

# **Front-End UI/UX Mini Project**

## **Event Calendar**

- **Submitted By:**
  - *2462060 Darain Brit A*  
[darain.brit@btech.christuniversityuniversity.in](mailto:darain.brit@btech.christuniversityuniversity.in)
  - *2463021 Gutha Nihitha*  
[gutha.nihitha@btech.christuniversity.in](mailto:gutha.nihitha@btech.christuniversity.in)
  - *2462142 San Maria Joby*  
[san.maria@btech.christuniversity.in](mailto:san.maria@btech.christuniversity.in)
- **Course:** *UI/UX Design Fundamentals*
- **Instructor Name:** *Mr. Dhiraj*
- **Institution:** *Christ University*
- **Date of Submission:** *26/09/2025*

## Abstract

- This project is a simple static event calendar website created for the **UI and UX Design** course at Christ University.
- It allows the user to add a new event in the calendar, specify the date and time, provide details of the event, view all the events, and navigate between months.
- Seamless navigation with previous/next month buttons.
- Organize events by type (Personal, Work, Meeting, Reminder, Other)
- There is **no backend**.

## Objectives

- Design a user-friendly interface using modern UI principles
- Develop a fully responsive layout using only HTML and CSS
- Implement structured HTML5 semantic elements
- Apply CSS styling for branding, layout, and responsive behavior
- Use of JavaScript to ensure proper event handling.
- Ensure accessibility and readability across devices using Bootstrap.

## 4. Scope of the Project

- Focused on front-end design only
- Intended for desktop, tablet, and mobile viewports
- Used only open-source tools and pure code (no libraries)

## Tools & Technologies Used

Tool/Technology	Purpose
<b>HTML5</b>	Markup and content structure
<b>CSS3</b>	Styling and layout management
<b>VS Code</b>	Code editor
<b>Chrome DevTools</b>	Testing and debugging
<b>JavaScript</b>	Event Handling and calendar logic
<b>jQuery</b>	DOM Manipulation and Event Handling
<b>Bootstrap</b>	Responsive grid system and components

## HTML Structure Overview

- Used semantic tags: <header>, <nav>, <main>, <section>, <footer>

## CSS Styling Strategy

- Used external CSS file (style.css)
- Organized with comments and sections
- Techniques used:
  - Flexbox and Grid for layout
  - Media Queries for responsiveness
  - CSS Variables for theme customization
  - Hover effects and transitions
  - Mobile-first design approach

## Key Features

Feature	Description
<b>Responsive Design</b>	Adapts seamlessly to all screen sizes
<b>Smooth Navigation</b>	Fixed top nav with anchor links
<b>Project Cards</b>	Flex-based layout with hover effects
<b>Accessible Fonts &amp; Colors</b>	High contrast and readable typography
<b>Local Storage</b>	Events persist between browser sessions
<b>Event List View</b>	See all events in a organized list format

## Challenges Faced & Solutions

Challenge	Solution
<b>Overlapping elements on small screens</b>	Used media queries to stack elements
<b>Difficulty aligning items using float</b>	Shifted to Flexbox and Grid
<b>Typography scaling issue</b>	Used relative units (em/rem) instead of px

## Outcome

- Achieved a clean, consistent, and visually engaging front-end layout
- All key components function as intended using HTML, CSS, JavaScript, Bootstrap and jQuery.
- Learned about layout responsiveness and UI hierarchy in depth
- Mobile-optimized with adjusted grid sizes and touch-friendly interfaces
- Clean, minimal, and easy-to-read layout
- Styled using CSS for better visual appeal
- Collapsible navigation and modal dialogs
- Fully static — works by just opening in a browser



## Future Enhancements

- Integrate animations or transitions
- Theme toggler (light/dark mode)

