



# Inbox Genius - your next productive email client

FYP23034

Lee Lok Chi Gigi (3035782889)

Cheung Yan Ting (3035786988)

Brandon Tay Jian Wei (3035767102)

Yeung Adrian Justin Tsoi Chun (3035780491)

# OUTLINE

01 | BACKGROUND

02 | OBJECTIVE

03 | SOLUTION

04 | METHODOLOGY

05 | DEMO

06 | TECHNICALS

# 01

## BACKGROUND

Why such an idea?

---



**Do you check your  
email every day?**

# BACKGROUND

## MASSIVE USE OF EMAIL

- Email as their primary medium at work: 82%
- Emails received: 71/day



## TOOL TO SIMPLIFY EMAIL MANAGEMENT

- Assign importance ratings to emails
- Enable users to prioritize

# 02

## OBJECTIVE

What's our aim?

---



**How much time do  
you spend on emails  
per day?**

# OBJECTIVE

## EFFICIENCY IN READING EMAILS

- Time spent: 2.6 hours/day
- Overwhelmed by inbox volume: 67%



## PREVENT MISSING IMPORTANT EMAILS

- Overwhelming number of unimportant emails



## SECURITY & PRIVACY

- Cyber attacks in email: 75%
- E.g. phishing





# 03

## SOLUTION

How to achieve the objective?

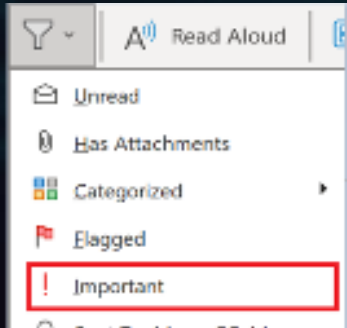
---



# COMPETITORS

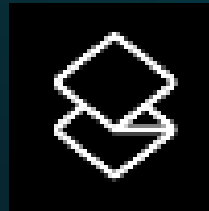
## VANILLA OUTLOOK

- Important Tag → set manually



## SUPERHUMAN

- Split Inbox → by searching domain names



## CLEAN EMAIL

- Advanced filters → age of the email, size of the attachment, cc'd emails
- Unsubscribe function



# SOLUTION

## EMAIL CATEGORIZATION

- 3 levels of importance
- Collect user behaviour → time spent, times clicked
- Whitelist settings
- Natural language input



## EMAIL AUTOMATION

- Generate iCalendar files
- Smart Search

## EMAIL SUMMARIZATION

- Summarizes emails
- Daily Summary

Send us an email at  
[fyp23034@outlook.com](mailto:fyp23034@outlook.com) !

# 04

## Methodology

How are we building it?

