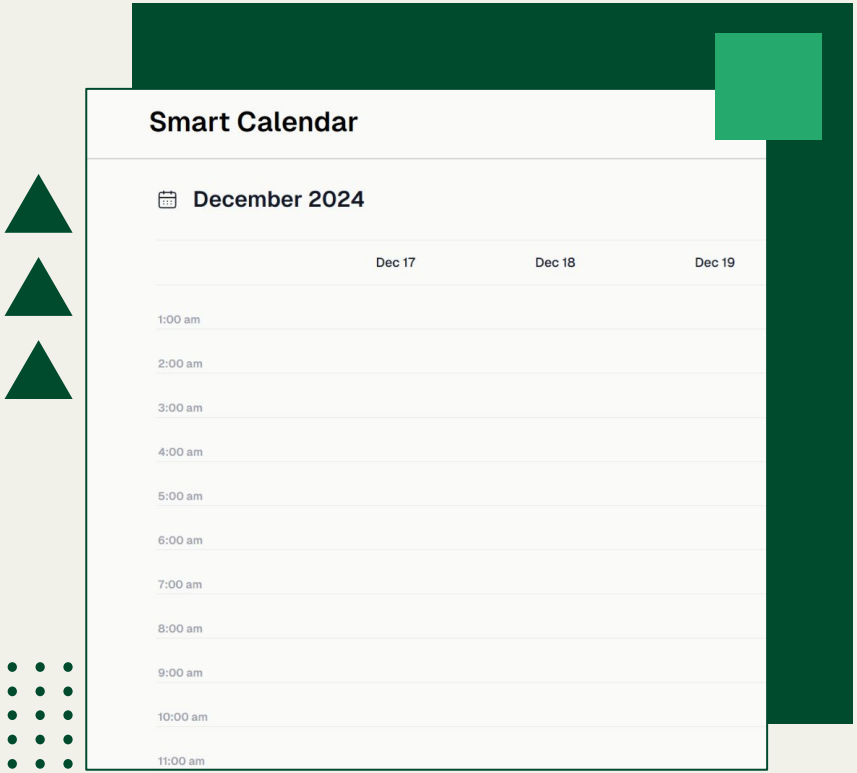
Smart Calendar

Daniel Lumbu, Raymond Kim w. Will Beddow

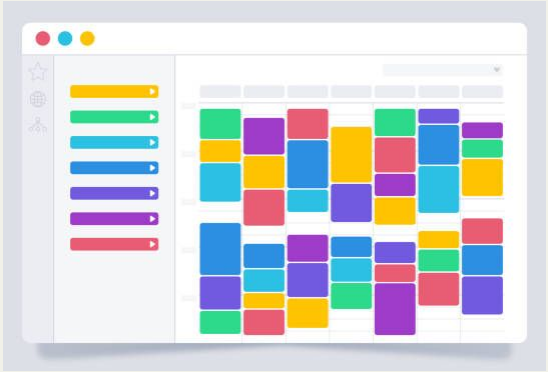


Contents

Problem

IMAGINE...

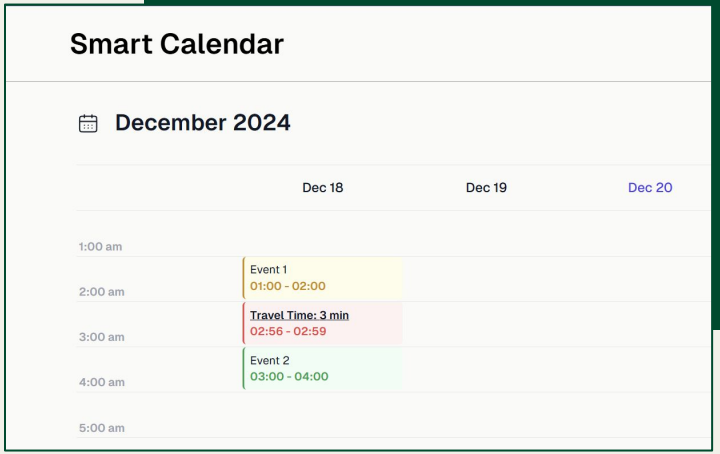
- Your calendar looked like this:



- How do I get to each event?
- How long will it take?
- When should I leave?

What if your calendar was smart?

Solution



LESS MENIAL WORK

The calendar automatically computes all travel routes and times for you.

TIME EFFICIENT

Spend just a few seconds to compute all the travel information you need.

MORE ACCURATE RESULTS

No more misjudging travel times and arrival times. Never be late to another event!