## Sample Code

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Event Calendar</title>
7   <link href="https://cdnjs.cloudflare.com/ajax/libs/bootstrap/5.3.0/css/bootstrap.min.css" rel="stylesheet">
8   <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css" rel="stylesheet">
9   <style>
10    body {
11      background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
12      min-height: 100vh;
13      font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
14    }
15
16    .calendar-container {
17      background: rgba(255, 255, 255, 0.95);
18      border-radius: 20px;
19      box-shadow: 0 20px 40px rgba(0, 0, 0, 0.1);
20      backdrop-filter: blur(10px);
21      margin: 20px 0;
22    }
23
24    .calendar-header {
25      background: linear-gradient(45deg, #4facfe 0%, #00f2fe 100%);
26      color: white;
27      border-radius: 20px 20px 0 0;
28      padding: 30px;
29      text-align: center;
30    }
31
32    .calendar-nav {
33      display: flex;
34      justify-content: space-between;
35      align-items: center;
36      margin-bottom: 20px;
37    }
38
39    .nav-btn {
40      background: rgba(255, 255, 255, 0.2);
41      border: none;
42      color: white;
43      width: 50px;
44      height: 50px;
45      border-radius: 50%;
46      cursor: pointer;
47      transition: all 0.3s ease;
48      display: flex;
49      align-items: center;
50      justify-content: center;
51    }
52
53    .nav-btn:hover {
54      background: rgba(255, 255, 255, 0.3);
55      transform: scale(1.1);
56    }
57
58    .calendar-grid {
59      display: grid;
60      grid-template-columns: repeat(7, 1fr);
61      gap: 1px;
62      background: #e9ecef;
63      margin: 20px;
64      border-radius: 10px;
65      overflow: hidden;
66    }
67
68    .day-header {
69      background: #495057;
70      color: white;
71      padding: 15px 5px;
72      text-align: center;
```

```

73     font-weight: bold;
74     font-size: 0.9em;
75 }
76
77 .calendar-day {
78     background: #fff9f9;
79     padding: 10px 5px;
80     min-height: 100px;
81     border: none;
82     cursor: pointer;
83     transition: all 0.3s ease;
84     position: relative;
85     display: flex;
86     flex-direction: column;
87 }
88
89 .calendar-day:hover {
90     background: #fff9f9;
91     transform: translateY(-2px);
92     box-shadow: 0 5px 15px rgba(0, 0, 0, 0.1);
93 }
94
95 .calendar-day.other-month {
96     background: #fff9f9;
97     color: #6c757d;
98 }
99
100 .calendar-day.today {
101     background: linear-gradient(45deg, #ff6b6b, #ffa500);
102     color: white;
103     font-weight: bold;
104 }
105
106 .day-number {
107     font-weight: bold;
108     margin-bottom: 5px;
109 }
110
111 .event {
112     background: linear-gradient(45deg, #4facfe, #00f2fe);
113     color: white;
114     padding: 2px 6px;
115     margin: 1px 0;
116     border-radius: 10px;
117     font-size: 0.75em;
118     cursor: pointer;
119     transition: all 0.2s ease;
120     white-space: nowrap;
121     overflow: hidden;
122     text-overflow: ellipsis;
123 }
124
125 .event:hover {
126     transform: scale(1.05);
127     box-shadow: 0 2px 8px rgba(0, 0, 0, 0.2);
128 }
129
130 .modal-content {
131     border: none;
132     border-radius: 20px;
133     overflow: hidden;
134 }
135
136 .modal-header {
137     background: linear-gradient(45deg, #4facfe 0%, #00f2fe 100%);
138     color: white;
139     border: none;
140 }
141
142 .form-control, .form-select {
143     border-radius: 10px;
144     border: 2px solid #e9ecef;
145     transition: all 0.3s ease;
146 }
147
148 .form-control:focus, .form-select:focus {
149     border-color: #4facfe;
150     box-shadow: 0 0 0.2rem rgba(79, 172, 254, 0.25);
151 }
152
153 .btn-primary {
154     background: linear-gradient(45deg, #4facfe, #00f2fe);
155     border: none;
156     border-radius: 10px;
157     padding: 10px 30px;
158     transition: all 0.3s ease;
159 }
160
161 .btn-primary:hover {
162     transform: translateY(-2px);
163     box-shadow: 0 5px 15px rgba(79, 172, 254, 0.4);
164 }
165
166 .btn-danger {
167     background: linear-gradient(45deg, #ff6b6b, #ff5722);
168     border: none;
169     border-radius: 10px;
170 }
171
172 .date-picker-container {

```

