



Team Amazon

Michigan State University

Amazon Email Improvement Tool

Project Plan

Fall 2023

Amazon Contacts

Zach Arnold

DJ Martin

Rashmi Priya

Team Amazon

Nafisa Ahmed

Will Gamba

Annika Karlstrom

Jack Sleeman

Byzjon Speights

Michael Than

Table of Contents

- Table of Contents 2
- Executive Summary 3
- Functional Specifications..... 4
- Design Specifications 5
- Overview 5
 - Home Page 5
 - Email Analysis Page 6
 - Email Empathy Dropdown..... 7
 - Email Clarity Dropdown..... 8
 - Email Possible Duplicates Dropdown 8
 - Email Summary Dropdown 9
 - Email Templates Page 9
 - Email Templates Popup 10
 - Administrative Menu Page..... 11
- Technical Specifications 14
 - System Architecture 14
 - Overview..... 14
 - System Components 15
 - Development Environments/Languages 16
- Risk Analysis 18
 - Categorizing, Testing and Training Data 18
 - Data Storage 18
 - Project Expense 18
- Schedule..... 19

Executive Summary

Headquartered in Seattle, Washington, Amazon is one of the world's leading technology companies and part of the Big Five American Information Technology companies. The vast internet-based enterprise was founded in 1994 by Jeff Bezos originally as an online marketplace for books. Since then, Amazon has become an all-encompassing online retailer, web service provider, parent company to over 100 subsidiaries, and has taken the second spot on the Fortune 500 list. Amazon either sells products directly or acts as the middleman between their sellers and millions of global customers. Amazon's global marketplace hosts over 6.3 million sellers and caters to more than 150 million users, all while striving to ensure that the highest standards of customer obsession and earning trust are maintained. With such demand, Amazon is working to ensure such standards are met when sending millions of email communications.

To ensure the company's renowned standards are met, Amazon is exploring new ways to write and send better-quality emails by developing a review and analysis platform. Email communications provide a better user experience and promote Amazon's ideals of earning their customer's trust. To maintain their deep sense of customer loyalty, Amazon provides personalized email communications to sellers to inform and enhance their marketplace experience. On their current platform, content creators spend time and resources creating thorough communications that undergo a series of checks to confirm that the communication meets standards. Only after the email is reviewed can it be sent. Having a series of checks ensures that only high-quality, satisfactory emails that are both informational and meet Amazon standards are produced.

To support Amazon's goal of improving the quality of emails, our software will help Amazon analyze and check the multitude of emails written by content creators. By utilizing Machine Learning, the software will generate detailed reviews and confidence scores for each aspect being tested in the email. The analysis of each email will give feedback on improvements and summarize the content, making it easier for creators to find similar templates. Additionally, to increase efficiency, the software will determine if duplicates or similar templates exist and are ready for content use. A filter templates feature is also included to provide access to previously written emails without having to submit an email to the software. This process will allow creators to work quicker when sending email communications and enhance the user experience with quick, high-quality content.

Functional Specifications

Amazon is keen on producing the highest standards for not only their millions of customers, but their sellers as well. They are constantly working towards making changes in order to earn the trust of their sellers. Amazon does this through managing thousands of emails that need to be sent out daily. Taking care of all these emails, however, can be a lot of work. There are many Amazon teams along with copious amounts of code that go into creating high-quality emails, which comes with many setbacks. For example, the emails that get sent out to sellers can possibly be repetitive or struggle to deliver the right tone, as does any communication through text. Email creators always have the best intent when it comes to writing these emails, but it is a challenge to understand all the workarounds at the company and it is hard to receive feedback from everyone.

When a seller receives email communication from Amazon, it can shape their perspective on the company. In order for a seller to be satisfied, Amazon needs to carefully craft each and every email that gets sent to them. The goal of our project is to help Amazon improve the quality of the emails they send out to their sellers by providing a web application that provides instant feedback on how an email creator can improve their email. By making a user-friendly web application, Amazon employees will find the process of producing a well-written email much simpler and will be able to communicate their thoughts with their sellers in the best way.

Our program will evaluate new email content to compare to thousands of pre-existing Amazon email templates. Once the new email is analyzed, users will be taken to an analysis page that presents feedback on the email. This feedback includes the classification of the email according to its objective. This page will also include several scores to assess the clarity and empathy of the email, where the user can expand upon each category and find where exactly within their email they can improve in either section. The user will also be directed towards any possible duplicate emails that may exist in comparison to their own, as well as a short summary of their email for quick future reference. The email templates page provides a way for users to quickly access the existing database of Amazon email templates, as an alternative to producing their own new email. This project's objective is to provide detailed descriptions to Amazon content creators on how to develop the best quality email.

Design Specifications

Overview

The Email Improvement Tool is a web application that is designed to elevate automated emailing. It is exclusively designed for Amazon employees, providing many resources for those working with it. At its core, it's an email analysis tool, which categorizes emails, evaluates for empathy and clarity, provides a summary, and searches through other Amazon email templates for potential duplicates. This tool will also grant Amazon content creators access to the database of email templates. For higher-level management, an administrative menu page offers centralized control of the website and its features. The web application works as an optimization tool for Amazon communication, ensuring a positive and customer-centric experience.

