Vibe coding experiment on a technology migration project:

Start Date: 03/17/2025

Start Time: 10:15 AM

**ChatGpt Prompt:**

There is a standalone java application with dependency on java mail and Apache velocity that runs as a batch job, reads message details to be sent from database, using this data from table and velocity templates with static content and placeholders for key data like emailFrom, emailTo, Subject, caseNumber, and caseTitle, it sends an email to the user. I noticed that this application is part of a large monolithic application with a dozen more modules. I would like to replace this email system with JavaScript based solution. Please recommend a technology roadmap for this migration.

**ChatGpt Response:**

You have a few JavaScript options depending on environment and preference:

**Platform** Node.js + Express

**When to Choose** You need server-side batch processing

**Stack**  Node.js, Express.js, Nodemailer

**ChatGpt Prompt:**

I want to move email message database to Kafka

**ChatGpt Response:**

Great decision! Moving email messages from a relational database to **Kafka** decouples your system, improves scalability, and aligns with modern **event-driven architecture** principles.

Here’s a technology **roadmap** for migrating your **email message database to Kafka**, integrated with your new **JavaScript-based email service**.

**Old Flow:**

[Database Table] → [Java Batch Job] → [Velocity] → [JavaMail]

**New Flow with Kafka:**

[Producer System / Legacy App] → [Kafka Topic: email-messages] → [JavaScript Consumer Service] → [Template Engine] → [Nodemailer] → Email

**Summary of What to Add:**

| **Feature** | **New/Modified Files** | **Notes** |
| --- | --- | --- |
| Kafka client setup | /config/kafkaConfig.js | Broker URLs, group IDs |
| Kafka consumer logic | /kafka/consumer.js | Message listener, deserialization |
| Email service | /services/emailService.js | Template rendering + Nodemailer |
| App entry point | app.js | Initialize consumer, error handling |
| Logger | /utils/logger.js | Centralized logging |

10.50 AM

**Development in Windsurf IDE**

**Prompt:**

Start boilerplate or example code to get going with Nodemailer + Handlebars + Kafka Client + Kafka Consumer Service

**Outcome:**

Windsurf planned for 9 seconds, used 23 tokens, and created project structure with all the dependencies in 20 seconds. Also gave instructions and CLI commands to install dependencies.

**Prompt:**

There are two apache velocity templates with extension .vm, convert them to template understood by Nodemailer and add to directory src\templates

**Outcome:**

Created 2 .hbs files in and example.js file in 4 seconds.

After cleaning up .hbs templates and removing some message part that was suggested by windsurf but not originally present in velocity template, the project was ready to be shipped by 11.10 AM.