```

205     @media (max-width: 768px) {
206         .calendar-grid {
207             margin: 10px;
208         }
209
210         .calendar-day {
211             min-height: 80px;
212             padding: 5px 2px;
213         }
214
215         .day-header {
216             padding: 10px 2px;
217             font-size: 0.8em;
218         }
219
220         .event {
221             font-size: 0.65em;
222             padding: 1px 4px;
223         }
224
225         .nav-btn {
226             width: 40px;
227             height: 40px;
228         }
229     }
230 }
231 </style>
232 </head>
233 <body>
234     <div class="container-fluid">
235         <div class="row justify-content-center">
236             <div class="col-12 col-lg-10">
237                 <div class="calendar-container">
238                     <div class="calendar-header">
239                         <div class="calendar-nav">
240                             <div class="calendar-nav">
241                                 <button class="nav-btn" id="prevMonth">
242                                     <i class="fas fa-chevron-left"></i>
243                                 </button>
244                                 <div>
245                                     <h2 id="currentMonth" class="mb-0"></h2>
246                                     <p class="mb-0" id="currentYear"></p>
247                                 </div>
248                                 <button class="nav-btn" id="nextMonth">
249                                     <i class="fas fa-chevron-right"></i>
250                                 </button>
251                             </div>
252                             <div class="d-flex justify-content-center gap-3 flex-wrap">
253                                 <button class="btn btn-light" data-bs-toggle="modal" data-bs-target="#addEventModal">
254                                     <i class="fas fa-plus"></i> Add Event
255                                 </button>
256                                 <button class="btn btn-light" data-bs-toggle="modal" data-bs-target="#datePickerModal">
257                                     <i class="fas fa-calendar"></i> Go to Date
258                                 </button>
259                                 <button class="btn btn-light" data-bs-toggle="modal" data-bs-target="#eventsListModal">
260                                     <i class="fas fa-list"></i> View All Events
261                                 </button>
262                             </div>
263                         </div>
264                         <div class="calendar-grid" id="calendarGrid">
265                             <!-- Calendar will be generated here -->
266                         </div>
267                     </div>
268                 </div>
269             </div>
270         </div>
271
272         <!-- Add Event Modal -->
273         <div class="modal fade" id="addEventModal" tabindex="-1">
274             <div class="modal-dialog">
275                 <div class="modal-content">
276                     <div class="modal-header">
277                         <h5 class="modal-title">
278                             <i class="fas fa-plus-circle me-2"></i>Add New Event
279                         </h5>
280                         <button type="button" class="btn-close btn-close-white" data-bs-dismiss="modal"></button>
281                     </div>
282                     <div class="modal-body">
283                         <form id="eventForm">
284                             <div class="mb-3">
285                                 <label for="eventTitle" class="form-label">Event Title</label>
286                                 <input type="text" class="form-control" id="eventTitle" required>
287                             </div>
288                             <div class="mb-3">
289                                 <label for="eventDate" class="form-label">Date</label>
290                                 <input type="date" class="form-control" id="eventDate" required>
291                             </div>
292                             <div class="mb-3">
293                                 <label for="eventTime" class="form-label">Time</label>
294                                 <input type="time" class="form-control" id="eventTime">
295                             </div>
296                             <div class="mb-3">
297                                 <label for="eventDescription" class="form-label">Description</label>
298                                 <textarea class="form-control" id="eventDescription" rows="3"></textarea>
299                             </div>
300                             <div class="mb-3">
301                                 <label for="eventCategory" class="form-label">Category</label>
302                                 <select class="form-select" id="eventCategory">
303                                     <option value="personal">Personal</option>
304                                     <option value="work">Work</option>
305                                     <option value="meeting">Meeting</option>
306                                 </select>
307                             </div>
308                         </form>
309                     </div>
310                 </div>
311             </div>
312         </div>
313     </body>
314 </html>

```