---



# FRONTEND TECHNOLOGIES



**Figma**



**ReactJS**



**Tailwind CSS**

# BACKEND TECHNOLOGIES



**Flask**



**Redis**



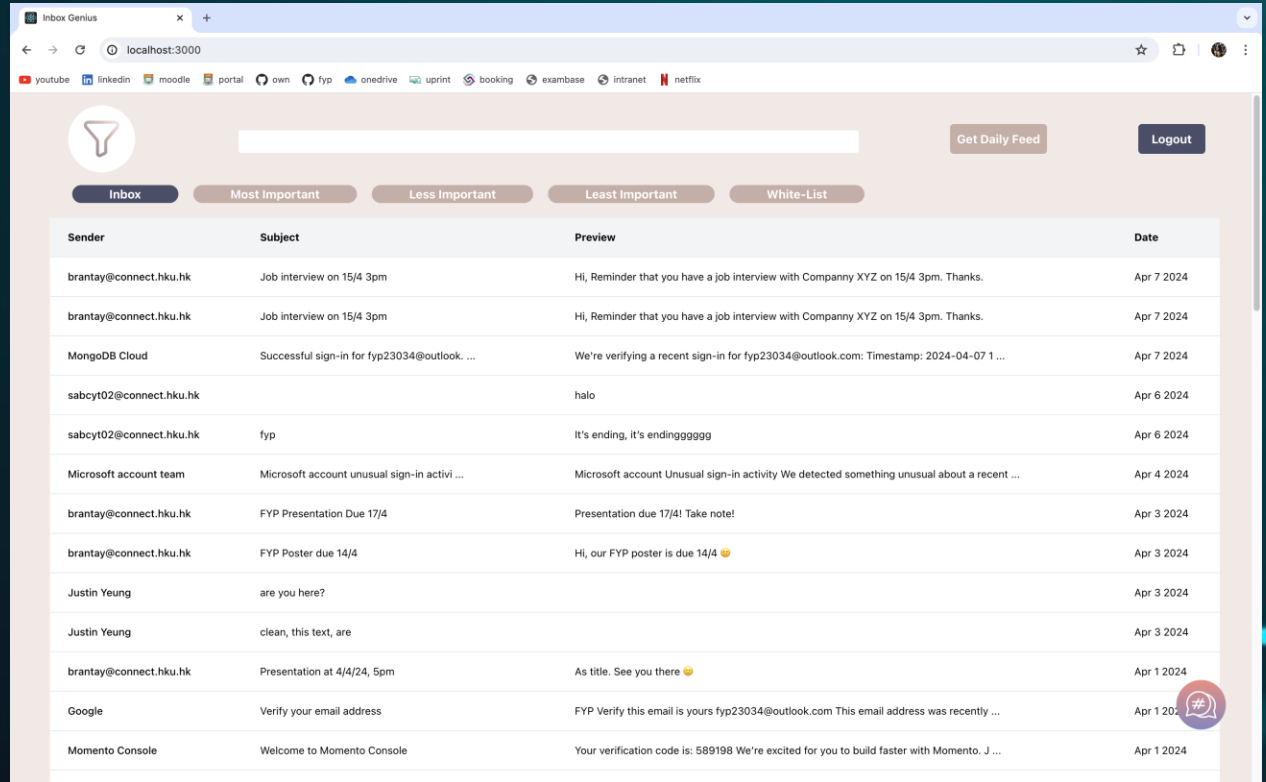
**Postman**



**MongoDB**

# 05

## Demo



The screenshot displays the Inbox Genius web application interface. At the top, there is a navigation bar with a funnel icon, a search bar, a "Get Daily Feed" button, and a "Logout" button. Below the navigation bar, there are five filter tabs: "Inbox", "Most Important", "Less Important", "Least Important", and "White-List". The main content area is a table of emails with the following columns: "Sender", "Subject", "Preview", and "Date".

Sender	Subject	Preview	Date
brantay@connect.hku.hk	Job interview on 15/4 3pm	Hi, Reminder that you have a job interview with Company XYZ on 15/4 3pm. Thanks.	Apr 7 2024
brantay@connect.hku.hk	Job interview on 15/4 3pm	Hi, Reminder that you have a job interview with Company XYZ on 15/4 3pm. Thanks.	Apr 7 2024
MongoDB Cloud	Successful sign-in for fyp23034@outlook. ...	We're verifying a recent sign-in for fyp23034@outlook.com: Timestamp: 2024-04-07 1 ...	Apr 7 2024
sabcyt02@connect.hku.hk		halo	Apr 6 2024
sabcyt02@connect.hku.hk	fyp	It's ending, it's endingggggg	Apr 6 2024
Microsoft account team	Microsoft account unusual sign-in activi ...	Microsoft account Unusual sign-in activity We detected something unusual about a recent ...	Apr 4 2024
brantay@connect.hku.hk	FYP Presentation Due 17/4	Presentation due 17/4! Take note!	Apr 3 2024
brantay@connect.hku.hk	FYP Poster due 14/4	Hi, our FYP poster is due 14/4 📅	Apr 3 2024
Justin Yeung	are you here?		Apr 3 2024
Justin Yeung	clean, this text, are		Apr 3 2024
brantay@connect.hku.hk	Presentation at 4/4/24, 5pm	As title. See you there 📅	Apr 1 2024
Google	Verify your email address	FYP Verify this email is yours fyp23034@outlook.com This email address was recently ...	Apr 1 2024
Momento Console	Welcome to Momento Console	Your verification code is: 589198 We're excited for you to build faster with Momento. J ...	Apr 1 2024



# 06

## TECHNICAL DISCUSSION



# ACCOMPLISHMENTS: BACK-END

## Completed in Sem 1:

- OAuth 2.0 authorization flow with Microsoft EntraID
- Database design & implementation
- Development and testing of core APIs

## Completed in Sem 2:

- Development and testing of additional APIs
- Integration with frontend service and AI module
- SPF and DMARC checks
- Database caching with Redis
- Database read replication
- API documentation

Development

Infrastructure

# COMPLETED API ROUTES

## Semester 1:

1. **GET:** /email/{id}
2. **GET:** /email/getByCategory
3. **PUT:** /metrics/recordTime/{id}
4. **PUT:** /metrics/recordClick/{id}
5. **POST:** /pref/updatePreferences
6. **GET:** /pref/getPreferences

Total: 6 APIs

## Semester 2:

1. **GET:** /email?page={pageNum}
2. **POST:** /email/changeCategory
3. **GET:** /email/getSummary/{id}
4. **GET:** /email/search
5. **GET:** /email/smartSearch
6. **GET:** /email/dailySummary
7. **GET:** /email/generateICS

