

Table of Contents

| | |
|---|-----------|
| 1. Project Information | 4 |
| 1.1. Company Acting Director | 4 |
| 1.2. Project Team | 4 |
| 2. Project Overview | 4 |
| 3. User Manual | 5 |
| 3.1. Data Engineering | 5 |
| 3.1.1. Data Engineering 2022 T3 Overview | 5 |
| 3.1.2. Physical Data Model | 5 |
| 3.1.3. Data Dictionary | 6 |
| 3.1.3.1. ITEM table | 6 |
| 3.1.3.2. ITEM_PRICE_CURRENT table | 7 |
| 3.1.3.3. ITEM_PRICE_HIST table | 7 |
| 3.1.3.4. USER table | 8 |
| 3.1.3.5. USER_ORDER table | 8 |
| 3.1.3.6. ORDER_ITEM table | 8 |
| 3.1.3.7. RECOMMEND_ITEM table | 9 |
| 3.1.4. Data Platform Setup | 9 |
| 3.1.5. Access the Cloud Database | 17 |
| 3.1.6. Create Database Schema and Load Initial Data | 18 |
| 3.1.7. Source Data | 19 |
| 3.1.7.1. Woolworths and Coles Weekly Price Data | 19 |
| 3.1.8. Data Ingestion | 24 |
| 3.1.8.1. Weekly Price Data | 24 |
| 3.1.8.2. User Buying Transaction Data | 24 |
| 3.1.8.3. Item Images | 24 |
| 3.1.9. Data Cleansing and Transformation | 25 |
| 3.1.10. Verification of Data Loading | 25 |
| 3.2. Data Engineering | 26 |
| 3.2.1. Prediction model process flow | 26 |
| 3.2.1. The Process flow in details | 26 |
| 4. Completed Deliverables | 28 |
| 5. Roadmap | 33 |
| 6. Open Issues | 34 |
| 7. Lessons Learned | 34 |
| 8. Product Development Life Cycle | 35 |
| 8.1. New Tasks | 35 |
| 8.2. Definition of Done | 36 |
| 8.3. Task Review | 36 |
| 8.4. Testing | 36 |
| 8.5. Branching Strategy | 36 |
| 9. Product Architecture | 37 |
| 9.1. Tech Stack | 37 |

| | |
|--|-----------|
| 10. Source Code..... | 38 |
| 11. Login Credentials | 38 |
| 11.1. <i>About our training data for Machine Learning model.....</i> | <i>38</i> |
| 11.2. <i>Machine Learning Research.....</i> | <i>39</i> |
| 12. Appendices..... | 39 |

1. Project Information

1.1. Company Acting Director

1.2. Project Team

2. Project Overview

The DiscountMate project seeks to empower consumers with the ability to make their lives easier by providing reliable information regarding discounted items of interests from various supermarket chains, affording them the opportunity to save potentially hundreds of dollars off their weekly grocery shopping with minimal effort.

Item information is updated regularly through data collection techniques such as web-scraping, prices are then compared between various providers for the same item and those with the highest potential for savings are presented to consumers.

Additionally, the project aims to employ machine learning and data analysis techniques to identify patterns and predict future discount opportunities, providing consumers with relevant recommendations according to their interests and purchase history.

History is obtained through user interaction methods such as item searches, checking off items on a shopping list to indicate a purchase, and scanning purchase receipts by utilizing optical character recognition (OCR) technology and extracting necessary data.

3. User Manual

3.1. Data Engineering

3.1.1. Data Engineering 2022 T3 Overview

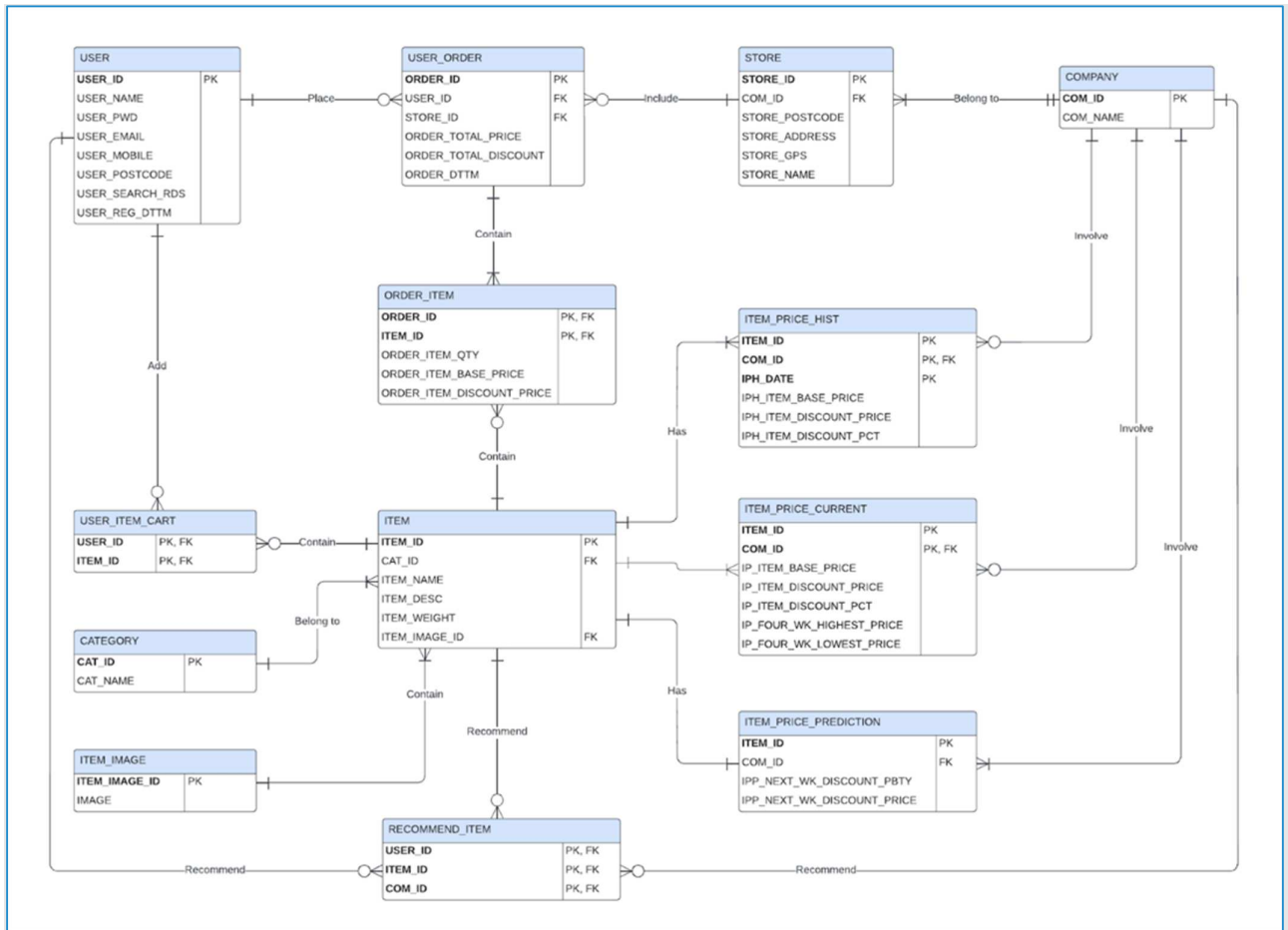
The goal of the data engineering team this trimester was to develop the data platform for providing operation use data for the front-end team and analytic use data for the data science team. Data use by the front-end team involved weekly price, discount, image and description information of items listed on Woolworths and Coles web site. On the order side, historical item price detail is saved and formatted, together with user buying behaviour data, are provided to data science team for their machine learning algorithms use. Those data are used to predict item price's trend and perform buying item recommendation for the system user.

The developed data platform is operating by a MySQL cloud database which hosted on Google Cloud Platform (GCP). Data model is designed in third normal form to store the formatted and calculated data. The source data of the data platform are from weekly web scrapping item's related information from Woolworths and Coles web site, and from Woolworths provided dataset. Python and SQL are the major programming languages in the development of Extract, Transform, Load (ETL) jobs for data import, loading of the platform. The ETL jobs are deployed and scheduled in a ETL server which operated in a Windows server hosted on GCP.

For the Woolworths provided dataset. It included over 60 customer invoices with about 300 line-items. It became the test data for studying user buying behaviour and later according to that, developed the buying recommendation algorithm.

3.1.2. Physical Data Model

Below is the physical data model of front-end use, in Crow's foot Entity-Relation diagram.



[DiscountMate Database ER Diagram v1.0.pdf](#)

(github.com/discountmate/dm_app/tree/main/database/design_documents/)

3.1.3. Data Dictionary

Tables created in the data platform database are in fact more than the table shown in the diagram 1. It is because beside tables supporting front-end use, there are staging tables, formatted operational tables and database views that used in the ETL process.