```

305         <option value="reminder">Reminder</option>
306         <option value="other">Other</option>
307     </select>
308 </div>
309 <div class="mb-3">
310     <div class="form-check">
311         <input class="form-check-input" type="checkbox" id="recurYearly">
312         <label class="form-check-label" for="recurYearly">
313             <i class="fas fa-redo-alt me-2"></i>
314             <strong>Repeat every year</strong>
315             <small class="text-muted d-block">Perfect for birthdays, anniversaries, and holidays</small>
316         </label>
317     </div>
318 </div>
319 </form>
320 </div>
321 <div class="modal-footer">
322     <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Cancel</button>
323     <button type="button" class="btn btn-primary" id="saveEvent">
324         <i class="fas fa-save me-1"></i>Save Event
325     </button>
326 </div>
327 </div>
328 </div>
329 </div>
330
331 <!-- Date Picker Modal -->
332 <div class="modal fade" id="datePickerModal" tabindex="-1">
333     <div class="modal-dialog modal-sm">
334         <div class="modal-content">
335             <div class="modal-header">
336                 <h5 class="modal-title">
337                     <i class="fas fa-calendar-alt me-2"></i>Go to Date
338                 </h5>
339                 <button type="button" class="btn-close btn-close-white" data-bs-dismiss="modal"></button>
340             </div>
341             <div class="modal-body text-center">
342                 <input type="date" class="date-picker" id="gotoDate">
343             </div>
344             <div class="modal-footer">
345                 <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Cancel</button>
346                 <button type="button" class="btn btn-primary" id="gotoDate">Go</button>
347             </div>
348         </div>
349     </div>
350 </div>
351
352 <!-- Events List Modal -->
353 <div class="modal fade" id="eventsListModal" tabindex="-1">
354     <div class="modal-dialog modal-lg">
355         <div class="modal-content">
356             <div class="modal-header">
357                 <h5 class="modal-title">
358                     <i class="fas fa-list me-2"></i>All Events
359                 </h5>
360                 <button type="button" class="btn-close btn-close-white" data-bs-dismiss="modal"></button>
361             </div>
362             <div class="modal-body">
363                 <div id="eventsList" class="events-list">
364                     <!-- Events will be listed here -->
365                 </div>
366             </div>
367         </div>
368     </div>
369 </div>
370
371 <!-- Event Details Modal -->
372 <div class="modal fade" id="eventDetailsModal" tabindex="-1">
373     <div class="modal-dialog">
374         <div class="modal-content">
375             <div class="modal-header">
376                 <h5 class="modal-title">
377                     <i class="fas fa-info-circle me-2"></i>Event Details
378                 </h5>
379                 <button type="button" class="btn-close btn-close-white" data-bs-dismiss="modal"></button>
380             </div>
381             <div class="modal-body" id="eventDetailsContent">
382                 <!-- Event details will be shown here -->
383             </div>
384             <div class="modal-footer">
385                 <button type="button" class="btn btn-danger" id="deleteEvent">
386                     <i class="fas fa-trash me-1"></i>Delete Event
387                 </button>
388                 <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Close</button>
389             </div>
390         </div>
391     </div>
392 </div>
393
394 <script src="https://cdnjs.cloudflare.com/ajax/libs/bootstrap/5.3.0/js/bootstrap.bundle.min.js"></script>
395 <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.7.0/jquery.min.js"></script>
396
397 <script>
398     class EventCalendar {
399         constructor() {
400             this.currentDate = new Date();
401             this.events = this.loadEvents();
402             this.selectedEventId = null;

```