Home Page

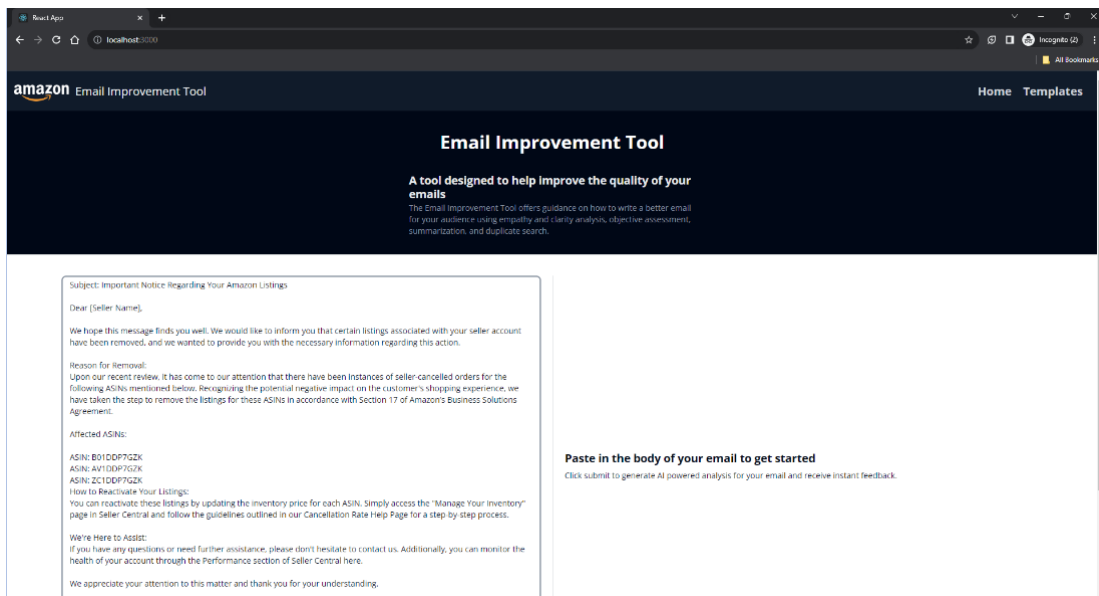


Figure 1 – Email Improvement Tool Home page, initial web page

Upon opening the Email Improvement Tool, users will be greeted by the Home page, as illustrated in Figure 1. At the top of this page, text will be displayed to explain the purpose and functionality of the application.

To initiate the analysis process, user input is required. A text box on the left side of the page allows users to conveniently input the email they wish to analyze. Once the user has entered the email text, the analysis process can commence by simply pressing the submit button.

Email Analysis Page

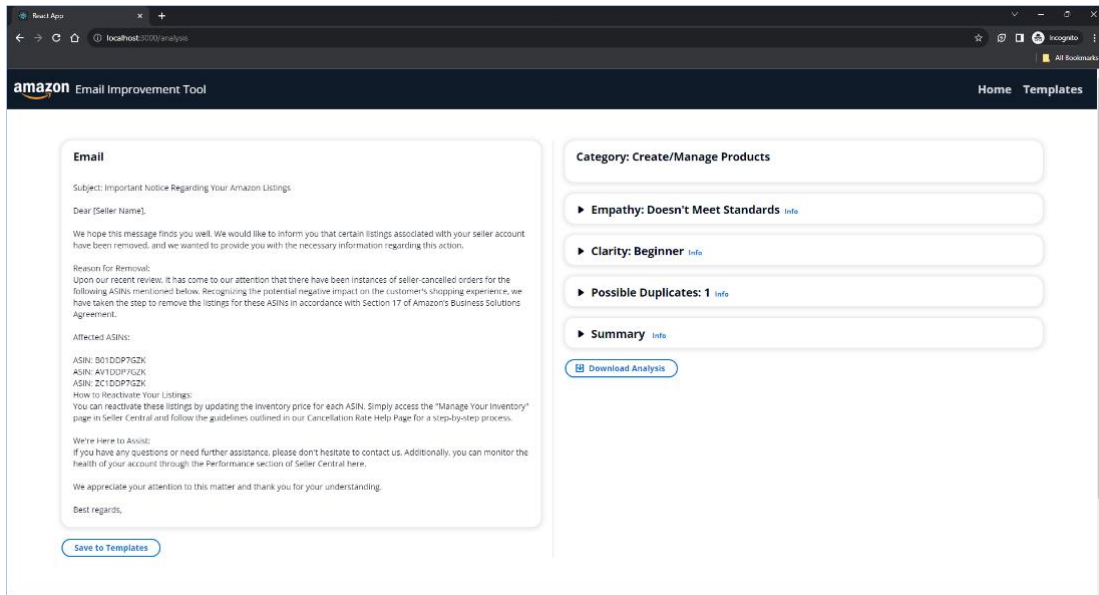


Figure 2 – Email Information Page, Analysis of the email

Upon the submission of the user's email, the system will redirect the user to the web page depicted in Figure 2. This page will present an analysis of the given email. The left-hand section of the page will be exclusively dedicated to the user's initially submitted email. The other half of the page will display the analysis of the email.

The analysis will include key information such as the category of the email within the Amazon system, the conveyed empathy in the email, the clarity of the communication, possible duplicates found within the database, and a concise summary of the email.

The design ensures a smooth and user-friendly experience, allowing users to effortlessly navigate through the analysis and take necessary actions such as saving the email to the database or downloading the detailed analysis report in a PDF format.

