

# MailGPT: A GPT-Integrated Email Service for Dynamic Content Generation

Yosep Kim<sup>1</sup>, Dohyun Bu<sup>2</sup>, Sumin Lim<sup>3</sup>, Miseo Joung<sup>4</sup>,

October 5, 2023

<sup>1,2,3,4</sup>Sungkyunkwan University  
<https://mailgpt.intueri.cloud>

## Abstract

In the digital communication era, email remains a pivotal tool for professional interactions. Despite its prevalence, composing a proficient email that encapsulates clarity, conciseness, and the appropriate tone can be complex. Miscommunications in such contexts can inadvertently lead to undesired consequences. MailGPT introduces a novel solution by integrating the GPT API to facilitate dynamic email content creation suited to user intent. This system is technologically underpinned by React for frontend design, Docker for deployment, and SQLite for data management, establishing it as a potential game-changer in digital communication. Its architecture is modular, ensuring adaptability, and incorporates a user-friendly workflow. With the collaborative expertise of team members like Yosep Kim, Dohyun Bu, Sumin Lim, and Miseo Joung, MailGPT seeks to redefine the realm of email drafting.

## 1 Introduction

In the current digital age, email remains a primary medium for professional communication. Yet, for many, crafting an effective email—balancing clarity, conciseness, and the desired tone—can be a daunting task. The challenges of email writing are compounded when one has to ensure consistency and professionalism across myriad scenarios, from addressing a superior or a client to communicating with a colleague.

### 1.1 Motivation

The essence of these challenges stems from the inability of individuals to quickly adapt their writing styles to different contexts. A miscommunication or a misinterpreted tone can have unintended consequences, affecting relationships and business outcomes. Therefore, it's imperative to strike the right balance, ensuring that the content of the email aligns perfectly with its intent.

## 1.2 Solution: MailGPT

Enter MailGPT: a revolutionary approach to email writing. This service integrates the powerful language capabilities of the GPT API, offering dynamic email content generation tailored to the user's requirements. By simply specifying the target and mood of the email, users are provided with content that matches their intent, eliminating the common pain points of email drafting.

## 1.3 Technical Backbone

Underpinning MailGPT is a robust technological framework. It leverages the capabilities of React for frontend development, ensuring a responsive and intuitive user interface. Docker facilitates easy deployment, ensuring that the application remains scalable and resilient. The data backbone of the application, including email drafts and user preferences, is managed through SQLite, a versatile and efficient database solution. The fusion of these technologies, coupled with the intelligence of the GPT API, positions MailGPT as a forerunner in transforming digital communication.

# 2 System Architecture

The system architecture of our application is robust and designed to handle the modern demands of real-time processing. Built on a modular structure, it offers scalability, maintainability, and ease of enhancements.

## 2.1 Components

Our system is divided into three core components, each dedicated to performing its task efficiently:

- **Frontend:** Developed using React, a popular React framework known for its server-side rendering capabilities. It ensures a seamless user experience, catering to a responsive design that is adaptable across devices of varying sizes.
- **Backend:** Developed using Django, a plug to play framework for building web applications. This component is responsible for the system's core logic, including its interaction with SQLite and the GPT API. It plays a pivotal role in data processing and content generation.
- **Database:** We've opted for SQLite as our primary data store.

## 2.2 Workflow

The workflow is straightforward and user-centric. Upon accessing the platform, users are prompted to provide the target email and their desired mood or tone for the email content. This leads to a sequence of operations:

The frontend, harnessing the capabilities of React, utilizes a predefined template to generate a unique prompt based on the user's input. This prompt is securely stored in SQLite for future reference. Following this, the backend, powered by Django, dispatches the

prompt to the GPT API. The GPT API, using its advanced natural language processing capabilities, crafts an appropriate email content tailored to the user's requirements. This content is returned in Markdown format, allowing users to view a well-structured preview.

If the user is satisfied with the generated content, a confirmation is obtained. Subsequently, the content is sent to the designated target email address using SMTP, ensuring timely and reliable delivery.

## 3 Technical Details

### 3.1 Frontend

The frontend, developed using the sophisticated React framework, serves as the gateway to our application. It's the first point of contact for users, offering them an intuitive interface to input their requirements. Built with a responsive design, it ensures accessibility across a myriad of devices, from desktops to smartphones. It is designed to be both visually appealing and functional, enhancing user engagement and satisfaction.