```

403     this.monthNames = [
404       'January', 'February', 'March', 'April', 'May', 'June',
405       'July', 'August', 'September', 'October', 'November', 'December'
406     ];
407     this.dayNames = ['Sun', 'Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat'];
408
409     this.init();
410   }
411
412   init() {
413     this.renderCalendar();
414     this.bindEvents();
415     this.setDefaultDate();
416   }
417
418   loadEvents() {
419     const saved = localStorage.getItem('calendarEvents');
420     return saved ? JSON.parse(saved) : {};
421   }
422
423   saveEvents() {
424     localStorage.setItem('calendarEvents', JSON.stringify(this.events));
425   }
426
427   bindEvents() {
428     $('#prevMonth').click(() => this.previousMonth());
429     $('#nextMonth').click(() => this.nextMonth());
430     $('#saveEvent').click(() => this.saveEvent());
431     $('#gotoDate').click(() => this.goToSelectedDate());
432
433     // Use event delegation for delete button since it's in a modal
434     $(document).on('click', '#deleteEvent', () => this.deleteEvent());
435
436     // Set today's date as default for new events
437     $('#addEventModal').on('shown.bs.modal', () => {
438       const today = new Date();
439       $('#eventDate').val(today.toISOString().split('T')[0]);
440     });
441
442     // Update events list when modal is shown
443     $('#eventsListModal').on('shown.bs.modal', () => {
444       this.renderEventsList();
445     });
446
447     // Clear selected event when event details modal is hidden
448     $('#eventDetailsModal').on('hidden.bs.modal', () => {
449       this.selectedEventId = null;
450     });
451
452     // Refresh calendar when event list modal is hidden
453     $('#eventsListModal').on('hidden.bs.modal', () => {
454       this.renderCalendar();
455     });
456   }
457
458   setDefaultDate() {
459     const today = new Date();
460     const year = today.getFullYear();
461     const month = String(today.getMonth() + 1).padStart(2, '0');
462     const day = String(today.getDate()).padStart(2, '0');
463     const todayStr = `${year}-${month}-${day}`;
464     $('#gotoDate').val(todayStr);
465   }
466
467   previousMonth() {
468     this.currentDate.setMonth(this.currentDate.getMonth() - 1);
469     this.renderCalendar();
470   }
471
472   nextMonth() {
473     this.currentDate.setMonth(this.currentDate.getMonth() + 1);
474     this.renderCalendar();
475   }
476
477   goToSelectedDate() {
478     const selectedDate = $('#gotoDate').val();
479     if (selectedDate) {
480       this.currentDate = new Date(selectedDate);
481       this.renderCalendar();
482       $('#datePickerModal').modal('hide');
483     }
484   }
485
486   getEventsForDate(dateStr) {
487     const events = [];
488
489     // Parse the date string to avoid timezone issues
490     const [year, month, day] = dateStr.split('-').map(num => parseInt(num));
491     const targetMonth = month; // Already 1-based from our date string
492     const targetDay = day;
493
494     // Add direct events for this date
495     if (this.events[dateStr]) {
496       this.events[dateStr].forEach((event, index) => {
497         events.push({
498           event: event,
499           originalDate: dateStr,
500           originalIndex: index

```

```

500         originalIndex: index
501     });
502 });
503 }
504
505 // Add yearly recurring events
506 Object.keys(this.events).forEach(originalDate => {
507     // Parse original date to avoid timezone issues
508     const [origYear, origMonth, origDay] = originalDate.split('-').map(num => parseInt(num));
509
510     // Check if month and day match (but different year)
511     if (origMonth === targetMonth && origDay === targetDay && originalDate !== dateStr) {
512         this.events[originalDate].forEach((event, index) => {
513             if (event.recurYearly) {
514                 events.push({
515                     event: event,
516                     originalDate: originalDate,
517                     originalIndex: index,
518                     isRecurring: true
519                 });
520             }
521         });
522     }
523 });
524
525 return events;
526 }
527
528 renderCalendar() {
529     const year = this.currentDate.getFullYear();
530     const month = this.currentDate.getMonth();
531
532     $('#currentMonth').text(this.monthNames[month]);
533     $('#currentYear').text(year);
534
535     const firstDay = new Date(year, month, 1);
536     const lastDay = new Date(year, month + 1, 0);
537     const startDate = new Date(firstDay);
538     startDate.setDate(startDate.getDate() - firstDay.getDay());
539
540     let html = '';
541
542     // Day headers
543     this.dayNames.forEach(day => {
544         html += `<div class="day-header">${day}</div>`;
545     });
546
547     // Calendar days
548     const currentDate = new Date(startDate);
549     for (let week = 0; week < 6; week++) {
550         for (let day = 0; day < 7; day++) {
551             // Use local date string to avoid timezone issues
552             const year = currentDate.getFullYear();
553             const month = String(currentDate.getMonth() + 1).padStart(2, '0');
554             const dayNum = String(currentDate.getDate()).padStart(2, '0');
555             const dateStr = `${year}-${month}-${dayNum}`;
556
557             const isCurrentMonth = currentDate.getMonth() === this.currentDate.getMonth();
558             const isToday = this.isToday(currentDate);
559
560             let dayClass = 'calendar-day';
561             if (!isCurrentMonth) dayClass += ' other-month';
562             if (isToday) dayClass += ' today';
563
564             const dayEvents = this.getEventsForDate(dateStr);
565             let eventsHtml = '';
566             dayEvents.forEach((eventData, index) => {
567                 const event = eventData.event;
568                 const eventKey = eventData.originalDate + '_' + eventData.originalIndex;
569                 eventsHtml += `<div class="event" onclick="calendar.showEventDetails('${eventData.originalDate}', ${eventData.originalIndex})" title="${event.`
570             });
571
572             html += `
573             <div class="${dayClass}" onclick="calendar.selectDate('${dateStr}')">
574                 <div class="day-number">${currentDate.getDate()}</div>
575                 ${eventsHtml}
576             </div>
577             `;
578
579             currentDate.setDate(currentDate.getDate() + 1);
580         }
581     }
582
583     $('#calendarGrid').html(html);
584 }
585
586 isToday(date) {
587     const today = new Date();
588     return date.toDateString() === today.toDateString();
589 }
590
591 selectDate(dateStr) {
592     $('#eventDate').val(dateStr);
593     $('#addEventModal').modal('show');
594 }
595

```

