Calendar App

Submitted by Aditi Srivastava

1. Project Overview

- A personal calendar app with REST and CLI interfaces.
- Allows users to create and query scheduled events.
- Time zone-aware design for accuracy across locales.
- Includes a simple HTML/Bootstrap frontend.

2. Key Features – What the App Supports

- Add new events with title, start, and end time.
- List events for:
  - Today
  - Remaining part of the day
  - A specified date
  - Find next available time slots of specified duration.
- Works across time zones.
- Functional via REST API, CLI, and browser UI.

3. Why Time zone Support Matters

- Users may be located in different time zones.
- Events may be scheduled for another timezone (e.g., meeting in PST from EST).
- Ensures events show up correctly in the user’s local time.
- Prevents overlaps and scheduling errors when traveling or working globally.

4. Overall Architecture – High Level Design

It is a modular and layered architecture.

- Model: Event class with validation and serialization.
- Service: Business logic (CalendarService).
- Storage: Persistence via file-based storage (can be swapped with database or in-memory implementation).
- API: REST endpoints using Javalin.
- CLI: Console-based interaction using Scanner.
- UI: HTML/JS-based frontend.

5. Code Flow

Adding An Event

- User provides title, start time, end time, and timezone.
- Frontend/CLI sends request to backend.
- EventFactory.fromRequest(...) validates input and creates Event.
- CalendarServiceImpl.addEvent(...) checks for conflicts and saves, if valid.
- Event is persisted to events.json via EventFileStorage.

5. Code Flow

Listing Events

- User selects date and timezone.
- API routes call CalendarServiceImpl.listEventsForDay(...).
- Date is parsed and time zone is applied.
- All events are filtered by matching time window.
- Response is returned as a sorted list of events.

6. Why This Design Works - Extensibility & Maintainability

- Extensible:
  - Storage layer (interface) allows plugging DBs or in-memory stores.
  - Timezone parsing uses Java’s native libraries – easily swappable.
- Maintainable:
  - Each class has a single responsibility.
  - Event validation is centralized in EventFactory.
- Testable:
  - Extensive JUnit 5 and Mockito-based unit tests..

7. Key Files & Folders – Code Organization

- com.calendar.model: Event class.
- com.calendar.api: REST endpoints (EventController).
- com.calendar.service: Business logic (CalendarServiceImpl).
- com.calendar.util: Utilities like DateTimeUtil and JsonUtil.
- com.calendar.storage: Storage interfaces and file-based impl.
- resources/public/calendar.html: Simple Bootstrap-based UI.