Email Empathy Dropdown

▼ **Empathy: Doesn't Meet Standards** [Info](#)

- The email subject line could be more clear and empathetic. Consider rephrasing to convey empathy and provide clarity on the purpose of the email.
- The opening paragraph comes across as abrupt. Consider rephrasing to establish rapport, express empathy, and provide context before mentioning the removal of listings.
- The reason for removal paragraph is vague. Provide specific examples of seller-cancelled orders to help the seller understand the issue.
- The affected ASINs paragraph only includes examples. List the actual ASINs affected to help the seller identify them.
- The call to action could be more clear and empathetic. Rephrase to clearly explain the steps the seller needs to take while expressing support.
- The closing could be more empathetic. Rephrase to reassure the seller and convey willingness to assist.

Subject: Assistance with Your Amazon Listings

Dear [Seller Name],

We hope all is well with you. We are reaching out to inform you that some of your listings were removed from Amazon recently. We understand this may be disappointing, and want to provide guidance on next steps so you can quickly get your listings reactivated.

Reason for Removal:
After reviewing your account, we noticed multiple seller-cancelled orders for ASIN B01234, ASIN A1234, and ASIN Z1234 over the past month. To ensure a positive customer experience, these listings were removed according to Section 17 of the Amazon Business Solutions Agreement.

Here is what you can do to reactivate your listings:
Please update the inventory price for each affected ASIN through the "Manage Your Inventory" page in Seller Central. Our Cancellation Rate Help Page has step-by-step instructions to guide you through this process.

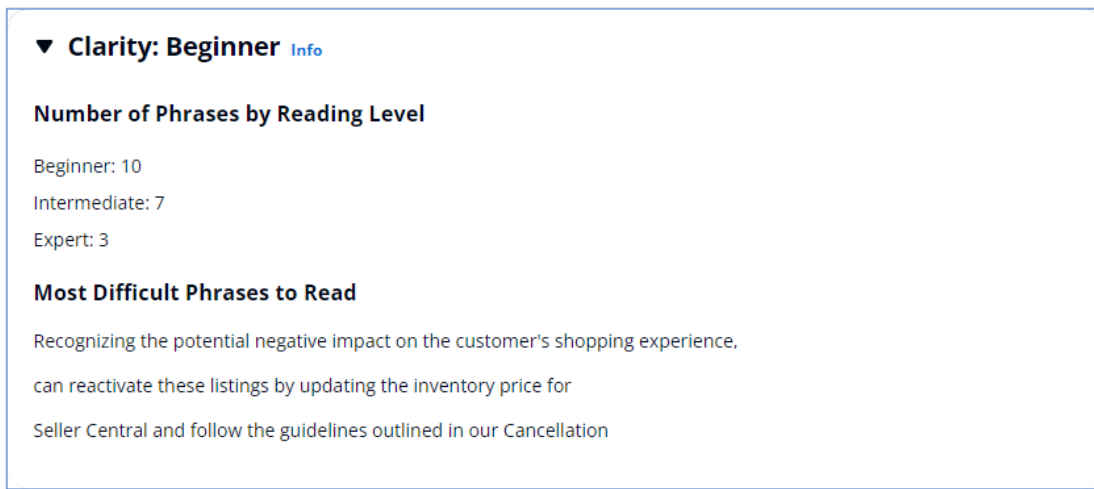
Please let us know if you have any other questions. We are here to help get your listings back up and running smoothly. Thank you for your cooperation and we look forward to assisting you.

Best regards,
[Sender Name]
Amazon

Figure 3 - Email Empathy Dropdown

The empathy dropdown will initially indicate if the email aligns with Amazon's standards. When the user clicks the dropdown, it will provide specific feedback on sentences that do not meet standards and suggest improvements. Additionally, users will be able to view a fully written improved version of the email for reference.

Email Clarity Dropdown



▼ **Clarity: Beginner** [Info](#)

Number of Phrases by Reading Level

Beginner: 10
Intermediate: 7
Expert: 3

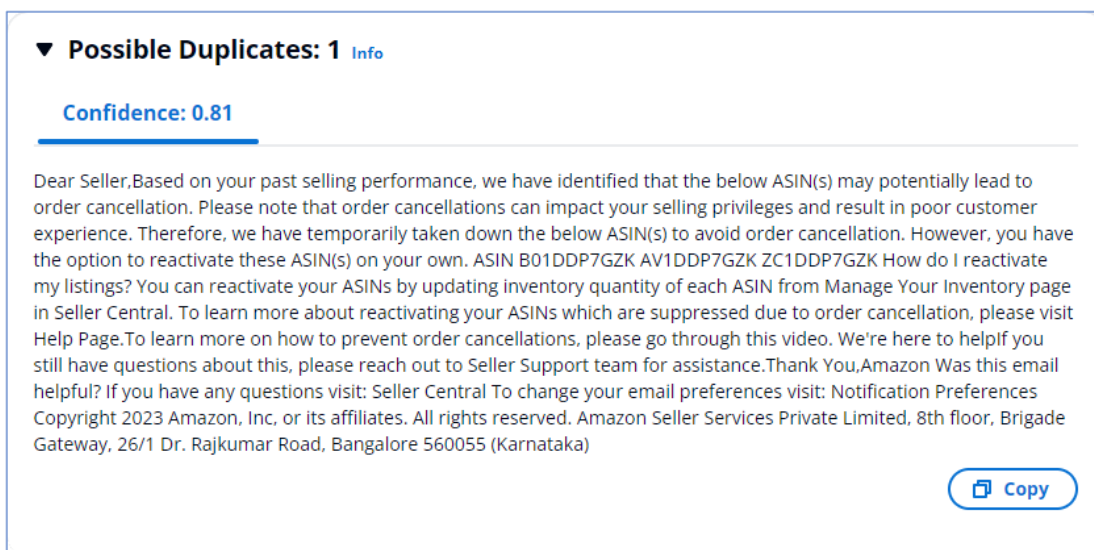
Most Difficult Phrases to Read

Recognizing the potential negative impact on the customer's shopping experience,
can reactivate these listings by updating the inventory price for
Seller Central and follow the guidelines outlined in our Cancellation