```

596 saveEvent() {
597   const title = $('#eventTitle').val().trim();
598   const date = $('#eventDate').val();
599   const time = $('#eventTime').val();
600   const description = $('#eventDescription').val().trim();
601   const category = $('#eventCategory').val();
602   const recurYearly = $('#recurYearly').prop('checked');
603
604   if (!title || !date) {
605     alert('Please fill in the required fields (Title and Date)');
606     return;
607   }
608
609   const event = {
610     id: Date.now() + Math.random(),
611     title,
612     date,
613     time,
614     description,
615     category,
616     recurYearly
617   };
618
619   if (!this.events[date]) {
620     this.events[date] = [];
621   }
622
623   this.events[date].push(event);
624   this.events[date].sort((a, b) => (a.time || '').localeCompare(b.time || ''));
625
626   this.saveEvents();
627   this.renderCalendar();
628   $('#eventForm')[0].reset();
629   $('#addEventModal').modal('hide');
630   const message = recurYearly ? 'Yearly recurring event saved successfully!' : 'Event saved successfully!';
631   this.showToast(message, 'success');
632 }
633
634 showEventDetails(dateStr, eventIndex) {
635   if (!this.events[dateStr] || !this.events[dateStr][eventIndex]) {
636     this.showToast('Event not found!', 'error');
637     return;
638   }
639
640   const event = this.events[dateStr][eventIndex];
641   this.selectedEventId = { date: dateStr, index: eventIndex };
642
643   const formattedDate = new Date(dateStr).toLocaleDateString('en-US', {
644     weekday: 'long',
645     year: 'numeric',
646     month: 'long',
647     day: 'numeric'
648   }));
649
650   const timeStr = event.time ? new Date(`1970-01-01T${event.time}`).toLocaleTimeString('en-US', {
651     hour: '2-digit',
652     minute: '2-digit'
653   }) : 'All day';
654
655   const recurringBadge = event.recurYearly ? '<span class="badge bg-success ms-2"><i class="fas fa-redo-alt me-1"></i>Yearly</span>' : '';
656
657   const html = `
658     <div class="event-details">
659       <h4 class="text-primary">${event.title}${recurringBadge}</h4>
660       <div class="mb-2">
661         <i class="fas fa-calendar text-muted me-2"></i>
662         <strong>Date:</strong> ${formattedDate}
663       </div>
664       <div class="mb-2">
665         <i class="fas fa-clock text-muted me-2"></i>
666         <strong>Time:</strong> ${timeStr}
667       </div>
668       <div class="mb-2">
669         <i class="fas fa-tag text-muted me-2"></i>
670         <strong>Category:</strong> <span class="badge bg-secondary">${event.category}</span>
671       </div>
672       ${event.recurYearly ? `
673         <div class="mb-2">
674           <i class="fas fa-info-circle text-muted me-2"></i>
675           <strong>Recurring:</strong> <span class="text-success">This event repeats every year on ${new Date(dateStr).toLocaleDateString('en-US', { month: 'long', year: 'numeric' })}</span>
676         </div>
677       ` : ''}
678       ${event.description ? `
679         <div class="mb-2">
680           <i class="fas fa-align-left text-muted me-2"></i>
681           <strong>Description:</strong>
682           <div class="mt-1 p-2 bg-light rounded">${event.description}</div>
683         </div>
684       ` : ''}
685     </div>
686   `;
687
688   $('#eventDetailsContent').html(html);
689   $('#eventDetailsModal').modal('show');
690 }
691
692 deleteEvent() {

```

