### **Step 1: File Handling and Email Processing**

* 1) File Picker Implementation:
  + Implement a file picker in your application to allow users to select and upload files.
* 2) Automated File Downloading:
  + Set up a script to automatically download files from the GCM.navigator URL on the first of each month.
* 3) File Export and Save:
  + Once the file is downloaded, export and save it to a specified file location (awaiting your confirmation on this location).
* 4) Pulling Legal Markup and SBSDR Word Document:
  + Create a script to pull the legal markup from the specified file location.
  + Similarly, pull the SBSDR Word document from the same file location.
* 5) Email Processing:
  + Implement an email automation system to process 10 different types of emails:
    - 6 emails with legal markup and GCM files.
    - 2 emails with only SBSDR files.
    - 2 emails without attachments.

6) Set up a reminder system to send follow-up emails if no response is received.

* 7) Forex Rates Update:
  + Implement a system to update Forex rates, possibly via web scraping from a designated website.

### **Step 2: Email Template Creation and Dynamic Content**

* Build Email Templates:
* 1) Create 10 different email templates for retrieving reports. These templates should have slight differences as per your requirements.
* 2) Dynamic Date Insertion:
  + Implement a feature to dynamically update the date in each email to reflect a 1-year period ending in the previous month.
* 3) Attachment Handling in Emails:
  + Configure the system to attach the required files to the emails:
    - 6 emails with 2 types of attachments (file from website and legal markup).
    - 2 emails with SBSDR attachments.
    - 2 emails without attachments.
* 4) Dynamic Submission Date:
  + Update the submission date in each email to be 3 business days after the email is sent.
* 5) Automated Reminder Emails:
  + Set up an automated system to send reminder emails if there is no response within 2 days, and again on the 3rd day.

### **Implementation Notes:**

* Programming Language and Tools: Consider using a language like Python or Node.js for this project, as they offer robust libraries for web scraping, file handling, and email automation.
* Email API: Utilize an email service API like SendGrid, Mailchimp, or SMTP libraries for sending and receiving emails.
* Web Scraping: If needed, use web scraping libraries (like BeautifulSoup in Python or Cheerio in Node.js) for Forex rates updates.
* Secure Storage: Ensure secure storage and handling of files, especially if they contain sensitive information.
* Testing and Validation: Thoroughly test each component to ensure it functions correctly and handles edge cases, such as missing data or server downtime.

This project requires careful planning and implementation, especially considering the handling of potentially sensitive data and the need for precise timing

in automated tasks. Make sure to adhere to best practices in coding, security, and data protection throughout the development process.