Total: 7 APIs

# INTEGRATION WITH FRONTEND AND AI SERVICES

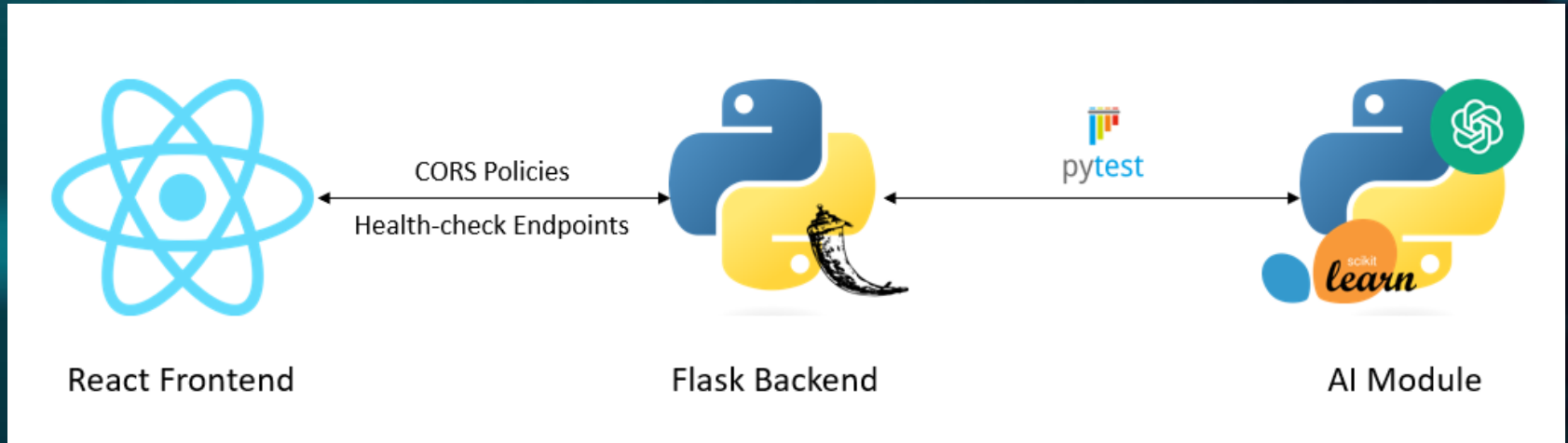


Figure 1: Integration Flow Between Services

# MORE ON CORS POLICIES

- Custom headers on the backend require a CORS preflight check

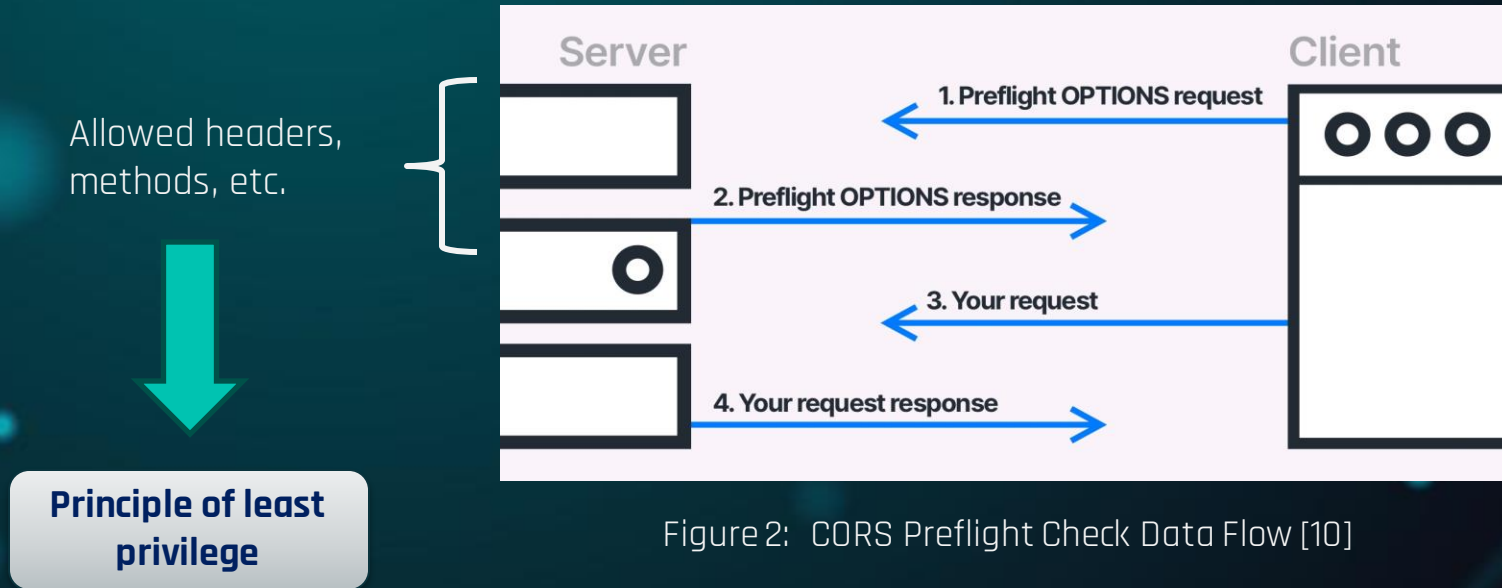


Figure 2: CORS Preflight Check Data Flow [10]

