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

**Project report**

**Event calender**

• Submitted By:

o Team Members-

Aderu Raghavendra bhanu teja (2462014)

Sharon Cheriyan (2462146)

Sreedev Binoj (2462152)

o College-E-mail id- [aderu.raghavendra@btech.christuniversity.in](mailto:aderu.raghavendra@btech.christuniversity.in), [Sharon.cheriyan@btech.christuniversity.in](mailto:Sharon.cheriyan@btech.christuniversity.in), [sreedev.binoj@btech.christuniversity.in](mailto:sreedev.binoj@btech.christuniversity.in) .

• Course: UI/UX Developer

• Instructor Name: MS.NAGAVEENA

• Institution: Christ University

• Date of Submission: 26/09/2025

• Abstract:

The Event Calendar project is a web-based mini-application developed using **HTML, CSS, JavaScript, jQuery, and Bootstrap**. The project enables users to view a dynamic monthly calendar, navigate across months, and add, update, or remove events on specific dates. With its interactive design and simple interface, it provides a practical solution for managing schedules and important dates without requiring server-side processing.

This project highlights the integration of front-end technologies to create a responsive and user-friendly application. The use of **modal pop-ups for event management, hover effects for better interaction, and weekend highlighting** makes the calendar more intuitive. By balancing simplicity and functionality, this mini-project offers a real-world utility while serving as an excellent learning exercise in web development.

• Objectives :

The primary objective of this mini-project is to design a simple, responsive, and interactive event calendar application using front-end technologies. The project emphasizes improving technical skills while delivering practical functionality.

**Specific Objectives**

1. To design a clear and responsive calendar layout using HTML and Bootstrap.
2. To allow navigation between months dynamically with JavaScript.
3. To enable users to add and manage events through a modal-based form.
4. To store event details temporarily using JavaScript objects.
5. To provide visual cues such as weekend highlights and event dots for better usability.
6. To implement hover and click interactions with CSS and jQuery.
7. To ensure compatibility across different devices using responsive design.
8. To use modal pop-ups for user-friendly event input.
9. To manage events efficiently with add, update, and delete options.
10. To serve as a learning tool for understanding DOM manipulation and front-end integration.

• Scope of the Project :

The scope of this project is focused on developing a **client-side event calendar application** that provides essential scheduling features. It helps users visually organize their plans without needing complex databases or server-side scripting. The calendar can be used in personal websites, small projects, or academic exercises.

Though it is currently designed as a front-end tool, the project can be expanded into a fully functional calendar management system with features such as user accounts, persistent storage (databases), and notifications. This scalability ensures that the project is not only a mini-project but also a foundation for larger real-world applications.

• Tools & Technologies Used :

| **Tool / Technology** | **Purpose** |
| --- | --- |
| **HTML5** | Structures the form elements semantically and ensures accessibility. |
| **CSS3** | Styles the form, error messages, layout, and improves visual appeal. |
| **JavaScript** | Handles validation logic, event handling, and form submission control. |
| **jQuery** | Simplifies DOM manipulation, event handling, and error message display. |
| **Bootstrap 5** | Provides a responsive layout, prebuilt components, and consistent styling. |
| **Web Browser** | Used for testing, debugging, and displaying the form during development. |

• HTML Structure Overview :

The HTML structure of the Event Calendar is designed to be clean, modular, and responsive. At the top, the project begins with a **container** that holds the main calendar layout, including a heading (<h1>) for the project title and navigation controls for switching between months. These navigation buttons (Prev and Next) are placed inside a flexbox (d-flex justify-content-between) to ensure alignment, while the current month and year are dynamically displayed in the center. Below this, a fixed **row of day headers** (Sun to Sat) provides a clear structure for the calendar grid.

The main **calendar grid** is dynamically filled inside the <div id="calendar"> using JavaScript. Each day is represented as a div with the class .calendar-day, which is styled and made interactive. Additionally, a **Bootstrap modal** is integrated into the structure to manage event creation. The modal contains a date display, an input field for event details, and a save button. This modular structure separates the navigation, calendar grid, and event form clearly, making the layout intuitive and easy to maintain.