For the full detail of the tables please see the referred data dictionary document in Github:

[DiscountMate Data Dictionary v1.0.xlsx](#)

(github.com/discountmate/dm_app/tree/main/database/design_documents/)

3.1.3.1. ITEM table

This is the master table of items from both Woolworths and Coles. Where Woolworths items are ITEM_ID < 1000001 and Coles items are ITEM_ID > 1000001. Table updated weekly.

| Column | Data Type | Remark |
|---------|-----------|--|
| ITEM_ID | int | ItemCode in Woolworths, DiscountMate system generated code for Coles items |

| | | |
|---------------|---------------|--|
| CAT_ID | int | |
| ITEM_NAME | varchar(255) | |
| ITEM_DESC | varchar(1000) | |
| ITEM_WEIGHT | varchar(50) | |
| ITEM_IMAGE_ID | text | |

3.1.3.2. ITEM_PRICE_CURRENT table

Current week item price table that contained both Woolworths and Coles items. Table updated weekly.

| Column | Data Type | Remark |
|--------------------------|--------------|---|
| ITEM_ID | int | |
| COM_ID | int | |
| IP_ITEM_BASE_PRICE | decimal(8,2) | Item price of the current week |
| IP_ITEM_DISCOUNT_PRICE | decimal(8,2) | Amount saved - difference between IP_FOUR_WK_HIGHEST_PRICE and IP_ITEM_BASE_PRICE |
| IP_ITEM_DISCOUNT_PCT | decimal(4,2) | Percentage of IP_ITEM_DISCOUNT_PRICE over IP_FOUR_WK_HIGHEST_PRICE |
| IP_FOUR_WK_HIGHEST_PRICE | decimal(8,2) | The highest price of the item during last 4 weeks |
| IP_FOUR_WK_LOWEST_PRICE | decimal(8,2) | The lowest price of the item during last 4 weeks |

3.1.3.3. ITEM_PRICE_HIST table

Historical item price table that saves weekly item price. Updated weekly.

| Column | Data Type | Remark |
|-------------------------|--------------|---|
| ITEM_ID | int | |
| COM_ID | int | |
| IPH_DATE | date | The effective date of the weekly item price |
| IPH_ITEM_BASE_PRICE | decimal(8,2) | Item price of that effective date |
| IPH_ITEM_DISCOUNT_PRICE | decimal(8,2) | Amount saved - difference between the item's IP_FOUR_WK_HIGHEST_PRICE and IPH_ITEM_BASE_PRICE |
| IPH_ITEM_DISCOUNT_PCT | decimal(4,2) | Percentage of IPH_ITEM_DISCOUNT_PRICE over the item's IP_FOUR_WK_HIGHEST_PRICE |

3.1.3.4. USER table

Table for storing user information.

| Column | Data Type | Remark |
|-----------------|--------------|----------------------------|
| USER_ID | int | |
| USER_NAME | varchar(255) | Login name |
| USER_PWD | varchar(255) | Login password |
| USER_EMAIL | varchar(255) | |
| USER_MOBILE | varchar(50) | |
| USER_POSTCODE | smallint | |
| USER_SEARCH_RDS | varchar(255) | Store searching radius |
| USER_REG_DTTM | datetime | User registration datetime |

3.1.3.5. USER_ORDER table

Table contained invoice header. Data source from the transaction datafile provided by Woolworths.

| Column | Data Type | Remark |
|----------------------|---------------|----------------|
| ORDER_ID | varchar(30) | Invoice Number |
| USER_ID | int | |
| STORE_ID | int | |
| ORDER_TOTAL_PRICE | decimal(10,2) | |
| ORDER_TOTAL_DISCOUNT | decimal(10,2) | |
| ORDER_DTTM | datetime | |

3.1.3.6. ORDER_ITEM table

Table contained invoice line-items. Data source from the transaction datafile provided by Woolworths.

| Column | Data Type | Remark |
|---------------------------|--------------|----------------|
| ORDER_ID | varchar(30) | Invoice Number |
| ITEM_ID | int | |
| ORDER_ITEM_QTY | smallint | |
| ORDER_ITEM_BASE_PRICE | decimal(8,2) | |
| ORDER_ITEM_DISCOUNT_PRICE | decimal(8,2) | |

3.1.3.7. RECOMMEND_ITEM table

Machine learning algorithm calculates the recommended buying item for each user base on users buying behaviour and the latest item price. This table store a list of recommended items from each company for every user. Updated weekly.

| Column | Data Type | Remark |
|---------|-----------|--------|
| USER_ID | int | |
| ITEM_ID | int | |
| COM_ID | int | |

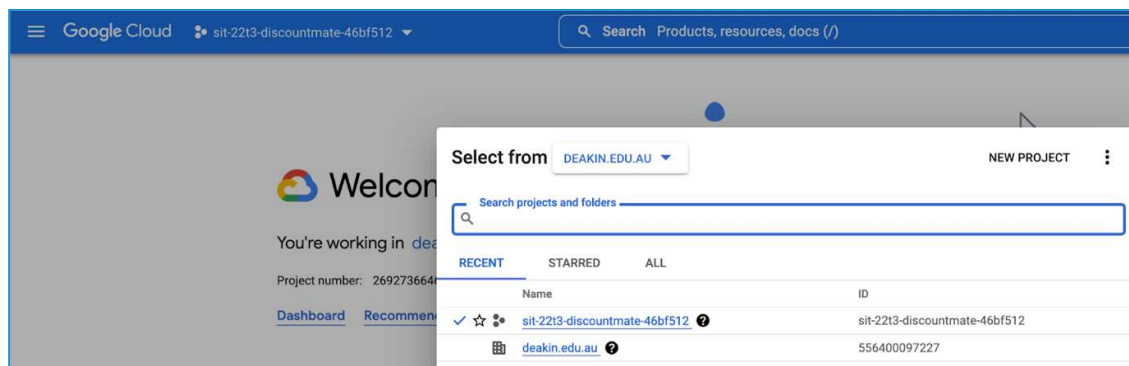
3.1.4. Data Platform Setup

DiscountMate project data platform is operating in MySQL database which hosted in Google Cloud Platform. Please follow below steps to setup the database for project use.

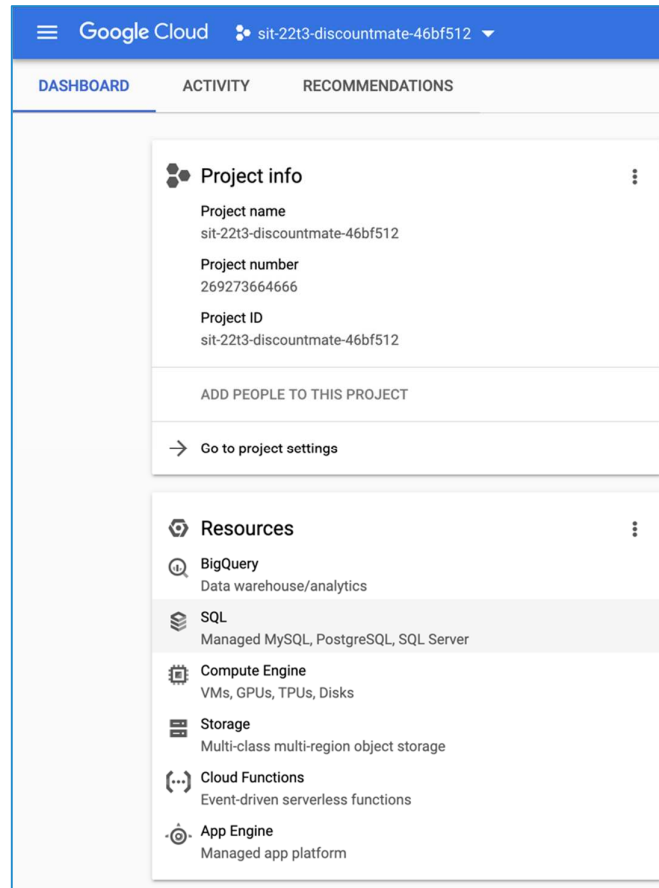
1. Run web browser to access GCP management console at:

<https://console.cloud.google.com>

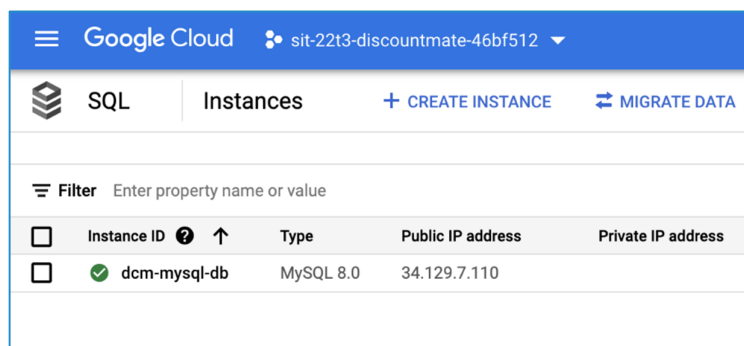
4. Once signed in, select the drop-down box that appears at the top left of the browser window (look to the right of the Google Cloud branding) At the very top of the small panel that pops up, make sure that the 'Select from' field is set to deakin.edu.au

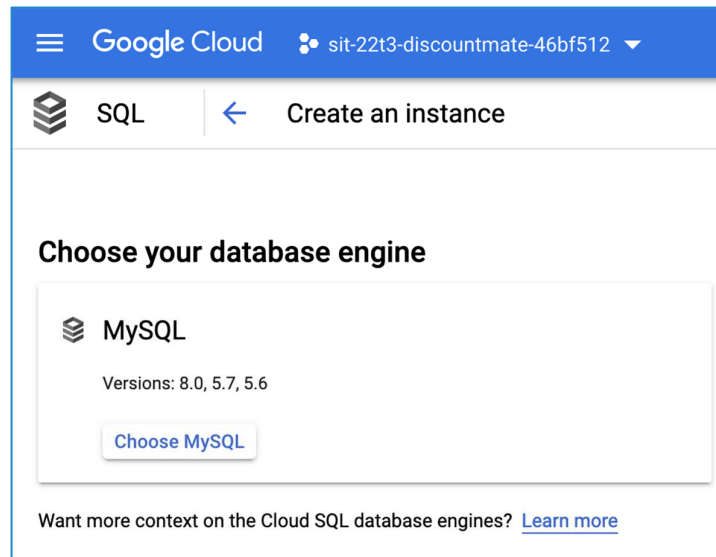


6. In the project page, click **SQL** under **Resources**.



7. Click **+ CREATE INSTANCE**. Then click **Choose MySQL** to create a MySQL database instance.





8. In the MySQL instance creation page. Fill information on the below circle fields. After completed all fields, click **CREATE INSTANCE**.

