🔹 1. gmail\_client.py

This file is a wrapper around Gmail API for authentication and fetching emails.

Functions & Their Purpose

• \_\_init\_\_(self, credentials=None) → Initializes the Gmail client with credentials if available.

• \_build\_service(self) → Builds the Gmail API service object using credentials.

• get\_auth\_url(self) → Generates the Google OAuth 2.0 authorization URL where the user gives permission.

• get\_credentials\_from\_code(self, code) → Exchanges the OAuth code (from callback) for actual credentials.

• set\_credentials(self, credentials\_json) → Accepts stored JSON credentials and initializes the Gmail client.

• get\_recent\_emails(self, max\_results=10) → Fetches a list of recent emails from the Gmail inbox (with metadata like From, Subject, Date, Snippet).

• get\_email\_content(self, message\_id) → Gets full email content (including sender, recipient, subject, body).

• \_get\_email\_body(self, message) → Extracts plain-text or HTML body from email.

• \_decode\_body(self, data) → Decodes base64-encoded Gmail message body into text.

✅ In short: This file handles Gmail login + fetches emails.

🔹 2. main.py

This is your FastAPI backend entry point.

It connects Gmail + LLaMA3 (via MCPClient) + exposes REST APIs.

Workflow

• Sets up FastAPI app with CORS enabled (so frontend can call it).

• Initializes:

o MCPClient → For translation (Marathi).

o GmailClient → For Gmail API integration.

• Maintains a dictionary user\_credentials to store authenticated Gmail sessions.

Endpoints

• POST /translate-to-marathi

→ Translates any given text into Marathi using LLaMA3 (Ollama).

• GET /gmail/auth-url

→ Returns Google OAuth 2.0 authorization link.

• GET /auth/callback

→ Handles Google redirect, exchanges code for credentials, stores them.

• GET /gmail/check-auth

→ Checks if Gmail user is authenticated.

• GET /gmail/recent-emails

→ Fetches recent emails from authenticated user's inbox.

• POST /gmail/email-content

→ Gets specific email content by message ID.

• POST /gmail/translate-email

→ Gets email content + translates its body to Marathi.

✅ In short: This file is the backend server with all APIs exposed.

🔹 3. mcp\_client.py

This file is a helper client for Ollama (LLaMA3 model) to perform translation.

Functions

• \_\_init\_\_(self, temperature=0) → Initializes Ollama API details.

• translate\_to\_marathi(self, text) → Sends text + translation prompt to LLaMA3 model and returns Marathi output.

✅ In short: This file connects to local LLaMA3 running in Ollama for translations.

🔄 Overall Project Workflow

• User clicks “Connect Gmail” (Frontend calls → /gmail/auth-url).

• Backend (GmailClient.get\_auth\_url) → returns Google OAuth link.

• User authenticates with Google → Google redirects to /auth/callback.

• Backend exchanges code → tokens and stores credentials in user\_credentials.

• Now user can:

• Call /gmail/recent-emails → Get list of recent emails.

• Call /gmail/email-content → Get details of a specific email.

• Call /gmail/translate-email → Get full email content + Marathi translation.

• Translation is done by MCPClient which sends prompt to Ollama (LLaMA3).