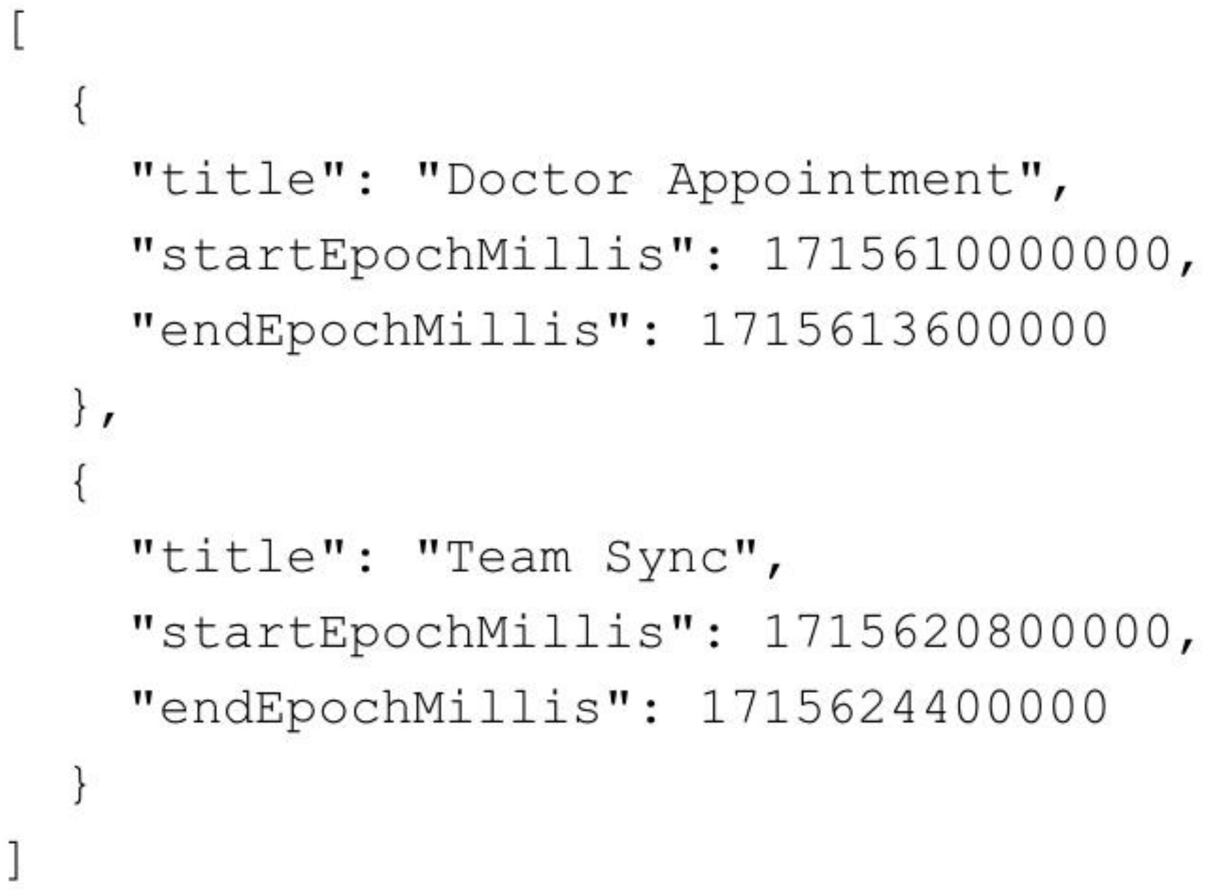
8. How to Run

- Backend (REST):
  - mvn clean install package
  - java -jar target/calendar-app-1.0.0.jar
- CLI Mode:
  - java -jar target/calendar-app-1.0.0.jar CLI
- Frontend:
  - Open **localhost:8081/calendar.html** in your browser. Ensure backend is running on port 8000.

9. How to Test & Verify

- Run all unit tests:
  - mvn test
- Tests cover:
  - JSON utilities
  - Date parsing and timezone handling
  - Event creation/validation
  - Event service logic
  - Seed data can be added in events.json.