LIVE DEMO:

Smart Calendar (Vercel)

Development Timeline

Our Website Uses:

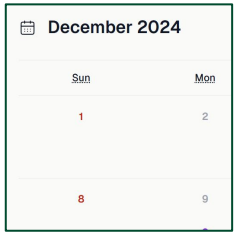
React

JavaScript Library

NextJS

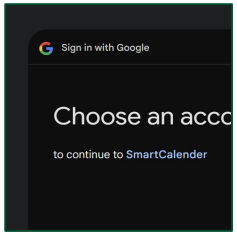
React Framework

Website Components



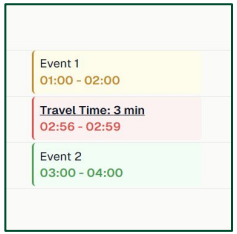
FRONT-END UI

+



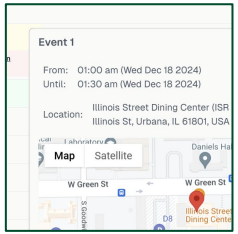
GOOGLE AUTHORIZATION

+



CALENDAR EVENT FETCHING

+



ROUTE CALCULATION/DISPLAY

Front-End: Key Features

CALENDAR UI

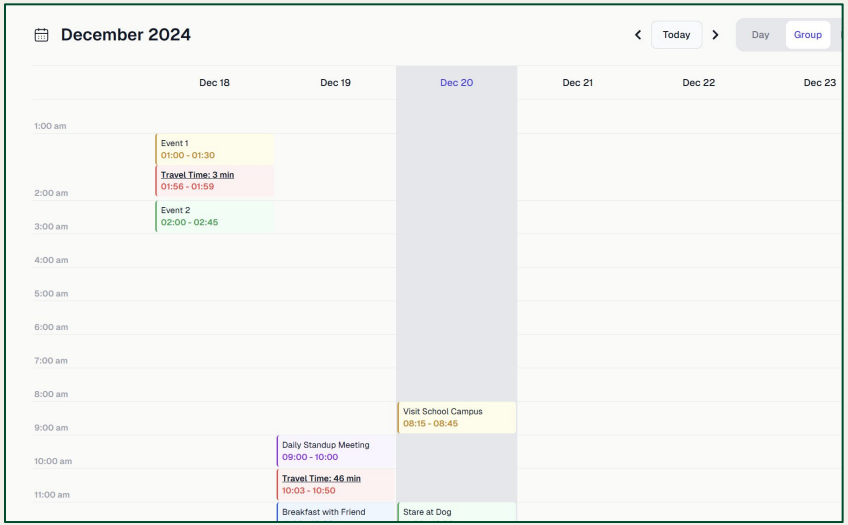
- Implemented partially using React Calendar module
- Day/week/month views, color-coded events, flexible layout
- Utilizes React States & Tailwind CSS for intuitive navigation/responsiveness.

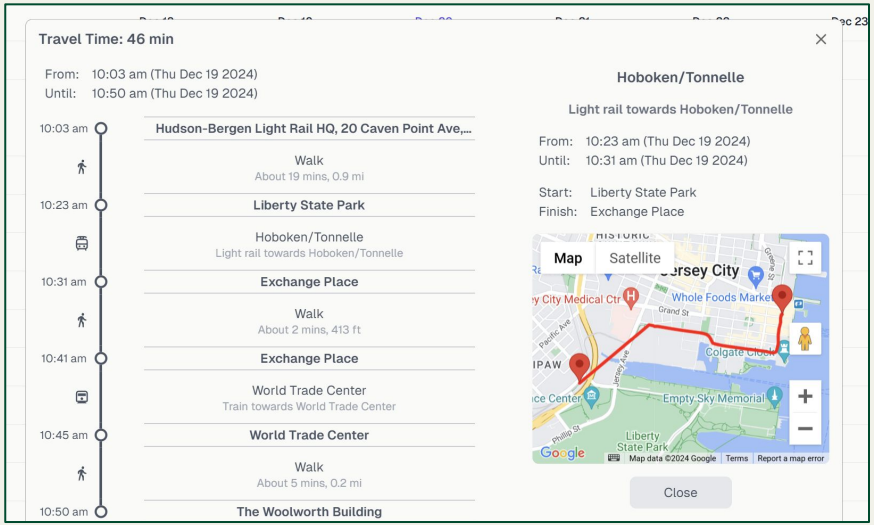
MODULAR DATA INPUT

- Events/pop-ups are populated by list of events/travels, which is filled by back-end API calls
- Utilizes NextJS Server & Client Components to handle API calls & component rendering.