Google Cloud sit-22t3-discountmate-46bf512 Search sit-22t3-discountmate-46bf512

< Create a MySQL instance

Instance info

Instance ID *
dcm-mysql-db

Use lowercase letters, numbers, and hyphens. Start with a letter.

Password *
dusit2022t3 [GENERATE](#)

Set a password for the root user. [Learn more](#)

☐ No password

[PASSWORD POLICY](#)

Database version *
MySQL 8.0

Choose a configuration to start with

These suggested configurations will pre-fill this form as a starting point for creating an instance. You can customize as needed later.

☐ Production
Optimized for the most critical workloads. Highly available, performant, and durable.

☒ Development
Performant but not highly available, while reducing cost by provisioning less compute and storage.

Summary

| | |
|---------------------------|--|
| Region | australia-southeast2 (Melbourne) |
| DB Version | MySQL 8.0 |
| vCPUs | 2 vCPU |
| Memory | 8 GB |
| Storage | 100 GB |
| Network throughput (MB/s) | 500 of 2,000 |
| Disk throughput (MB/s) | Read: 48.0 of 240.0 Write: 48.0 of 144.0 |
| IOPS | Read: 3,000 of 15,000 Write: 3,000 of 9,000 |
| Connections | Public IP |
| Backup | Automated |
| Availability | Single zone |
| Point-in-time recovery | Enabled |

| | Production | Development |
|-----------------------------|------------------|-------------|
| Availability | Highly Available | Single Zone |
| vCPU | 4 | 2 |
| Memory | 26 GiB | 8 GiB |
| Storage | 100 GiB | 100 GiB |
| Automatic storage increases | Enabled | Enabled |
| Automated backups | Enabled | Enabled |
| Point-in-time recovery | Enabled | Enabled |
| Maintenance order | Later | Any |

[^ COLLAPSE DETAILS](#)

Choose region and zonal availability

For better performance, keep your data close to the services that need it. Region is permanent, while zone can be changed any time.

Region

australia-southeast2 (Melbourne) ▼

Zonal availability

☒ **Single zone**
In case of outage, no failover. Not recommended for production.

☐ **Multiple zones (Highly available)**
Automatic failover to another zone within your selected region. Recommended for production instances. Increases cost.

[▼ SPECIFY ZONES](#)

Choose region and zonal availability

For better performance, keep your data close to the services that need it. Region is permanent, while zone can be changed any time.

Region

australia-southeast2 (Melbourne) ▼

Zonal availability

☒ Single zone

In case of outage, no failover. Not recommended for production.

☐ Multiple zones (Highly available)

Automatic failover to another zone within your selected region. Recommended for production instances. Increases cost.

▼ SPECIFY ZONES

Storage capacity

10 - 65,536 GB. Higher capacity improves performance, up to the limits set by the machine type. Capacity can't be decreased later.

☐ 10 GB

☒ 20 GB

☐ 100 GB

☐ 200 GB

☐ Custom

☒ Enable automatic storage increases

If enabled, whenever you are nearing capacity, storage will be incrementally (and permanently) increased. [Learn more](#)

▼ ADVANCED ENCRYPTION OPTIONS

Connections

Choose how you want your source to connect to this instance, then define which networks are authorized to connect. [Learn more](#)

You can use the Cloud SQL Proxy for extra security with either option. [Learn more](#)

Instance IP assignment

☐ Private IP

Assigns an internal, Google-hosted VPC IP address. Requires additional APIs and permissions. Can't be disabled once enabled. [Learn more](#)

☒ Public IP

Assigns an external, internet-accessible IP address. Requires using an authorized network or the Cloud SQL Proxy to connect to this instance. [Learn more](#)

Authorized networks

You can specify CIDR ranges to allow IP addresses in those ranges to access your instance. [Learn more](#)

ADD NETWORK

Data Protection
Automatic backups enabled. Point-in-time recovery (via binary logs) disabled. Instance deletion protection enabled. ▼

Maintenance
Updates may occur any day of the week. Maintenance timing set to 'Later.' ▼

Flags
No flags set. ▼

Labels
No labels set ▼

^ HIDE CONFIGURATION OPTIONS

CREATE INSTANCE CANCEL

9. The instance of the MySQL database is then up and running. Please remember the password you assigned. It will be using to access the database with root account in order to create database user access for your teammates.

10. In order to enable access from your local computer, you need to enable Cloud SQL Admin API from below page:

<https://console.developers.google.com/apis/api/sqladmin.googleapis.com/overview?project=269273664666>

11. After you are done, you should see the screen as below. In next step you will need to add authorized network of this database in order to allow access to it from your local computer.

Google Cloud sit-22t3-discountmate-46bf512

Search Products, resources, docs (/)

APIs & Services

Enabled APIs & services

Library

Credentials

OAuth consent screen

Page usage agreements

API/Service Details

DISABLE API

Cloud SQL Admin API
API for Cloud SQL database instance management

By Google Enterprise API

Service name: sqladmin.googleapis.com

Type: Public API

Status: Enabled

METRICS QUOTAS CREDENTIALS

Select Graphs: 4 Graphs

Filters: Versions: v1 and v1beta4 Credentials: Compute Engine default s... Methods: 102 options selected

Traffic by response code

No data is available for the selected time frame.

UTC+11 Oct 27, 2022 Nov 3, 2022 Nov 10, 2022

12. Go back to the SQL dashboard. Click your database instance ID you just created.

Google Cloud sit-22t3-discountmate-46bf512

SQL Instances + CREATE INSTANCE ⇄ MIGRATE DATA

Filter Enter property name or value

| | Instance ID ? ↑ | Type | Public IP address | Private IP address |
|-------------------------------------|-----------------|-----------|-------------------|--------------------|
| <input checked="" type="checkbox"/> | dcm-mysql-db | MySQL 8.0 | 34.129.7.110 | |

13. Mark down the public IP address. You will use it for your database access.

14. Click **Connections** from the left panel. Under Authorized networks, click **ADD NETWORK**.

The screenshot shows the Google Cloud console interface for a Cloud SQL instance. The top navigation bar includes the Google Cloud logo, the project name 'sit-22t3-discountmate-46bf512', and a search bar. The left sidebar is titled 'SQL' and contains a 'PRIMARY INSTANCE' section with links to Overview, Query insights (marked 'NEW'), Connections (selected), Users, Databases, Backups, Replicas, and Operations. The main content area is titled 'Connections' and shows the instance 'dcm-mysql-db' (MySQL 8.0). It has three tabs: 'NETWORKING' (active), 'SECURITY', and 'CONNECTIVITY TESTS'. The 'NETWORKING' tab contains instructions on how to connect, a section for 'Instance IP assignment' with 'Private IP' (unchecked) and 'Public IP' (checked) options, and an 'Authorized networks' section with a dropdown menu showing 'Oscar' and an 'ADD NETWORK' button. At the bottom, there is an 'App Engine authorization' section.

Google Cloud sit-22t3-discountmate-46bf512 Search sit-22t3-discoun

SQL

PRIMARY INSTANCE

- Overview
- Query insights **NEW**
- Connections**
- Users
- Databases
- Backups
- Replicas
- Operations

Connections

All instances > dcm-mysql-db

✓ **dcm-mysql-db**

MySQL 8.0

NETWORKING SECURITY CONNECTIVITY TESTS

Choose how you want your source to connect to this instance, then define which networks are authorized to connect. [Learn more](#)

You can use the Cloud SQL Proxy for extra security with either option. [Learn more](#)

Instance IP assignment

☐ Private IP
Assigns an internal, Google-hosted VPC IP address. Requires additional APIs and permissions. Can't be disabled once enabled. [Learn more](#)

☒ Public IP
Assigns an external, internet-accessible IP address. Requires using an authorized network or the Cloud SQL Proxy to connect to this instance. [Learn more](#)

Authorized networks

You can specify CIDR ranges to allow IP addresses in those ranges to access your instance. [Learn more](#)

Oscar ✓

ADD NETWORK

App Engine authorization

All apps in this project are authorized by default. You can use [Cloud IAM](#) to authorize apps in other projects. [Learn more](#)

Release Notes

15. Under Edit network. You could enter IP for your specific computer access, or you could follow below screen to allow all TCP/IP access to that database.

