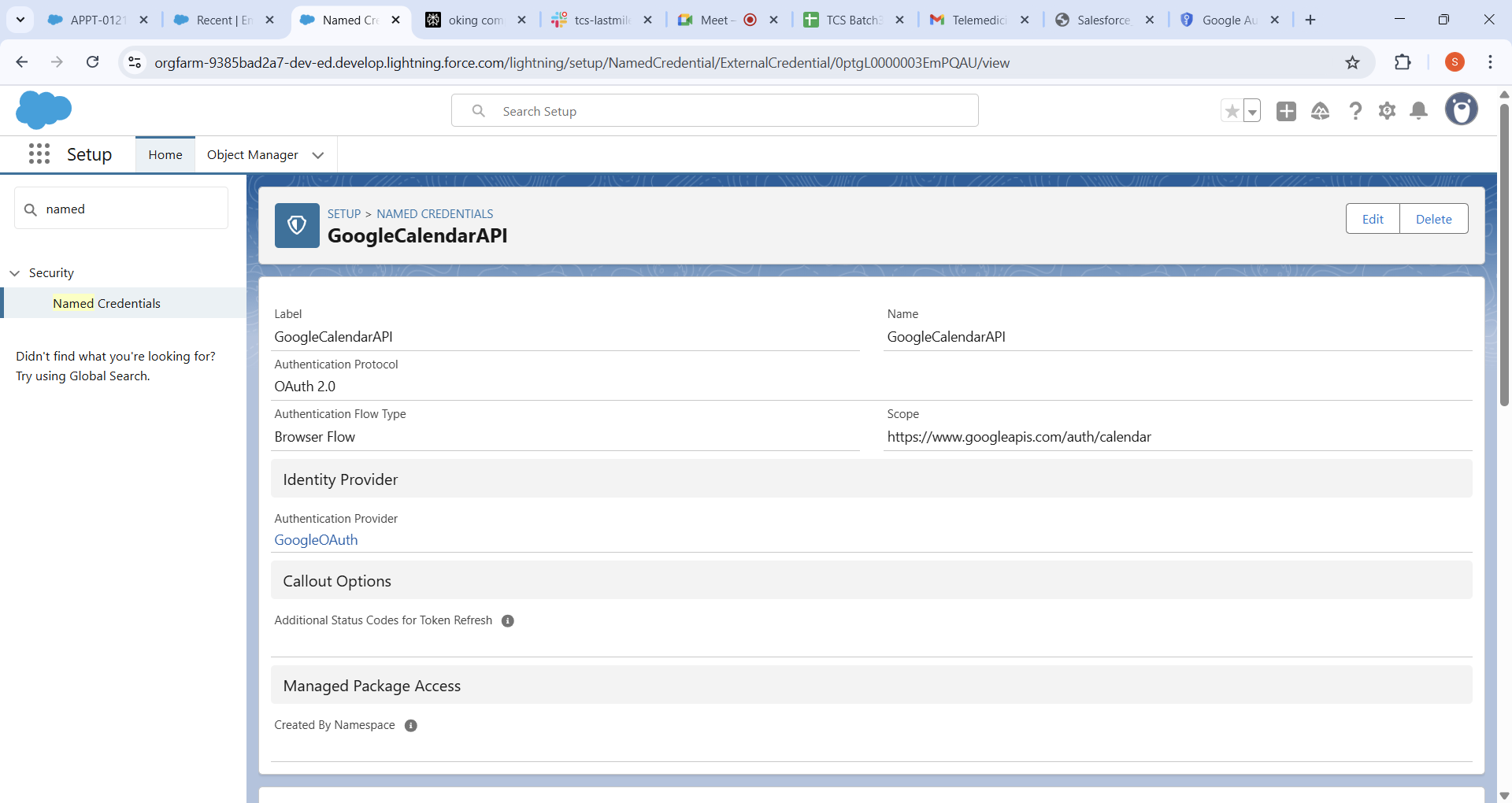
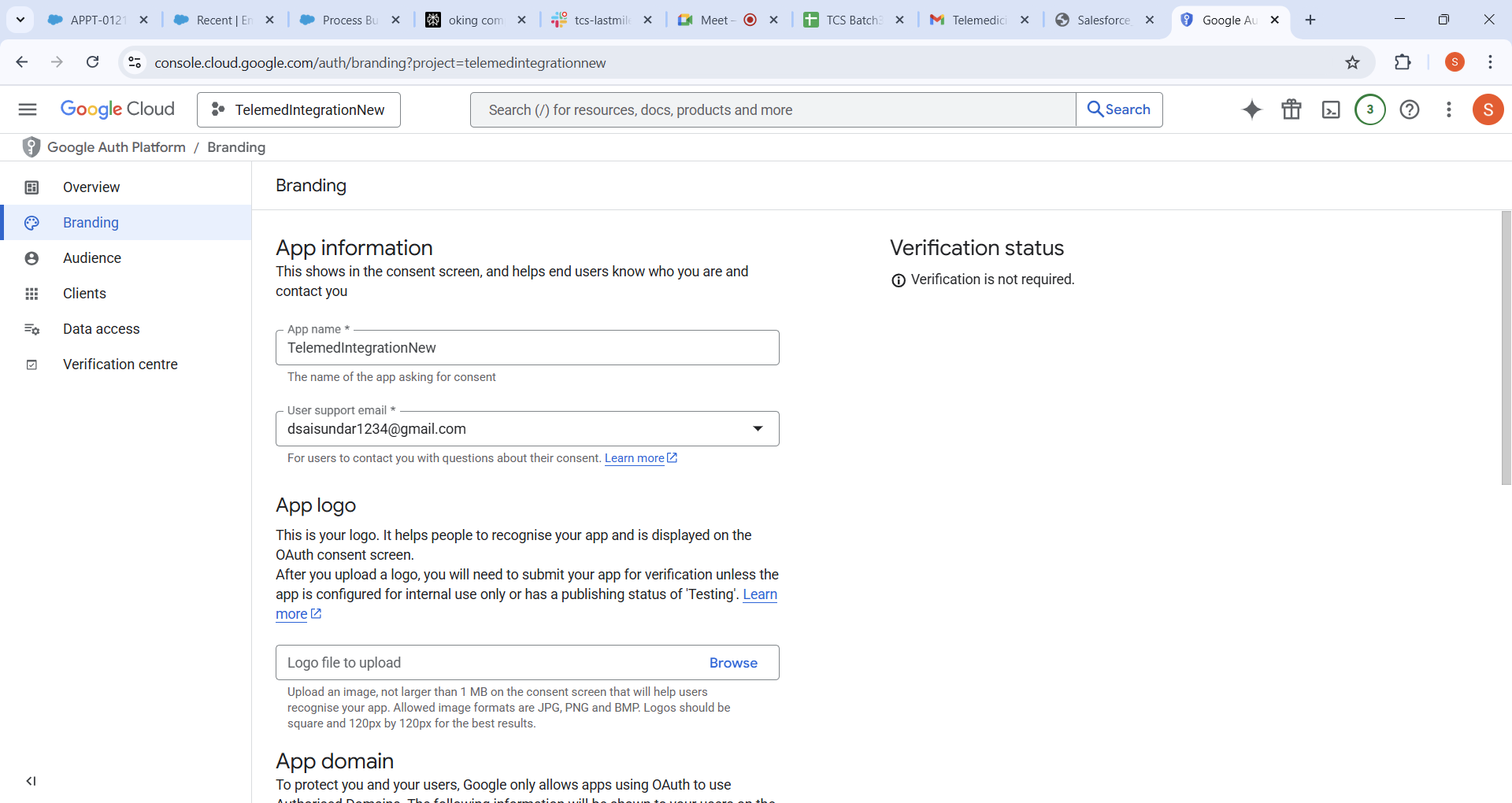
**Phase7- Integration Access Google meet Link formation and process flows**

