Google Calendar Integration Setup Guide

🔐 OAuth Setup Instructions

Step 1: Create a Google Cloud Project

1. Go to Google Cloud Console
2. Click "Create Project" or select an existing project
3. Give it a name like "FocusFlow-Scheduler"

Step 2: Enable Google Calendar API

1. In your project, go to "APIs & Services" → "Library"
2. Search for "Google Calendar API"
3. Click on it and press "Enable"

Step 3: Create OAuth 2.0 Credentials

1. Go to "APIs & Services" → "Credentials"
2. Click "Create Credentials" → "OAuth client ID"
3. If prompted, configure the OAuth consent screen first:
   - Choose "External" user type
   - Fill in app name: "FocusFlow"
   - Add your email as support email
   - Add authorized domains (for testing: localhost)
   - Add scopes: `../auth/calendar.readonly`
   - Add test users (your email)
4. For OAuth client ID:
   - Application type: "Web application"
   - Name: "FocusFlow Web Client"
   - Authorized JavaScript origins:
     - `http://localhost` (for local testing)
     - `http://localhost:8080`
     - Your production domain later
   - Authorized redirect URIs:
     - `http://localhost`
     - `http://localhost:8080`

Step 4: Get Your Credentials

1. After creating, you'll get:
   - **Client ID**: Something like `123456789-abcdef.apps.googleusercontent.com`
   - **Client Secret**: Keep this secure (not needed for frontend-only)
2. Also create an API Key:
   - Go to "Create Credentials" → "API Key"
   - Restrict it to Google Calendar API
   - Copy the API key

Step 5: Update the Application

In the `index.html` file, replace these lines:

javascript

```javascript
// Around line 340-341
const GOOGLE_CLIENT_ID = 'YOUR_CLIENT_ID.apps.googleusercontent.com';
const GOOGLE_API_KEY = 'YOUR_API_KEY';
```

With your actual credentials:

javascript

```javascript
const GOOGLE_CLIENT_ID = '123456789-abcdef.apps.googleusercontent.com';
const GOOGLE_API_KEY = 'AIzaSyB_your_actual_api_key_here';
```

Step 6: Test the Integration

1. Open `index.html` in a web browser
2. You should see a "Connect Google Calendar" button
3. Click it to authorize access
4. Your upcoming calendar events will appear
5. Events with "deadline" keywords will be highlighted for import

🚀 How It Works

Features Implemented:

1. **OAuth Authentication**: Secure Google sign-in
2. **Event Fetching**: Retrieves next 10 events from primary calendar
3. **Deadline Detection**: Automatically identifies events with deadline keywords:
    - "deadline", "due", "submit", "delivery", "final"
4. **One-Click Import**: Import Google Calendar deadlines as tasks
5. **Auto-Fill**: Clicking import pre-fills the task form

Security Notes:

- Client-side only implementation (no backend needed)
- Uses Google's official OAuth 2.0 flow
- Only requests read-only calendar access
- No data is stored except in localStorage

🎯 Smart Deadline Detection

The app intelligently scans your Google Calendar for:

- Event titles containing deadline keywords

- Event descriptions with deadline-related terms
- All-day events marked as deadlines
- Recurring deadlines (shows next occurrence)

🔧 Troubleshooting

Common Issues:

1. **"Invalid Origin" Error**
   - Make sure your domain is in authorized JavaScript origins
   - For local testing, use `http://localhost:8080` and run a local server
2. **"Access Blocked" Error**
   - Ensure you've added yourself as a test user
   - Check that OAuth consent screen is configured
3. **No Events Showing**
   - Verify Google Calendar API is enabled
   - Check that you have upcoming events in your primary calendar
   - Ensure proper scopes are authorized

Running a Local Server:

For security reasons, OAuth requires a proper origin. Use Python's simple server:

bash

```bash
# Python 3
python -m http.server 8080

# Then open: http://localhost:8080/index.html
```

Or with Node.js:

bash

```bash
npx http-server -p 8080
```

📝 For the Hackathon Demo

1. **Pre-setup**: Have the Google Cloud project ready before the demo
2. **Test Account**: Use a Google account with sample calendar events
3. **Demo Events**: Create calendar events with obvious deadlines:
   - "Project Alpha - Final Deadline"
   - "Submit Hackathon Project - Due"
   - "Q4 Report Delivery"

4. **Show the Flow**:
    - Click "Connect Google Calendar"
    - Authorize (one-time)
    - Show events appearing
    - Click "Import as Task" on a deadline
    - Show how it pre-fills the form
    - Add the task and show it scheduled with AI

This integration makes FocusFlow stand out by:

- Reducing manual entry
- Catching deadlines you might forget
- Seamlessly integrating with existing workflows
- Using real OAuth (not mock data)

Note: Keep your API keys secure. For production, implement additional security measures like domain restrictions and API key rotation.