### 3.2 Backend

At the heart of our application lies the backend, which orchestrates a symphony of operations:

1. **Interaction with SQLite:** Even though SQLite serves as our primary database, we leverage SQLite for certain relational data requirements. It provides a robust and efficient platform for data storage and retrieval.
2. **Sending the prompt to the GPT API:** The core functionality. Based on user inputs, the system creates a custom prompt. This is dispatched to the GPT API, awaiting its magic of content generation.
3. **Receiving and returning the generated content to the frontend:** Once the GPT API generates the content, it's the backend's responsibility to receive it, process it if needed, and relay it to the frontend. This ensures that users receive a swift response, enhancing their overall experience.

### 3.3 Data Flow Diagrams

The intricate interplay between the frontend and backend of MailGPT is aptly captured using detailed data flow diagrams. These diagrams, while complex in their representation, serve as a beacon for understanding the delicate choreography of data exchange and processing that ensues within our system. The multifarious activities and communication pathways, showcasing the convergence of various system modules, are intricately laid out in these diagrams.

Our decision to utilize such expressive visuals stems from the quintessential need for clear and precise elucidation of the interactions between different system components. For developers, stakeholders, and even advanced users, these diagrams provide invaluable

insights into the internal workings of our platform. They unravel the layered intricacies and underscore the robustness of our application’s design.

Furthermore, while the user interface presents a polished and seamless experience to the end-users, it’s the underlying data flows and transactions, meticulously visualized in our diagrams, that fortify the application’s reliability and performance. Hence, these diagrams become an instrumental tool, not just for design and development, but also for troubleshooting, optimization, and further enhancements to our cutting-edge platform.

### 3.3.1 GPT API Data Flow

As depicted in Figure 1, the GPT API data flow underpins MailGPT’s operation, demonstrating a meticulously orchestrated interaction among various components.

Upon a user’s request, the system channels the call from the client terminal to our EC2 instance, initially interfacing with the Nginx for efficient load balancing. The request subsequently traverses the React framework for preliminary processing and then to Django, an ASGI framework ensuring rapid processing and data integrity. At the crux of this sequence, the GPT API server, fortified with advanced algorithms, processes and contextualizes the request, transforming it into coherent outputs.

The processed data, rich with insights, retraces its steps, passing through Django and React for final optimization before Nginx ensures its prompt return to the user. This streamlined flow epitomizes efficiency and reliability in MailGPT’s infrastructure.

**GPT API Data Flow**

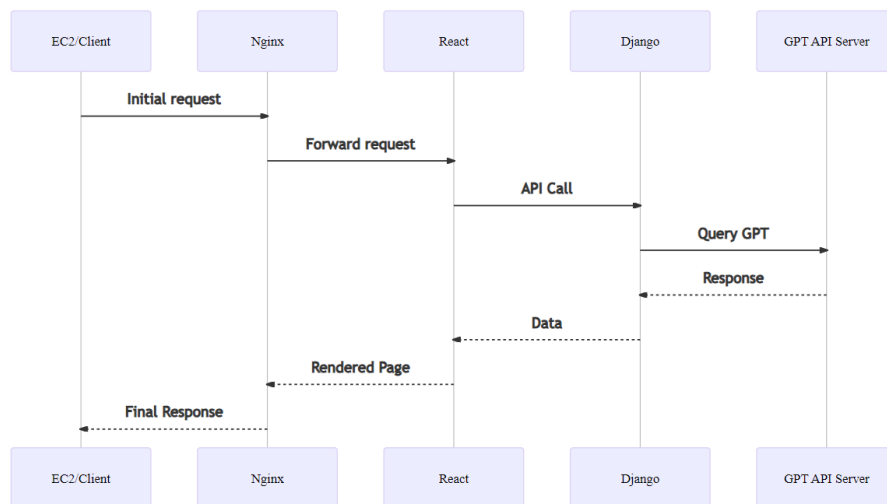


Figure 1: Data flow diagram illustrating the GPT API communication process

### 3.3.2 smtplib Data Flow

The symphony of our email notification system finds its muse in the data flow depicted in Figure 2. Like a familiar tune, the journey originates with the client’s sincere request. This beacon of communication is then acknowledged by the frontend, initiating a series of orchestrated movements. The backend, acting as the maestro, liaises with the email

server, crafting a melodious email notification. As the server resonates with the acknowledgment of successful dispatch, the backend, not missing a beat, alerts the frontend. The finale witnesses the frontend conveying this melodious success back to the awaiting client.

