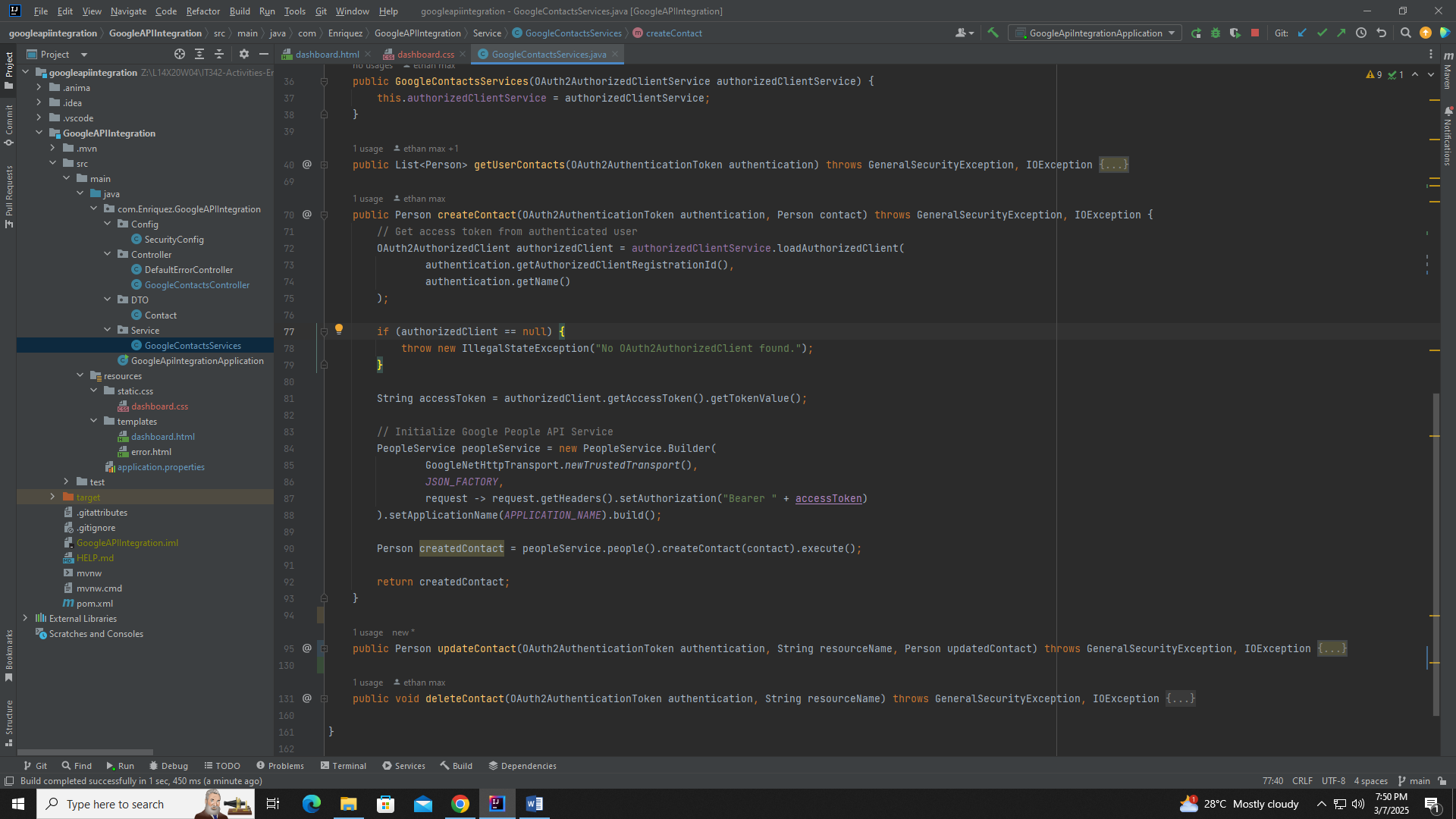
BASIC CRUD OPERATIONS

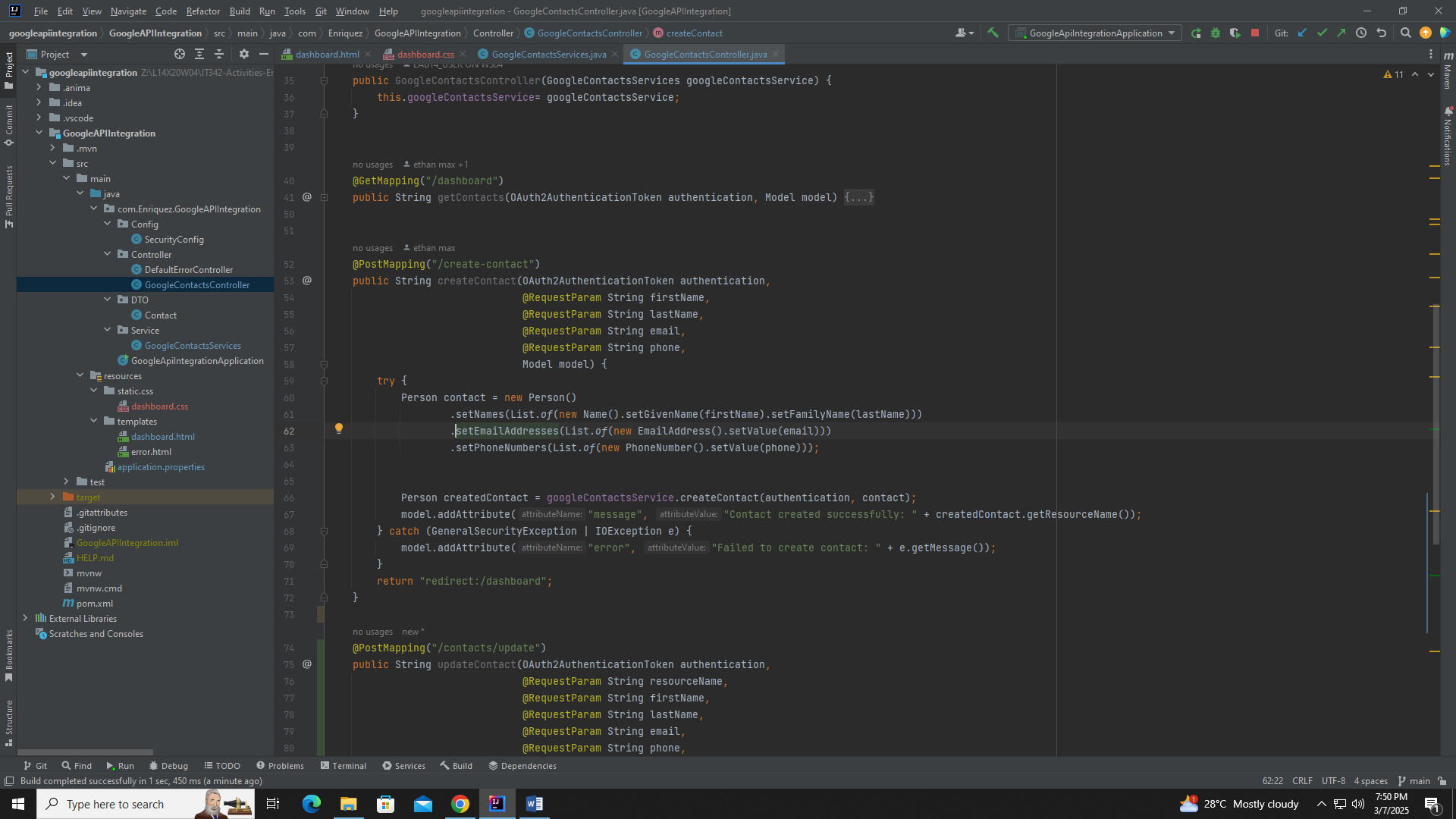
Create

-Service



Explanation: Get the Access Token from the authenticated user (Google account). Pass through the If-statement for checking the value o fthe Access Token. Initalize the Access Token. Initialize the Google People API service then execute the Create Function from the Google People API library. Returns an object Created Contact