# SPF & DMARC CHECKS

- SPF, DMARC: Email authenticity protocols used in spam filtering => Can be somewhat imprecise
- SPF & DMARC certificates obtainable via recursive DNS queries to the sender's mail server

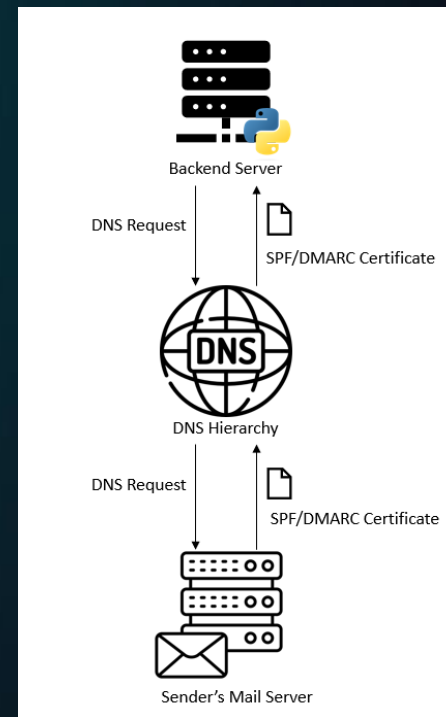


Figure 4: DNS Querying in SPF/DMARC Checks

# DATABASE INFRASTRUCTURE IMPROVEMENTS

**Problem:** Repeated querying of database slows down performance

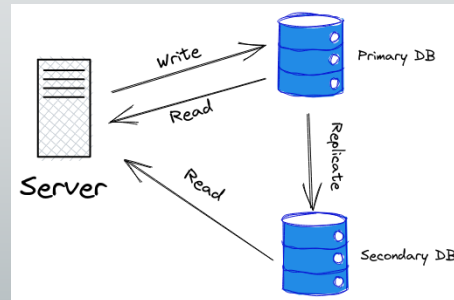
## **Solution 1: Database Caching**

- Redis used as an in-memory caching solution



## **Solution 2: Read Replication**

- Async copies of primary database instance



# REDIS: CACHE-ASIDE PATTERN

- Most common caching pattern; optimizes read queries
- For our application, TTL = 1 hour

**Result:**



~40%

...reduction in API response times

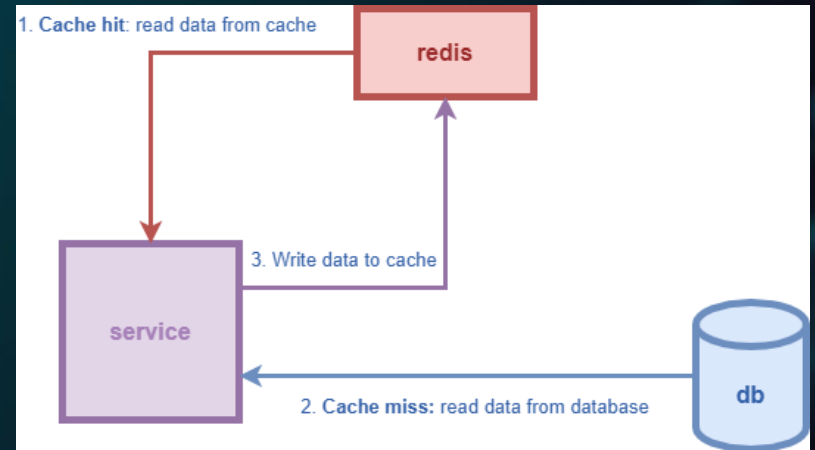


Figure 3: Redis Cache-aside Pattern in Action [11]