10. Sample Seed Data - Example events.json



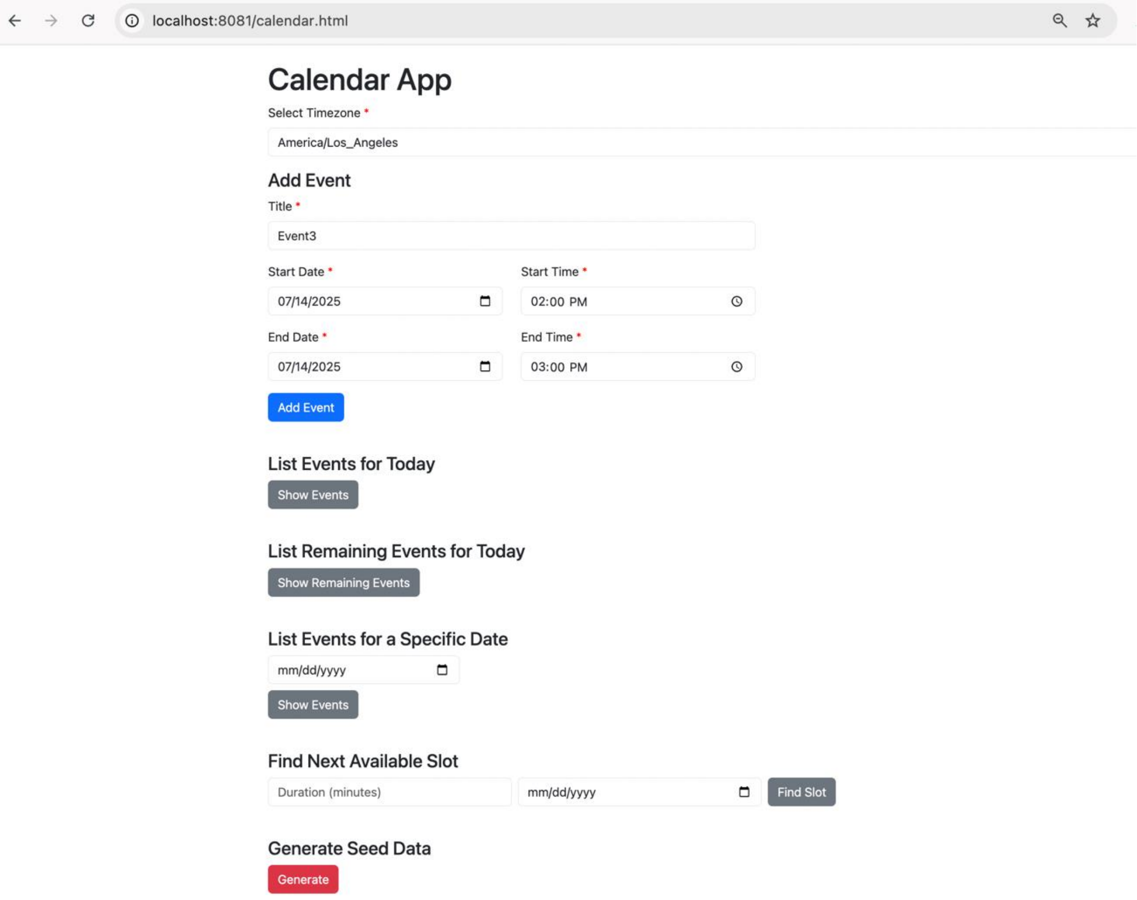
```json
[
  {
    "title": "Doctor Appointment",
    "startEpochMillis": 1715610000000,
    "endEpochMillis": 1715613600000
  },
  {
    "title": "Team Sync",
    "startEpochMillis": 1715620800000,
    "endEpochMillis": 1715624400000
  }
]
```

10. Sample Seed Data - Example events.json (contd.)

- Dynamic Seed Data Generator:
  - Features like “today’s events”, “remaining events”, and “next available slot” rely on the current date and time.
- A static JSON file cannot adapt to real-time changes or demonstrate dynamic behavior.
- The generator provides fresh, relevant seed data every time.
- Trigger via UI or CLI.
  - Button on the frontend labeled “Generate Seed Data”
- Calls a backend endpoint: GET /generate-seed-data
- On success, events are saved to events.json and UI displays a confirmation message.