Edit network

Name

All Access

Use [CIDR notation](#)

Network *

0.0.0.0/0

Example: 199.27.25.0/24

16. Click **SAVE** to confirm the change and wait some seconds for the change to take effect. You could then try to access the cloud database from your local computer.

Reference Documents:

- [MySQL Cloud DB Creation on Google Cloud Platform.docx](#)


3.1.5. Access the Cloud Database

There are multiple ways to access the cloud database you created:


1. Install MySQL Workbench (or any other SQL client that able to connect MySQL) in your computer. Then use below detail to setup the database connection:

| | |
|-----------|--|
| Hostname: | [the public ip you have marked down] |
| Username: | root |
| Password: | [the password you assigned in the database creation] |


2. Without using SQL client, you could access the database through GLOUD SHELL in GCP management console. Please follow below simple steps.
 - I. Login to GCP management console.
 - II. Click your project id. Go to the SQL dashboard.
 - III. Click your created database instance id.
 - IV. Under Connect to this instance, click **OPEN CLOUD SHELL**.

 **Connect to this instance**

Public IP address

:


Connection name

sit-22t3-discountmate-46bf512:australia-southeast2:dcm-mysql-db


Need help connecting?

Review the documentation to learn about the many ways to connect to your instance.

[Learn more](#)

To connect using gcloud,

[OPEN CLOUD SHELL](#)

To learn about connecting with a Compute Engine VM,

[START TUTORIAL](#)

- V. In the terminal, press **Enter** and then type in your root password.

```

CLOUD SHELL
Terminal (sit-22t3-discountmate-46bf512) X + -
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to sit-22t3-discountmate-46bf512.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
s300054233@cloudshell:~ (sit-22t3-discountmate-46bf512) $ gcloud sql connect dcm-mysql-db --user=root --quiet
Allowlisting your IP for incoming connection for 5 minutes...done.
Connecting to database with SQL user [root].Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 433015
Server version: 8.0.26-google (Google)

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>

```

- VI. Please execute your SQL command / query to the cloud database in this terminal command line.

3.1.6. Create Database Schema and Load Initial Data

Before creating database schema, please create a database and related database user accounts in the MySQL cloud DB. Detail commands could be searched from Internet.

Once the project database is created in MySQL, please use the below script file to create database schema accordingly.

[DiscountMate-db-schema-initial-data.sql](#)

3.1.7. Source Data

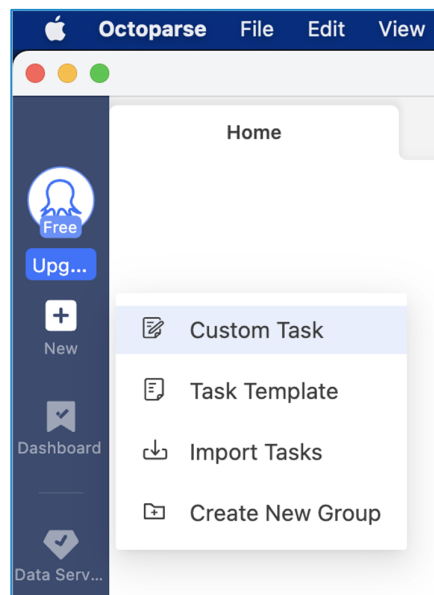
Source data of the data platform are as below:

1. Weekly item price. From web scrapping tool weekly exported CSV datafile.
2. User buying transactions. From user scanned receipt and Woolworths provided test transaction data file.
3. Item images. Download from Internet.

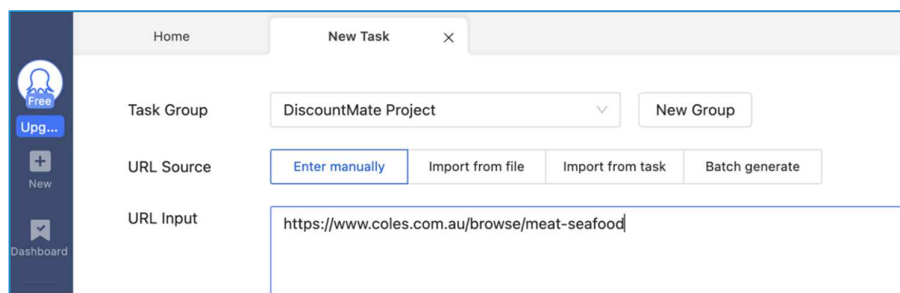
3.1.7.1. Woolworths and Coles Weekly Price Data

The data are generating weekly from the web scrapping tool Octoparse. Please follow below steps for your weekly source data generation:

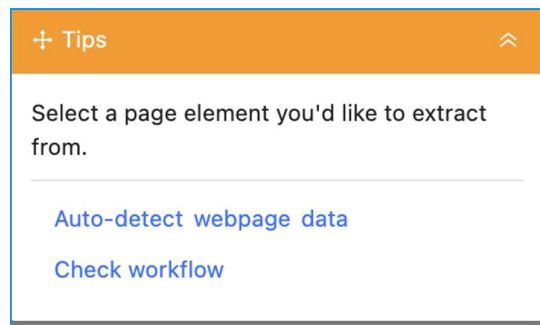
1. Download and install Octoparse software to your Windows or Mac computer.
2. Run the software, register your free version account.
3. Click **New** on the left menu. Then click **Custom Task**.



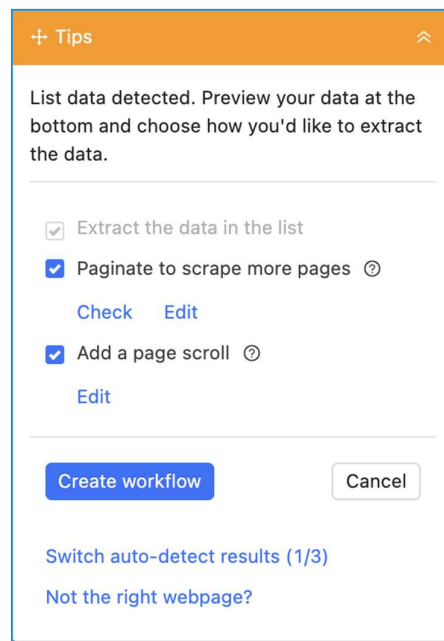
4. Paste the target URL in the **URL Input** like below screen. Then click **SAVE**.



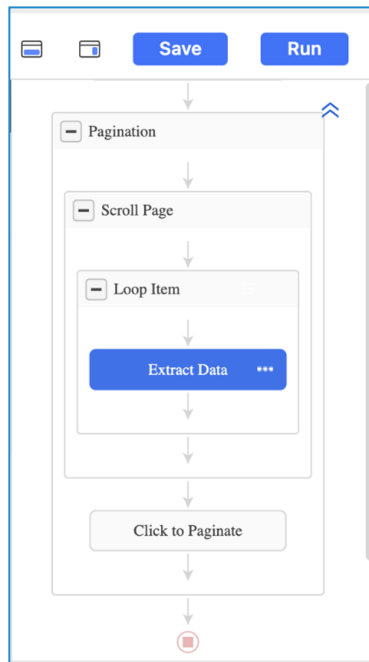
5. On the workflow screen. Click **Auto-detect webpage data**.



6. After selecting the correct scraping detail, click **Create workflow**.



7. At the top right corner, click RUN to execute the workflow.



8. After job completed, click Export Data, choose the export file format and target export location. Then the data file will be exporting accordingly.
9. Repeat step 3 – 8 to export all categories items from Woolworths and Coles web site.

In detail, from Woolworths and Coles website, we will extract:

- I) Item Code (which is extracted from the item URL for Woolworths website)

| No. | Title_URL | shelfproducttileimagewrapper_URL |
|-----|-------------------------|--|
| 1 | Ambrosia Apple Each | https://www.woolworths.com.au/shop/productdet... |
| 2 | Apple Bravo Punnet 750g | https://www.woolworths.com.au/shop/productdet... |
| 3 | Apple Royal Gala Each | https://www.woolworths.com.au/shop/productdetails/90622/ambrosia-apple |
| 4 | Apricot Fresh Each | https://www.woolworths.com.au/shop/productdet... |

(Item Code is not available on Coles web site so we will match Coles item name with ITEM_NAME column in ITEM table for getting the ITEM_ID to use)

- II) Item Description
- III) Item Price

Items in 12 categories (below URL sources) will be exported: Fruit and vegetables, meat, seafood and deli, bakery, freezer, pet, baby, dairy, eggs, and fridge, drinks, liquor, pantry, health and beauty, and household.