• CSS Styling Strategy :

The CSS styling strategy for the Event Calendar emphasizes **simplicity, clarity, and user interaction**. The base design uses a soft background color (#f0f4f8) for a light and professional feel, while each day cell (.calendar-day) is styled with a white background, borders, and hover effects for interactivity. Weekends are visually separated from weekdays by applying a light yellow background (.weekend), which improves readability and helps users quickly identify non-working days. Smooth hover transitions (background-color 0.3s ease) make the calendar feel modern and responsive to user interaction.

In addition to styling the calendar grid, **event indicators** are introduced using small blue dots (.event-dot) placed at the bottom-right corner of day cells. These dots provide users with a clear visual cue for days containing scheduled events without cluttering the interface. The modal inherits Bootstrap’s default styling but is customized with spacing and a clean layout for better usability. Overall, the CSS strategy focuses on balancing **aesthetic appeal and functional clarity**, ensuring that the calendar looks engaging while remaining easy to use across devices.

• **JavaScript Overview :**

The JavaScript code is responsible for the **functionality and interactivity** of the Event Calendar. Using JavaScript’s Date object, the program calculates the **number of days in a month, the first day of the week, and dynamically generates the calendar grid**. Navigation buttons (Prev and Next) are linked to functions that update the current month, allowing seamless browsing across different months. The JavaScript also handles rendering weekends with special classes and managing blank spaces for alignment at the beginning of each month.

A significant part of the JavaScript is dedicated to **event management**. Each day cell is clickable and triggers a Bootstrap modal that allows users to add or update event details. Events are temporarily stored in a JavaScript object, where the date acts as a key, ensuring quick retrieval. Event dots are displayed automatically on days with scheduled events, maintaining a clean calendar view. Additionally, jQuery is used to simplify DOM manipulation and event handling. This structure ensures that the application is not only interactive but also efficient, modular, and easy to expand in the future.

• Key Features :

1. Interactive and responsive calendar layout.
2. Month navigation with previous and next buttons.
3. Weekend highlighting for better scheduling awareness.
4. Modal form for adding and managing events.
5. Visual event indicators using dots.
6. Dynamic rendering with JavaScript and jQuery.
7. Smooth user interface using Bootstrap components.
8. Cross-browser and device compatibility.

• Challenges Faced & Solutions :

* **Challenge:** Handling dynamic date generation for each month.
  + **Solution:** Used JavaScript Date() objects to calculate first day and number of days.
* **Challenge:** Retaining events after re-rendering the calendar.
  + **Solution:** Stored events in a JavaScript object and reloaded them dynamically.
* **Challenge:** Making weekends visually distinct.
  + **Solution:** Added conditional CSS classes for weekends.
* **Challenge:** Providing simple yet functional event input.
  + **Solution:** Integrated Bootstrap modal for smooth interaction.