**smtplib Data Flow**

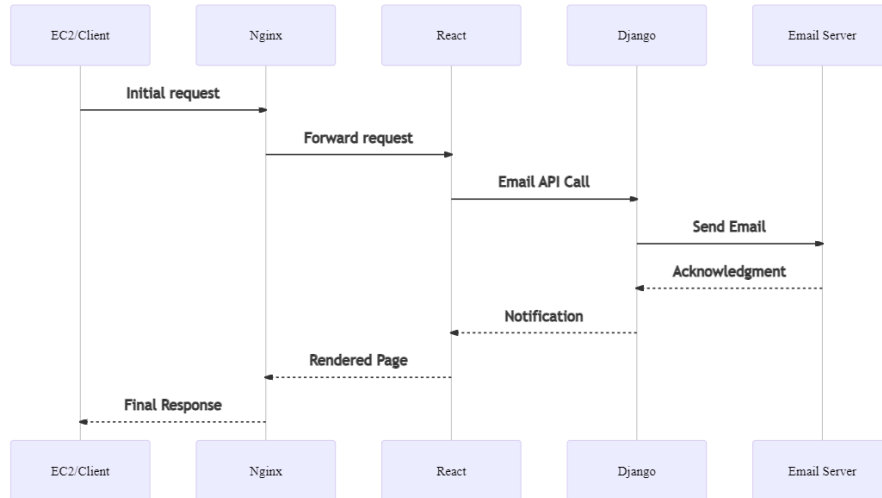


Figure 2: Data flow diagram illustrating the smtplib communication process

### 3.4 Conclusion

Our platform’s technical tapestry is woven with precision, where every thread represents a service or technology that contributes to the grand design. By employing avant-garde tools and technologies and ensuring a harmonious flow of data, we present to our users a masterpiece of efficiency and reliability. From the warm embrace of the frontend to the rhythmic cadence of the backend, every facet of our application has been chiseled to perfection, promising an experience that is both memorable and unparalleled.

### 3.5 Detailed Database Design

Our database comprises three core tables: *LOGIN*, *HISTORY*, and *SETTINGS*. Their inter-relationships and attributes are illustrated in the ER Diagram (Figure 3).

#### 3.5.1 LOGIN Table

The *LOGIN* table stores user login details. Key attributes include:

- **ID:** A unique identifier for each user.
- **Username:** The login name used by the user.
- **Password:** The encrypted password.
- **LastLogin:** Timestamp of the user’s last login.

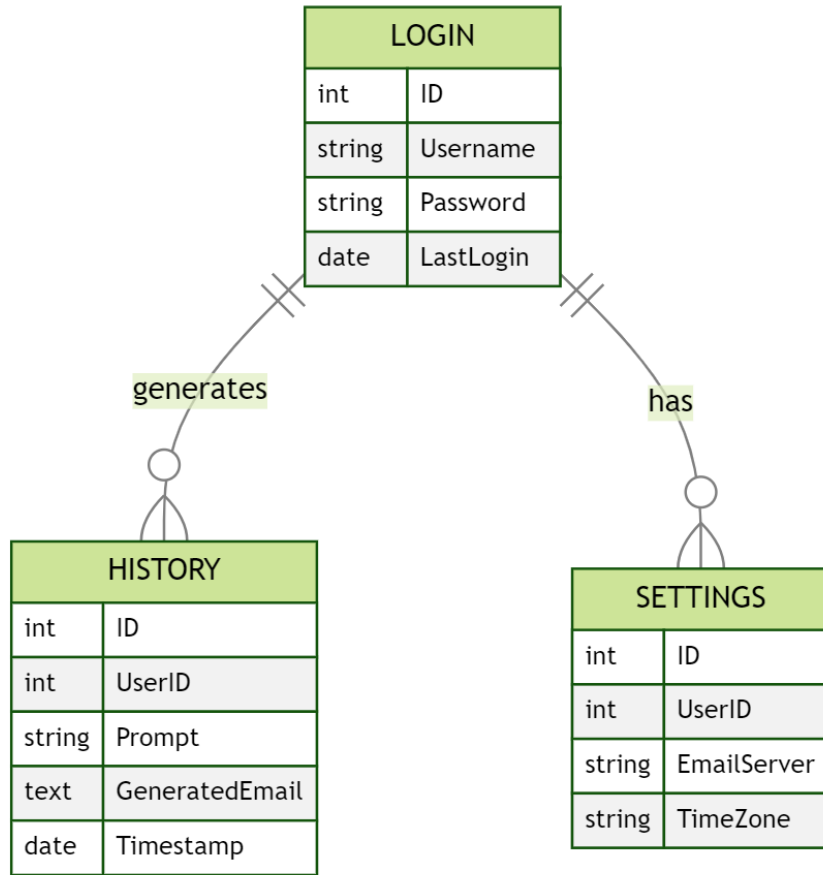


Figure 3: Entity-Relationship Diagram for the MailGPT Database

### 3.5.2 HISTORY Table

The *HISTORY* table keeps a record of all email prompts and the generated content based on them. Key attributes include:

- **ID:** A unique identifier for each record.
- **UserID:** Foreign key relating back to the *LOGIN* table.
- **Prompt:** The initial email prompt provided by the user.
- **GeneratedEmail:** The content produced by the GPT API.
- **Timestamp:** The exact time the email content was generated.