**1.Named Credentials:**

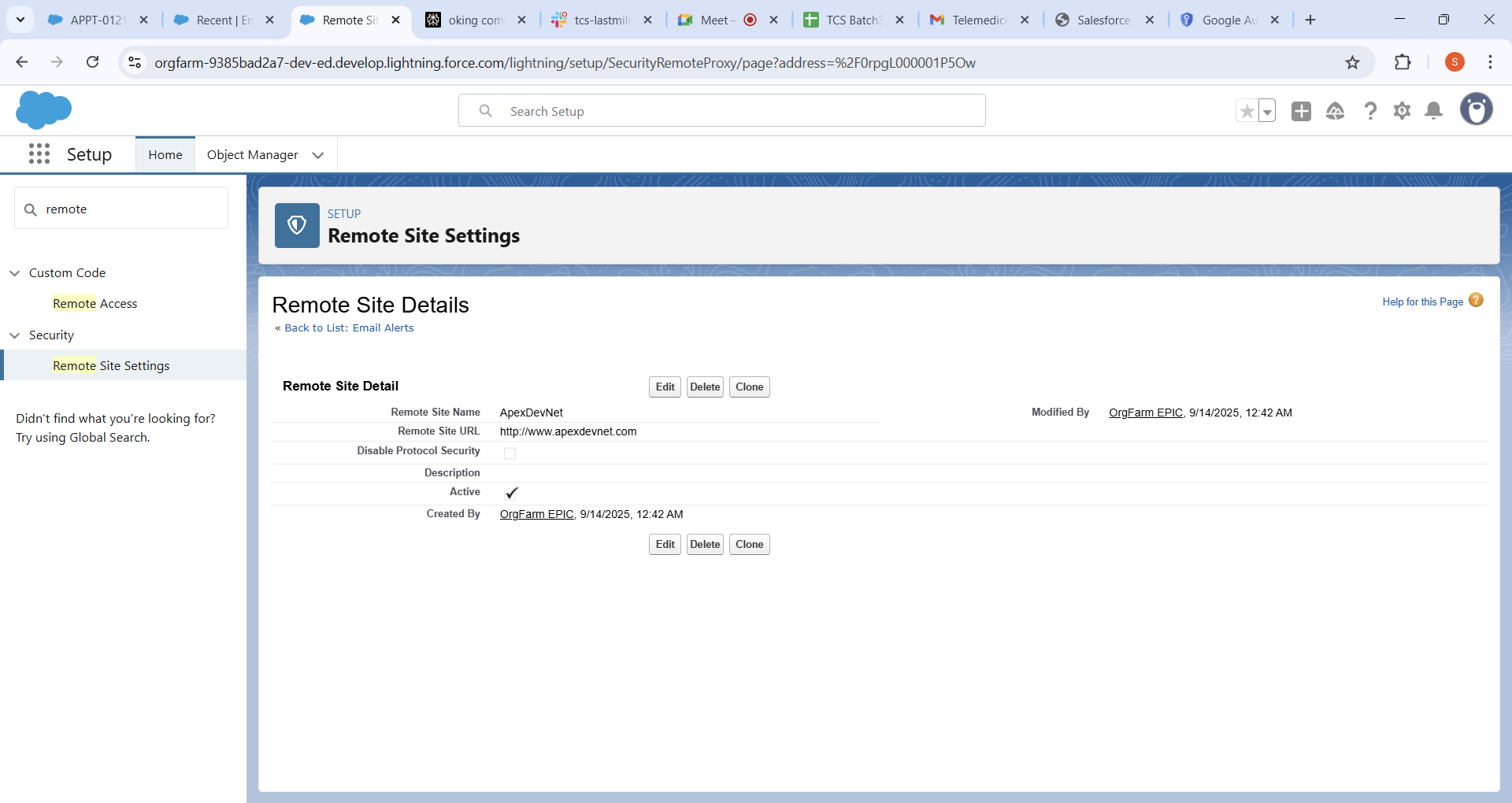


Created the GoogleAPI in the Named Credential for the access for the GoogleCloud



Named Credential configured in Salesforce to authenticate and securely connect to the Google Calendar API. The Named Credential includes the endpoint URL, OAuth 2.0 settings, and references the Google Auth Provider for seamless token management.