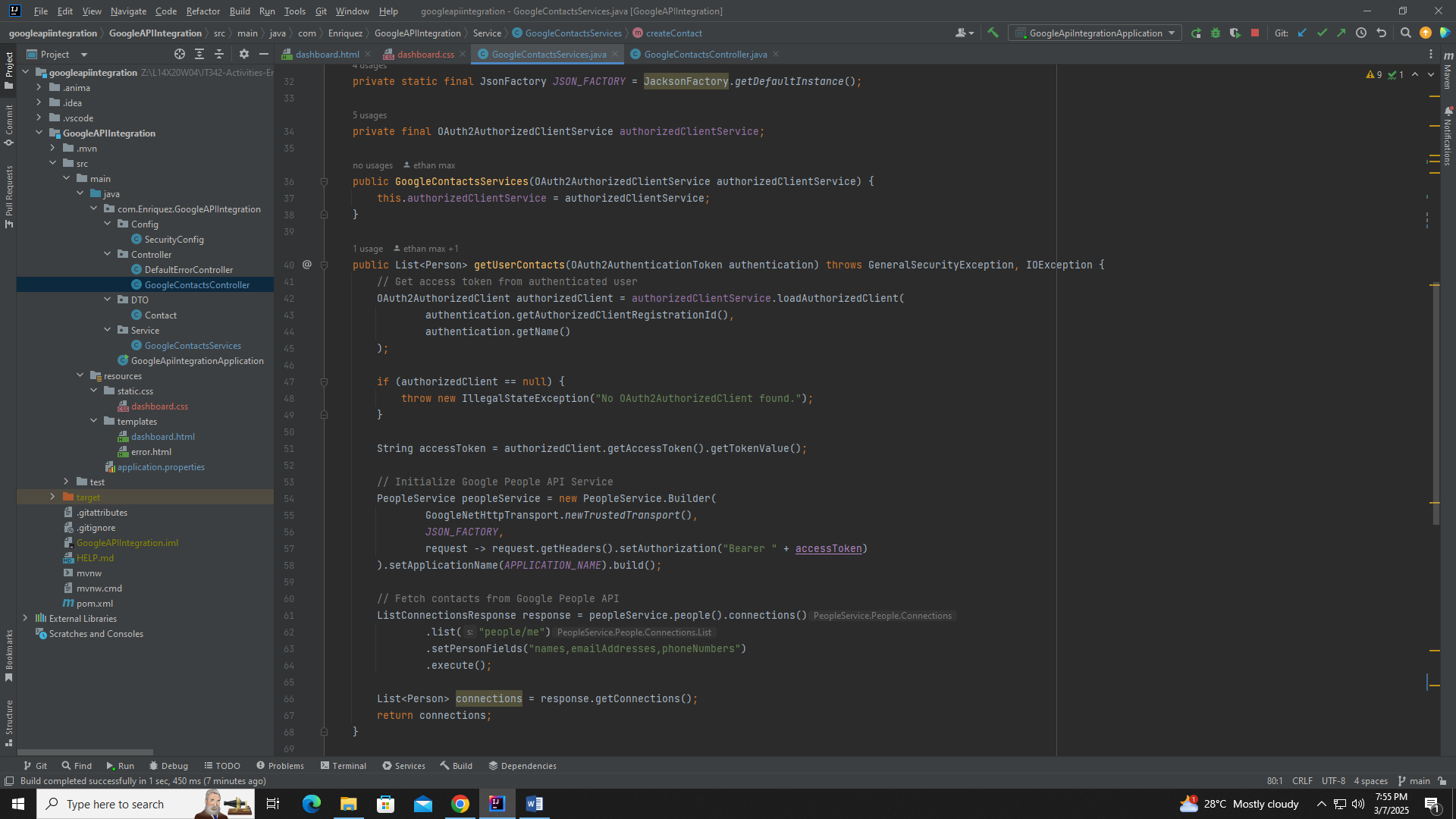
Struggle: Initializing the Google People API Service from the Google People Documentation.

-Controller

Explanation: Initialize the URL parameters for firstname, lastname, email, phone according to the model. Try-Catch to pass the parameters to the getter-setters from the model, then call the function createContact from GoogleContactsServices. If it catches an error it sends a message “Failed to create Contact”. After Try-Catch block it redirects back to the dashboard.html

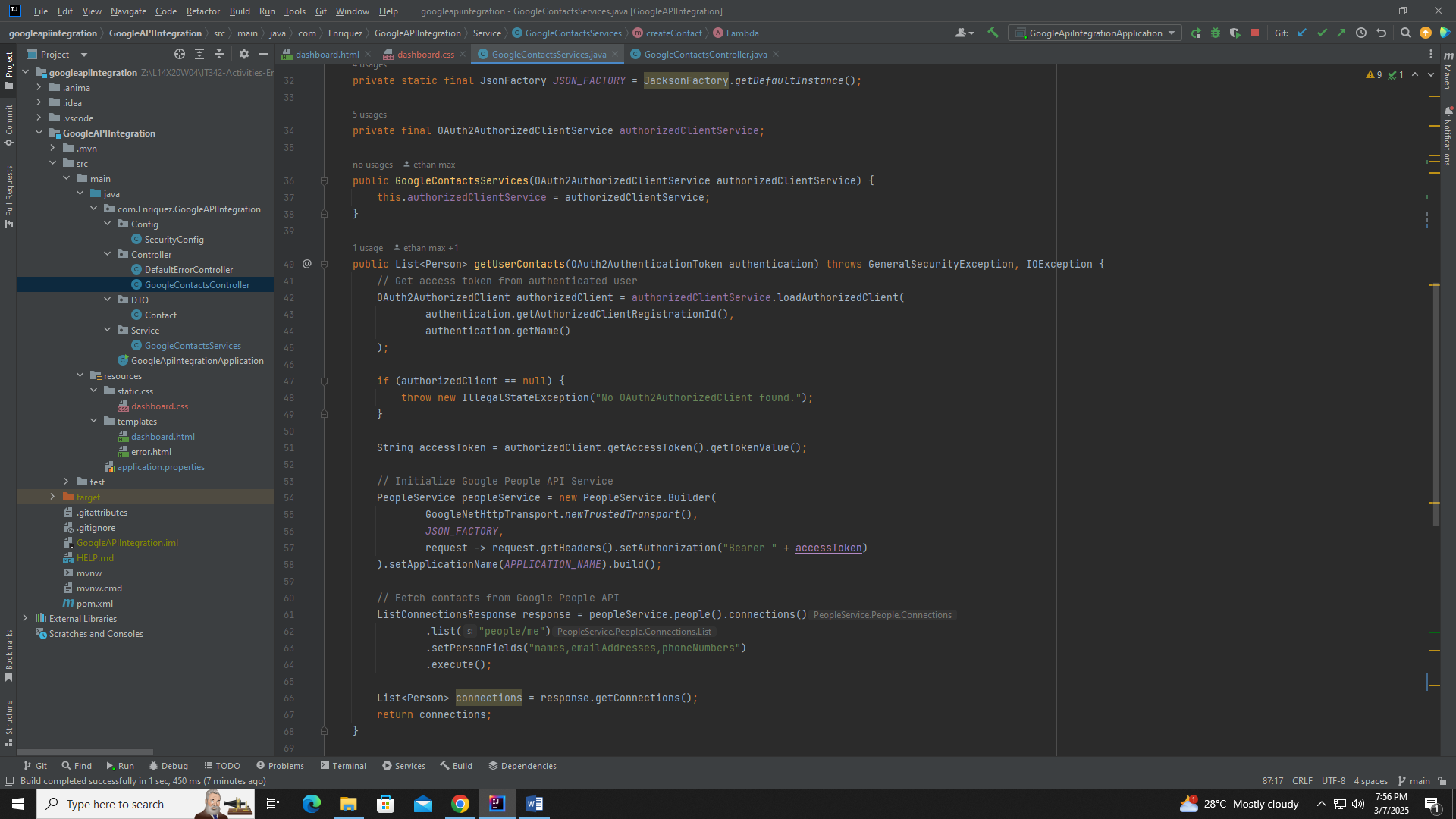
Struggle: Passing paramaters to the Google People API and avoiding a 401 error.