11(a). Screenshots – UI & CLI Snapshots

calendar.html with form inputs



Calendar App

Select Timezone *: America/Los_Angeles

Add Event

Title *: Event3

Start Date *: 07/14/2025

Start Time *: 02:00 PM

End Date *: 07/14/2025

End Time *: 03:00 PM

List Events for Today

List Remaining Events for Today

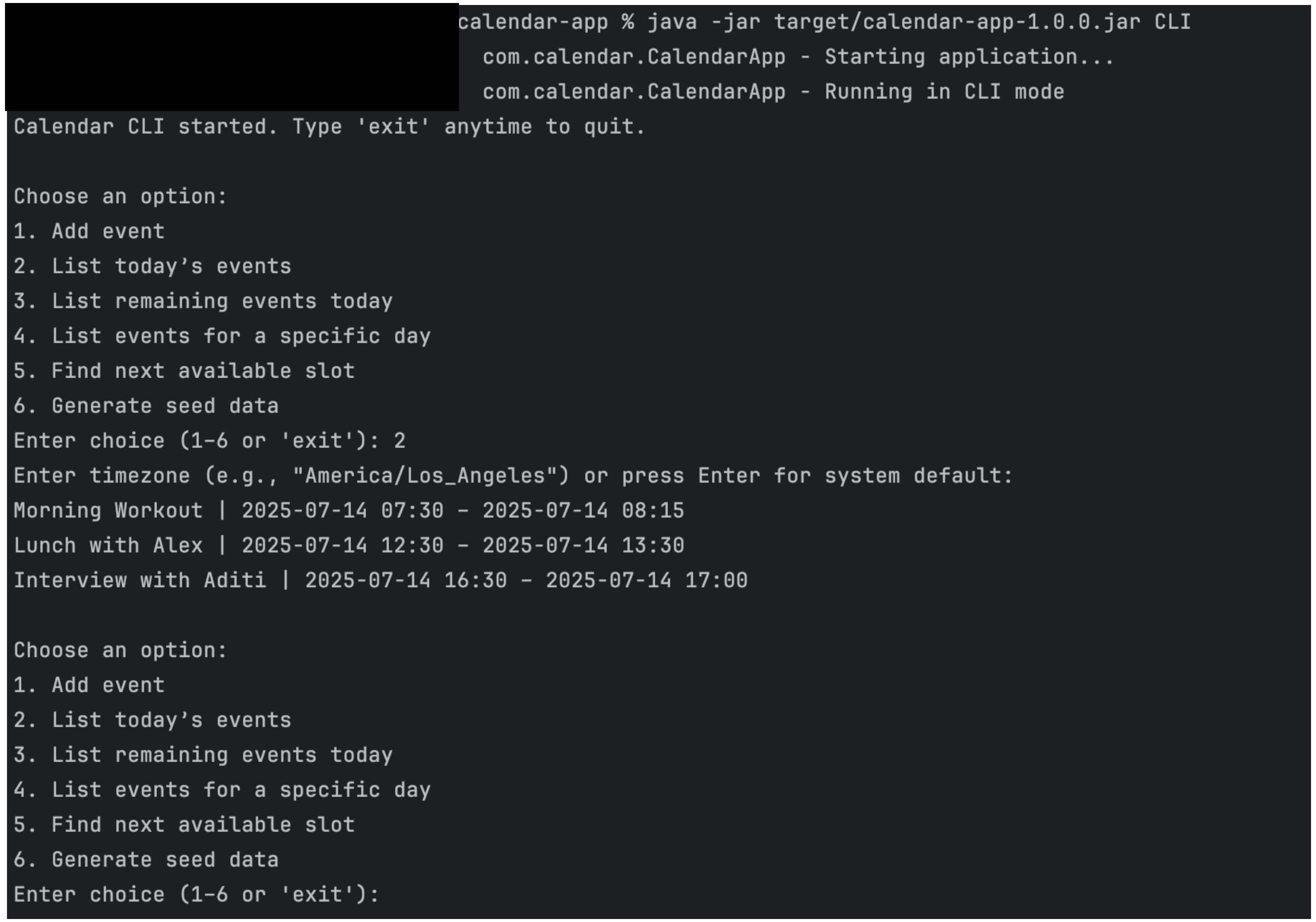
List Events for a Specific Date

Find Next Available Slot

Generate Seed Data

11(b). Screenshots – UI & CLI Snapshots

CLI asking for event details



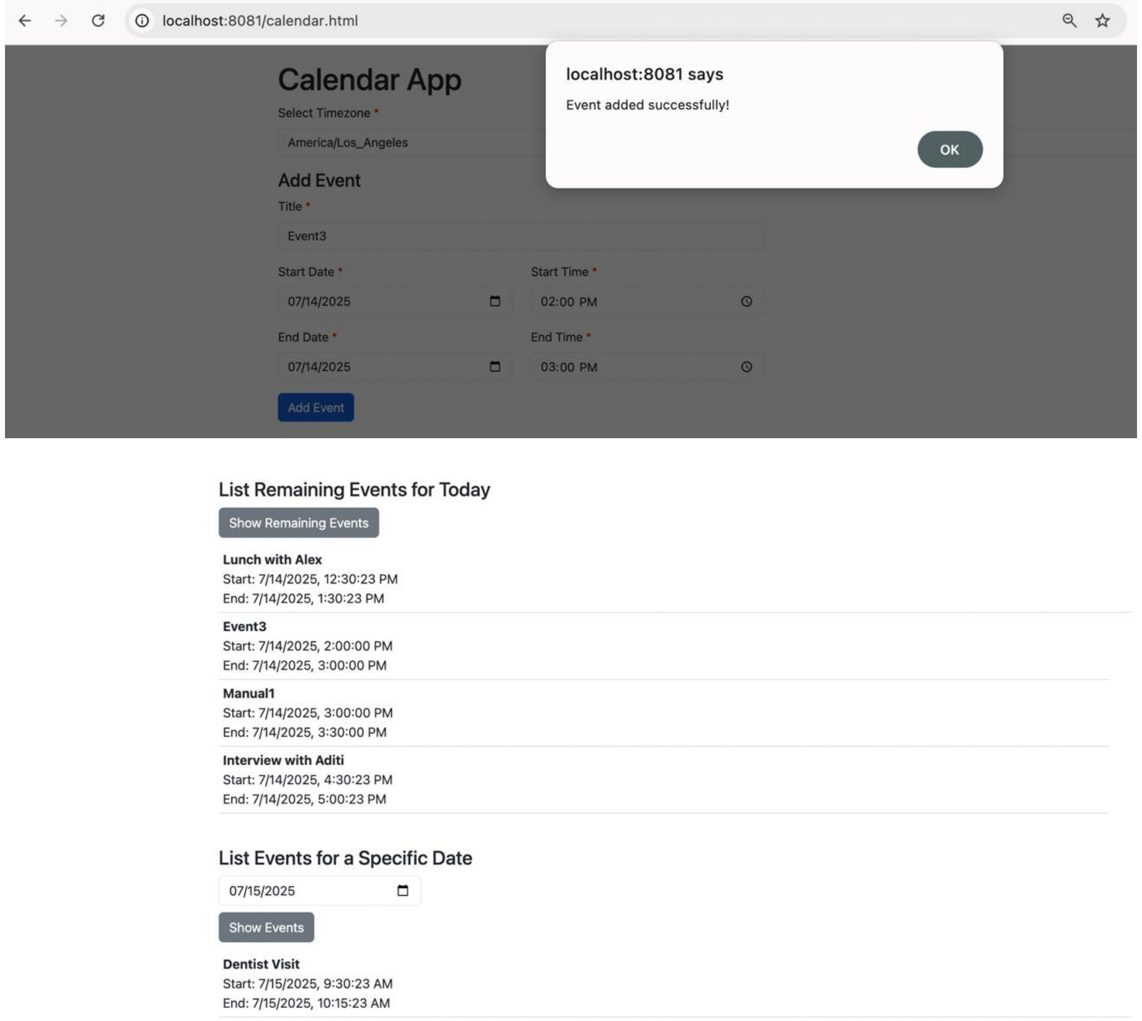
```
[redacted]calendar-app % java -jar target/calendar-app-1.0.0.jar CLI
           com.calendar.CalendarApp - Starting application...
           com.calendar.CalendarApp - Running in CLI mode
Calendar CLI started. Type 'exit' anytime to quit.

Choose an option:
1. Add event
2. List today’s events
3. List remaining events today
4. List events for a specific day
5. Find next available slot
6. Generate seed data
Enter choice (1-6 or 'exit'): 2
Enter timezone (e.g., "America/Los_Angeles") or press Enter for system default:
Morning Workout | 2025-07-14 07:30 - 2025-07-14 08:15
Lunch with Alex | 2025-07-14 12:30 - 2025-07-14 13:30
Interview with Aditi | 2025-07-14 16:30 - 2025-07-14 17:00

Choose an option:
1. Add event
2. List today’s events
3. List remaining events today
4. List events for a specific day
5. Find next available slot
6. Generate seed data
Enter choice (1-6 or 'exit'):
```

11(c). Screenshots – UI & CLI Snapshots

REST API response in browser



localhost:8081 says

Event added successfully!