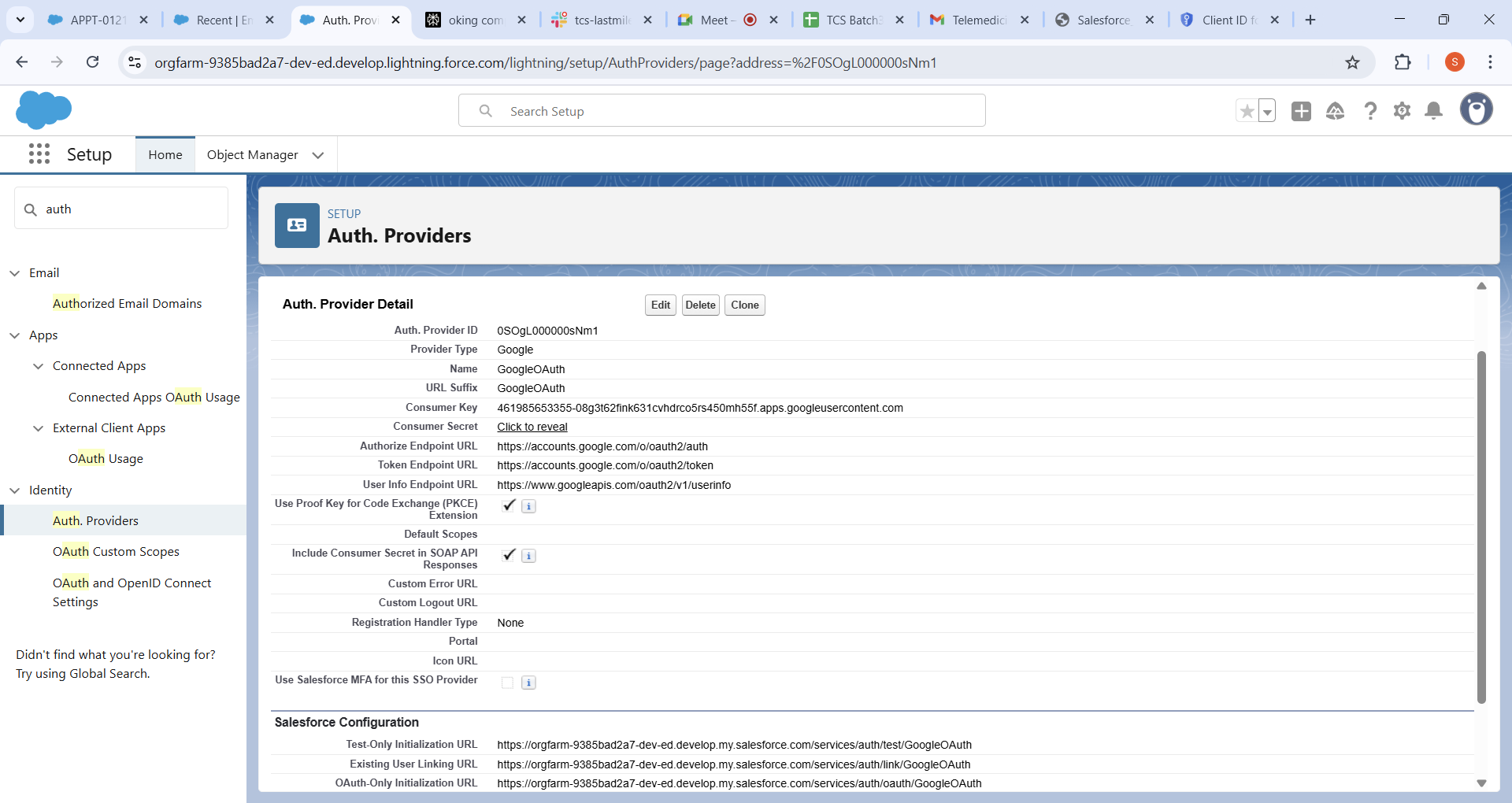
2.**Remote Site Settings:**

****

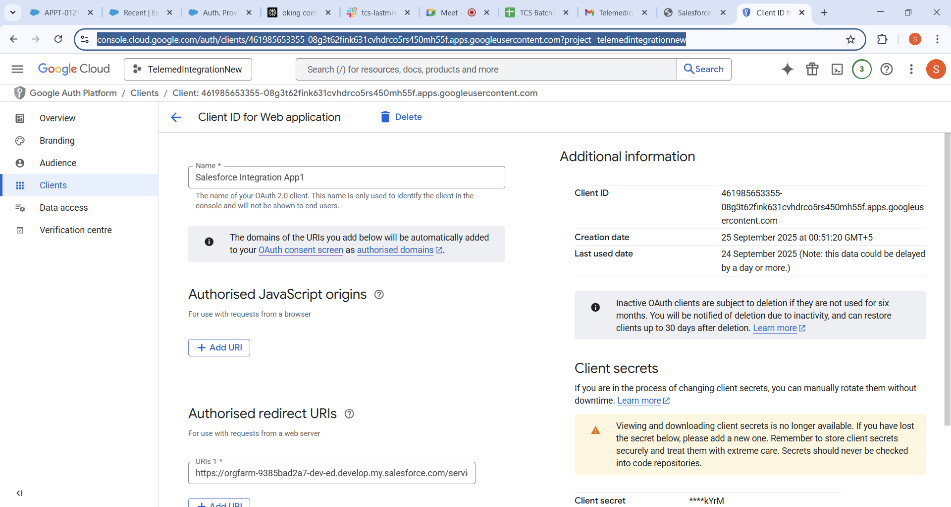
This image lists the Remote Site Settings currently registered in Salesforce, including the Google API URL. This setting allows Salesforce Apex code to make HTTP callouts to external endpoints like Google's calendar API, ensuring secure communication without Salesforce callout restrictions.

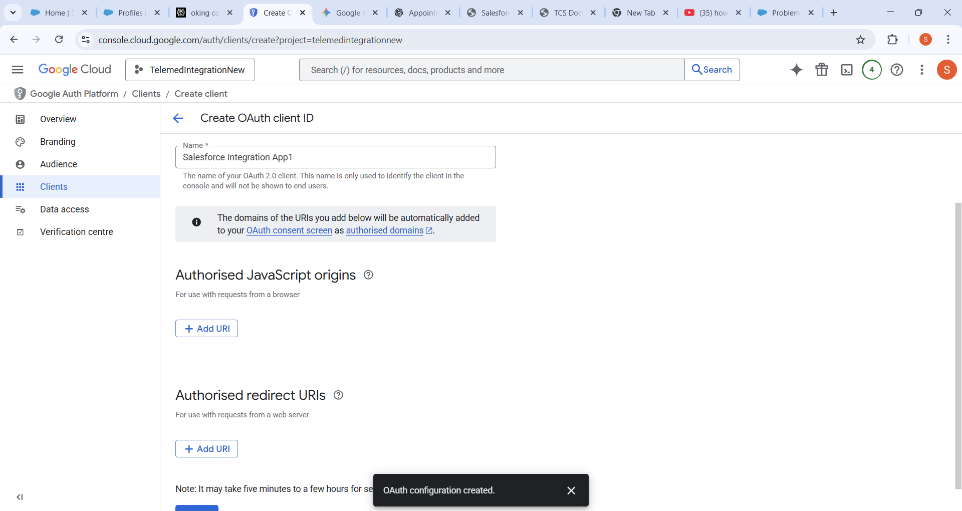
* List page showing all configured remote sites
* The remote site URL you added to allow callouts

3.**OAuth & Authentication Provider:**

****

Here is the Authentication Provider setup screen showing the Google OAuth 2.0 configuration.





This provider facilitates OAuth authentication between Salesforce and Google, specifying Client ID, Client Secret, and the Salesforce callback URLs to handle OAuth tokens for accessing Google services.

**4.Apex Callout Code:**

public with sharing class GoogleMeetIntegration {

public static String createGoogleMeet(TeleAppointment\_\_c appt) {

HttpRequest req = new HttpRequest(); req.setEndpoint('callout:GoogleCalendarAPI/calendar/v3/calendars/primary/events?conferenceDataVersion=1');

req.setMethod('POST');

req.setHeader('Content-Type', 'application/json');

Map<String, Object> event = new Map<String, Object>{

'summary' => 'Telemedicine Appointment',

'start' => new Map<String,String>{ 'dateTime' => String.valueOf(appt.Appointment\_Date\_\_c) },

'end' => new Map<String,String>{ 'dateTime' => computeEndDateTime(appt) },

'attendees' => new List<Map<String,String>>{

new Map<String,String>{'email' => getDoctorEmail(appt)},

new Map<String,String>{'email' => getPatientEmail(appt)}

},

#Some more Code is there just not added

}

private static String computeEndDateTime(TeleAppointment\_\_c appt) {

if(appt.Appointment\_Date\_\_c == null) return null;

Integer duration = appt.Duration\_Minutes\_\_c != null ? appt.Duration\_Minutes\_\_c.intValue() : 30;

return appt.Appointment\_Date\_\_c.addMinutes(duration).formatGMT('yyyy-MM-dd\'T\'HH:mm:ss\'Z\'');

}

private static String getDoctorEmail(TeleAppointment\_\_c appt) {

if(appt.Doctor\_\_c == null) return null;

User doctorUser = [SELECT Email FROM User WHERE Id = :appt.Doctor\_\_c LIMIT 1];

return doctorUser != null ? doctorUser.Email : null;