Read

-Service 

Explanation: Get the Access Token from the authenticated user (Google account). Pass through the If-statement for checking the value of the Access Token. Initalize the Access Token. Initialize the Google People API service. Then fetch the list of contacts through object response. This function returns object connections

-Struggle: Getting the List of Contacts and passing it to the parameter for the controller

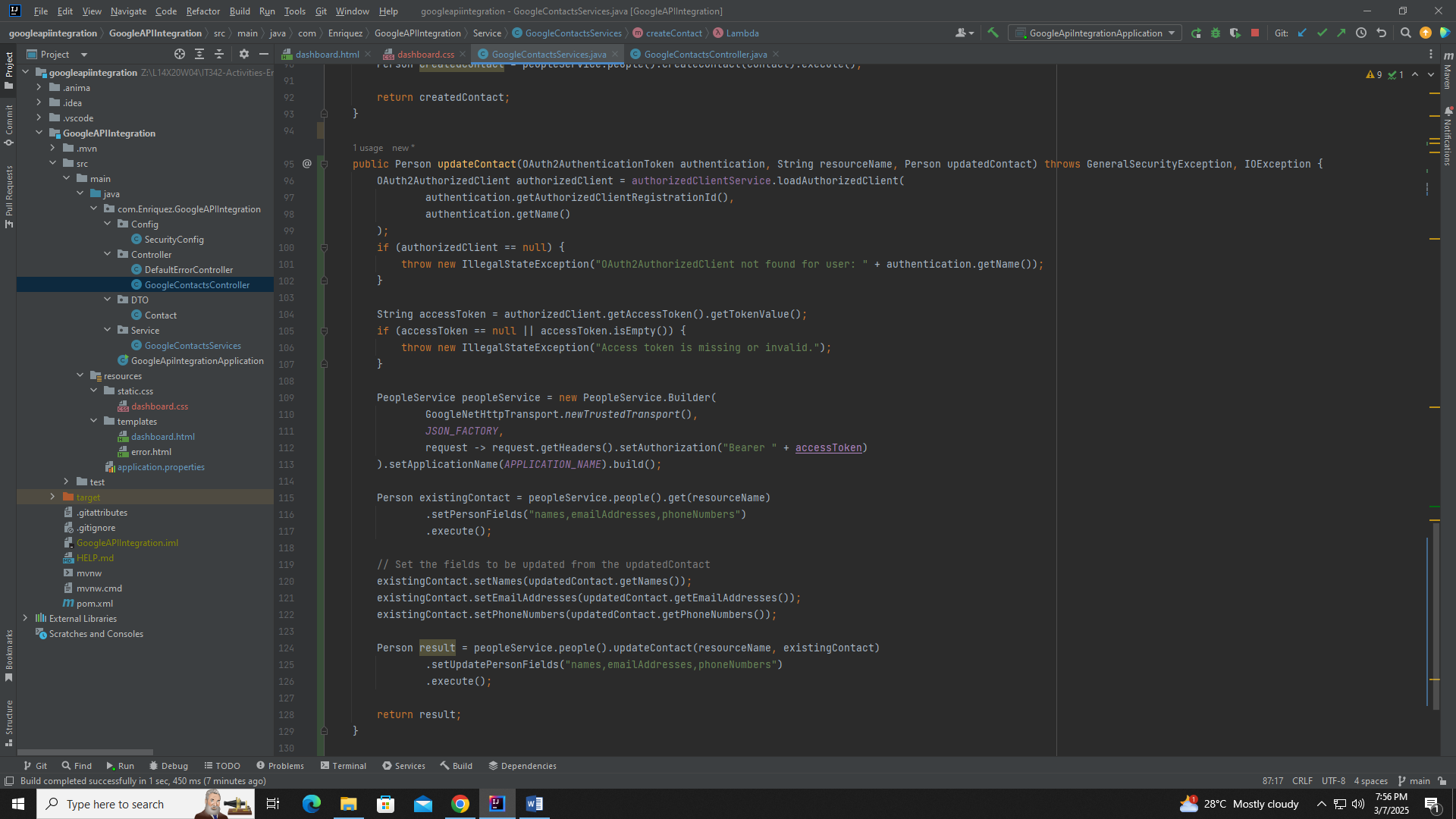
-Controller 

Explanation: The try block attempts to retrieve the user's contacts from the googleContactsService.getUserContacts(authentication). If successful, it adds the contacts list to the model, making it available for display in the frontend. The catch block handles any security or I/O errors: GeneralSecurityException: Occurs if there's an issue with Google API authentication or encryption. IOException: Occurs if there is a network error or API request failure.