| Category | URL |
|----------------------|---|
| Fruit & Veg | https://www.woolworths.com.au/shop/browse/fruit-veg?sortBy=Relevance |
| Meat, Seafood & Deli | https://www.woolworths.com.au/shop/browse/meat-seafood-deli?sortBy=Relevance |
| Bakery | https://www.woolworths.com.au/shop/browse/bakery?sortBy=Relevance |
| Freezer | https://www.woolworths.com.au/shop/browse/freezer?sortBy=Relevance |
| Pet | https://www.woolworths.com.au/shop/browse/pet?sortBy=Relevance |
| Baby | https://www.woolworths.com.au/shop/browse/baby?sortBy=Relevance |
| Dairy, Eggs & Fridge | https://www.woolworths.com.au/shop/browse/dairy-eggs-fridge?sortBy=Relevance |
| Drinks | https://www.woolworths.com.au/shop/browse/drinks?sortBy=Relevance |
| Liquor | https://www.woolworths.com.au/shop/browse/liquor?sortBy=Relevance |
| Pantry | https://www.woolworths.com.au/shop/browse/pantry?sortBy=Relevance |
| Health & Beauty | https://www.woolworths.com.au/shop/browse/health-beauty?sortBy=Relevance |
| Household | https://www.woolworths.com.au/shop/browse/household?sortBy=Relevance |

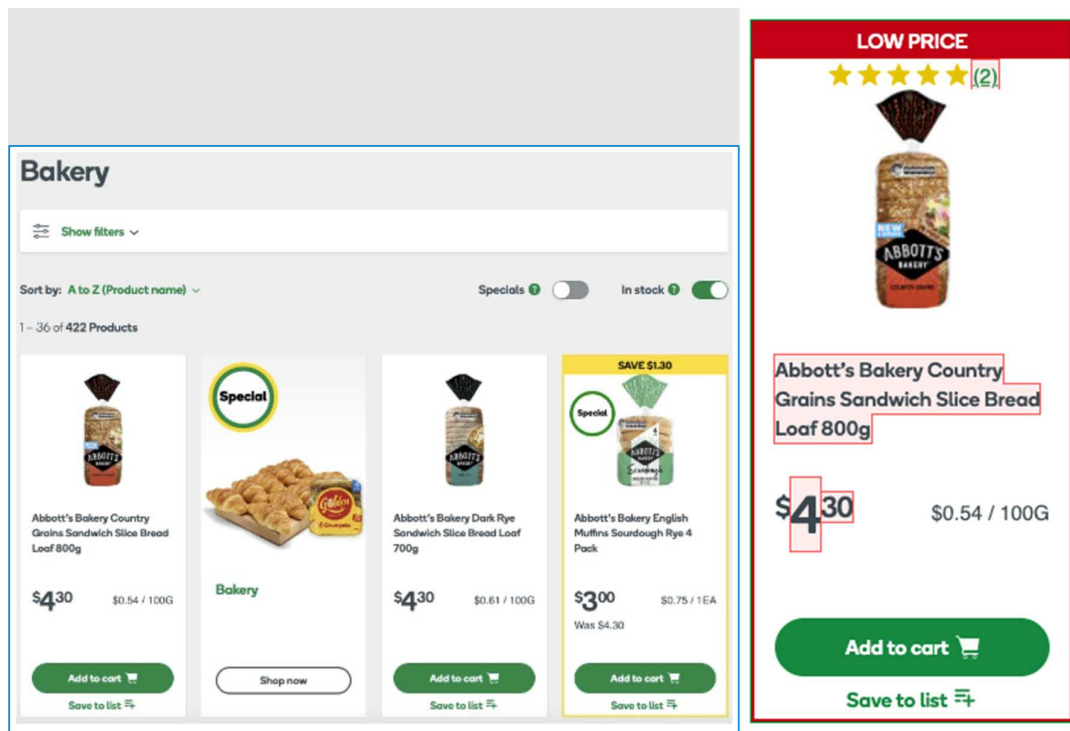
To perform the task, Octoparse tasks are using to scrape required data from Woolworths and Coles website weekly on Thursday. Since supermarket has been announcing new promotion every Wednesday and weekly catalogue lasts for 7 days, we will then have the most updated data that is effective for a week. The 12 categories of items are divided into 5 parsing tasks with reasonable batch size to meet the requirement (limited to 10,000 lines) in order to use this software freely (below screenshot).

| Task Name | Status | Next Run | Time Spent |
|-------------------------------|--|----------|-----------------------------|
| Batch1 My Group 7 days ago | <div>Run</div> <div>Completed 8452 lines (21 duplic...</div> | Not Set | - |
| Batch2 My Group 7 days ago | <div>Run</div> <div>Completed 4853 lines (85 duplic...</div> | Not Set | 38m 49s (24 minutes ago) |
| Batch3 My Group 7 days ago | <div>Run</div> <div>Completed 4622 lines</div> | Not Set | - |
| Batch4 My Group 7 days ago | <div>Run</div> <div>Completed 4620 lines</div> | Not Set | 24m 46s (a few seconds a... |
| Batch5 My Group 7 days ago | <div>Run</div> <div>Completed 4623 lines (5 duplica...</div> | Not Set | 28m 33s (an hour ago) |
| | | Not Set | - |
| | | Not Set | 33m 10s (13 minutes ago) |
| | | Not Set | 31m 19s (an hour ago) |

For each task we first need to supply a list of URLs that we want to go through and set up rules for data parsing.



We will start with a loop through a list of Woolworths URLs that we want to parse from. Once we are on page 1 of the website, for example, Bakery.



We choose only in stock items by triggering the In-stock toggle. Then, software will loop through all the items on this page and extract required data and move on to next page. It

will loop items and extract data again till the last page. And it moves on to the next category on the URL list until the task is completed.

3.1.8. Data Ingestion

3.1.8.1. Weekly Price Data

After collecting exported data files of Woolworths and Coles weekly item price, data ingestion python scripts are used to import the source data file to the staging tables of the data platform. Staging tables are serving as the landing place of the imported data. Staging tables data type are all in text format to reduce the chance of import error due to data quality issue.

Python script execution: The following collected Woolworths and Coles weekly item price data has been converted to pandas before being connected to the database and executing loops to run each row within the converted tabular form (pandas) and insert those values to the respective table columns.

Python scripts:

- [etl 01 import wooly datafile mysql.py](#)
- [etl 02 import coles datafile mysql.py](#)

3.1.8.2. User Buying Transaction Data

Similar to weekly item price data, data ingestion python scripts are used to import the source data file to the staging tables of the data platform.

Python script:

[etl 03 import txn datafile mysql.py](#)

3.1.8.3. Item Images

Unlike the above two data source, import of item images involved more manual steps:

1. Manual download item images from Internet.
2. Manual resize image files to 300 x 300 pixels.
3. Execute python script to batch convert image files to base64 string.
4. Compose insert statement to import image base64 string to ITEM_IMAGE table.
5. Compose update statement to update the image corresponding item record in ITEM table ITEM_IMAGE_ID column.

After creating a 'txt' and saving it to a variable, then add the newly created variable within the open function as in write mode for opening a file for writing, creates a new file if it does

not exist, or truncate the file if it exists. Basically, getting a list of all files with the ".jpeg" extension in the current directory has been performed. After opening the image directory, read the file content as binary data and encode the binary data as a Base64 string, then print it out.

3.1.9. Data Cleansing and Transformation

Once all staging tables are imported with latest source data, data cleansing and transformation job could be executed. Please run below python scripts in the sequential order stated.

Python scripts:

- [etl 04 insert item from transaction.py](#)
- [etl 05 insert item price current.py](#)
- [etl 06 insert item price history.py](#)
- [etl 07 insert user order.py](#)
- [etl 08 insert order item.py](#)

3.1.10. Verification of Data Loading

After running all data cleansing and transformation python scripts, below tables in database need to verify the correctness of the imported data against source data files:

1. ITEM_PRICE_CURRENT (source data: weekly item price data files)
2. ITEM_PRICE_HIST (source data: weekly item price data files)
3. ITEM_IMAGE (source data: downloaded item images)
4. USER_ORDER (source data: transaction data files)
5. ORDER_ITEM (source data: transaction data files)

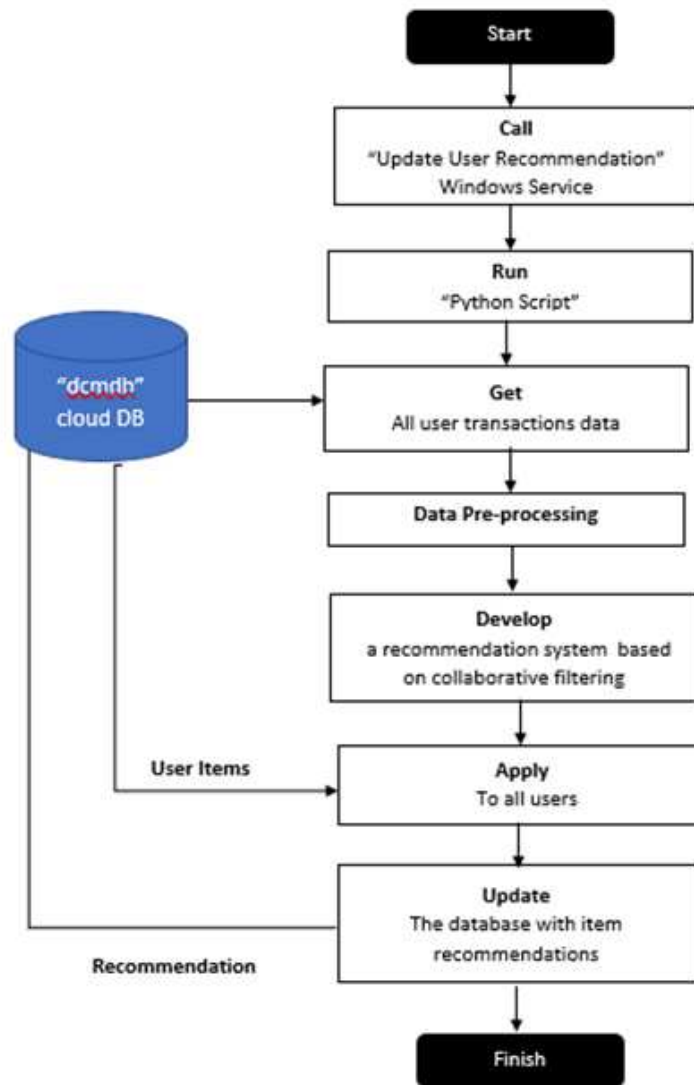
It is recommended to use 2 verification methods to check if there are any different in the value between source data files and the above database tables:

- i. Check number of records
- ii. Checksum of the price / invoice number fields from the dataset.