Figure 4 - Email Clarity Dropdown

The clarity dropdown will initially provide the user with the reading level of the email, between the three options, beginner, intermediate, or expert. When the user opens the dropdown, the user will be shown two sections: “Number of Phrases By Reading Level”, which displays the total count of phrases in each category (beginner, intermediate, and expert), and “Most Difficult Phrases to Read”, showing the most complex phrases within the email.

Email Possible Duplicates Dropdown



▼ **Possible Duplicates: 1** [Info](#)

Confidence: 0.81

Dear Seller, Based on your past selling performance, we have identified that the below ASIN(s) may potentially lead to order cancellation. Please note that order cancellations can impact your selling privileges and result in poor customer experience. Therefore, we have temporarily taken down the below ASIN(s) to avoid order cancellation. However, you have the option to reactivate these ASIN(s) on your own. ASIN B01DDP7GZK AV1DDP7GZK ZC1DDP7GZK How do I reactivate my listings? You can reactivate your ASINs by updating inventory quantity of each ASIN from Manage Your Inventory page in Seller Central. To learn more about reactivating your ASINs which are suppressed due to order cancellation, please visit Help Page. To learn more on how to prevent order cancellations, please go through this video. We're here to help if you still have questions about this, please reach out to Seller Support team for assistance. Thank You, Amazon Was this email helpful? If you have any questions visit: Seller Central To change your email preferences visit: Notification Preferences Copyright 2023 Amazon, Inc. or its affiliates. All rights reserved. Amazon Seller Services Private Limited, 8th floor, Brigade Gateway, 26/1 Dr. Rajkumar Road, Bangalore 560055 (Karnataka)

[Copy](#)

Figure 5 - Email Possible Duplicates Dropdown

The possible duplicates dropdown will display the total number of similar or possible duplicates found reflecting the input email. Upon opening the dropdown, it will showcase different emails based on confidence levels. Users will be able to click on a specific confidence level to reveal the body of the corresponding email. The user will also be able to copy the duplicate email with the copy button.

Email Summary Dropdown

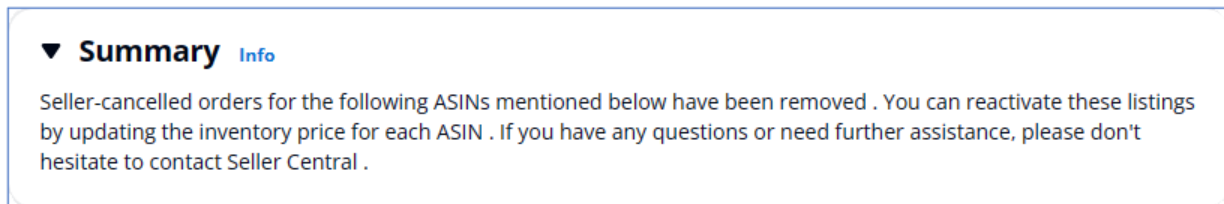


Figure 4 – Summary of Email Dropdown, summarizing the given email

When the summary dropdown is clicked, the user will be presented with a quick and concise summary of the email. The summary will be given to the user to view for easy comprehension of the written email.

Email Templates Page

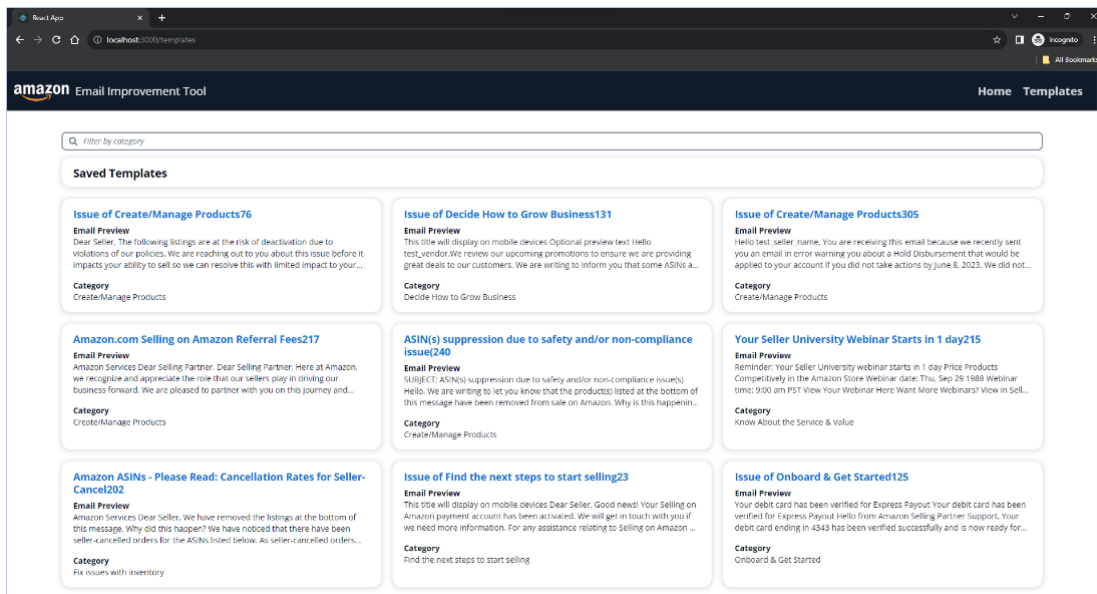


Figure 7 – Email Templates Page, showing all email templates in the database

The email templates page will present all stored email templates, each of which is labeled by the name of the email. Within each template is the body and its associated category. The navigation process is made simple with a filter by category feature, which allows the user to efficiently locate templates for their selected category.