# SYSTEM ARCHITECTURE

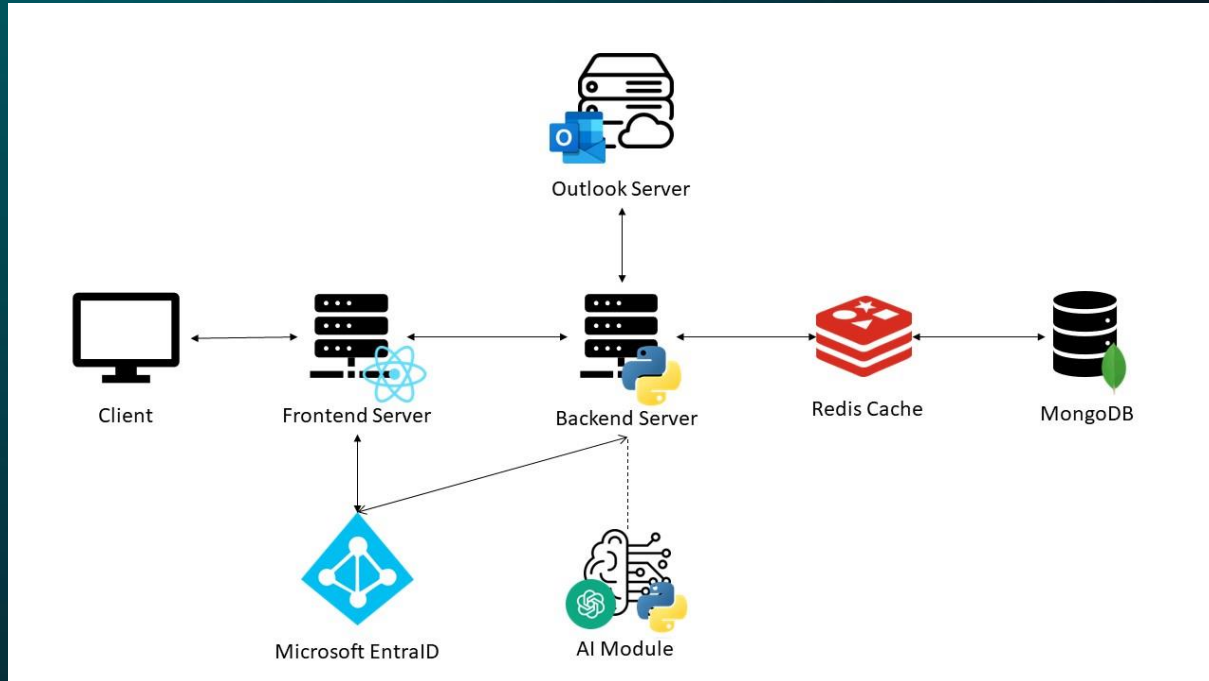


Figure 5: Overall System Architecture

# List of Functions

Email  
Summarization

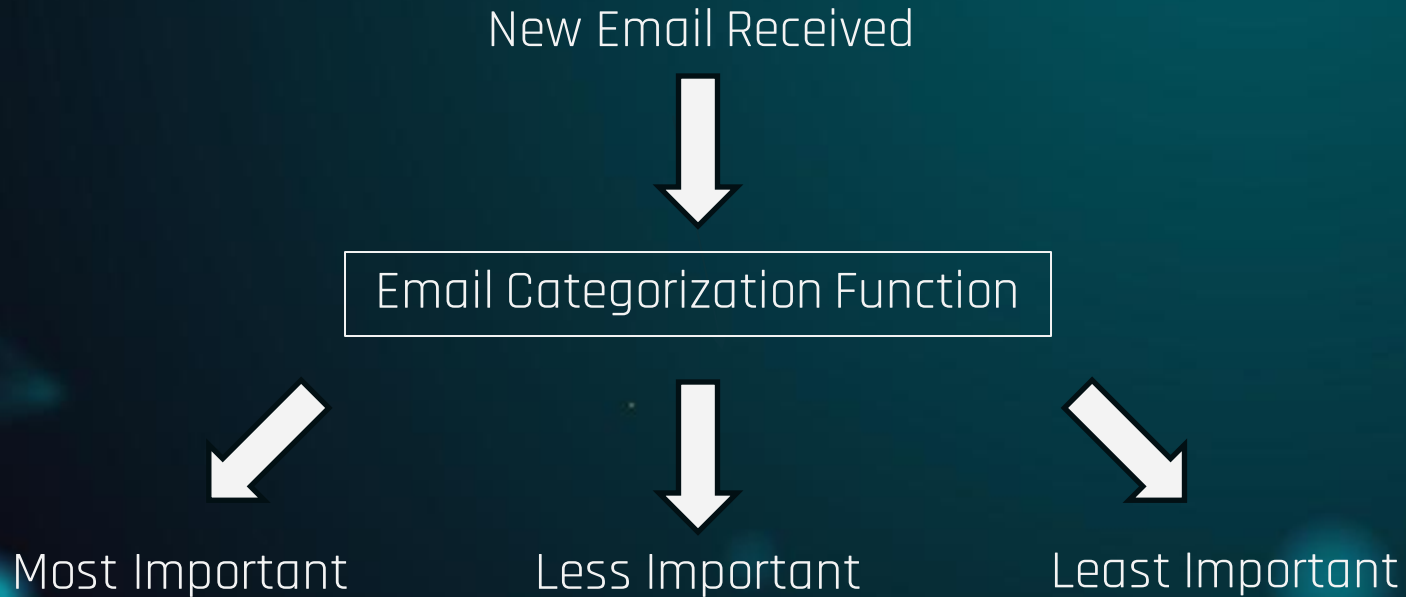
Daily  
Summary

Email Categorization

Smart  
Search

Generate  
iCalendar file

# Recap: Email Categorization



# Email Categorization

Similar to old email + Old email is important -> New Email Important

New Email's Importance Score

= Similarity(New Email, Old Email A) × Old Email A's Estimated Importance  
+ Similarity(New Email, Old Email B) × Old Email B's Estimated Importance  
+ Similarity(New Email, Old Email C) × Old Email C's Estimated Importance  
+ ...

