 26/09/2025

# Index

## Abstract

This project presents an interactive calendar application designed and developed using HTML5, CSS3, and JavaScript. The application features a dynamic monthly calendar view with event management capabilities, live clock and date display, and a responsive design optimized for desktop and mobile devices. The primary goal was to create a functional, user-friendly calendar system that allows users to add, view, and manage events with persistent data storage. The application demonstrates modern front-end development practices, including DOM manipulation, local storage implementation, and responsive design principles. The final outcome is a robust, interactive calendar interface suitable for personal event management, showcasing strong UI/UX principles and practical coding skills.

## Objectives

- Design an intuitive calendar interface using modern UI/UX principles
- Develop interactive event management functionality with JavaScript
- Implement responsive design for optimal viewing across all devices
- Create persistent data storage using browser local storage
- Integrate real-time clock and date display features
- Ensure accessibility and usability across different user scenarios

## Scope of the Project

- **Inclusions:**
    i. Interactive monthly calendar view with navigation
    ii. Event creation, editing, and deletion functionality
    iii. Color-coded event categories (Work, Personal, Birthday, Other)
    iv. Real-time clock and date display
    v. Responsive design for desktop, tablet, and mobile
    vi. Local storage for data persistence
    vii. Modern CSS animations and transitions
- **Exclusions:**
    i. No server-side integration or database connectivity
    ii. No user authentication or multi-user support

iii. No recurring event functionality

iv. No email notifications or reminders

## Tools & Technologies Used

| Tool/Technology | Purpose |
|---|---|
| HTML5 | Markup and content structure |
| CSS3 | Styling and layout |
| VS Code | Code editor |
| Chrome DevTools | Testing and debugging |
| JavaScript | Interactive functionality and DOM manipulation |
| jQuery | DOM manipulation and event handling |
| Bootstrap | UI components and responsive grid system |

## HTML Structure Overview

- Semantic HTML5 structure with proper document outline
- Main sections:
    1. Live clock and date display header
    2. Calendar navigation controls (previous/next month)
    3. Monthly calendar grid with day cells
    4. Event management sidebar with form controls
    5. Event list display area
- Bootstrap grid system for responsive layout

- Form elements for event creation with input validation
- Modal dialogs for event editing and confirmation
- Accessibility features including ARIA labels and semantic markup

## CSS Styling Strategy

- Modern gradient background design with blue-to-white transition
- CSS organized with logical grouping and clear commenting structure
- **Techniques Used:**
    i. CSS Grid and Flexbox for layout structure and alignment
    ii. CSS animations including pulse effect for current date highlighting
    iii. Smooth transitions for hover effects and interactions
    iv. Custom color schemes for different event categories
    v. Responsive design using media queries and flexible units
    vi. Modern box-shadow and border-radius for visual depth

## Key Features

| Feature | Description |
|---|---|
| Interactive Calendar | Monthly view with clickable date cells and navigation |
| Event Management | Add, edit, and delete events with form validation |
| Real-time Updates | Live clock and current date display |
| Color Coding | Visual distinction between event categories |
| Data Persistence | Events saved to local storage across sessions |
| Responsive Design | Adapts to all screen sizes and orientations |

| Feature | Description |
|---|---|
| Visual Feedback | Hover effects and current date highlighting |
| Category Filtering | Organized event display by type |

## Challenges Faced & Solutions

| Challenge | Solution |
|---|---|
| Dynamic calendar generation | Implemented JavaScript date calculations and DOM manipulation |
| Event data persistence | Utilized localStorage API for client-side data storage |
| Responsive calendar grid layout | Used CSS Grid with flexible sizing and media queries |
| Real-time clock updates | Implemented setInterval for continuous time display updates |
| Event collision handling | Created stacking system for multiple events on same date |
| Cross-browser compatibility | Used standard JavaScript APIs and tested across browsers |

## Outcome

- Successfully delivered a fully functional interactive calendar application
- Achieved seamless event management with intuitive user interface design
- Implemented responsive design that works across all device types
- Created persistent data storage ensuring user events are retained
- Gained comprehensive understanding of DOM manipulation and event handling
- The application is ready for deployment and real-world usage scenarios