Email Templates Popup

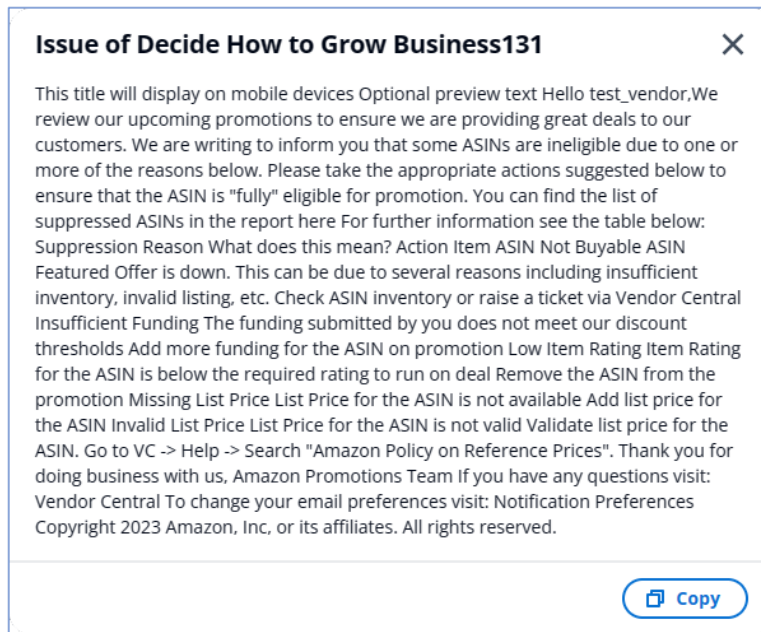


Figure 8 – Email Templates Popup

Upon clicking on one of the templates on the Email Templates page, shown in Figure 7, a small popup will appear. The popup is designed to give users a comprehensive view of the entire body of the template. As well as a way for the users to copy the template with the use of the copy button.

Administrative Menu Page

The Administrative Menu Page is designed exclusively for administrative purposes. This page will encompass an array of monitoring features and visualizations of website data, costs, and specific lambda function data. Additionally, it will host functionalities enabling direct access to the database, empowering administrators to seamlessly modify, delete, or append data within the database.

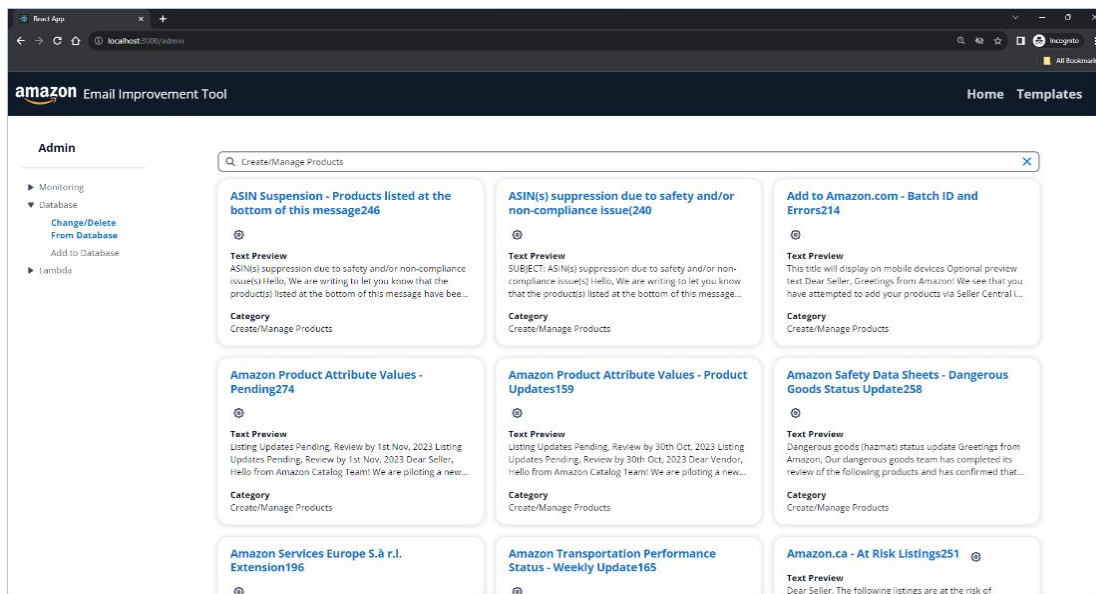


Figure 9 - Modifying Database, Administrative Menu page

A key feature within the administrative menu page will be the data manipulation section, where users can dynamically alter and remove information. Within this section, administrative users will have the capability to pressure all available templates, affording them the flexibility to modify various elements within each template. This includes the ability to change the name of the email, adjust the category, and edit the body of the email to suit their specific needs.

