**User signs up through Google OAuth**

**The user visits your website and chooses to sign up or log in through Google. The frontend makes a request to the Google OAuth API and the user is redirected to the Google sign-in page. After the user successfully signs in and authorizes your app, Google will redirect the user back to your website with an authorization code.**

**This authorization code is then sent to your backend server, which exchanges it for an access token by making a POST request to the Google OAuth API. Once the backend server has the access token, it can use it to access the user's Google profile information.**

**The backend server then checks the Users table in your database to see if a user with the given Google profile ID already exists. If not, it creates a new user record with the Google profile ID and email. It also stores the access token, refresh token, expiry time, and token type in the OAuthTokens table.**

**This whole process is the same for Facebook OAuth, but using the Facebook APIs and user's Facebook profile ID instead.**

**User logs in through Google OAuth**

**When the user logs in through Google, the process is largely the same as signing up. The frontend sends the authorization code to the backend server, which exchanges it for an access token and gets the user's profile information.**

**The server then checks the Users table for a user with the given Google profile ID. If it finds one, it updates the OAuthTokens table with the new access token, refresh token, expiry time, and token type. If it doesn't find a user with the given profile ID, it treats it as a new signup and creates a new user.**

**User starts a session**

**When a user logs in, the backend server creates a new session. It generates a session ID, which it sends to the frontend. The frontend stores this session ID in a cookie or local storage. The backend also creates a new record in the Sessions table with the session ID and the user's ID.**

**Each time the frontend makes a request to the backend, it includes the session ID. This allows the backend to authenticate the user and associate the request with the correct user.**

**User generates a new email**

**When the user fills out the form and clicks the Generate button, the frontend sends a POST request to the backend with the form data. The backend server handles this request by generating an email and returning it to the frontend.**

**The backend also creates a new record in the Requests table with the request data and the user's ID. And it creates a new record in the EmailsGenerated table with the email content, the time it was generated, and the user's ID.**

**User sends an email**

**If the user decides to send the generated email, the frontend sends another request to the backend, this time with the email content and the recipient's email address. The backend sends the email and updates the EmailsGenerated table with the time the email was sent.**