}

private static String getPatientEmail(TeleAppointment\_\_c appt) {

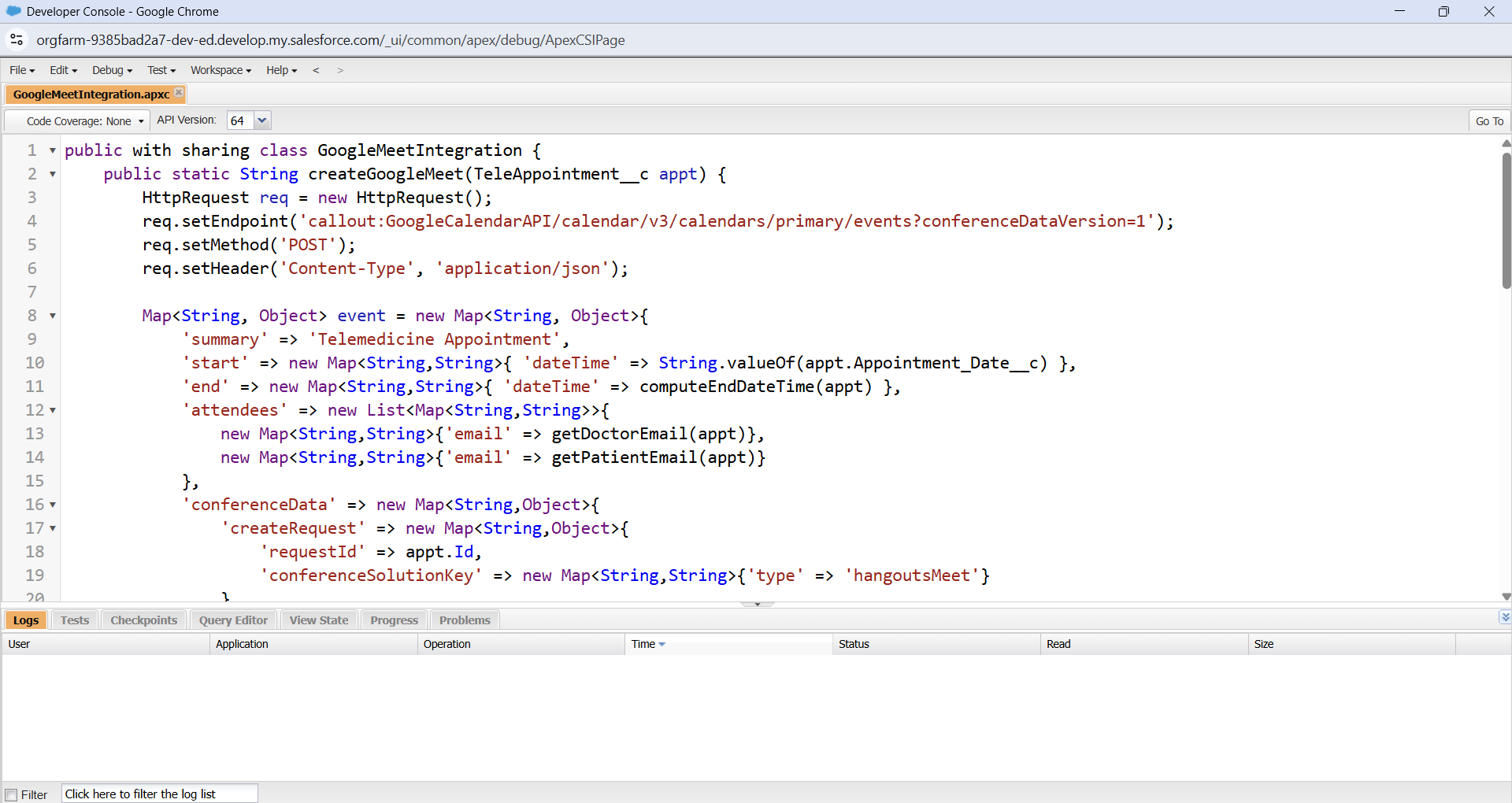
if(appt.patient\_\_c == null) return null;

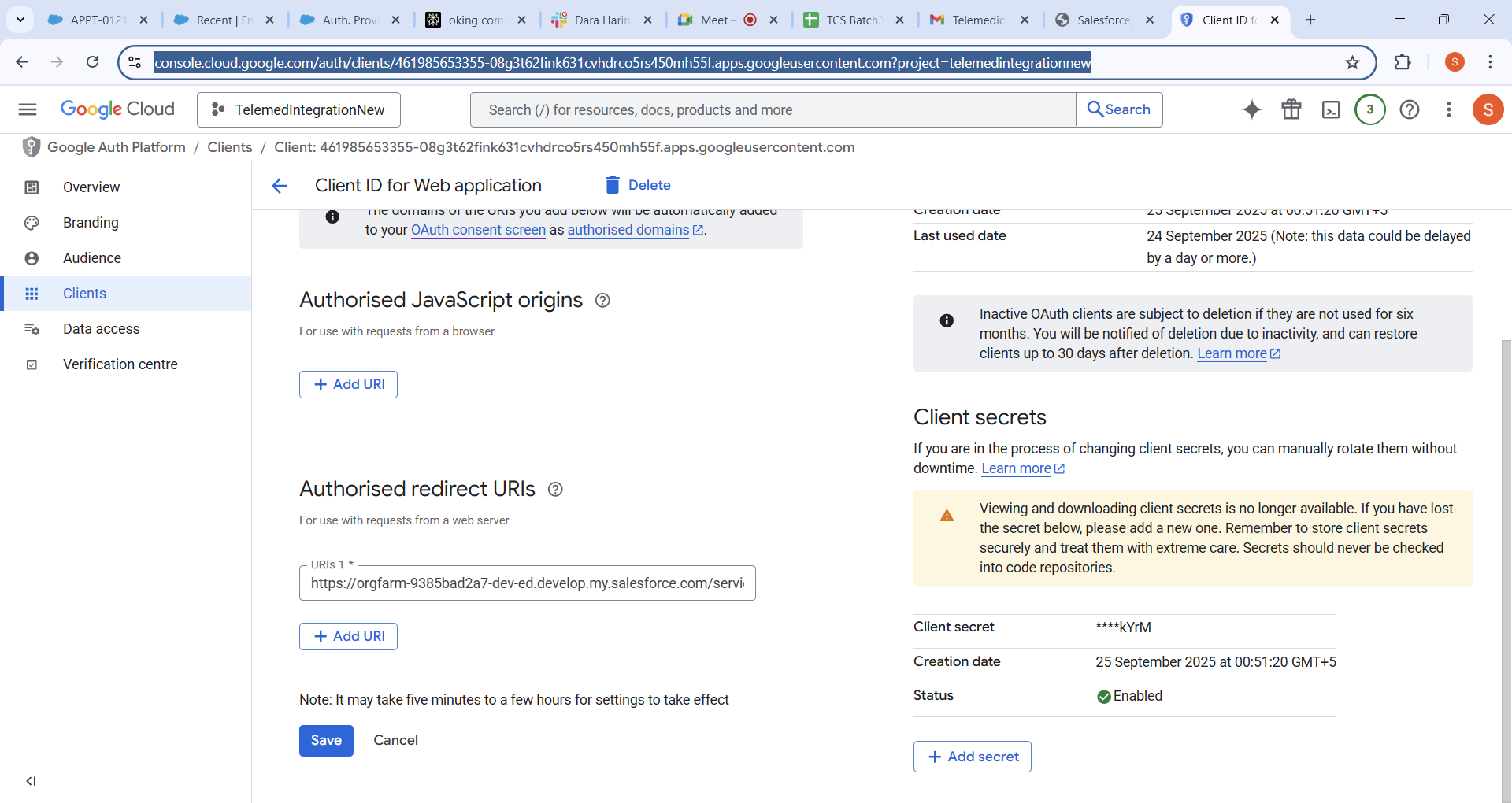
Patient\_\_c patient = [SELECT Patient\_Email\_\_c FROM Patient\_\_c WHERE Id = :appt.patient\_\_c LIMIT 1];

return patient != null ? patient.Patient\_Email\_\_c : null;