# Email Categorization

Estimating Old Email's Estimated Importance using regression model

Old Email's Estimated Importance

$$\begin{aligned} &= \beta_0 \\ &+ \beta_1 \times \text{Times Clicked} \\ &+ \beta_2 \times (\text{Time Spent/Character}) \end{aligned}$$

# Email Categorization

Estimating Old Email's Estimated Importance using regression model

#	A	B	C	D	E	F	
	email ID	Estimated Importance	Time Spent	Number of Characters	Time Spent ÷ Number of Characters	Times Clicked	
2	65427c84d747ca686fa73835		2	3	347	0.008645533	1
3	65427c85d747ca686fa73837		2	2	499	0.004008016	1
4	65427c84d747ca686fa73839		3	2	2077	0.000962927	1
5	65427c84d747ca686fa7383b		3	5	1763	0.002836075	1
6	65427c84d747ca686fa7383d		2	4	716	0.005586592	1
7	65427c84d747ca686fa7383f		1	3	12404	0.000241857	1
8	65427c88d747ca686fa73841		10	254	4502	0.056419369	4
9	65427c88d747ca686fa73843		8	30	471	0.063694268	1
10	65427c88d747ca686fa73845		7	40	855	0.046783626	1
11	65427c88d747ca686fa73847		6	60	752	0.079787234	2
12	65427c88d747ca686fa73849		3	7	1546	0.004527814	1
13	65427c88d747ca686fa7384b		2	6	725	0.008275862	1
14	65427c88d747ca686fa7384d		4	15	578	0.025951557	1
15	65427c88d747ca686fa7384f		6	46	1556	0.029562982	1
16	65427c88d747ca686fa73851		10	286	687	0.416302766	2
17	65427c88d747ca686fa73853		9	85	542	0.156826568	2
18	65427c88d747ca686fa73855		7	243	875	0.277714286	1
19	65427c88d747ca686fa73857		8	30	130	0.230769231	2
20	65427c88d747ca686fa73859		6	34	752	0.045212766	1
21	65427c88d747ca686fa7385b		4	20	452	0.044247788	1
22	65427c88d747ca686fa7385d		1	2	1438	0.001390821	1
23	65427c88d747ca686fa7385f		1	2	2132	0.000938086	1
24	65427c88d747ca686fa73861		1	2	5297	0.000377572	1
25	65427c88d747ca686fa73863		6	48	3313	0.014488379	1
26	65427c88d747ca686fa73865		8	42	350	0.12	2
27	65427c88d747ca686fa73867		4	23	486	0.047325103	1
28	65427c88d747ca686fa73869		1	2	1217	0.001643385	1
29	65427c88d747ca686fa7386b		6	56	3436	0.016298021	2
30	65427c88d747ca686fa7386d		10	125	765	0.163398693	3
31	65427c88d747ca686fa7386f		8	62	420	0.147619048	3
32	65427c88d747ca686fa73871		11	76	357	0.212885154	3

Figure 6: Regression process for estimating importance

# Email Categorization

Estimating Old Email's Estimated Importance using regression model

	I	J	K	L	M	N	O	P	Q	R
1	SUMMARY OUTPUT									
2										
3	<i>Regression Statistics</i>									
4	Multiple R	0.857457125								
5	R Square	0.735232721								
6	Adjusted R Square	0.716320772								
7	Standard Error	1.701575244								
8	Observations	31								
9										
10	ANOVA									
11		<i>df</i>	<i>SS</i>	<i>MS</i>	<i>F</i>	<i>Significance F</i>				
12	Regression	2	225.1235156	112.5617578	38.87662448	8.31956E-09				
13	Residual	28	81.07003274	2.895358312						
14	Total	30	306.1935484							
15										
16		<i>Coefficients</i>	<i>Standard Error</i>	<i>t Stat</i>	<i>P-value</i>	<i>Lower 95%</i>	<i>Upper 95%</i>	<i>Lower 95.0%</i>	<i>Upper 95.0%</i>	
17	Intercept	1.065369902	0.647342504	1.645759232	0.110994276	-0.260651106	2.391390909	-0.260651106	2.391390909	
18	Time Spent÷Number of Characters	15.6984298	3.532526452	4.443966666	0.000126557	8.462377386	22.93448221	8.462377386	22.93448221	
19	Times Clicked	1.997650356	0.435232029	4.589851438	8.50553E-05	1.106117959	2.889182753	1.106117959	2.889182753	
20										
21										

Figure 7: Regression results for estimating importance

# Email Categorization

Estimating Old Email's Estimated Importance using regression model

Old Email's Estimated Importance

= 1.07

+ 2.00 × Times Clicked

+ 15.70 × (Time Spent/Character)