## Future Enhancements

- Add recurring event functionality for repeating appointments
- Implement drag-and-drop interface for event rescheduling
- Add email notification system for upcoming events
- Integrate with external calendar services (Google Calendar, Outlook)
- Implement user authentication for multi-user support
- Add data export/import functionality (CSV, iCal formats)
- Include event search and filtering capabilities

## Sample Code

### a) Calendar Generation Function (script.js)

```
function showCalendar(month, year) {
    const firstDay = new Date(year, month).getDay();
    const daysInMonth = 32 - new Date(year, month, 32).getDate();
    const calendarBody = $("#calendar-body");
    calendarBody.empty();
    $("#monthAndYear").text(monthNames[month] + " " + year);
        let date = 1;
    for (let i = 0; i < 6; i++) {
        let row = $("<tr></tr>");
        // Calendar generation logic continues...
    }
}
```

## b) Event Management CSS (style.css)

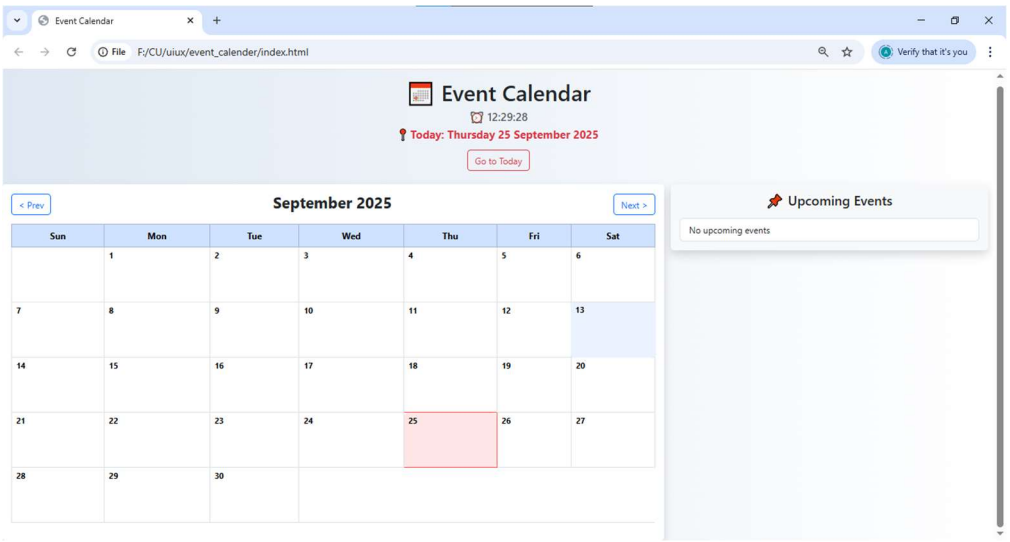
```
.event {
    display: block;
    padding: 2px 5px;
    border-radius: 6px;
    font-size: 12px;
    margin-top: 20px;
    text-align: left;
    color: white;
    font-weight: 500;
}

.event.Work { background: #0d6efd; }
.event.Personal { background: #198754; }
.event.Birthday { background: #ffc107; color: black; }
.event.Other { background: #6f42c1; }
```

## c) Live Clock Update Function (script.js)

```
function updateClock() {
    let now = new Date();
    let timeStr = now.toLocaleTimeString();
    document.getElementById("liveClock").innerText = "⏰ " + timeStr;
}
setInterval(updateClock, 1000);
```