Calendar App

Select Timezone *

America/Los_Angeles

Add Event

Title *

Event3

Start Date *
07/14/2025

Start Time *
02:00 PM

End Date *
07/14/2025

End Time *
03:00 PM

List Remaining Events for Today

Lunch with Alex
Start: 7/14/2025, 12:30:23 PM
End: 7/14/2025, 1:30:23 PM

Event3
Start: 7/14/2025, 2:00:00 PM
End: 7/14/2025, 3:00:00 PM

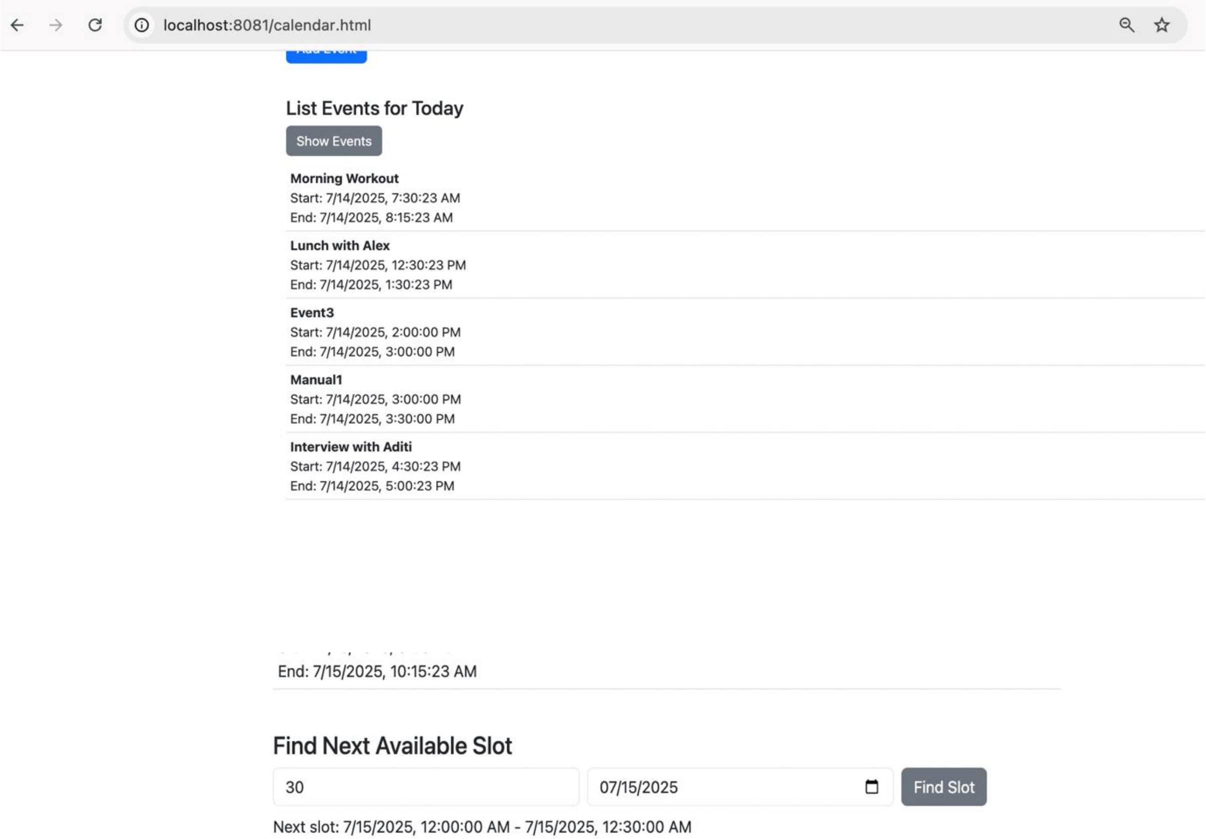
Manual1
Start: 7/14/2025, 3:00:00 PM
End: 7/14/2025, 3:30:00 PM

Interview with Aditi
Start: 7/14/2025, 4:30:23 PM
End: 7/14/2025, 5:00:23 PM

List Events for a Specific Date

07/15/2025

Dentist Visit
Start: 7/15/2025, 9:30:23 AM
End: 7/15/2025, 10:15:23 AM



List Events for Today

Morning Workout
Start: 7/14/2025, 7:30:23 AM
End: 7/14/2025, 8:15:23 AM

Lunch with Alex
Start: 7/14/2025, 12:30:23 PM
End: 7/14/2025, 1:30:23 PM

Event3
Start: 7/14/2025, 2:00:00 PM
End: 7/14/2025, 3:00:00 PM

Manual1
Start: 7/14/2025, 3:00:00 PM
End: 7/14/2025, 3:30:00 PM

Interview with Aditi
Start: 7/14/2025, 4:30:23 PM
End: 7/14/2025, 5:00:23 PM

End: 7/15/2025, 10:15:23 AM

Find Next Available Slot

30

07/15/2025

Next slot: 7/15/2025, 12:00:00 AM - 7/15/2025, 12:30:00 AM

12. Closing Thoughts – Final Notes

- Timezone support ensures event accuracy.
- Design enables easy future enhancements (recurring events, reminders).
- Fully testable and portable.
- Lightweight tech stack: Java 8, Javalin, Bootstrap, no frameworks.