# Email Categorization

Estimating  $\text{Similarity}(\text{New Email}, \text{Old Email})$  using regression model

$$\begin{aligned} & \text{Similarity}(\text{New Email}, \text{Old Email}) \\ = & \beta_0 \\ + & \beta_1 \times \text{text similarity between } \underline{\text{subject}} \\ + & \beta_2 \times \text{text similarity between } \underline{\text{body}} \end{aligned}$$

---

**Text Similarity  
Calculation:**

**Text Cleaning**

**TF-IDF and Cosine  
Similarity**

# Email Categorization

Estimating  $\text{Similarity}(\text{New Email}, \text{Old Email})$  using regression model

$\text{Similarity}(\text{New Email}, \text{Old Email})$

= 0.08

+ 1.19 × text similarity between subject

+ 2.71 × text similarity between body

# Email Categorization

New Email's Importance Score

$$\begin{aligned} &= \text{Similarity}(\text{New Email, Old Email A}) \times \text{Old Email A's Estimated Importance} \\ &+ \text{Similarity}(\text{New Email, Old Email B}) \times \text{Old Email B's Estimated Importance} \\ &+ \text{Similarity}(\text{New Email, Old Email C}) \times \text{Old Email C's Estimated Importance} \\ &+ \dots \end{aligned}$$

**Rank with Other  
Emails**

# Email Categorization

## NATURAL LANGUAGE COMMAND



01

Words may appear



02

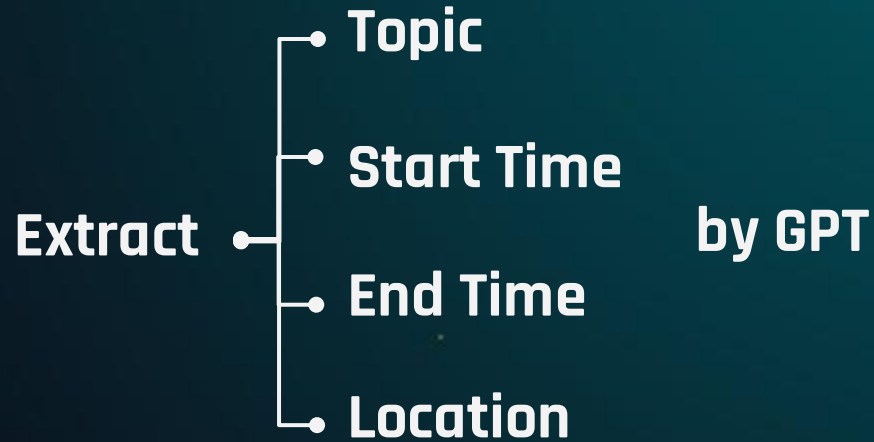
Preference of the email  
(indicated by a -100 to 100 rating)



03

Create fake email

# Generate iCalendar File



# Smart Search

**Search:**

"Interviews"



+ words may appear

"Interviews", "Meeting", "Candidates", "Assessment", "Application", ...



Words Processing & Lemmatize

"interview", "meet", "candidate", "assessment", "application", ...



match



Match

"meet dirk tutor position application"



Words Processing & Lemmatize

**Email:**

"Meet with Dirk for Tutor Position Application"

# Email Summarization & Daily Summary

**GPT Prompting**

**Daily Summary: Only Important Emails**



# Use of GPT

Email Summarization	<b>Yes</b>
Daily Summary	<b>Yes</b>
Smart Search	<b>Yes</b>
Email Categorization - Default	<b>No</b>
Email Categorization - Natural Language Command	<b>Yes</b>
Create iCalendar File	<b>Yes</b>

# MEASURING EFFICIENCY

Algorithm on deciding email importance (GPT v.s. Inbox Genius)

GPT disparity = |GPT rating - User answer|

Inbox Genius disparity = |Inbox Genius rating - User answer|



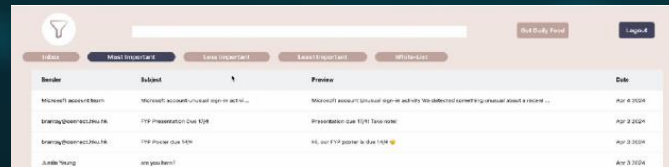
Number of Important Emails Classified Successfully (Vanilla Outlook v.s. Inbox Genius)