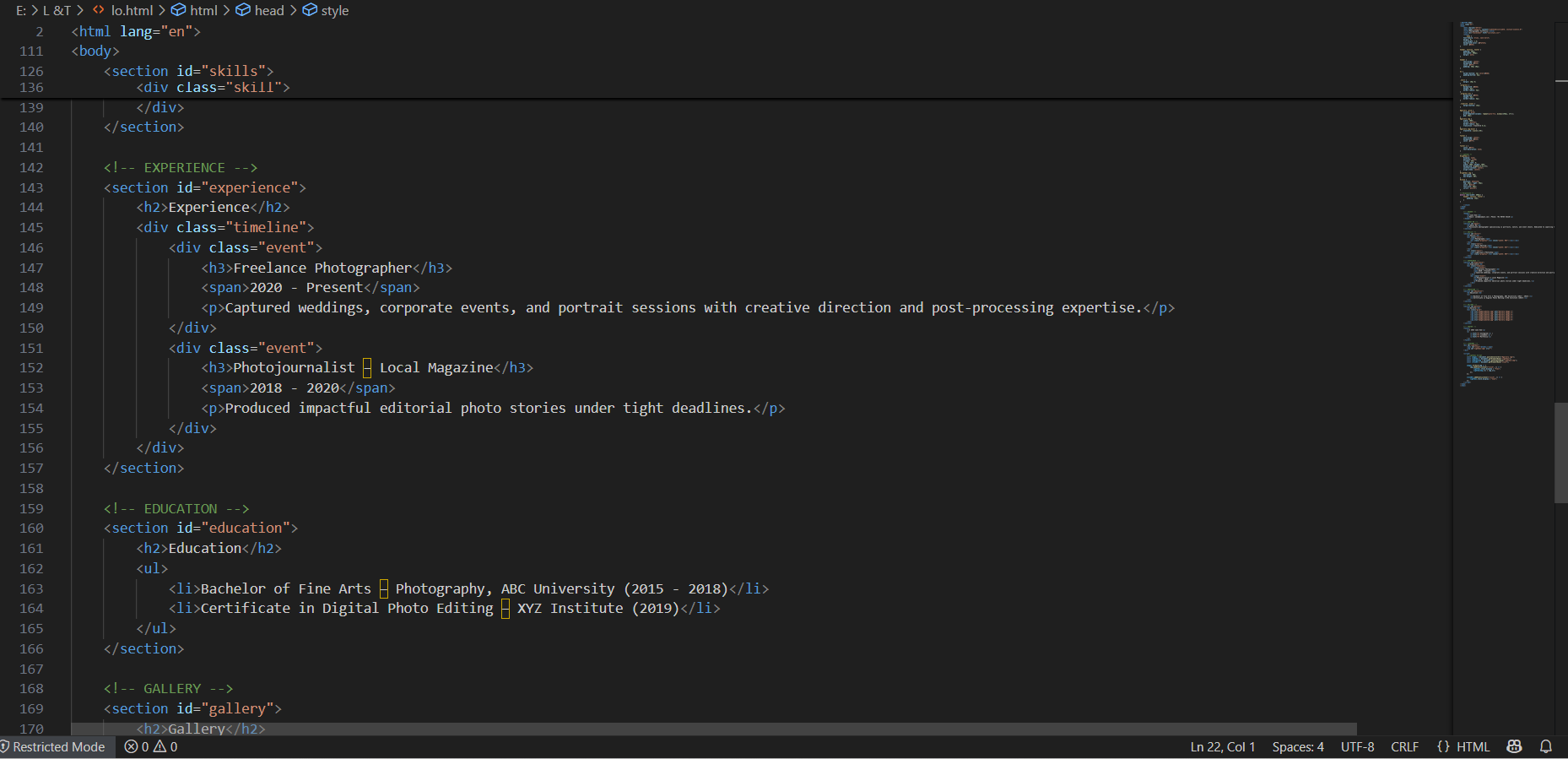
• Outcome :

The Event Calendar project successfully delivers a responsive and interactive tool that allows users to view a monthly calendar, navigate between months, and manage events through a simple modal interface. The inclusion of features such as weekend highlighting, hover effects, and event dots ensures a visually appealing and user-friendly experience. The project not only fulfills its core objective of offering an event management system but also serves as a practical demonstration of integrating multiple front-end technologies like HTML, CSS, JavaScript, jQuery, and Bootstrap. This outcome highlights the project’s ability to enhance productivity and provide real-world utility while showcasing the developers’ technical proficiency.

