Project Title	Book My Calendar
Technologies	MERN
Domain	Software as a service
Project Level	Difficult
Organization	INeuron Intelligence Private Limited

Table of Content

1. Problem Statement:

Design a web application **“Book My Calendar”** to manage multiple calendars at one place.

What is a calendar?
Book My Calendar is an application that allows you to schedule a virtual meeting or someone else to schedule a meeting. **Google Calendar** and **Outlook Calendar** are examples.

You might have used **Microsoft Teams** or **Google Meet** for scheduling meetings, and even attended some of the meetings at these platforms.

Now that you understand what a calendar is, let’s discuss some of the functionality of “Book My Calendar” that you will design.

1. Sign Up and Sign In using form or social accounts.
2. Integration with **Google Meet and Teams**.
3. Allow a user to create an event. Events can be shared with intended users so that users can book a slot in the calendar and join virtual meetings at respective platforms such as Google Meet or Microsoft Teams.
4. Allow account owners to define a workflow for events.

Integration with Google meet and Teams means that **“Book My Calendar”** can access calendar information and add new meetings in the calendar.

Event: Event is a specific time slot that can be used to schedule a meeting with the account owner or someone who is responsible for managing the **“Book My Calendar”** account.

Example: Consider that Harry has created an account at “Book My Calendar”. Now He has created an event called “Discussion with Harry”. Event page “Discussion with Harry” should be accessible using some of the urls such as “https://harry.bookmycalender.com/discussionwithharry”.

Let’s understand configuration for an event.

1. When Harry creates an event he must define the time length for each meeting under a specific event “Discussion with Harry”. Example Harry can decide each meeting scheduled under the event “Discussion with Harry” can be only for 15 minute or 20 minutes etc.
2. Harry can decide a specific start date and end date for an event “Discussion with Harry”. Example: 21st March 2022 to 21st May 2022.
3. Harry can decide how many days in a week users/customers can schedule a meeting for the event “Discussion with Harry”. Example: In a week we have seven day. Allowing a user to access an event at a specific date such as Harry only wants users to schedule meetings on

Monday to Friday, or may be on Alternate day. Design a complete facility to decide on which day event can be scheduled.

4. Harry must define start time and end time for an event “Discussion with Harry” on each day. Example: Event timing is “10:00 AM” to “02:00 PM”
5. Harry must define how many future days calendar should be visible to schedule meeting under event “Discussion with Harry” Example: Consider today is 05th April and Emma wants to schedule a meeting in event “Discussion with Harry” so she can only book calendar between 05th April to 10th April if Harry has defined only 5 day in future meeting can be scheduled.
6. Restrict a specific user to schedule a meeting multiple times within a day or may be on consecutive days.
7. Account owners can decide a meeting can only be scheduled before a certain hour. Example: Account owner has given a restriction such as 5 hour and Current timing is 01:00 PM. User “Thomas” can not schedule a meeting before 06:00 PM.
8. Account owners can decide the gap between two consecutive meetings. Example suppose the gap is 10 minute in that case a meeting is completed at 12:30 then a new meeting can only be scheduled after 12:40.

Let’s understand the workflow for an event.
Workflow is a set of handlers defined for specific events.

1. User/Customer must be notified 5 or 10 minute before the scheduled meeting.
2. How a user should be notified for scheduled meetings. Example email or call or message or whatsapp message.
3. How a user should receive a feedback form to submit after a meeting.

Note: if a meeting is scheduled at 01:00 PM in MS Team then a new meeting at Google meet can not be scheduled at the same time.

2. Project Evaluation metrics:

2.1. Code:

- You are supposed to write code in a modular fashion
- Safe: It can be used without causing harm.
- Testable: It can be tested at the code level.
- Maintainable: It can be maintained, even as your codebase grows.
- Portable: It works the same in every environment (operating system).
- You have to maintain your code on GitHub.
- You have to keep your GitHub repo public so that anyone can check your code.
- Proper readme file you have to maintain for any project development.
- You should include basic workflow and execution of the entire project in the readme file on GitHub.

- Follow the coding standards.

2.2. Database:

MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas.

2.3. Cloud:

You can use any cloud platform for this entire solution hosting like AWS, Azure or GCP

2.4. API Details or User Interface:

You have to expose your complete solution as an API or try to create a user interface for your model testing. Anything will be fine for us.

2.5. Logging:

Logging is a must for every action performed by your code, use the python logging library for this.

2.6. DevOps Pipeline:

Build complete Continuous Integration, Continuous Testing, and Continuous Deployment pipelines for multi stage such as test environments and production environment. Docker containers/ Kubernetes cluster must be used for deployment of applications.

2.7. Deployment:

Implementation of reverse proxy, load balancing, and security group is mandatory for deployed applications.

2.8. Solutions Design:

You have to submit complete solution design strategies in HLD, LLD, and Wireframe documents**.**

2.9. System Architecture:

You have to submit a system architecture design in your wireframe document and architecture document.

2.10. Optimization of solutions:

Try to optimize your solution on code level, architecture level, and mention all of these things in your final submission.

Mention your test cases for your project.

3. Submission requirements:

3.1. High-level Document:

You have to create a high-level document design for your project. You can reference the HLD form below the link.

Sample link: HLD Document Link

3.2. Low-level document:

You have to create a Low-level document design for your project; you can refer to the LLD from the link below.

Sample link: LLD Document Link

3.3. Architecture:

You have to create an Architecture document design for your project; you can refer to the Architecture from the link below.

Sample link:Architecture sample link

3.4. Wireframe:

You have to create a Wireframe document design for your project; refer to the Wireframe from the link below.

Demo link: Wireframe Document Link

3.5. Project code:

You have to submit your code to the GitHub repo in your dashboard when the final submission of your project.

Demo link: Project code sample link

3.6. Detail project report:

You have to create a detailed project report and submit that document as per the given sample.

Demo link: DPR sample link

3.7. Project demo video:

You have to record a project demo video for at least 5 Minutes and submit that link as per the given demo.

Demo link: Project sample link

3.8. The project LinkedIn a post:

You have to post your project details on LinkedIn and submit that post link in your dashboard in your respective field.

Demo link: Linkedin post sample link