If an error occurs, the message "Failed to fetch contacts: <error message>" is added to the model, allowing the frontend to display the issue. If successful, redirect back to endpoint /dashboard to go back to the dashboard.html file

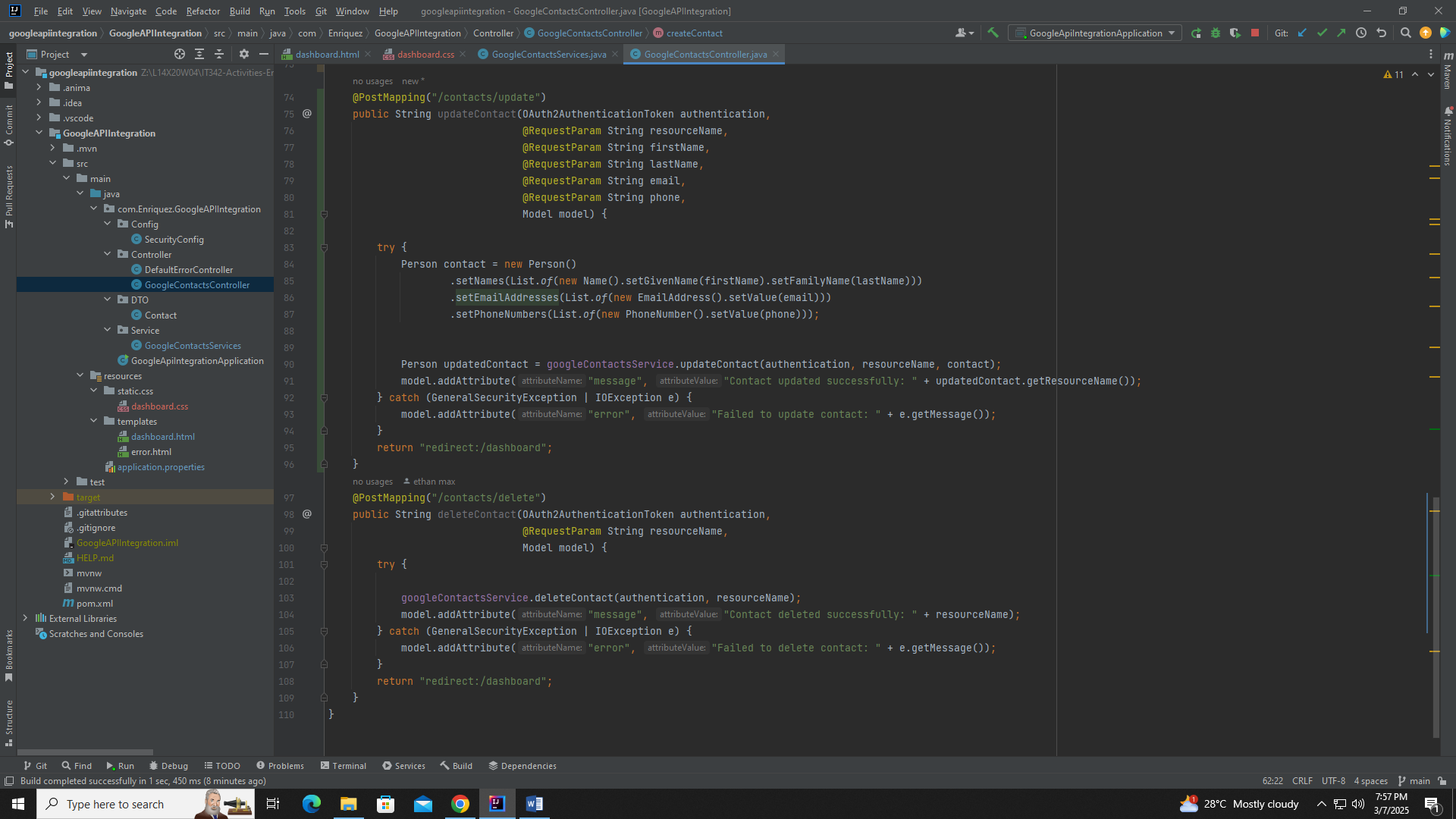
Struggle: Making the service function of retrieving contacts.

Update

-Service 

Explanation: Get the Access Token from the authenticated user (Google account). Pass through the If-statement for checking the value of the Access Token. Initalize the Access Token. Initialize the Google People API service. Then fetch the current contact to be updated through object Person existingContact that takes the variable resourceName. Then updates the values in existingContact with setters. Use Google People Service to send the object values to Google and redirect back to dashboard endpoint

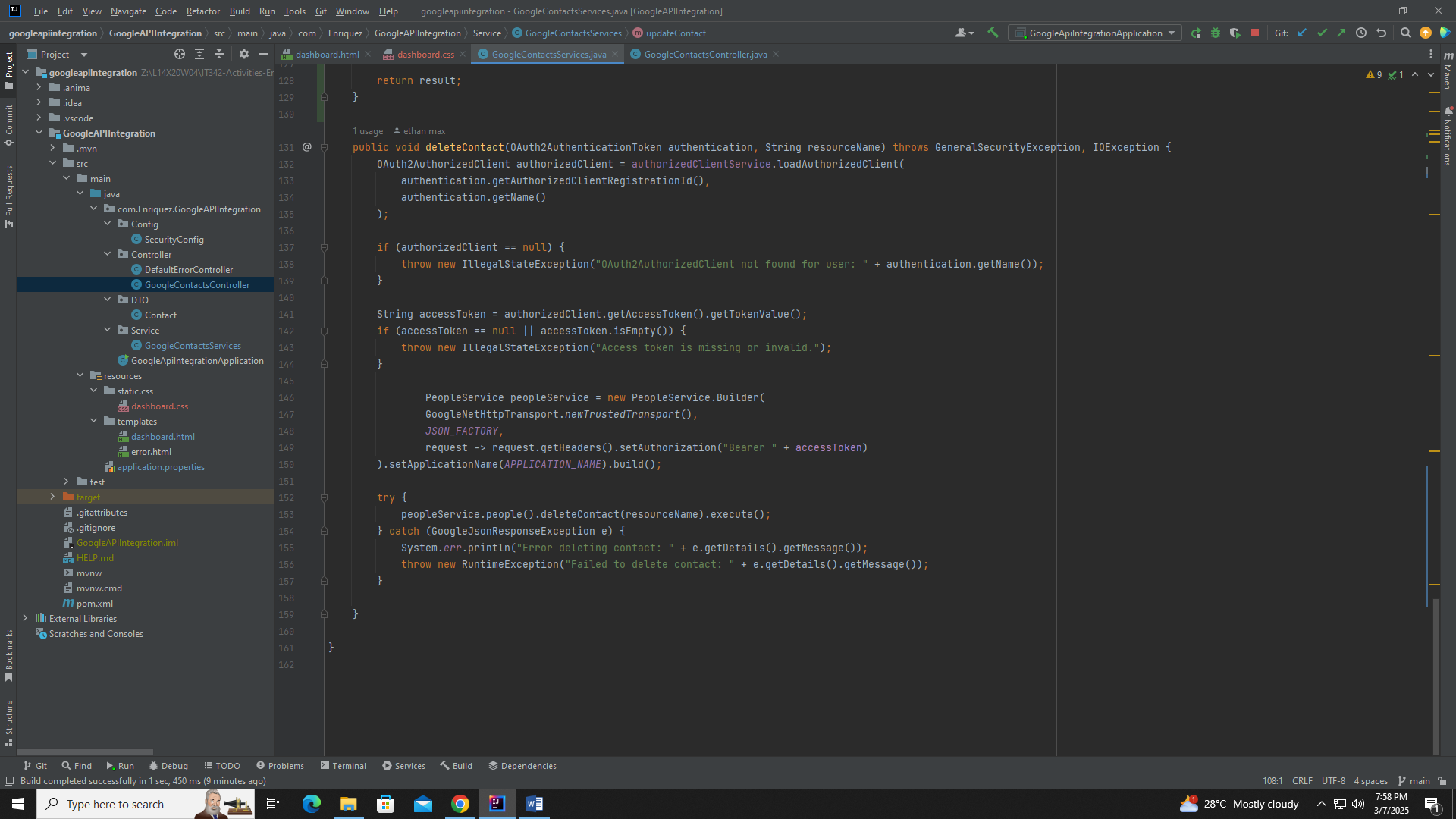
Struggle: Getting existing contact data with resourceName and then passing the object to Google People API.