```

692 ~ deleteEvent() {
693 ~   if (!this.selectedEventId) {
694 ~     this.showToast('No event selected for deletion', 'error');
695 ~     return;
696 ~   }
697 ~
698 ~   const { date, index } = this.selectedEventId;
699 ~   if (!this.events[date] || !this.events[date][index]) {
700 ~     this.showToast('Event not found', 'error');
701 ~     $('#eventDetailsModal').modal('hide');
702 ~     return;
703 ~   }
704 ~
705 ~   const event = this.events[date][index];
706 ~
707 ~   let confirmMessage = 'Are you sure you want to delete this event?';
708 ~   if (event.recurYearly) {
709 ~     confirmMessage = 'Are you sure you want to delete this yearly recurring event? This will remove it from all years.';
710 ~   }
711 ~
712 ~   if (confirm(confirmMessage)) {
713 ~     try {
714 ~       this.events[date].splice(index, 1);
715 ~       if (this.events[date].length === 0) {
716 ~         delete this.events[date];
717 ~       }
718 ~       this.saveEvents();
719 ~       $('#eventDetailsModal').modal('hide');
720 ~       setTimeout(() => {
721 ~         this.renderCalendar();
722 ~         this.showToast('Event deleted successfully!', 'success');
723 ~       }, 300);
724 ~       this.selectedEventId = null;
725 ~     } catch (error) {
726 ~       console.error('Error deleting event:', error);
727 ~       this.showToast('Error deleting event!', 'error');
728 ~     }
729 ~   }
730 ~ }
731 ~
732 ~
733 ~ renderEventsList() {
734 ~   let html = '';
735 ~   const allEvents = [];
736 ~   Object.keys(this.events).forEach(date => {
737 ~     this.events[date].forEach((event, index) => {
738 ~       allEvents.push({
739 ~         event: event,
740 ~         date: date,
741 ~         index: index
742 ~       });
743 ~     });
744 ~   });
745 ~   allEvents.sort((a, b) => new Date(a.date) - new Date(b.date));
746 ~
747 ~   if (allEvents.length === 0) {
748 ~     html = '<div class="text-center text-muted p-4"><i class="fas fa-calendar-times fa-3x mb-3"></i><p>No events found</p></div>';
749 ~   } else {
750 ~     allEvents.forEach(eventData => {
751 ~       const event = eventData.event;
752 ~       const date = eventData.date;
753 ~       const index = eventData.index;
754 ~
755 ~       const formattedDate = new Date(date).toLocaleDateString('en-US', {
756 ~         weekday: 'long',
757 ~         year: 'numeric',
758 ~         month: 'long',
759 ~         day: 'numeric'
760 ~       });
761 ~
762 ~       const timeStr = event.time ? new Date(`1970-01-01T${event.time}`).toLocaleTimeString('en-US', {
763 ~         hour: '2-digit',
764 ~         minute: '2-digit'
765 ~       }) : 'All day';
766 ~
767 ~       const recurringBadge = event.recurYearly ? '<span class="badge bg-success ms-2"><i class="fas fa-redo-alt me-1"></i>Yearly</span>' : '';
768 ~
769 ~       html += `
770 ~       <div class="event-item" onclick="calendar.showEventDetailsFromList('${date}', ${index})">
771 ~         <div class="d-flex justify-content-between align-items-start">
772 ~           <div>
773 ~             <h6 class="mb-1 text-primary">${event.title}${recurringBadge}</h6>
774 ~             <small class="text-muted">
775 ~               <i class="fas fa-calendar me-1"></i>${formattedDate}
776 ~               <i class="fas fa-clock ms-2 me-1"></i>${timeStr}
777 ~             </small>
778 ~             <p class="mb-1 mt-2">${event.description}</p>
779 ~             <small class="text-success"><i class="fas fa-info-circle me-1"></i>Repeats every year</small>
780 ~           </div>
781 ~           <span class="badge bg-secondary">${event.category}</span>
782 ~         </div>
783 ~       </div>
784 ~     `;
785 ~   }
786 ~ }
787 ~
788 ~ $('#eventsList').html(html);