After confirming the data correctness, data platform is ready to use by front-end team and data science team. Please inform the two team on the data readiness accordingly.

3.2. Data Engineering

3.2.1. Prediction model process flow



3.2.1. The Process flow in details

1. **Calling windows service and run the script.** The recommendation system is executed by calling associated window service scheduled to run in the application cloud server. Next, the python script is executed.
2. **Getting user's transactions data.** Collaborative filtering is a recommendation system that creates a prediction based on a user's previous behaviours. At this step, data related to behaviours is retrieved from the database, and this includes USER ID, ITEM ID and Name, COMPANY ID, Transaction date, Number of transactions, average item price, the current price, and the discount price.
3. **Data pre-processing.** This step includes:

- Handling NULL values. Particularly for the discount price which is used to compute the discount percentage. The value is considered zero in case of missing; means no discount.
 - Compute “item recent factor” which represents the difference in days between current date and the recent latest date an item is purchased.
- 4. Building the recommendation system.** The collaborative filtering system is based on finding the similarity between items based on certain criteria. For the given system, we decided to compute the similarity based on:
- Item transaction count.
 - Recent factor: the difference between transaction date and today in number of days.
 - DPCT or Discount percentage/ratio between the current and the discounted prices.

Based on app analysis, we believe that these criteria are most sufficient parameter combination to express correlation between items.

- 5. Applying the recommendation system to all users.** At this step, the system will compute the similarity scores between user items and all items. Items with highest mean score values are returned.
- 6. Updating the database.** The last step is writing updated item recommendations to the database to be displayed at the UIs.

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL POLYGLOT NOTEBOOK
PS C:\Users\61416\Desktop\projectCode> & C:/ProgramData/Anaconda3/envs/RQ3/python.exe c:/Users/61416/Desktop/projectCode/DiscountMe_Recommender.py
User Recommended Items..
  User ID Item ID Company ID Score
0 1003 60058 1 0.4010778687269961
1 1003 199418 1 0.4010778687269961
2 1003 345864 1 0.4010778687269961
3 1003 577860 1 0.4010778687269961
4 1003 807863 1 0.4010778687269961
5 1003 505074 1 0.39672296278546704
6 1003 120080 1 0.39672296278546704
7 1003 505069 1 0.39672296278546704
8 1003 728313 1 0.39672296278546704
9 1003 813764 1 0.39672296278546704
User Recommended Items..
  User ID Item ID Company ID Score
0 1004 117573 1 0.2876944080422449
1 1004 144329 1 0.2876944080422449
2 1004 170225 1 0.2876944080422449
3 1004 210245 1 0.2876944080422449
4 1004 134801 1 0.2830151753937079
5 1004 149963 1 0.28053960511755593
6 1004 162701 1 0.2749801297069234
7 1004 251007 1 0.2677454312808375
8 1004 841640 1 0.2677454312808375
9 1004 39319 1 0.2677454312808375
User Recommended Items..
  User ID Item ID Company ID Score
0 1001 608712 1 0.21738561893915317
1 1001 5554 1 0.20242511095500082
2 1001 144329 1 0.20242511095500082
3 1001 22033 1 0.20242511095500082
4 1001 33557 1 0.20242511095500082
5 1001 135552 1 0.20242511095500082
6 1001 83995 1 0.20129843003316678
7 1001 721116 1 0.20129843003316678
8 1001 41946 1 0.19990931837724005
9 1001 147197 1 0.19963049112562095
User Recommended Items..
  User ID Item ID Company ID Score
0 1002 168 1 0.17065239806053464
1 1002 927666 1 0.17065239806053464
2 1002 577861 1 0.17065239806053464
3 1002 117573 1 0.17065239806053464
4 1002 370814 1 0.14190388511261573
5 1002 33557 1 0.13461154900404812