-Controller 

Explanation: A Try-Catch block for error handling. Try creates a new person object with updated details then call the function for updating from the service file. Catch block sends an error message if it fails. Redirect back to /dashboard endpoint

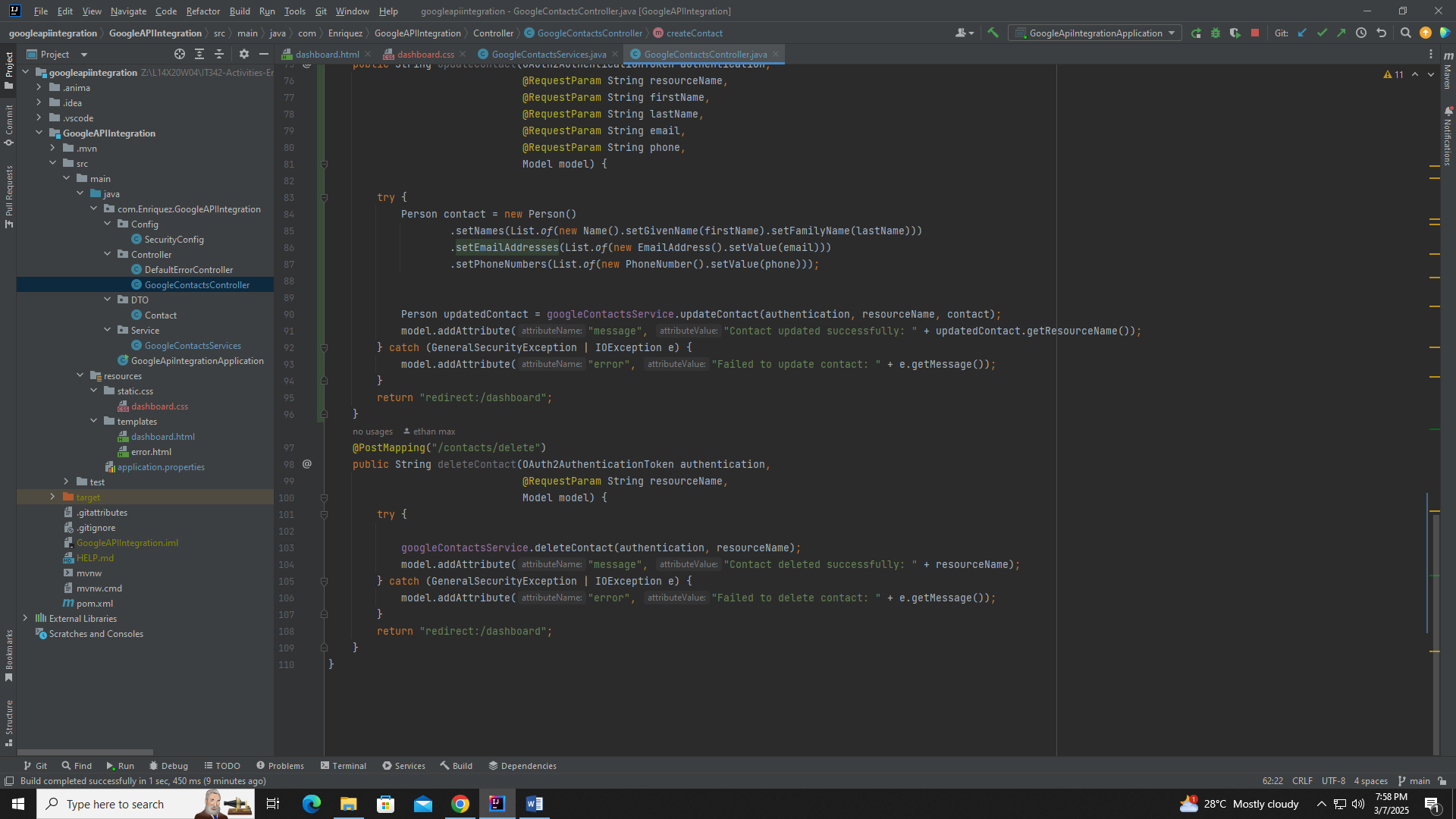
Struggle: Getting the Google Service to talk to the updated Contact Details from my newly created person Object

sDelete

-Service 

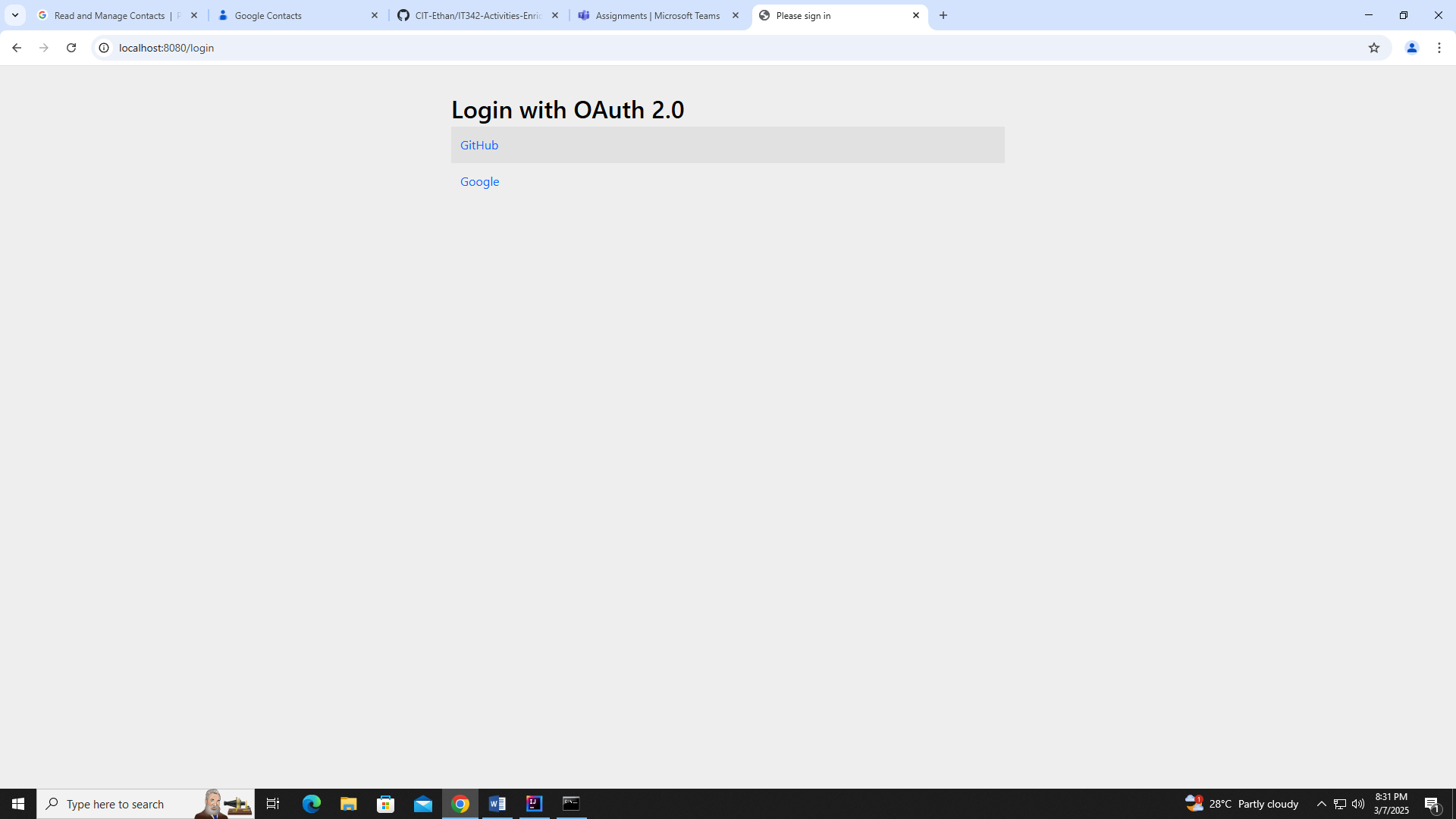
Explanation: Get the Access Token from the authenticated user (Google account). Pass through the If-statement for checking the value of the Access Token. Initalize the Access Token. Initialize the Google People API service. Try catch block that calls deleteContact function from Google People API. Catch sends an error message for failure.