}

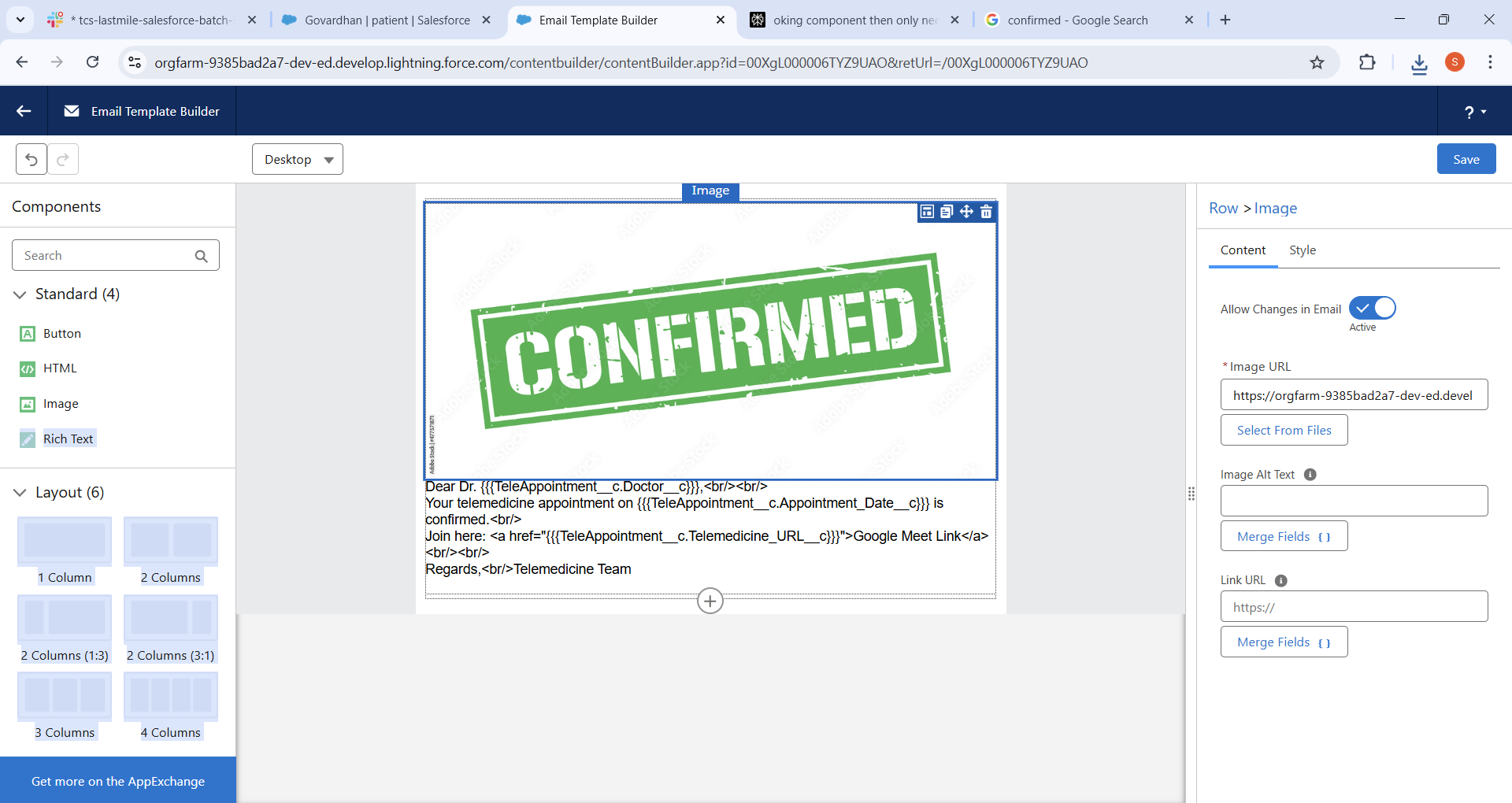
}



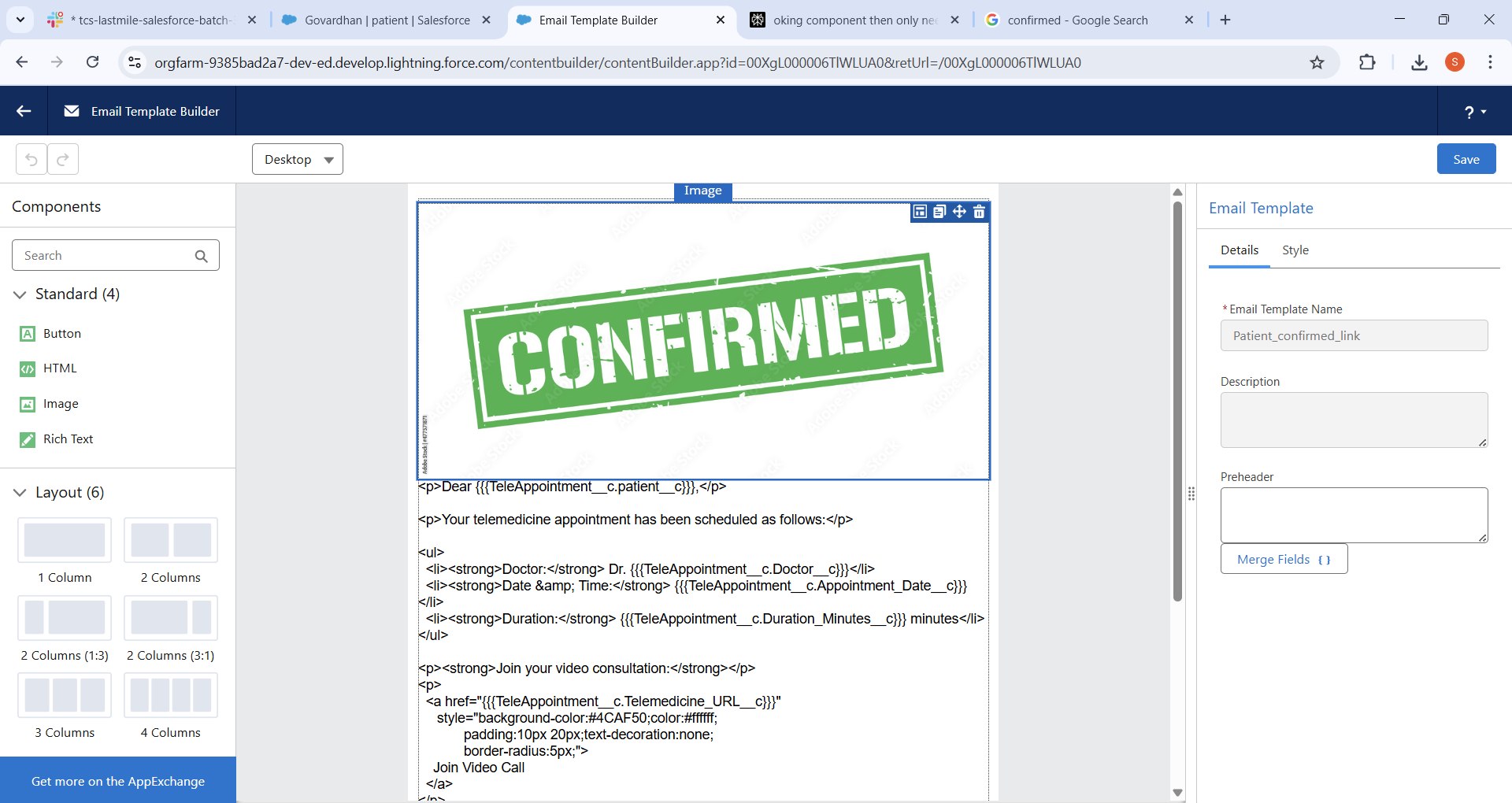


The screenshot captures the Salesforce Apex class method responsible for creating Google Meet links via HTTP REST callouts to the Google Calendar API. It demonstrates setting request endpoints, headers, body payloads, and processing JSON responses within Salesforce.

**5.Email Template for confirming the Meeting and Give the meet link:**



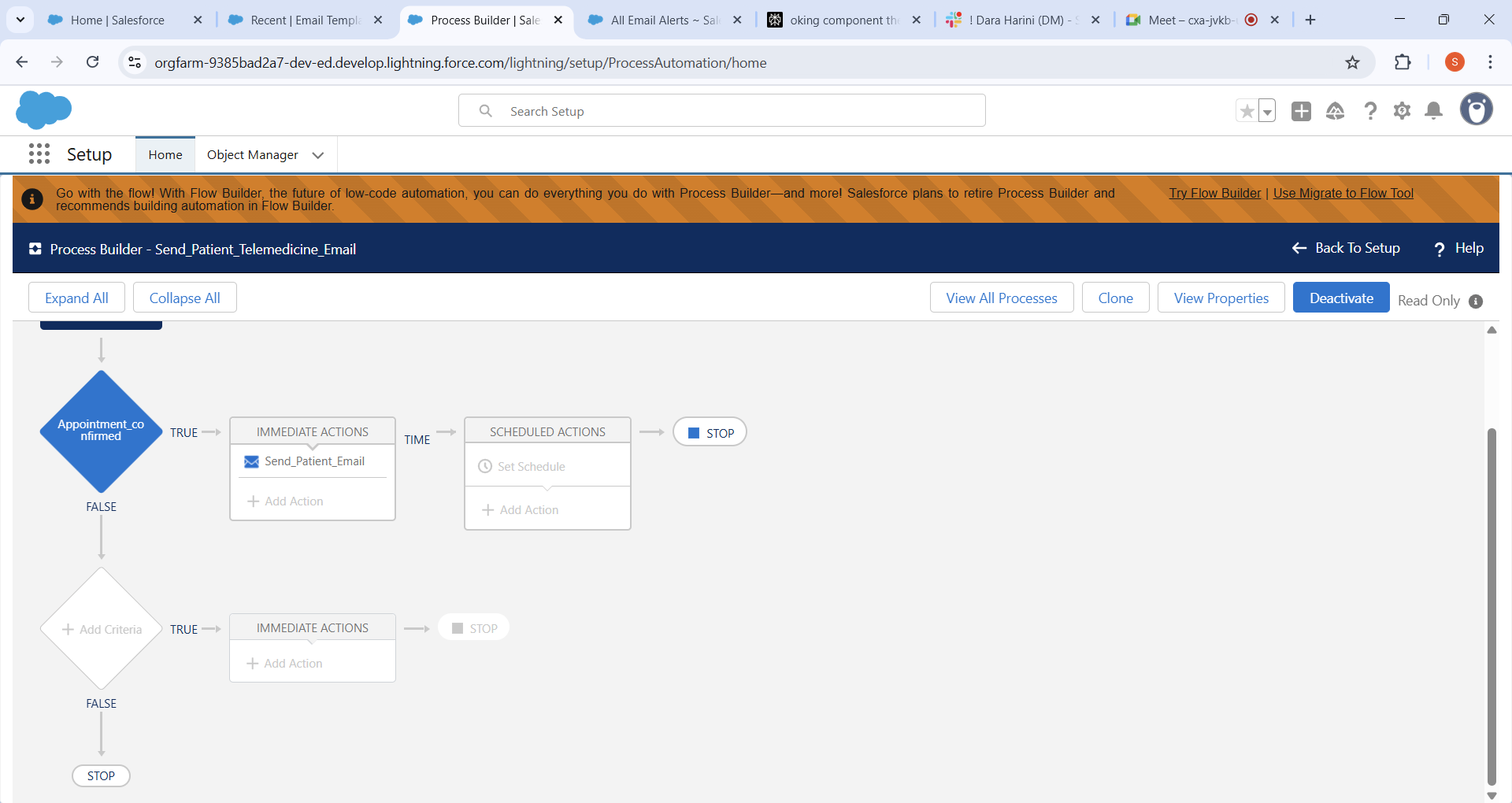
This is For the doctor for appointment is confirmed successfully and link is send successfully



