

# Flask Application with ChatGPT Integration

Dynamic Comments, Summaries, and  
Email Drafting

# Introduction

- This Flask application integrates with OpenAI's ChatGPT API to:
  - - Generate comments and feedback for products.
  - - Create subject lines and summaries dynamically.
  - - Analyze sentiment based on descriptions.
  - - Draft personalized email content.

# Features and Workflow

- 1. User provides input via a web form.
- 2. Flask routes handle API requests and responses.
- 3. ChatGPT generates:
  - - Comments
  - - Subject lines and summaries
  - - Sentiment analysis
  - - Email drafts
- 4. Results are displayed to the user in the web

# Core Code Structure

- 1. ``ask_chatgpt()``: Central function for API requests.
- 2. ``/process`` route:
  - - Receives JSON payload.
  - - Calls ChatGPT API using ``ask_chatgpt()``.
  - - Returns responses as JSON.
- 3. Frontend templates render results dynamically.

# Challenges and Solutions

- Challenges:
  - - Handling API errors or invalid responses.
  - - Ensuring fast response times.
- Solutions:
  - - Added error handling in ``ask_chatgpt()``.
  - - Optimized JSON parsing and data validation.

# Potential Enhancements

- - Add user authentication for secure access.
- - Store generated data in a database for future use.
- - Implement support for multiple languages.
- - Extend functionality for customized prompts.