Struggle: Making the button from the html file work with the service function

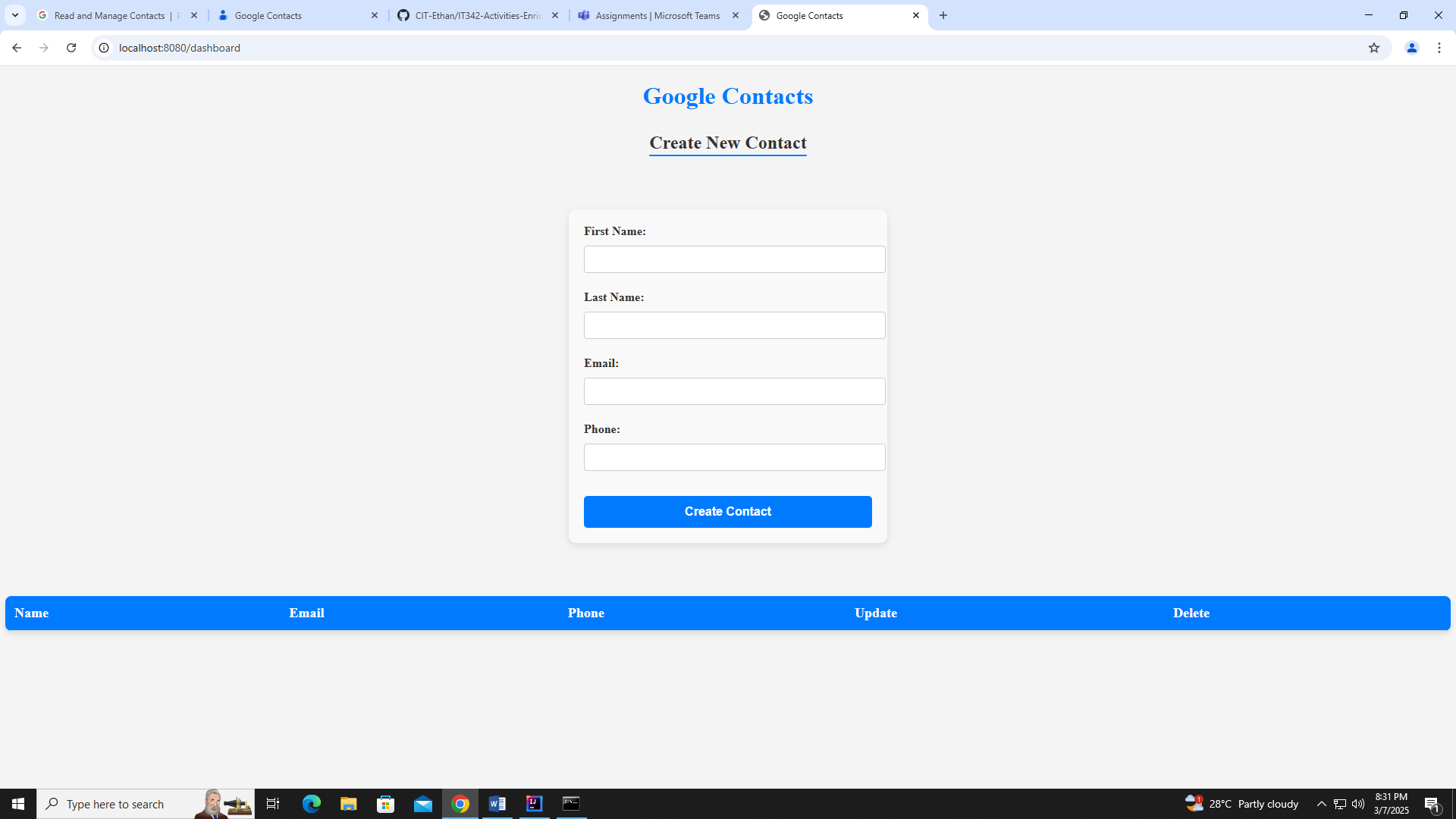
-Controller 

Explanation: attempts to delete a contact from Google Contacts using the googleContactsService.deleteContact() method, authenticated via OAuth2. If the deletion is successful, it adds a success message to the model with the deleted contact's resourceName. If an error occurs, such as an authentication issue (GeneralSecurityException) or an API/network failure (IOException), an error message is stored in the model. Final line redirects back to /dashboard endpoint