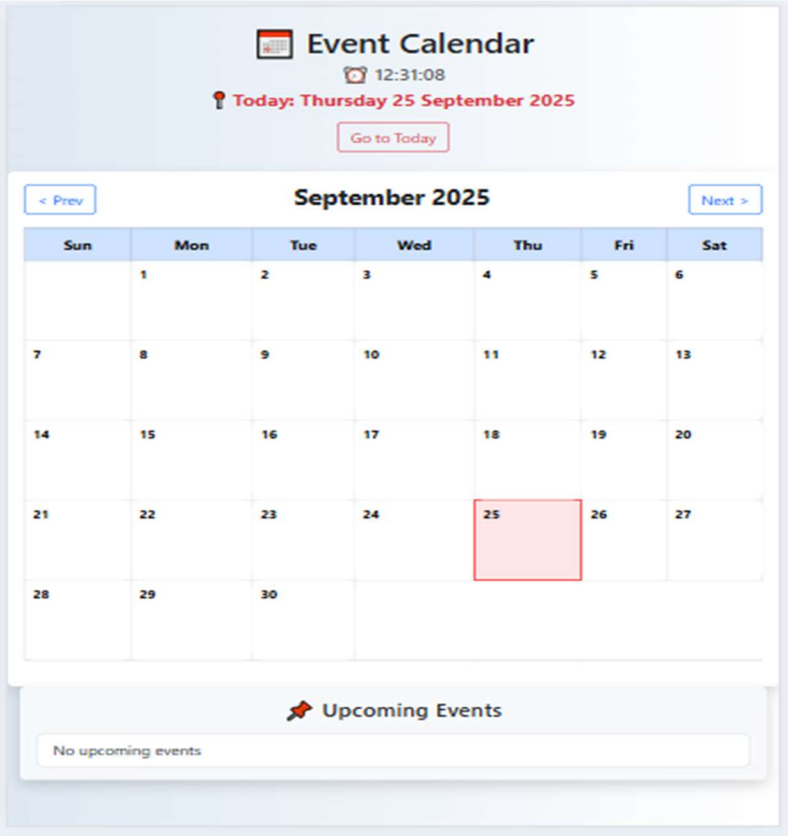
# Screenshots of Final Output

## a) Desktop Calendar View



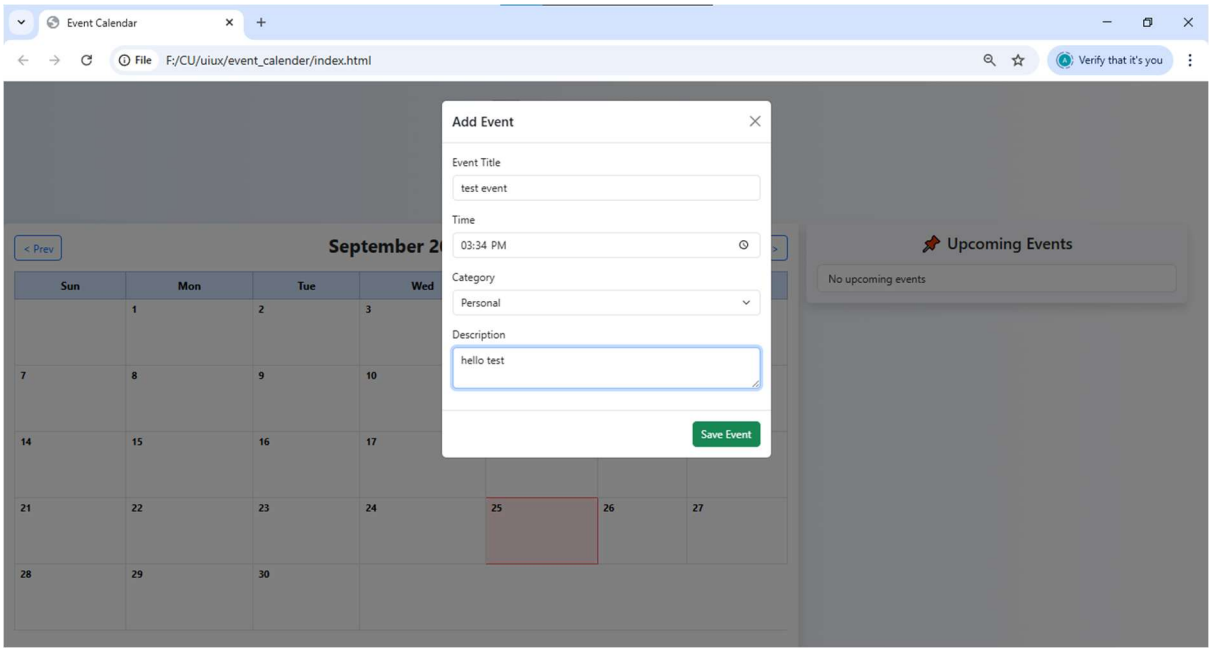

***Caption: Main calendar interface showing monthly view with events and navigation controls.***

## b) Mobile Responsive View



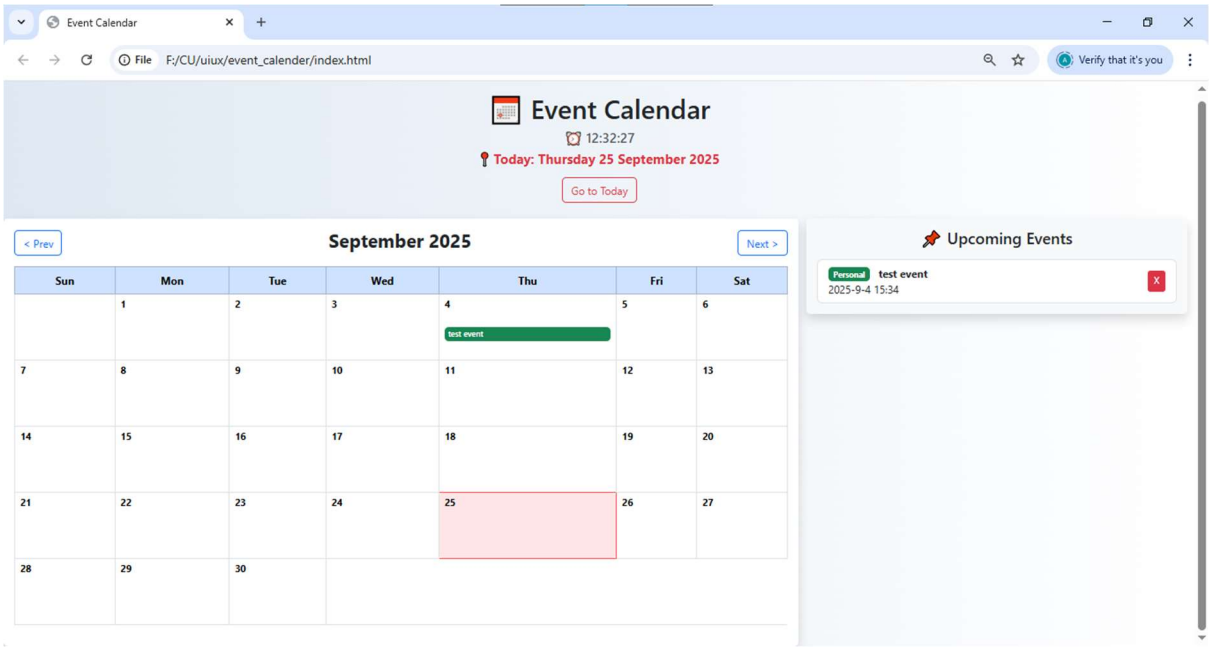

***Caption: Calendar application optimized for mobile devices with touch-friendly interface.***

## c) Event Management Interface




***Caption: Event creation and management sidebar with form controls and event list.***

### d) Current Date Highlighting




***Caption: Visual indication of current date with pulse animation and event display.***

## Conclusion

This interactive calendar application successfully demonstrates the integration of HTML5, CSS3, and JavaScript to create a functional, user-friendly event management system. The project strengthened skills in DOM manipulation, responsive design, and client-side data persistence. The application provides a solid foundation for personal event management with modern UI/UX principles and practical functionality. Future enhancements could expand the application's capabilities to include server integration, advanced scheduling features, and multi-user support for enterprise deployment.

## References


- L&T LMS: https://learn.lntedutech.com/Landing/MyCourse
- Mozilla Developer Network (MDN) Web Documentation
- Bootstrap Documentation for responsive design patterns
- jQuery API Documentation for DOM manipulation
- W3Schools JavaScript and CSS Reference guides