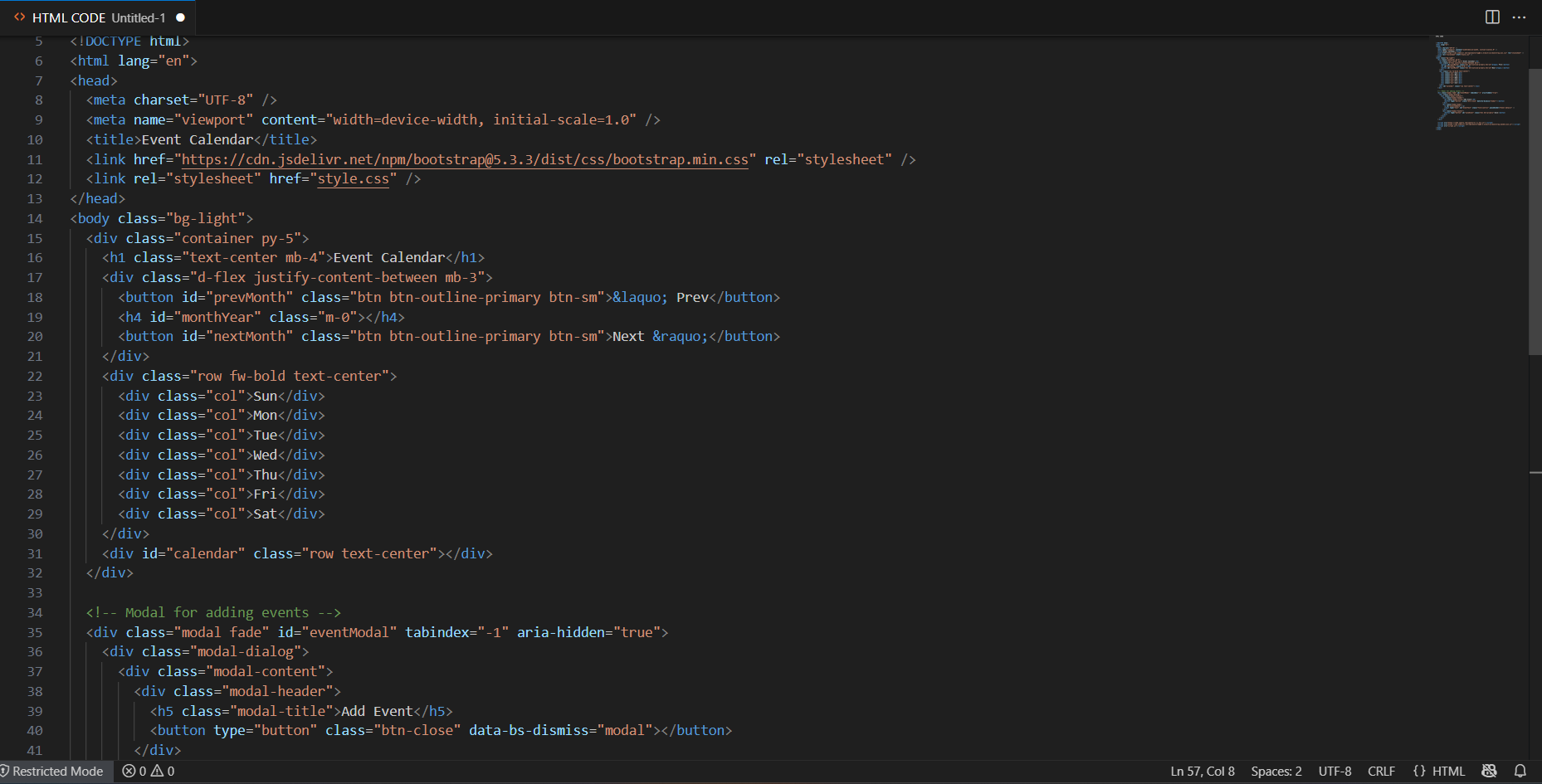
• Future Enhancements :

Beyond its current implementation, the project also opens the door for numerous feature enhancements. For instance, events can be stored persistently using **localStorage or a database**, ensuring data is not lost on refresh. The interface could be improved with **color-coded event categories, notifications, and reminders** to make scheduling more efficient. Integration with external services such as the **Google Calendar API** would enable synchronization with widely used platforms. Additionally, user authentication and multi-user support could transform this mini-project into a fully functional scheduling application. These enhancements would significantly extend its usability, making it scalable from a simple academic project into a robust real-world system.