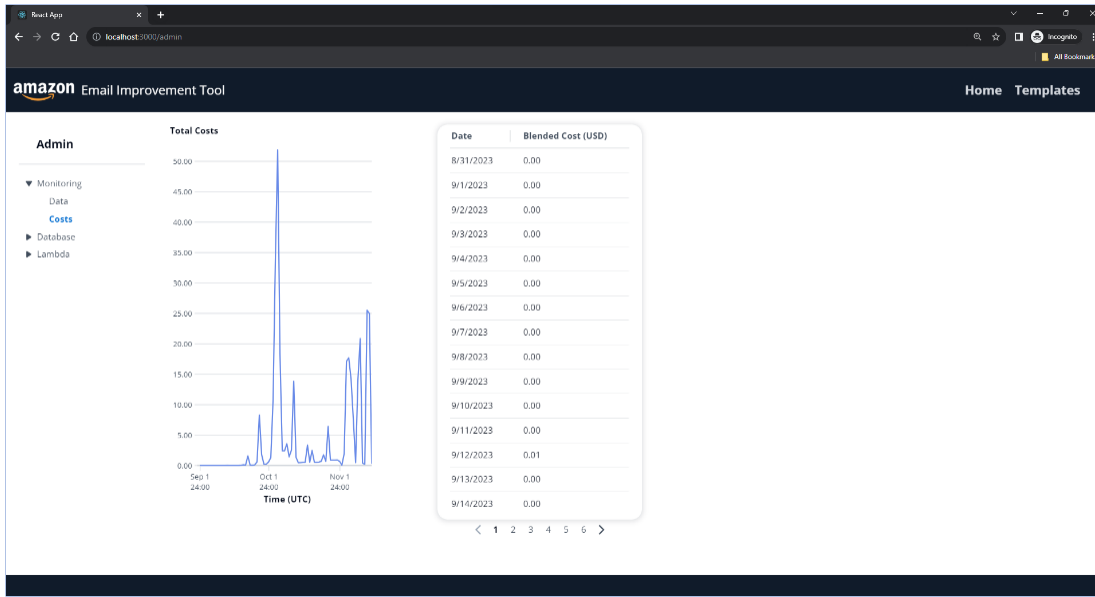


Figure 10 - Website Costs, Administrative Menu Page

An important addition to the Administrative Menu Page is the dedicated cost section. This feature will provide a comprehensive overview of the website’s total expenses, encompassing Machine Learning, API, DynamoDB, Endpoints, Lambda, and other associated AWS costs. The presentation will be both visual and detailed, presenting the costs through a line graph and corresponding table.

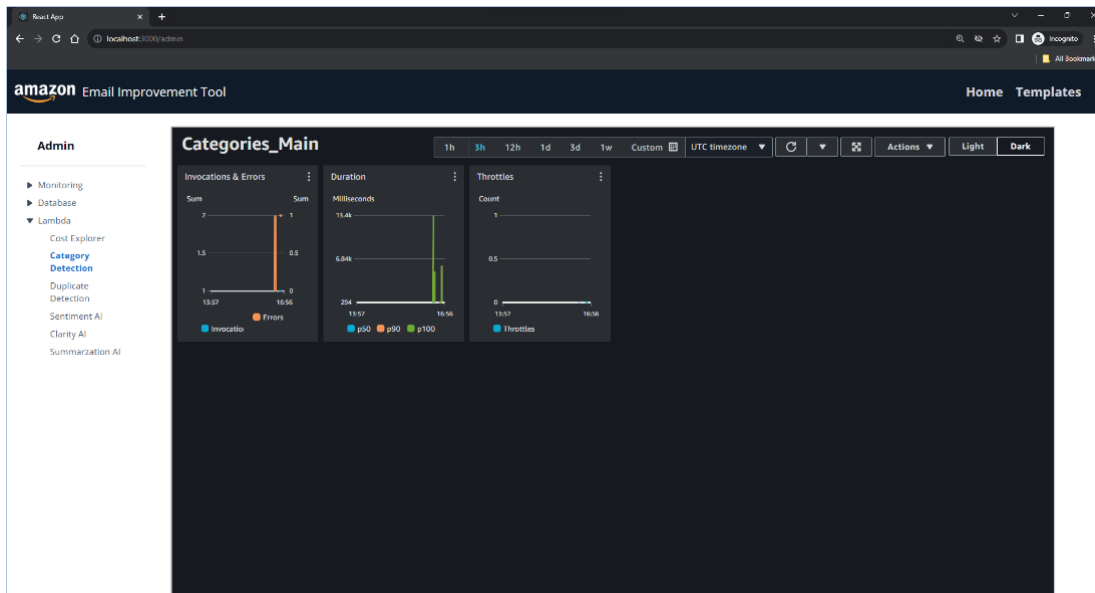


Figure 11 - Lambda Data, Administrative Menu Page

The graphs depicted above in Figure 9 are just a few among the several Lambda function graphs available for monitoring performance. This particular graph illustrates the duration of each Lambda call, highlighting instances where throttling may be necessary. Additionally, it provides insight into the invocation frequency of the function, showing moments when errors occur.

The Administrative Menu Page includes various supplementary features for administrative use. These functionalities cater to the nuanced needs of administrators, enhancing the overall management experience.