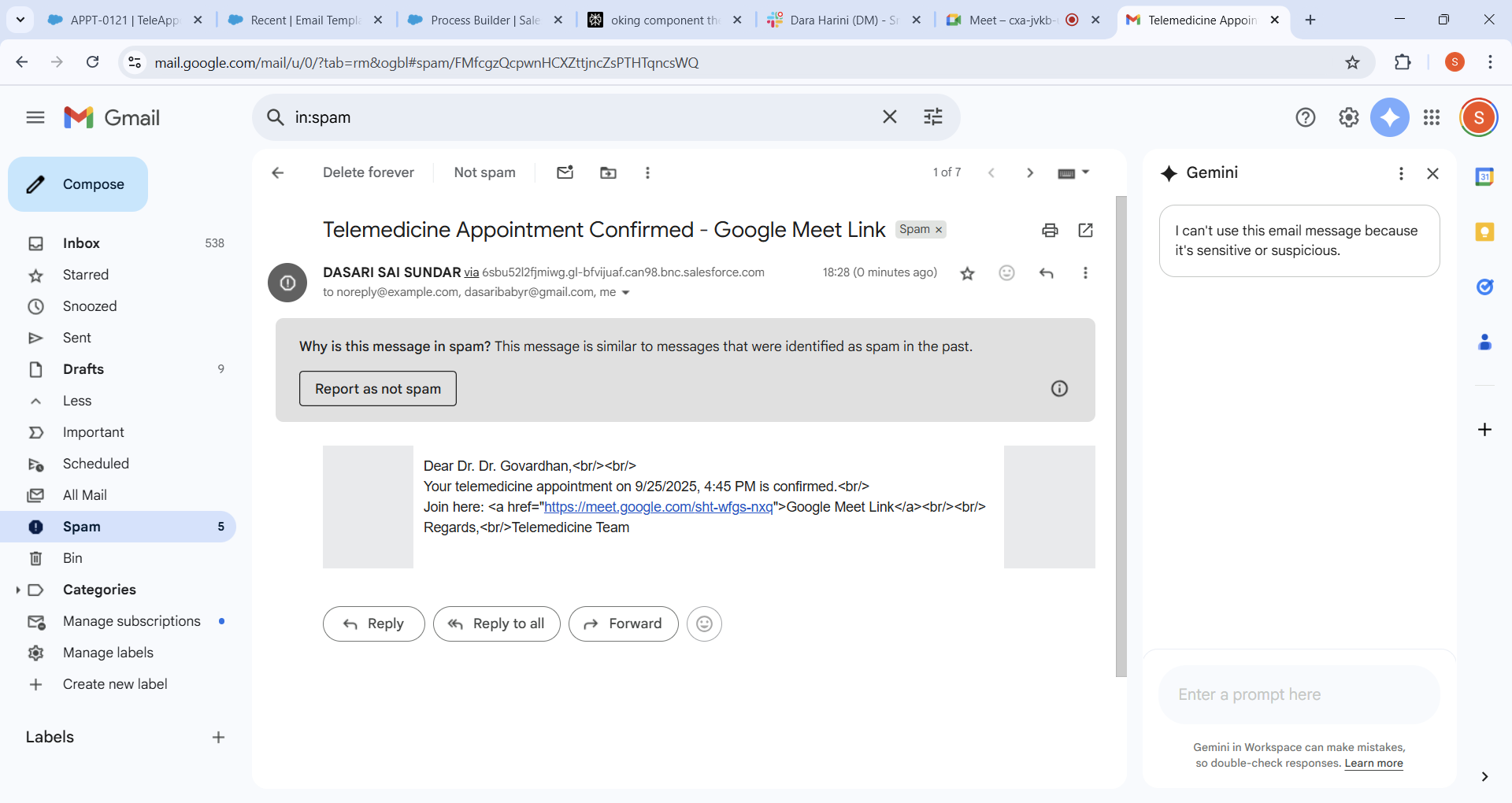
This is For the Patient for appointment is confirmed successfully and Google meet link is send successfully

This section displays the Lightning Email Template design used to send appointment confirmation emails containing Google Meet links to patients and doctors. The template utilizes merge fields from the TeleAppointment custom object for personalized content.