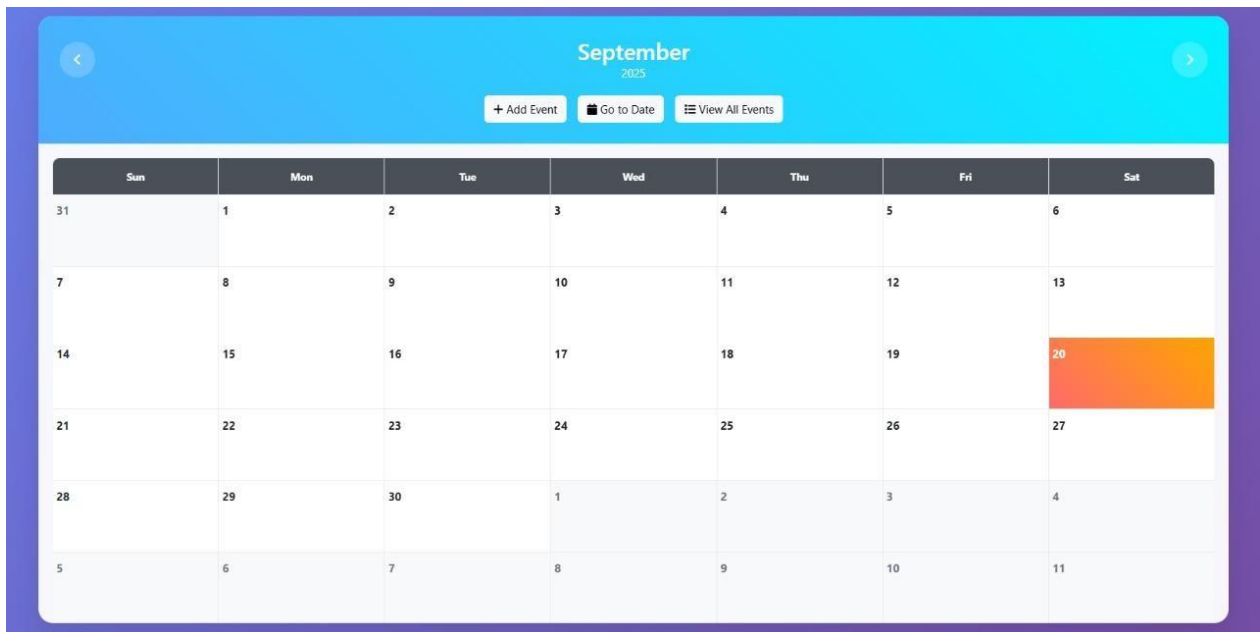
```

```

789   }
790
791   showEventDetailsFromList(dateStr, eventIndex) {
792
793     $('#eventsListModal').modal('hide');
794
795     setTimeout(() => {
796       |   this.showEventDetails(dateStr, eventIndex);
797     }, 300);
798   }
799
800   showToast(message, type = 'info') {
801
802     const toastClass = type === 'success' ? 'bg-success' : type === 'error' ? 'bg-danger' : 'bg-info';
803     const iconClass = type === 'success' ? 'fa-check-circle' : type === 'error' ? 'fa-exclamation-circle' : 'fa-info-circle';
804
805     const toast = `
806     |   <div class="toast-notification position-fixed top-0 end-0 m-3 p-3 ${toastClass} text-white rounded" style="z-index: 9999;"
807     |   |   <i class="fas ${iconClass} me-2"></i>${message}
808     |   </div>
809     `;
810
811     $('body').append(toast);
812
813     setTimeout(() => {
814       |   toast.fadeOut(() => toast.remove());
815     }, 3000);
816   }
817
818   let calendar;
819   $(document).ready(() => {
820     |   calendar = new EventCalendar();
821   });
822 </script>
823 </body>
824 </html>

```

## Screenshots of Final Output



## Conclusion

The calendar is fully functional and ready to use. Users can add events by clicking on any date, navigate between months, and manage their events with an intuitive interface that works seamlessly on both desktop and mobile devices.

## 12. References

- L&T LMS : <https://learn.Intedutech.com/Landing/MyCourse>