Technical Specifications

System Architecture

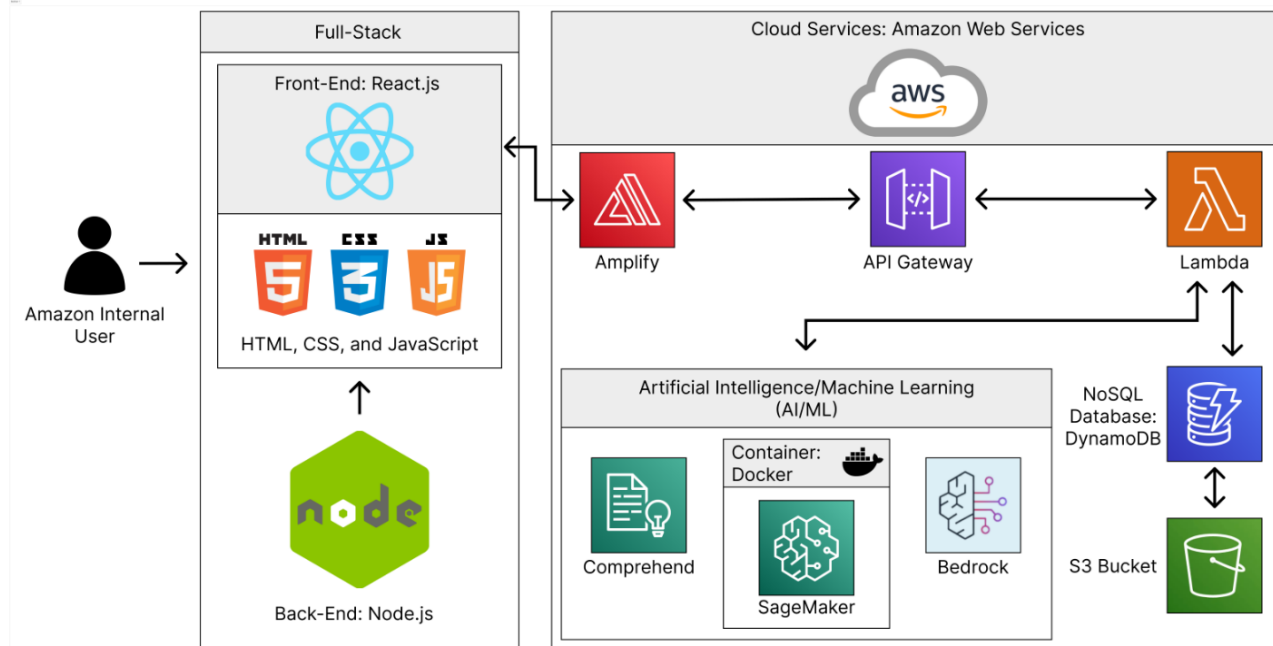


Figure 12 - System Architecture

Overview

The Email Improvement Tool is a React web application utilizing Amazon Web Services (AWS) to provide artificial intelligence and machine learning feedback to email templates.

Our application begins with the Node.js backend supporting the React front-end. Essentially, Node manages packages and helps build the React, so its build is ready for deployment. React is made up of HTML, CSS, and JavaScript, which will provide the interface the user is interacting directly with. The React and Node.js front-end will be relatively lightweight compared to the AWS suite of technologies leveraged in the tool.

Our front-end will be hosted on AWS Amplify; a service intended for application deployment. Amplify will generate a URL for our system, from which we will make calls to Amazon's API Gateway. This service will connect the many localized components of the email improvement tool to those resources associated with the cloud. The specific API Gateway will call its intended function in AWS Lambda, which will execute and return any relevant data.

The Lambda service provided by AWS will be especially critical, as it is key for communication with our artificial intelligence (AI), machine learning (ML), and database components.

Essentially, Lambda establishes the protocol desired for each of our API endpoints, connected via Gateway. Within said function, any queries will be made and standardized responses for the varying results will be established. The data will be returned through the API Gateway to the application.

If a Lambda function intends to query an AI or ML component of our platform it will be making a call to either Amazon SageMaker, Amazon Comprehend, or AWS Bedrock. If we are analyzing for clarity, duplicates, or category, Comprehend will be the service of choice, as it uses natural language processing (NLP). SageMaker, a more generalized AI engine is used to analyze an email and produce an associated summary. This model used a Docker Container. In the case of empathy analysis, AWS Bedrock is used to provide generative feedback. The results of the query will be returned to the user's instance of our application.

Lambda functions will also facilitate queries to our database, which will be hosted by DynamoDB. This database will be a NoSQL database, providing flexible and versatile possibilities for data storage, utilizing Amazon S3 buckets for email and analysis storage. This will be especially important as we support access to previous templates, in addition to checking our system for duplicate entries.

In summary, the email improvement tool will provide a React interface, deployed on AWS Amplify, that communicates through API Gateway, which in hand calls Lambda functions given the relevant data to DynamoDB, SageMaker, Comprehend, and Bedrock. DynamoDB supports our database, SageMaker works with summarization, Comprehend queries for clarity, duplicate and category analysis, and Bedrock hosts our empathy analysis model. The intended results will be returned to Lambda, as well as API Gateway, and displayed on our hosted React application.

System Components

AWS Amplify

AWS Amplify enables users to build, ship, and host full-stack applications on AWS, allowing for easy deployment and updating of our React web application.

API Gateway

AWS API Gateway is a fully managed service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs at any scale. The web application will use an API created for the Lambda functions to make requests.