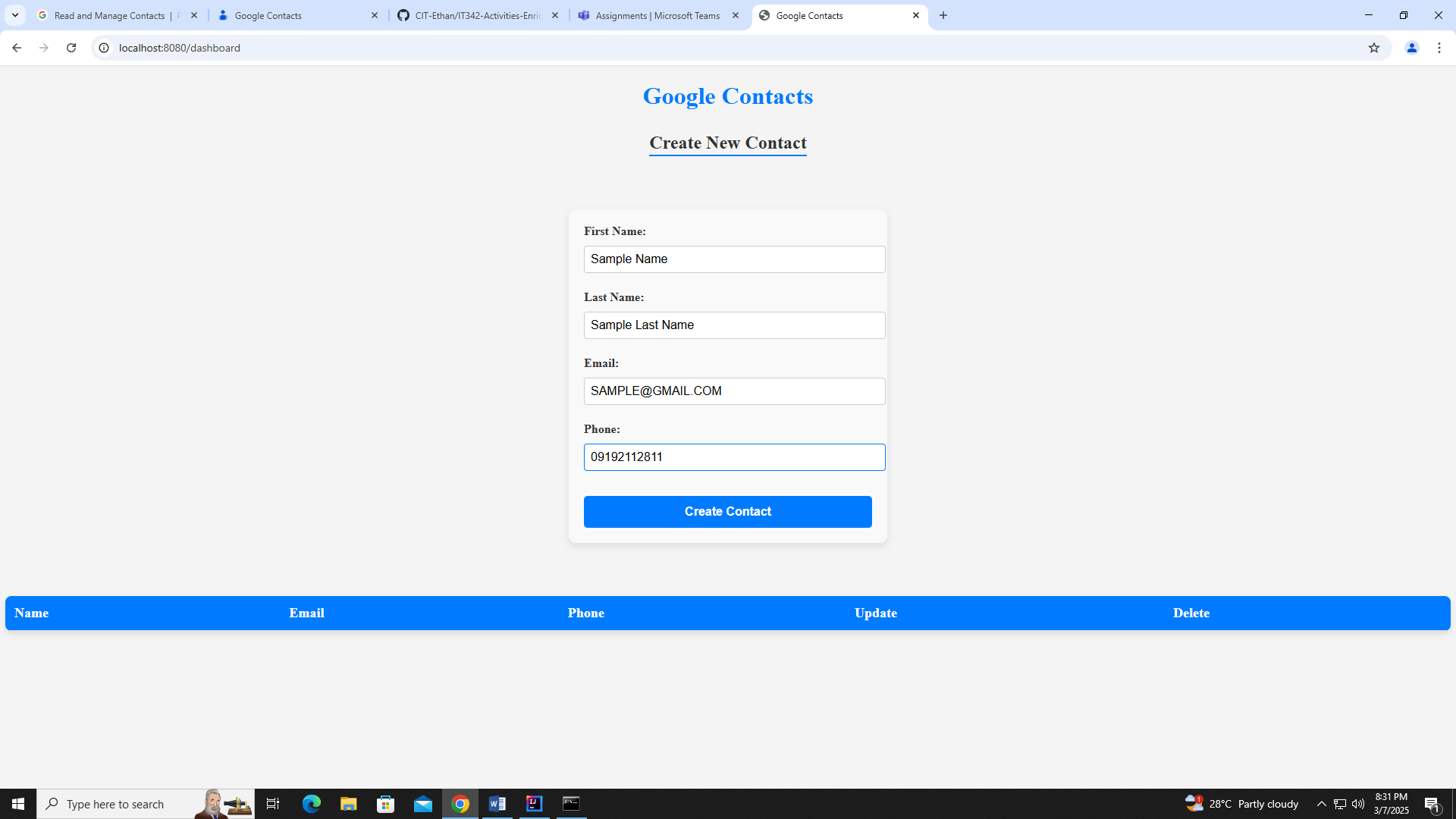
SCREENSHOTS

Login page  


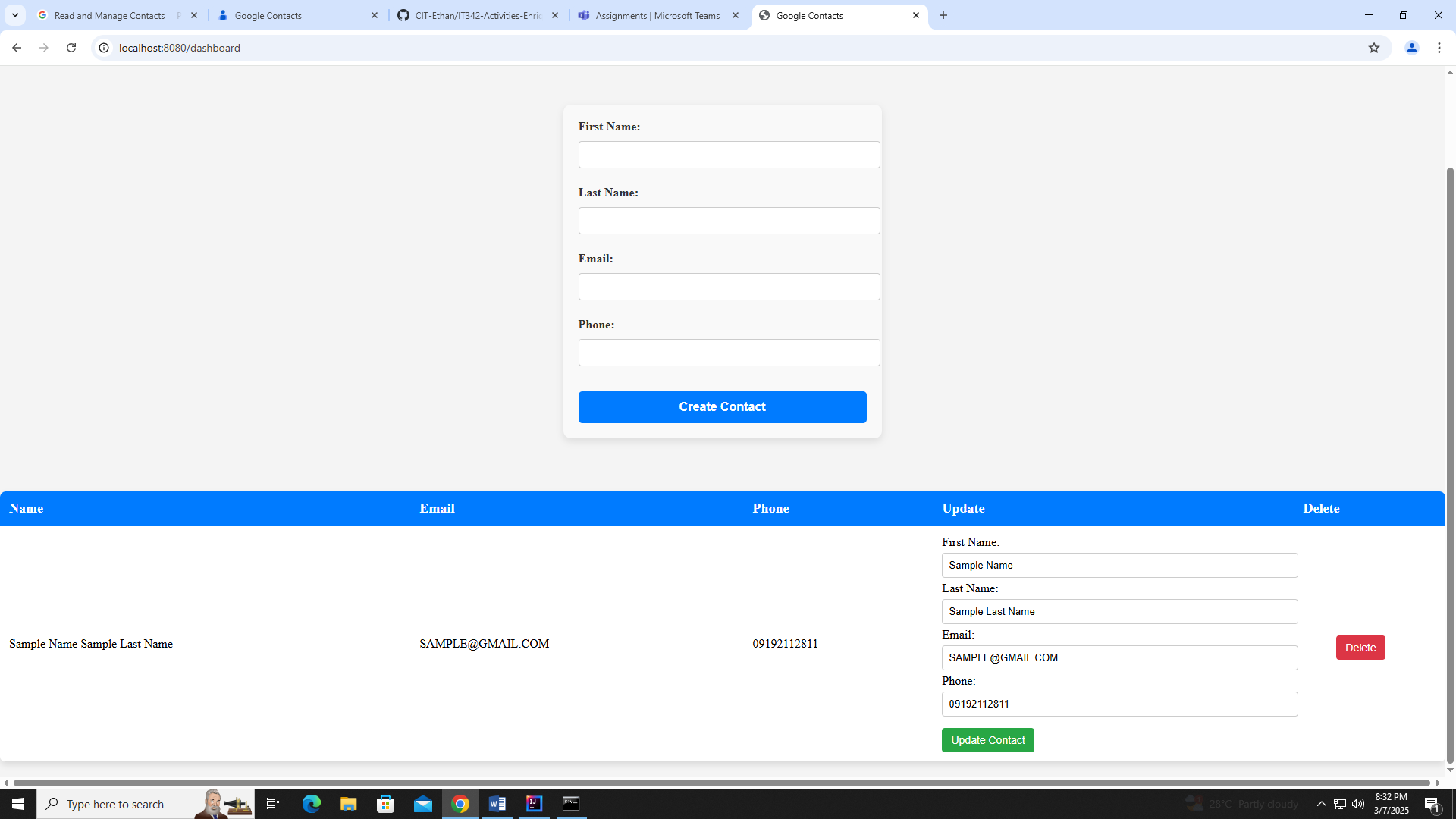
Dashboard