### 3.5.3 SETTINGS Table

The *SETTINGS* table holds configuration data for each user. Key attributes include:

- **ID:** A unique identifier for each settings record.
- **UserID:** Foreign key relating back to the *LOGIN* table.
- **EmailServer:** The user’s email server details.

- **TimeZone:** The user’s preferred timezone.

The relationships indicate that a user in the *LOGIN* table can have multiple entries in both the *HISTORY* and *SETTINGS* tables. This signifies that each user can generate multiple email contents and have multiple settings configurations.

## 4 Role and Time Line

### 4.1 Time Line

The Agile Scrum methodology’s Gantt chart showcases the progression of the project over three sprints, spanning three months. Tasks are divided among these sprints to ensure efficient and timely completion.

Task	Sprint 1	Sprint 2	Sprint 3
Time	Oct 1 - Oct 20	Oct 21 - Nov 10	Nov 11 - Dec 1
Frontend Base Setup	X		
Backend DB and API Setup	X		
Set Project Milestones	X		
Frontend-Backend Integration		X	
Backend Advanced Features		X	
Re-evaluate Milestones		X	
Finalize UI/UX			X
Backend Deployment			X
Project Closure			X

Table 1: Gantt Chart for Agile Scrum Project Timeline

### 4.2 Team Roles

In the journey of creating the MailGPT application, a team of adept professionals was assembled, each with specific roles and responsibilities. The table below delineates the key team members and their associated tasks:

Team Member	Role
Yosep Kim	Project Manager, Documentation, Backend (Cloud Infra), AI algorithm
Sumin Lim	Frontend (React, UX/UI)
Dohyun Bu	Backend (Django, GPT, DB), AI algorithm
Miseo Joung	Frontend (React, UX/UI)

Table 2: Roles and Responsibilities of the Team Members

Yosep Kim leads the project, overseeing management, backend infrastructure, and AI advancements. Dohyun Bu handles backend tasks, including APIs and AI. Sumin Lim manages the frontend using React and contributes to the backend. Miseo Joung specializes in UX/UI and backend APIs. Their combined expertise drives the MailGPT project.

## Conclusion

In the age of rapid digital communication, the efficiency and clarity of our electronic correspondences play a pivotal role in both professional and personal settings. **MailGPT**'s integration with the GPT API promises a transformative approach to email composition.

The traditional method of email writing often involves considerable time and cognitive effort. This can lead to unnecessary delays, potential misunderstandings, or even missed opportunities. By intelligently generating content based on user prompts, **MailGPT** not only simplifies this process but also offers a consistently high-quality output.

Moreover, as the GPT API continues to evolve and improve, the potential for **MailGPT**'s features will only grow. There's a possibility for broader applications such as customizing email styles based on recipient preferences, generating summaries of long emails for quick reading, and even predicting user intent to further streamline the email drafting process.

Additionally, for non-native speakers or those less confident in their writing abilities, **MailGPT** can act as an invaluable tool, ensuring clarity and coherence in their emails. This democratizes the process, allowing more individuals to communicate effectively, regardless of their writing expertise.

In conclusion, **MailGPT**'s innovative approach, driven by the power of the GPT API, promises not just an evolution in email composition, but a potential revolution in digital communication as a whole. As the digital age progresses, tools like **MailGPT** will become indispensable in bridging gaps, enhancing efficiency, and making sure that communication remains at the heart of all our interactions.

## References

- [1] A. Silberschatz, H. F. Korth, and S. Sudarshan, *Database system concepts*.
- [2] S. Chacon and B. Straub, "Pro Git," Sep. 2023.
- [3] Elvis Saravia, "Prompt Engineering A lecture by DAIR.AI," 2023.
- [4] I. John, "The Art of Asking ChatGPT for High-Quality Answers A Complete Guide to Prompt Engineering Techniques," 2023.
- [5] Stefan. Kottwitz, *LaTeX beginner's guide*. Packt, 2011.