Lambda

AWS Lambda is a serverless computing service that runs code in response to events on our application that seamlessly interacts with the other AWS services we're using.

Docker

Docker is an open platform for developing, shipping, and running applications. Docker enables separation between applications within infrastructure by creating containers. A container is a specific environment for a program to run within, complete with code and its dependencies.

Bedrock

AWS Bedrock is a generative artificial intelligence platform that enables users to quickly train and deploy their own generative AI models. ClaudeV2, one of the Bedrock models, will be used for empathy analysis and feedback on emails.

SageMaker

AWS SageMaker is a machine learning platform that enables users to quickly build, train, and deploy their own machine learning models. The email template classification model will be trained using this service.

Comprehend

AWS Comprehend uses natural language processing to extract insights about the content of documents by recognizing the entities, key phrases, language, sentiments, and other common elements. This service will be mainly used for its sentiment analysis.

DynamoDB

AWS DynamoDB is a fully managed NoSQL database service to store and retrieve any amount of data and serve any level of request traffic. It will be used to store email summaries.

S3 Buckets

S3 stands for simple storage service and is the cloud SQL storage service in AWS. It provides the ability to store, retrieve and access any data. We will use this to manage our training and testing data for the machine learning services.

Development Environments/Languages

Python

Python is the language used most for data science and machine learning and is well supported. It will act as our main model development language. Experiments and testing can be done in our own integrated development environments as well as in AWS.

React.js Framework

The React.js framework is an open-source JavaScript framework and library developed by Facebook (now known as Meta). It is being used to build the user interface of our web application.

Node.js Framework

Node.js is our runtime environment, supporting our React code. Node.js serves as our back end, based in JavaScript, and will also support our package management via the node package manager, also referred to as “NPM.”

Risk Analysis

Categorizing, Testing and Training Data

Difficulty: Medium

Description: It is difficult for us to plan out how we are going to handle the data. The goal is to classify email templates into 11 categories, but we may not have enough training data for each to properly train a model for them.

Mitigation: We have had difficulty getting the data delivered to us from our sponsor and we still are not sure exactly when they will be giving us this data. We will continue to push our sponsor to send us this data as quickly as possible, and once we receive it, we will then face the challenge of categorizing the data in a manner that is both helpful to our final product and meets the needs that have been laid out by our sponsor. This may include grouping some of the categories together.

Data Storage

Difficulty: Low

Description: Again, this risk comes with a general lack of knowledge within our group, this time with regard to databases. Our team has several options within the AWS services, and we must determine the optimal method for storing the data provided to us by Amazon for analyzing the emails.

Mitigation: For this risk, we will be consulting with our sponsor to learn as much as we can about their plan for what will be available to us with the training data, as well as doing individual research on the databases available to us through AWS and determine what we believe would be the best option for us.

Project Expense

Difficulty: Low

Description: Amazon has provided us with a \$100/month budget for using their AWS services which charge on a run-time basis. Although it is somewhat unlikely to lead to any issues, we still acknowledge the low difficulty risk of managing this budget and staying within the constraints our sponsor has set for us.

Mitigation: We will mitigate this risk by implementing cost-effective strategies with our usage of the AWS services to ensure that we stay below our monthly budget.

Schedule

Week 1 – (8/28 - 9/3)

- Initial team and client meetings
- Research AWS technologies

Week 2 – (9/4 - 9/10)

- Initial triage meeting with TM Griffin Klevering
- Delegate team member roles

Week 3 – (9/11 - 9/17)

- Status Report Presentation
- Present screen mockups to client.
- Work on Project Plan Document and Presentation
- Complete Home page layout

Week 4 – (9/18 - 9/24)

- Project Plan Presentation
- Complete Analysis page layout
- Research ML methodologies

Week 5 – (9/25 - 10/1)

- Make base connections between Lambda and the backend of web application
- Assess training models for SageMaker and Comprehend
- Receive first sample set of test data

Week 6 – (10/2 - 10/8)

- Work on Alpha Document and Presentation
- Deploy first Amplify version of application
- Integrate Lambda ML function APIs into web application
- Evaluate prototype model performances

Week 7 – (10/9 - 10/15)

- Alpha Presentation
- Migrate Home page to Cloudscape UI
- Fine tune ML models
- Allow for template analysis saving and storage

Week 8 – (10/16 - 10/22)

- Migrate Analysis page to Cloudscape UI
- Draft Administration page
- Allow for email analysis download

Week 9 – (10/23 - 10/29)

- Add duplicates confidence scores to duplicates dropdown on Analysis page
- Create Templates page layout
- Implement security features

Week 10 – (10/30 - 11/5)

- Subject Matter Expert meetings
- Migrate empathy analysis to AWS Bedrock
- Integrate existing email templates to the Templates page

Week 11 – (11/6 - 11/12)

- Work on Beta use cases and presentation
- Receive and implemented more data to database
- Categories ML implementation
- Application testing, troubleshooting, error handling and bug fixes

Week 12 – (11/13 - 11/19)

- Continued API troubleshooting and bug fixes
- Clean up front-end and implement final design changes
- Add more analyzed data to database
- Integrate Administrative page

Week 13 – (11/20 - 11/26)

- Final Amplify Deployment
- Beta Presentation
- Perform remaining bug fixes

Week 14 – (11/27 - 12/3)

- Create project video
- Final fixes and comments from sponsors

Week 15 – (12/4 - 12/10)

- Design Day
- Final Deliverables Due
- Present final product to clients