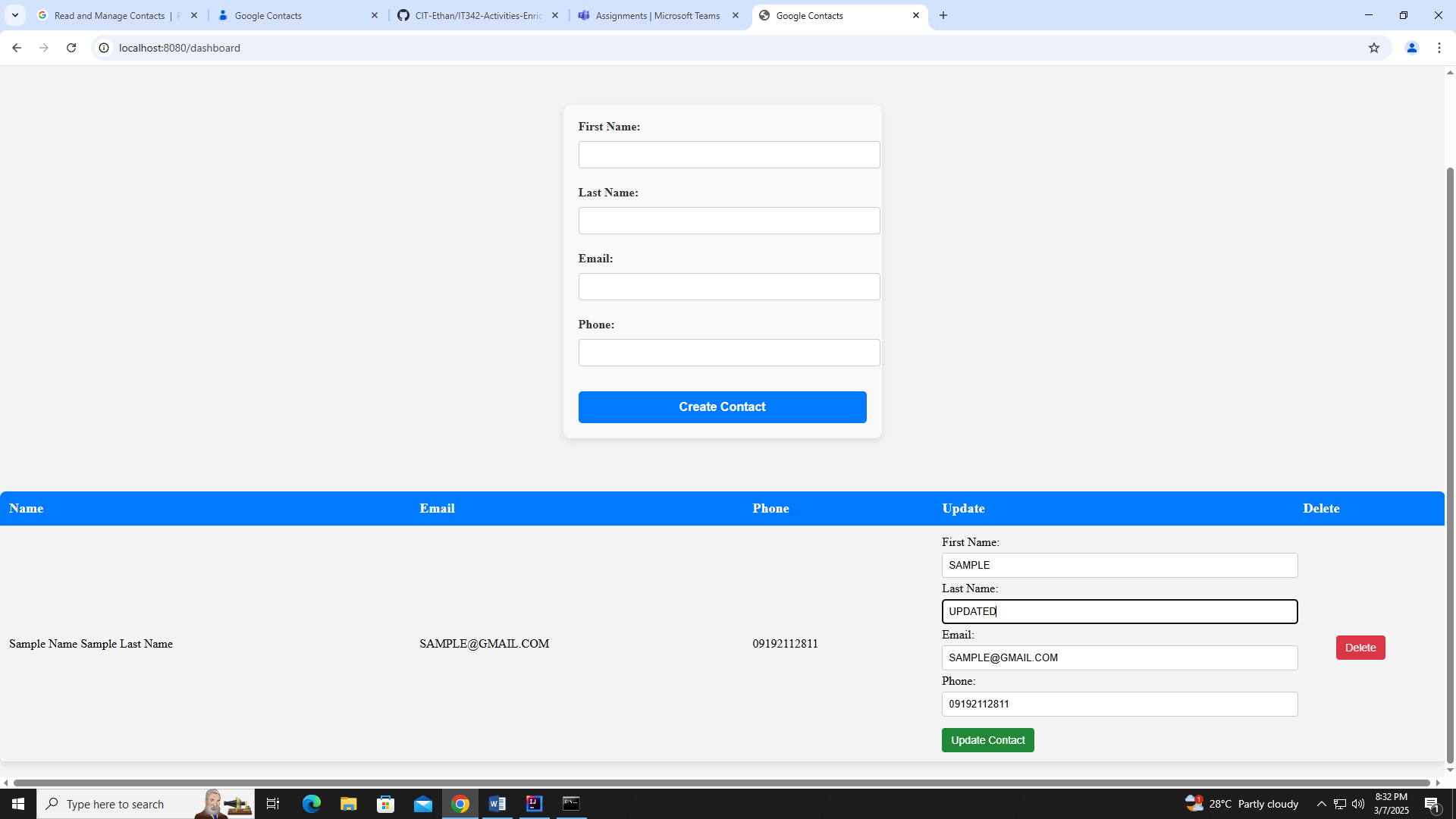


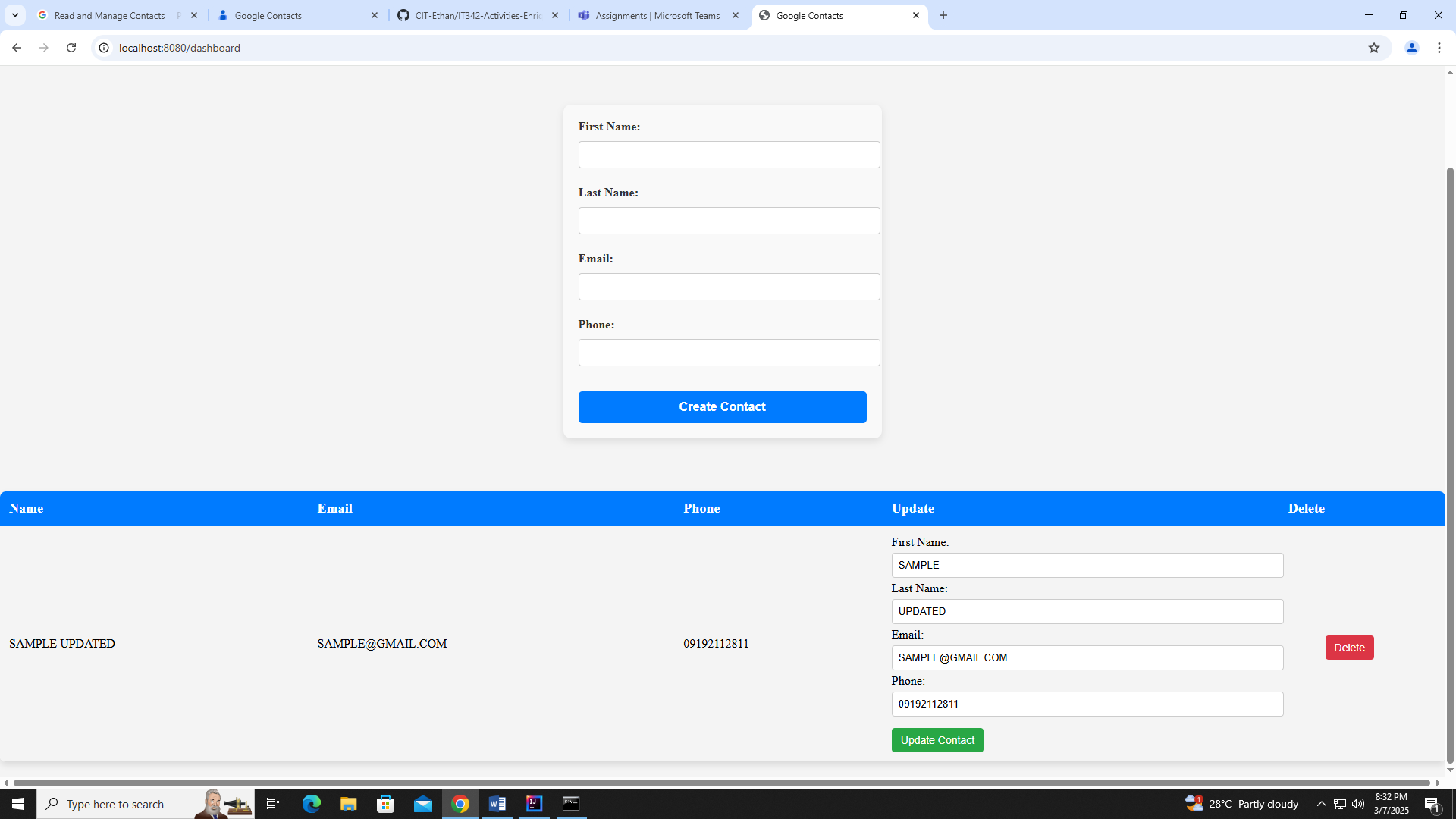
CREATE CONTACT



READ CONTACT LIST



UPDATE

After Update

DELETE