50 important emails are picked by our team

Fig. 1 Importance Tag in Outlook

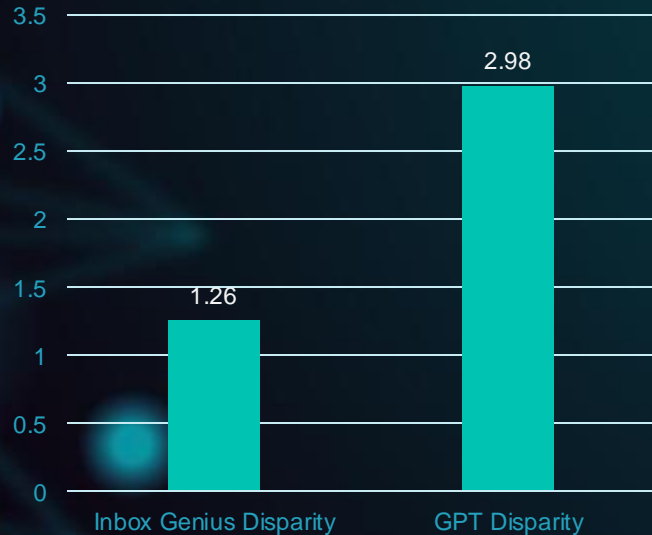


Fig. 2 Tagged in Inbox Genius

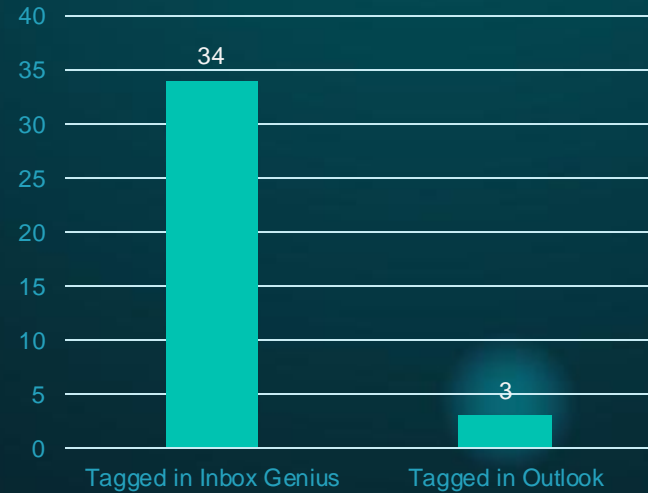


# RESULTS

## Algorithm on deciding email importance (GPT v.s. Inbox Genius)



## Percentage of Important Emails classified successfully (Vanilla Outlook v.s. Inbox Genius)



# 07

## Q & A

---



# REFERENCES

- [1] G. Christ, "Workers overwhelmed, distracted by email," HR Dive, <https://www.hrdive.com/news/workers-overwhelmed-distracted-by-email/637786/> (accessed Sep. 26, 2023).
- [2] "The Latest Work Email Statistics 2023 You Shouldn't Ignore," Gitnux, <https://blog.gitnux.com/work-email-statistics/#:~:text=11%20hours%20is%20the%20average,spends%20on%20email%20per%20week>, (accessed Sep. 26, 2023).
- [3] "Top 10 Cybersecurity Threats in 2023," Embroker, <https://www.embroker.com/blog/top-cybersecurity-threats/#:~:text=Over%2075%25%20of%20targeted%20cyberattacks,of%20%20stolen%20%20credentials%20and%20%20ransomware> (accessed Sep. 26, 2023).
- [4] B. Kopf, "The Power of Figma as a Design Tool," Designers. <https://www.toptal.com/designers/ui/figma-design-tool> (accessed Jan. 20, 2024).
- [5] "Stack Overflow trends," Stack Overflow, <https://insights.stackoverflow.com/trends?tags=reactjs%2Cvue.js%2Cangular%2Csvelte%2Cangularjs%2Cvuejs3> (accessed Oct. 17, 2023).
- [6] M. Grinberg, Flask web development: Developing web applications with Python. O'Reilly Media, Inc, USA, 2018.
- [7] "Scikit-learn vs TensorFlow: A Detailed Comparison," Simplilearn. [https://www.simplilearn.com/scikit-learn-vs-tensorflow-article#scikitlearn\\_vs\\_tensorflow\\_comparison](https://www.simplilearn.com/scikit-learn-vs-tensorflow-article#scikitlearn_vs_tensorflow_comparison) (accessed Jan. 20, 2024).

# REFERENCES

[8] "Why Use MongoDB and When to Use It?" MongoDB, <https://www.mongodb.com/why-use-mongodb#:~:text=MongoDB%20is%20an%20excellent%20choice,repository%20to%20a%20massive%20size> (accessed Jan. 20, 2024).

[9] M. Anicas, "An introduction to OAuth 2," Digital Ocean, <https://www.digitalocean.com/community/tutorials/an-introduction-to-oauth-2> (accessed Oct. 17, 2023).

[10] J. Pomykala, "What is CORS," SimpleLocalize, <https://simplelocalize.io/blog/posts/what-is-cors/> (accessed Apr. 11, 2024).

[11] T. B. Septian, "Redis - Cache-Aside Pattern," Blibli.com Tech Blog, <https://medium.com/blibliidotcom-techblog/redis-cache-aside-pattern-72fff2e4f927> (accessed Apr. 14, 2024).

[12] A. Maftai, "TF-IDF," Hyperskill, <https://hyperskill.org/learn/step/31428> (accessed Jan. 20, 2024)