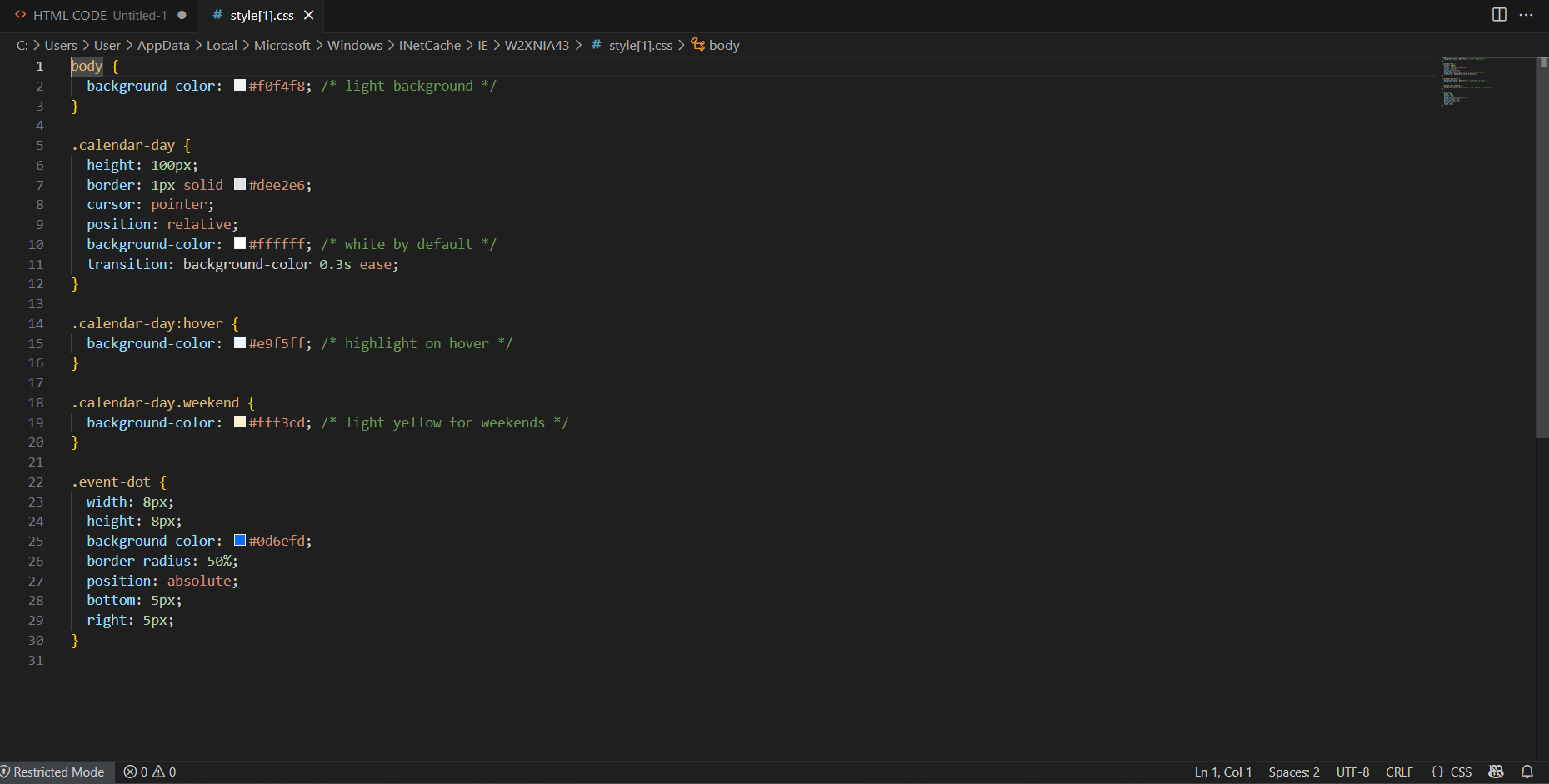
• Sample Code :