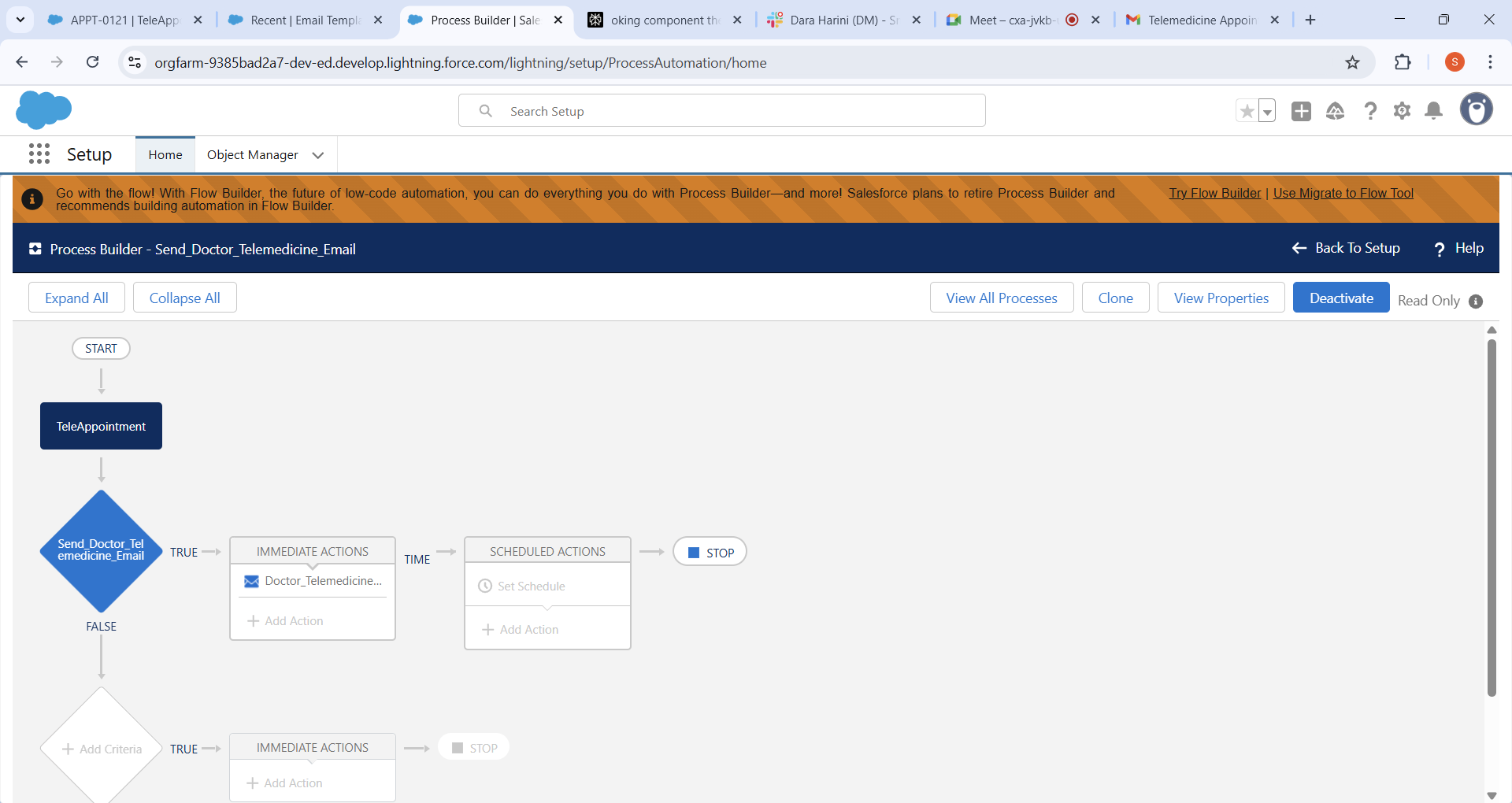
**6.Process Builder Screenshot**



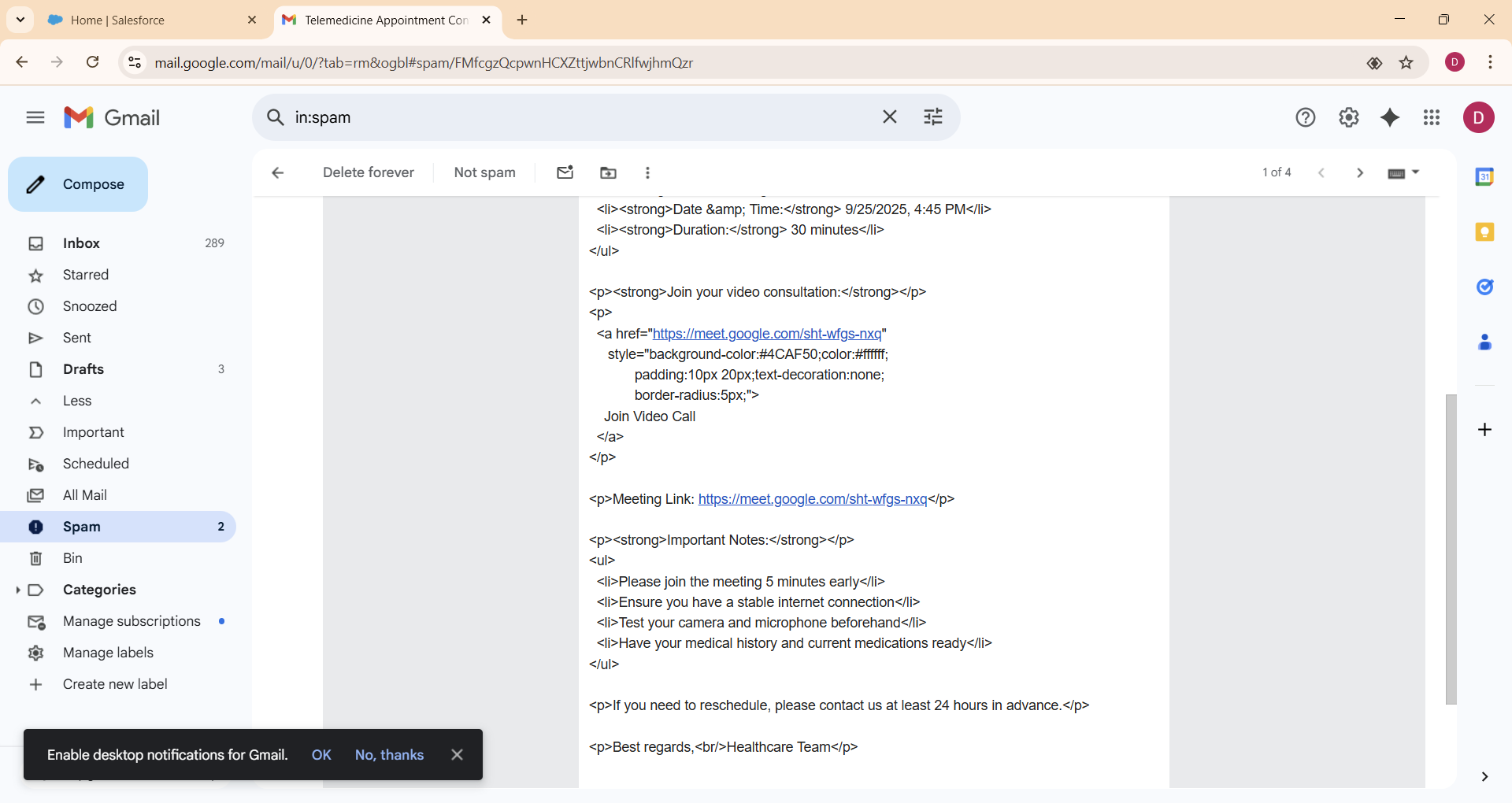
Process Builder is created for the patient link.



Received the mail at the patient email box

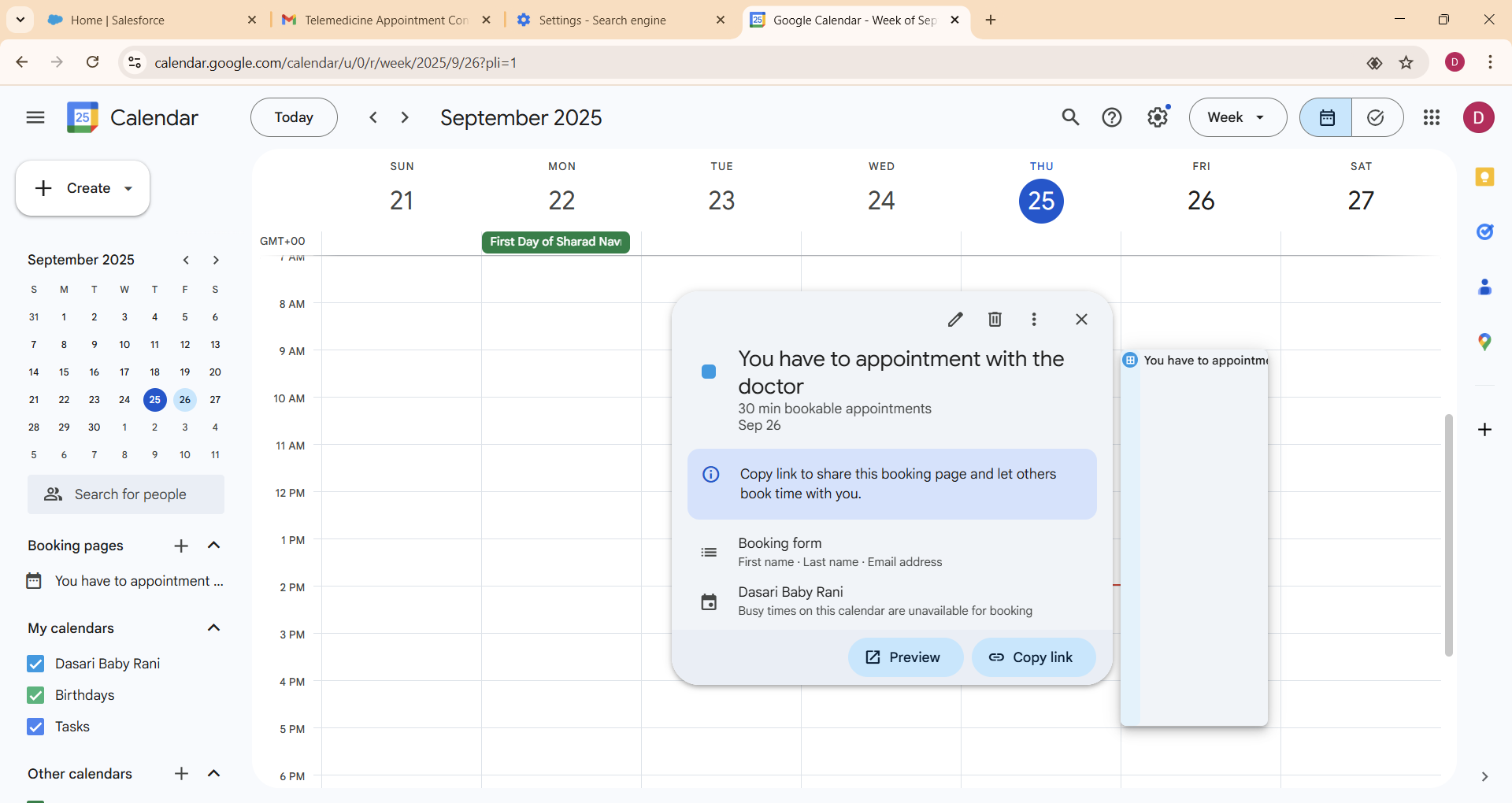


Process Builder is created for the Doctor link.



Received the email in the Doctor mail

This shows the Process Builder configuration used to automate sending of confirmation emails. It monitors the TeleAppointment object for status changes to 'Confirmed' and presence of the Google Meet link, triggering an email alert action to notify the corresponding user.



Google Calendar integration is successfully implemented using Named Credentials, OAuth authentication, and Apex REST callouts. Remote Site Settings ensure secure API access. Automated email notifications are triggered via Process Builder using Lightning Email Templates to notify users with Google Meet links. The system is now ready to streamline telemedicine appointments with seamless video conferencing features.