



## Technical Assignment: AI Email Summarizer Workflow

### Objective

This assignment evaluates your ability to design a small full-stack workflow automation project using Replit and OpenAI APIs. You'll integrate AI-powered summarization, data persistence, and a simple frontend dashboard.

### Scenario

You are tasked with creating a workflow tool that helps summarize incoming emails automatically. The workflow should load a mock list of emails, summarize and categorize them using OpenAI APIs, and store the results in a Neon PostgreSQL database using Drizzle ORM. A simple React dashboard should display the summarized data.

### Requirements

- 1 Load a mock list of emails (JSON or CSV) containing sender, subject, and body.
- 2 For each email, call the OpenAI API to generate a concise 2–3 sentence summary and assign a category (e.g., 'Meeting', 'Invoice', 'Support Request').
- 3 Store the summarized output in a Neon PostgreSQL database using Drizzle ORM.
- 4 Build a lightweight React dashboard to view summarized emails, filterable by category.
- 5 Allow the user to trigger 're-summarize' or 'delete' actions for individual entries.

### Tech Stack & Tools

- Frontend: React 18 + TypeScript (basic UI with list or table view)
- Backend: Node.js (Fastify or Express) + Drizzle ORM + Neon DB
- Automation Environment: Replit/Cursor/Copilot
- AI: OpenAI API (See API key below)

sk-proj-GD-lbJWWmktrmfBYj0zQBp6kE0vSs29cPlbxLaiAYRulcNFLoxcP23ZhExRUiL7A18pmTr26CT3BlkFJniiC8wUfnyyub8HClrKtXdAFtpWH9bQ9uI\_EEI3q0NiJ1zeiOcgODFEzdZCh0eRk2QChmcwZUA

### Deliverables

- A working Replit project link (or URL of hosted demo). Include credentials to access the demo.
- Source code in GitHub/GitLab/Bitbucket or Replit repository.
- A short README.md explaining setup, API configuration, and design decisions.

### Evaluation Focus

- OpenAI API usage: prompt design, batching, error handling.
- Drizzle/Prisma ORM + Neon/Supabase/Firebase DB integration: schema design and data handling.
- TypeScript correctness and modular code structure.
- Automation workflow logic and data flow.
- Frontend design clarity and React best practices.
- Creativity in applying AI and automation concepts.

### Bonus Ideas (Optional Extensions)

- Implement a 'keyword extraction' step using another OpenAI API call.

- Add a REST endpoint (/summaries/export) to download results as CSV.
- Extract item–price combinations from a PDF invoice attachment if an email includes one (simulate or use a sample PDF).

■ **Expected Completion Time:** 2–3 hours

Ensure your project runs smoothly in Replit or locally with clear setup instructions. Show creativity, structure, and a good grasp of the AI workflow pipeline.