POP-UP DISPLAYS

- Dynamic, digestible view of event/travel details
- Can be expanded/collapsed to show additional info
- Displays map routes utilizing Google Map API & polyline encoder/decoders.





Google: Data Flow & Key Features

1. USER LOGIN

- User logs in via Google sign-in, handled by Google OAuth2.0
- OAuth2.0 returns access token & refresh token
- Tokens are stored securely database
- Ensures secure/standardized user authentication

2. TOKEN VALIDATION

- Check for cookies containing session token to identify user
- Validates token by querying database entries
- access token with refresh token & Google API
- Establishes a connection to user's primary calendar using Google Calendar API

3. FETCHING EVENTS

- Connect to Google Calendar API via valid access token
- Retrieves range of events data from user's primary calendar
- Raw API data is processed into appropriate structure for front-end use
- Updates database during token refresh

Notification Setup Code

Saving Preferences

- User toggles notification preference, and saves settings
- Success feedback ("Settings saved!") is displayed
- Updates to settings are handled by API fetch calls

Global State Management

- Eliminates redundancy and ensures that all components see same state
- Converts raw event data into appropriate format
- Any changes are automatically reflected across all components using the context

Email Notifications:

- Fetch user and event data
- Convert route data to objects and send notifications via Nodemailer
- Sends detailed event/route notifications to user's email
- Configured with secure SMTP server to ensure reliable delivery

Route Finding & Displaying

GOOGLE DIRECTIONS API

- Input origin & destination from adjacent event data
- Specify arrival time to be start of latter event
- Use public transit routes if available, else choose walking or driving

GOOGLE GEOCODING API

- Geocoding: (Location Name) → (Coordinates)
- Used for displaying event locations on Google Maps
- Reverse Geocoding: (Coordinates) → (Location Name)
- Used for displaying route step locations

GOOGLE MAPS API

- Can set center/markers, and draw polylines with coordinates
- Display event location as single marker
- Display route as two markers, draw route by decoding polyline from Directions API

SERVER COMPONENT

Pros

- Can directly call “fetch” without API layer
- Runs JS in server, returns static HTML → Better perf.

Cons

- Cannot interact with client DOM → No React hooks
- Cannot access React contexts

CLIENT COMPONENT

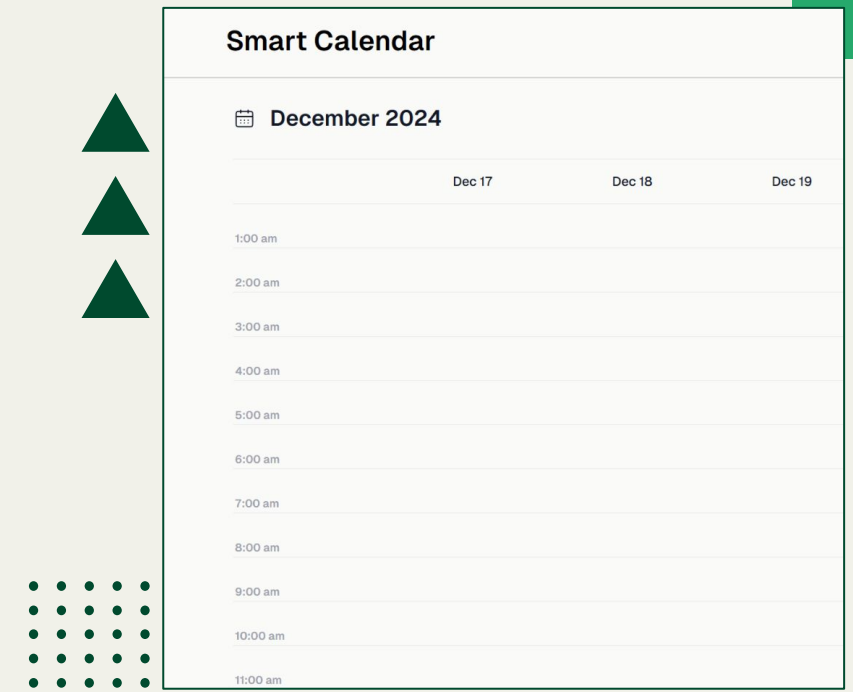
Pros

- Can directly interact with DOM → Use hooks (state, effects, event listeners)
- Can access React contexts

Cons

- Requires API layer (ex. useEffect)
- HTML has to be hydrated with JS before interaction

Thank You!


Smart Calendar
December 2024
Dec 17
Dec 18
Dec 19
1:00 am
2:00 am
3:00 am
4:00 am
5:00 am
6:00 am
7:00 am
8:00 am
9:00 am
10:00 am
11:00 am