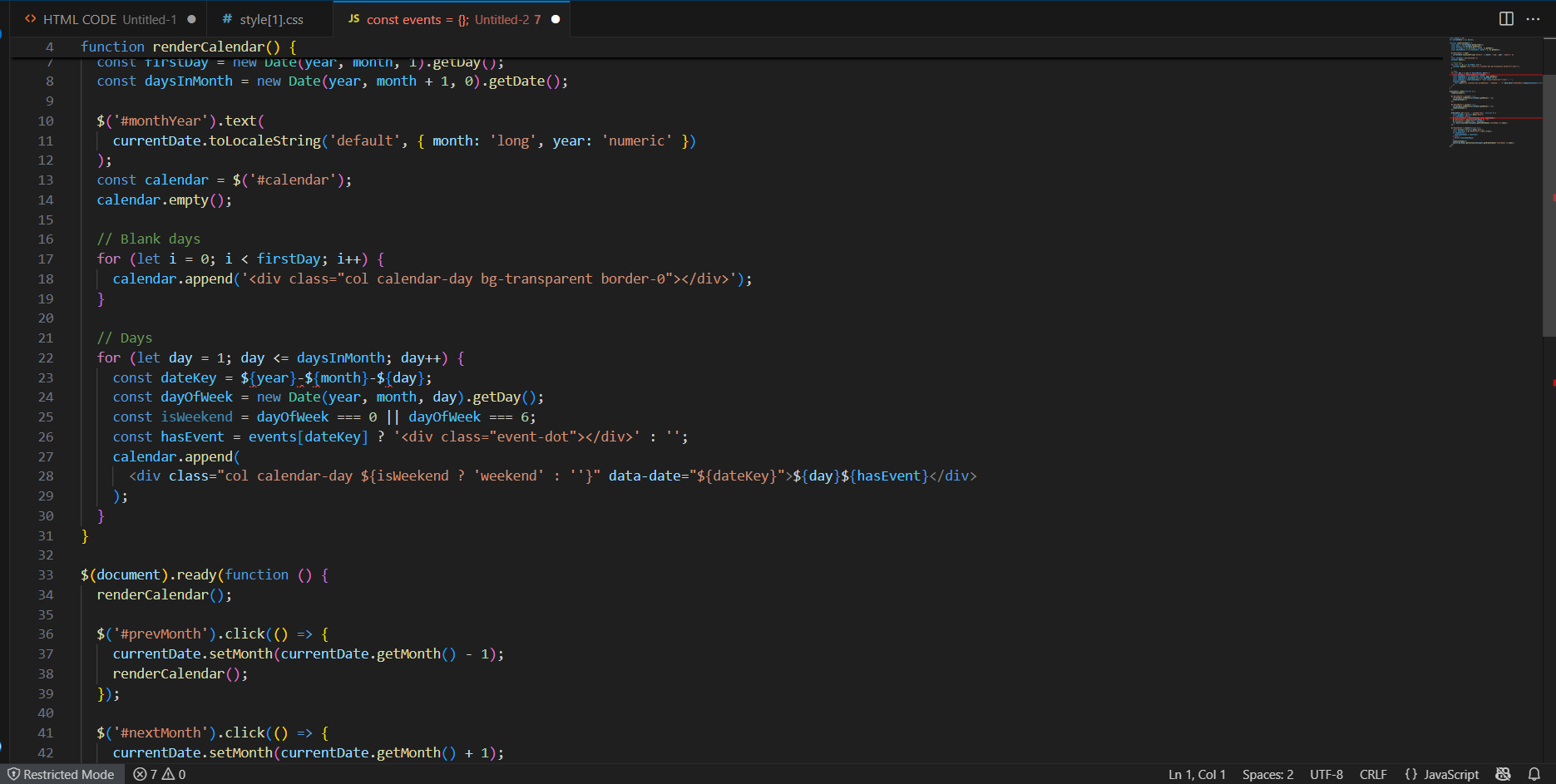
HTML structure



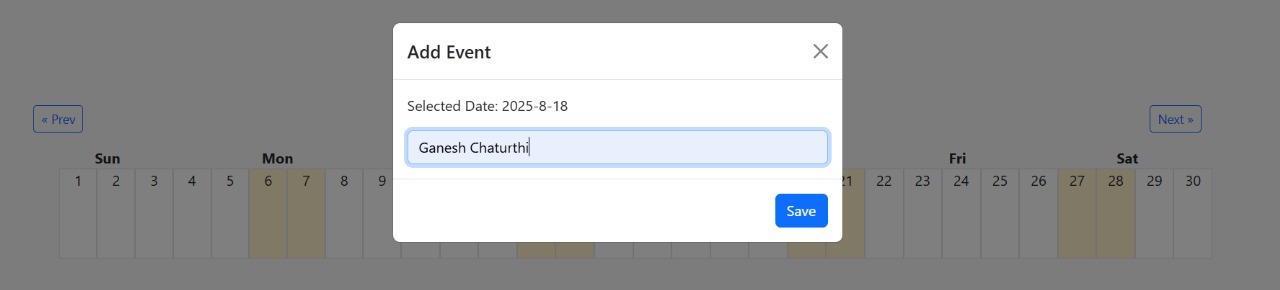
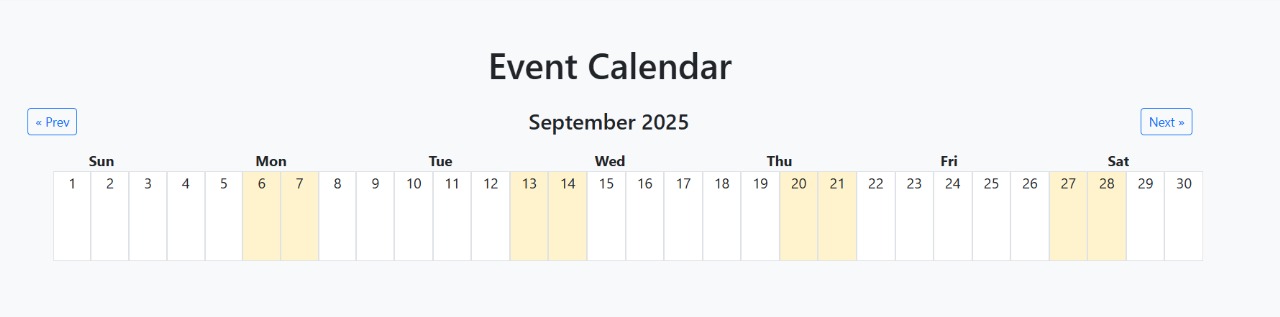
CSS structure

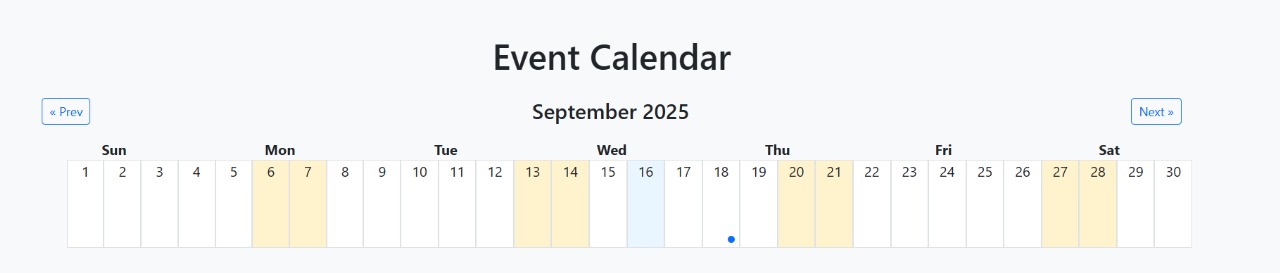


• Java script/Jquery code :



• Screenshot of output :





• Conclusion :

The Event Calendar project demonstrates how **HTML, CSS, JavaScript, jQuery, and Bootstrap** can be combined to create an interactive and practical web application. It offers a clean user interface, intuitive controls, and functional event management features. While the current version operates as a client-side mini-project, it lays the groundwork for advanced improvements such as persistent storage and real-time notifications. Overall, this project showcases both learning outcomes and practical utility in web development.

• References :

L&T LMS : https://learn.lntedutech.com/Landing/MyCourse