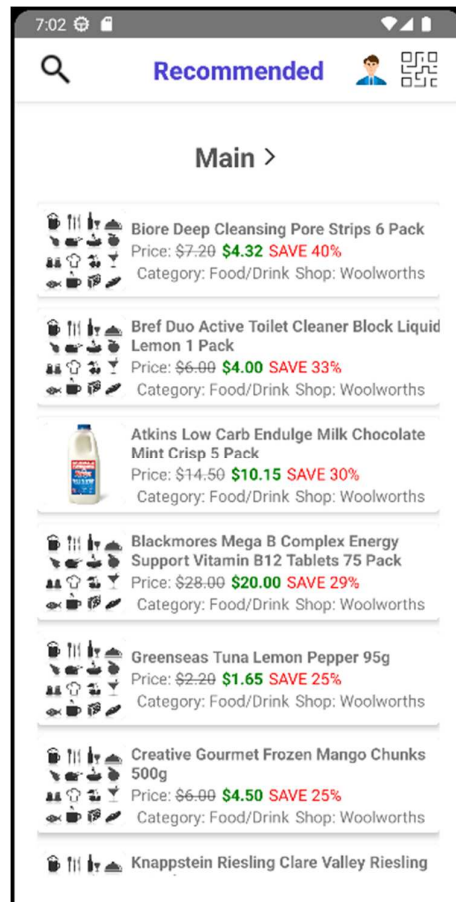
```

Source code: https://github.com/discountmate/dm_app/tree/main/Backend/util

4. Completed Deliverables

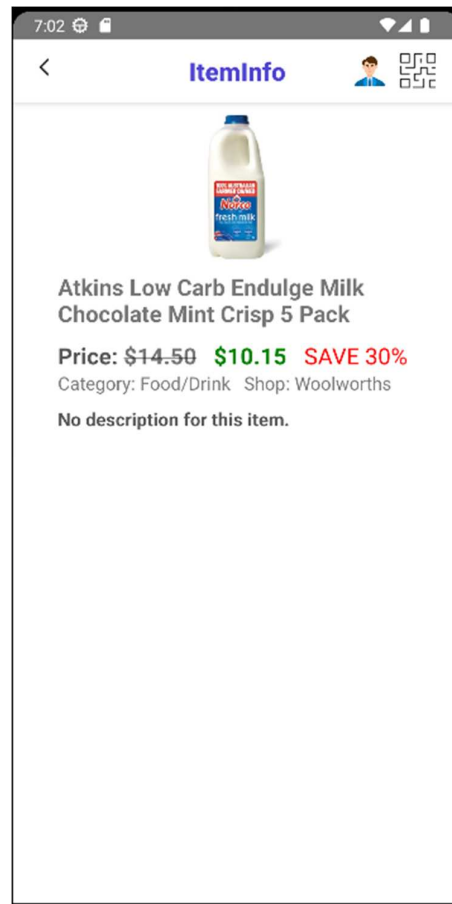
1. Recommended Items:

- Main page of the application after user successfully logs in.
- Displays item recommendations based on user's previous purchases and maximum possible savings (discounted prices)
- Displays product name, image, category of product, actual price and discounted price.
- Search icon at the top opens search items page.
- A person's avatar image opens user profile page.



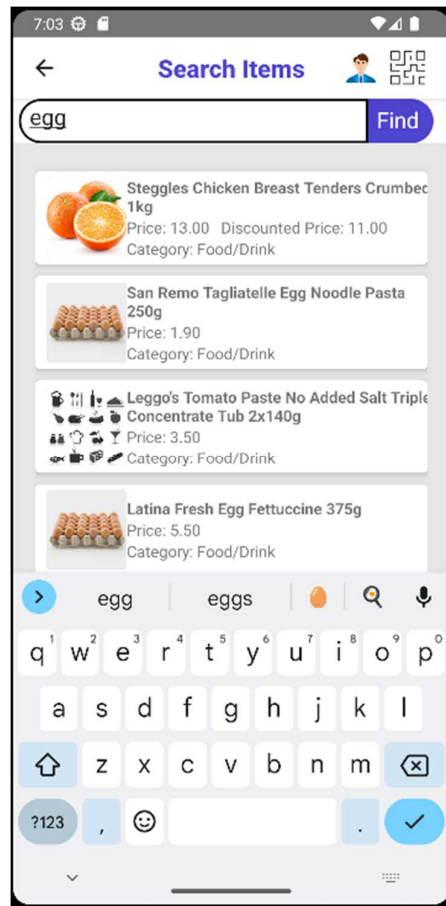
2. Item Info:

- Clicking any product on the recommended items page opens the item information page.
- Item image, name, base price, discounted price, percentage savings, category of product, other description (if available) are shown.
- Shows shop name where the selected product can be purchased.



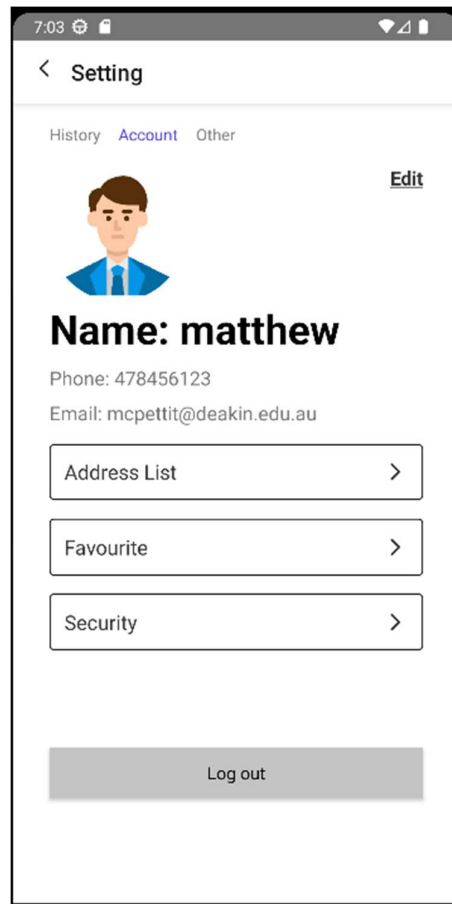
3. Search Items:

- a. System provides functionality to search items that are not displayed on the recommended items page.
- b. User enters item full description or partial word and clicks on Find button.
- c. Item details are retrieved from database and displayed on screen.
- d. Item image, description, base price, discounted price and category are shown.

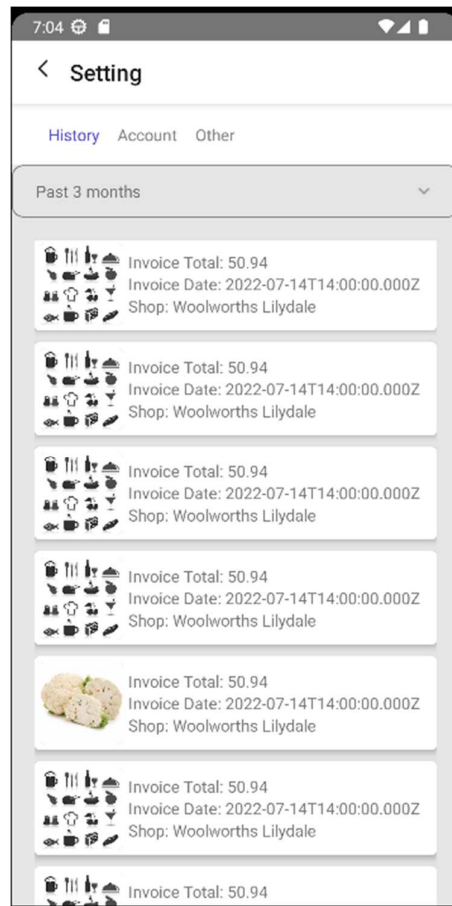


4. User Account:

- a. This page displays user details.
- b. Minor enhancement is done for this page from previous handover. Hardcoded values are now retrieved from database and displayed.



5. History:
- As per proposed production version of application, the user would upload item purchase receipts. These invoices are stored inside database.
 - Under the history page, these invoices are retrieved and displayed.
 - User gets an option to select from 3 options: Past 1 week, Past 1 month or Past 3 months.
 - Depending on time frame selected, invoice information is retrieved and displayed.
 - Invoice total, Invoice date-time and shop information is shown.



6. Item Price Data:
 - a. Enhanced web parsing function to expand the weekly product price data collection for Coles.

Coles | Batch 1

231

Data Extracted

Running
[Click to Paginate] Waiting for Ajax to load...

Duplicates: 0 line(s)
Time Spent: 40s
Avg. Speed: 342 lines/min

Task Overview
Data List
Event Log
Recent Runs

| # | Title | Title_URL | Image | Price | product__prcin... | product__prcin... | product__prcin... |
|----|----------------------|---------------------|----------------------|---------|-----------------------|-------------------|-----------------------|
| 1 | Cleaver's Beef M... | https://www.cole... | data:image/gif;ba... | | | | |
| 2 | Coles Organic Be... | https://www.cole... | data:image/gif;ba... | \$12.00 | | | \$24.00 per 1kg ... |
| 3 | Coles Tasmanian ... | https://www.cole... | data:image/gif;ba... | \$13.00 | | | \$40.00 per 1kg ... |
| 4 | Coles Finest Thic... | https://www.cole... | data:image/gif;ba... | \$6.50 | Save \$1.00 | Save \$1.00 | \$21.67 per 1kg ... |
| 5 | Coles Lamb Meat... | https://www.cole... | data:image/gif;ba... | \$6.94 | Final price is bas... | | \$7.00 per 1kg |
| 6 | Coles Beechwoo... | https://www.cole... | data:image/gif;ba... | \$27.90 | Final price is bas... | | \$9.00 per 1kg |
| 7 | Coles Graze Gras... | https://www.cole... | data:image/gif;ba... | \$20.46 | Final price is bas... | | \$62.00 per 1kg |
| 8 | Luv A Duck Rend... | https://www.cole... | data:image/gif;ba... | \$4.80 | Save \$1.20 | Save \$1.20 | \$24.00 per 1kg ... |
| 9 | Primo Gourmet C... | https://www.cole... | data:image/gif;ba... | \$7.00 | Save \$1.50 | Save \$1.50 | \$15.56 per 1kg ... |
| 10 | Rocco's Smokey ... | https://www.cole... | data:image/gif;ba... | \$8.00 | | | \$17.78 per 1kg |
| 11 | Coles Finest Beef... | https://www.cole... | data:image/gif;ba... | \$8.00 | Save \$1.00 | Save \$1.00 | \$16.00 per 1kg ... |

< 1 ... 8 9 10 11 12 >
Go to Page

7. Data availability:
 - a. Enhanced data wrangling function on cleaning raw product price data.
 - b. Transformed raw data to a well-structured dataset and load into the designed data model for the use of front-end application real-time item searching and displaying.
 - c. Supported data for item pricing trend analysis and machine learning on item recommendation.

1 • `select * from ITEM_PRICE_CURRENT;`

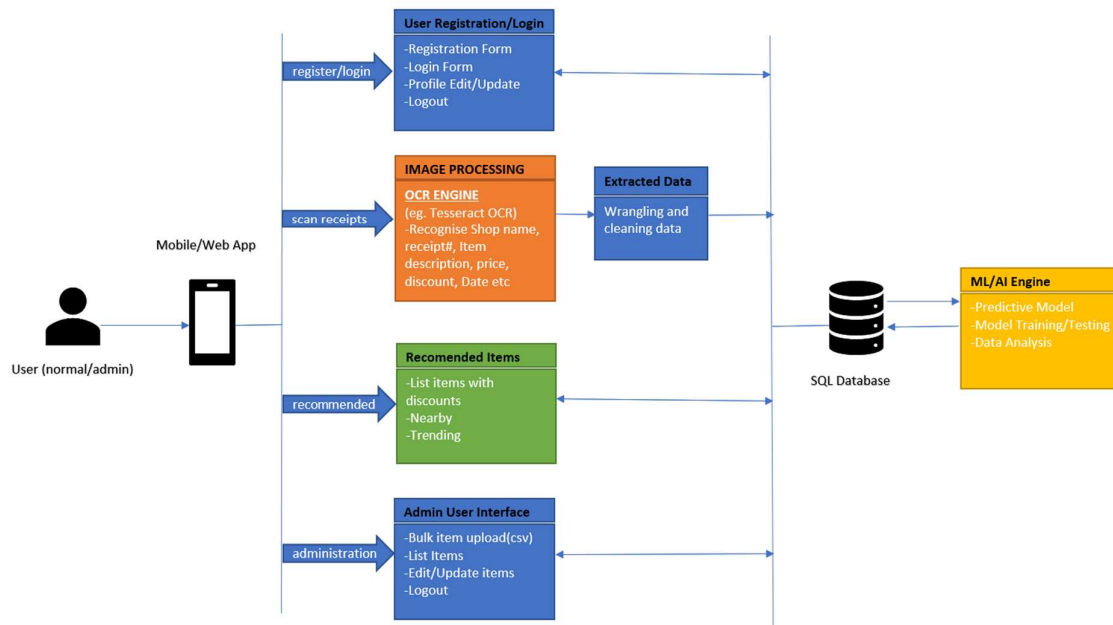
| ITEM_ID | COM_ID | IP_ITEM_BASE_PRICE | IP_ITEM_DISCOUNT_PRICE | IP_ITEM_DISCOUNT_PCT | IP_FOUR_WK_HIGHEST_PRICE | IP_FOUR_WK_LOWEST_PRICE |
|---------|--------|--------------------|------------------------|----------------------|--------------------------|-------------------------|
| 131 | 1 | 3.00 | 0.00 | 0.00 | 3.00 | 3.00 |
| 168 | 1 | 1.85 | 0.00 | 0.00 | 1.85 | 1.85 |
| 287 | 1 | 14.00 | 0.00 | 0.00 | 14.00 | 14.00 |
| 326 | 1 | 2.15 | 0.00 | 0.00 | 2.15 | 2.15 |
| 344 | 1 | 3.00 | 0.00 | 0.00 | 3.00 | 2.10 |
| 389 | 1 | 8.50 | 0.00 | 0.00 | 8.50 | 8.50 |
| 622 | 1 | 4.40 | 0.00 | 0.00 | 4.40 | 4.40 |
| 684 | 1 | 5.75 | 0.00 | 0.00 | 5.75 | 5.75 |
| 771 | 1 | 22.00 | 0.00 | 0.00 | 22.00 | 17.00 |
| 844 | 1 | 57.00 | -13.00 | -0.30 | 44.00 | 44.00 |
| 873 | 1 | 2.00 | 0.00 | 0.00 | 2.00 | 2.00 |
| 949 | 1 | 4.50 | 1.50 | 0.25 | 6.00 | 6.00 |
| 1079 | 1 | 6.00 | 0.00 | 0.00 | 6.00 | 6.00 |
| 1087 | 1 | 6.00 | 0.00 | 0.00 | 6.00 | 6.00 |
| 1100 | 1 | 3.75 | 3.75 | 0.50 | 7.50 | 7.50 |
| 1221 | 1 | 11.00 | 2.00 | 0.15 | 13.00 | 13.00 |
| 1240 | 1 | 16.00 | 0.00 | 0.00 | 16.00 | 13.00 |
| 1263 | 1 | 14.00 | 0.00 | 0.00 | 14.00 | 10.00 |
| 1469 | 1 | 8.20 | 0.00 | 0.00 | 8.20 | 8.20 |
