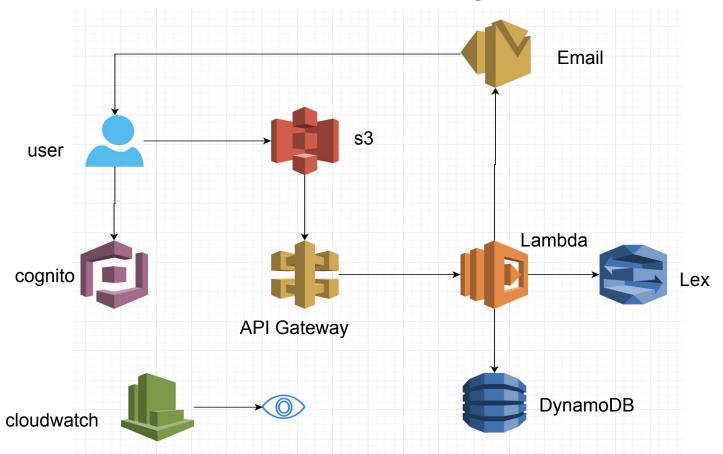
# Fashion Seeker

Junyue Wu, Bowen Tan, Jian Song, Tianchang Cheng

## Motivation

- Discount-driven shoppers are dissatisfied with checking promotion emails
- Promotion emails contains too much information not needed and users need one-click action for extracting the discount.

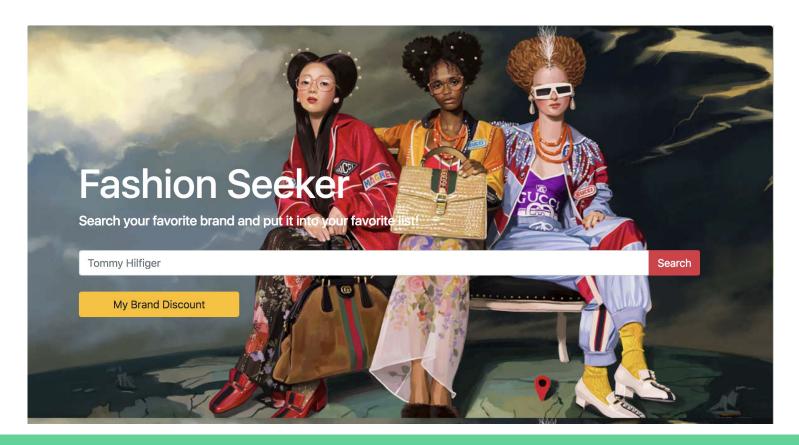
# Architecture Diagram



## **DynamoDB Structure**

```
User Table
                                         Discount Table
Username: "xxx".
                                    Url: "https://gap.com",
Urls: ["https://qap.com",
                                    Name: "GAP",
     "https://tommy.com"]
                                    Discount: ["50%..., 40%..."]
Username: "yyy",
                                    Url: "https://tommy.com",
Urls: ["https://tommy.com"]
                                    Name: "Tommy Hilfiger",
                                    Discount: ["20%..., 30%..."]
```

## **Live Demo**



## **API Gateway:**

#### /GET

Retrieve user's interested shopping brands and their latest discount information.

#### /POST

Search shopping brand by user input.

#### /PUT

Send an email to the user containing all discount information of interested brand.

## On Call Lambda Function:

#### **/Get All Shopping Brand and Discount**

Called after the authentication is done and the webpage is loading.

### /Search Shopping Brand

Called when user input keyword and click search in the frontend.

### /Insert New Shopping Brand & Get Discount

Called when a brand user searched is not recorded in the dynamoDB. After the insertion of new brand, get it's latest discount by crawling the url of the brand.

## **Automated Lambda Function:**

#### **/Crawl Latest Discount**

Triggered every 30 min to get the latest discount and update dynamoDB.

/Send Summary Email(New AWS Function)

Triggered every day to send summary email.

## **Architecture Detail**

- Use AWS S3 bucket to hold the static webpage.
- User AWS cognito to manage user login and user identities.
- Use AWS DynamoDB to do record favorite websites of the logged in user.
- Use AWS Lambda Function to achieve serverless architecture, which also support autoscaling.
- Enable CloudWatch to monitor the program and trigger the lambda function to get latest discount and make sure the discount information returned are up-to-date.
- Use SES to send discount information and link to user's registered emails.
- Use google search api to let user search brand or shopping site name.

# Question?