| 1471 | 1 | 3.60 | 0.00 | 0.00 | 3.60 | 3.60 |
| 1478 | 1 | 4.32 | 2.88 | 0.40 | 7.20 | 4.30 |
| 1500 | 1 | 13.00 | 0.00 | 0.00 | 13.00 | 13.00 |
| 1536 | 1 | 9.20 | 0.00 | 0.00 | 9.20 | 9.20 |
| 1563 | 1 | 4.00 | 0.00 | 0.00 | 4.00 | 4.00 |

5. Roadmap

6. Open Issues

- Performance
 - SQL queries containing many joins perform slow at times. Enhancement is required either to query data in bulk and store locally in frontend and retrieve information as needed.
 - Item search results retrieved from SQL queries limited to 100 results to increase application responsiveness and list rendering performance, further optimizations are necessary
- Stability
 - Current ETL server has connectivity issues with the Google Cloud MySQL database server effecting stability
 - Backend server crashing can be induced by errors relating to the frontend, stability enhancements are a necessary future step
- Uniformity
 - Identical items having different item codes across various supermarket chains lead to a decrease in the performance of data analysis and machine learning algorithms
 - Due to differences in developer habits, techniques and preferences combined with a modular approach to software development, not all sections of the source code are presented in the same style or implemented in the same manner
- Redundant code
 - Many source files obtained from previous handover contains many sections of unused functions, imports and commented out code, much of which has been removed however some remains
- Other
 - Backend server login check currently accepts any password for a user account, however this can be easily rectified by uncommenting the relevant hashed password variable and enabling hashing of passwords stored in the database
 - Various software version incompatibilities exist and are mentioned in the environment setup guide
 - Administrator accounts and the administration board from previous handover unutilized at this stage in development

9. Product Architecture



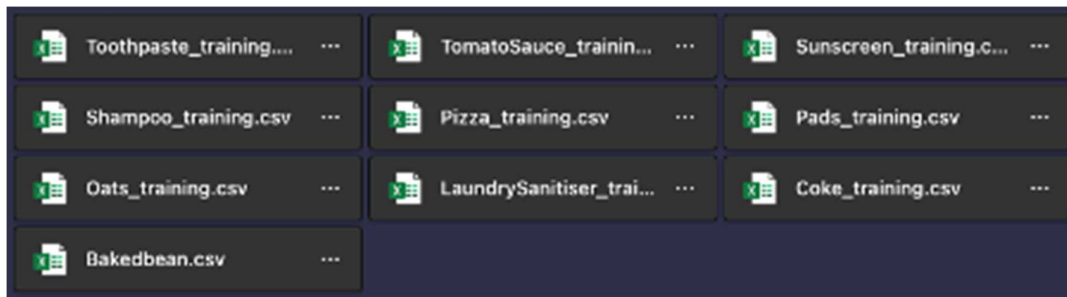
9.1. Tech Stack

| Required Software | |
|--|--|
| Git https://git-scm.com/downloads | Git is an application that facilitates interaction with the Github repository, used for obtaining the current project source code and committing any changes |
| Python 3.10.8 (Scikit-image fails on newer versions) https://www.python.org/downloads/release/python-3108/ | Python is a programming language utilized in data analysis and various backend scripts |
| Java JDK17 (newer versions currently incompatible) https://www.oracle.com/au/java/technologies/downloads/#java17 | Main application language in this project is JavaScript |
| NodeJS https://nodejs.org/en/download/ | Required backend server framework |
| Visual Studio Community Edition 2022 (used in this guide) https://visualstudio.microsoft.com/ Or Visual Studio Code https://code.visualstudio.com/ | Software development environment with support for many languages, used for developing both frontend and backend applications |
| Android Studio https://developer.android.com/studio | Android mobile application development environment, provides the ability to virtualize an android device for application testing purposes |
| MySQL Server (for local use) https://dev.mysql.com/downloads/mysql/ | Choosing to install the Developer Default options also includes MySQL Workbench |

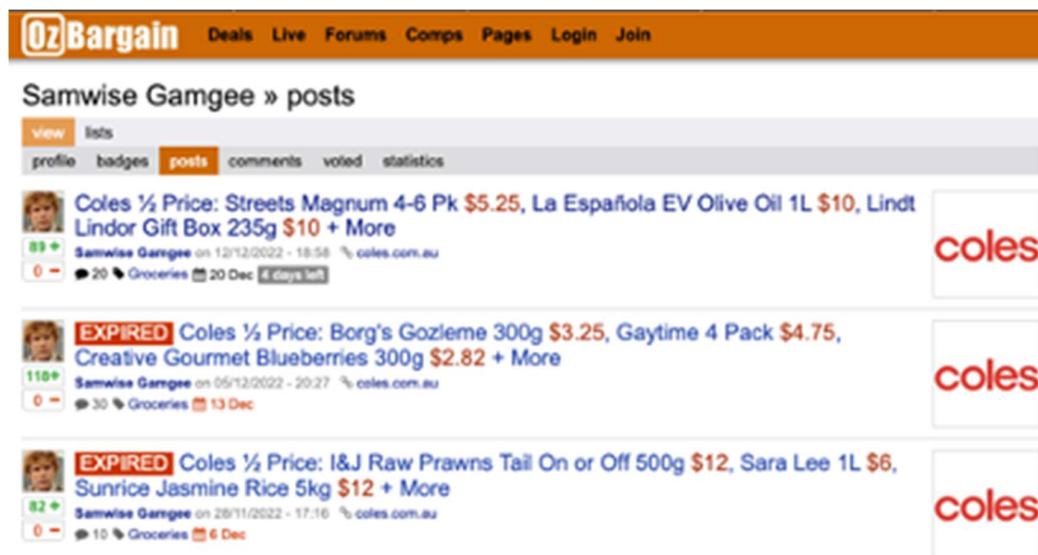
| | |
|---|---|
| MySQL Workbench (Skip if installing MySQL with Developer Default options ticked) https://www.mysql.com/products/workbench/ | Tool for connecting to and managing the MySQL database |
| MongoDB Compass https://www.mongodb.com/try/download/compass | Software for accessing the MongoDB database server, used for temporarily hosting receipt images before OCR processing |
| Tesseract https://github.com/UB-Mannheim/tesseract/wiki | Optical Character Recognition (OCR) engine used to extract information from user receipts |
| Postman https://www.postman.com/downloads/?utm_source=postman-home | API testing of requests and responses |

11.1. About our training data for Machine Learning model

Although we have thousands of records for items sold in Woolworths and Coles, we still do not have enough historical data for each item individually. Our first approach was to create dummy data with existing records. Data science team has created 10 datasets from different categories by applying discount rates manually with certain periods. However, due to the nature of dummy data (lack of time series properties) since all data is randomly created, the models did not fit well. Meanwhile, data team is still searching for a better dataset that we can apply time series models on.



Fortunately, with persevering and tireless research on the Internet, we have successfully found some useful information from Ozbargain website. The discount date, promotional price and base price in our Coca-Cola data were extracted and integrated to better interpret time series forecasting.



11.2. Machine Learning Research

In the “Time Series Forecasting Research” report attached in Other Relevant Information folder, we have briefly explained most of the key concepts for time series forecasting and introduced three most used models: SES, ARIMA and Auto ARIMA. We have also listed out the steps on how to manually find the parameters for ARIMA in order to explain the running logics of Auto ARIMA trained with the Coca-Cola dataset.

Unfortunately, results of the models are not satisfying and does not fit well. This could be a result of the data is not a good example of time series.