COMP4801 FYP Final Report

Name: Choi Tsz Long

UID: 3035705154

Group: fyp24112

Topic: E-commerce Web App for Toys

1. Project Introduction

1.1 Project Background

The toy market in Hong Kong has a high but undeveloped potential. According to research in 2019, HK was the 7th largest toy importer in the world, creating 1.5 million toy import trade value (World Integrated Trade Solution, 2019). Another example showing such market potential and people's purchasing power is the high popularity and high business volume of ACGHK, which is one of the largest events related to toys in Hong Kong. Every year it attracts tens of thousands of visitors and some spend a significant amount on buying toys there, as reported on the news (HK Ulifestyle, 2024). Unfortunately, the market's growth is limited by the following 2 major factors:

1.1.1 Insufficient online channels to buy or sell toys

Although online shopping is already popular nowadays, most local toy stores, especially those with smaller scales, do not have their own online store or provide any e-commerce services, and instead only sell toys at traditional brick-and-mortar stores or rely on other E-commerce platforms. This not only lowers the market's exposure to consumers, but also limits the market information accessible by consumers and creates an information asymmetry, hence encouraging speculative behaviours, which ultimately damages the market ecology and reduces consumers' confidence in buying toys in Hong Kong's local toy market (HK01, 2022).

1.1.2 Lack of specialization or localization for existing e-commerce platforms

Another factor is that existing e-commerce platforms where most consumers buy toys or sellers sell toys from are either not specialized for buying and selling toys or not localized enough for Hong Kong. Hence, the user experiences provided are not the most optimised. One example is Carousell, which allows its users to trade all categories of products or services, including digital products, clothes, luxuries, and even cars and properties, which are not specialized for toys. As there are too many product categories, the product recommendations are often inaccurate (See Fig 1).

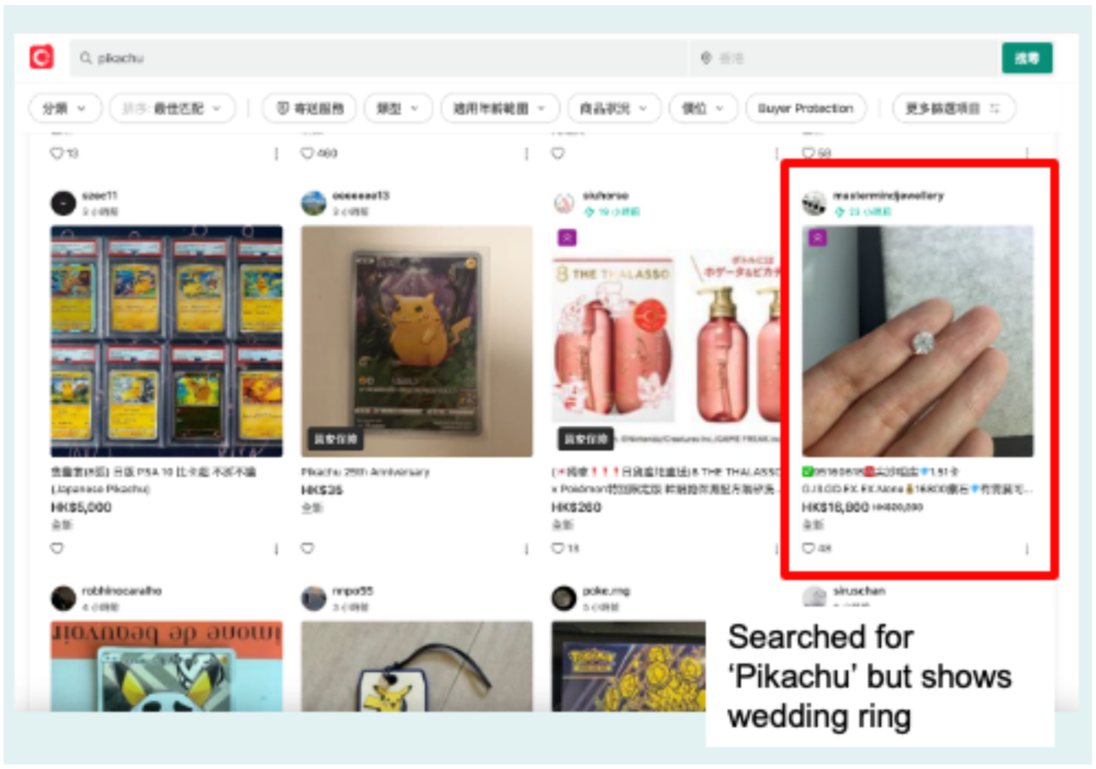


Fig 1: Example of Carousell displaying an irrelevant recommendation

Another popular platform for buying and selling toys is Taobao, which mainly serves users in Mainland China and is hence not localized to Hong Kong. For example, some sellers only offer delivery services to the Mainland, and users who live in Hong Kong need to pay for additional delivery charges. In addition, its website structure is complicated and not user-friendly for users using Taobao for the first time, e.g. it is divided into many sections or domains like Taobao, World Taobao and T Mall and the products sold from each section can be different, which is a result of the large scale and complex structure of Taobao’s website.

1.2 Project Objective

Therefore, the objective of this project is to create a one-stop E-commerce web app specialized for toys and overcome the mentioned two major factors hindering the market potential. The web app will include all essential features for both buying and selling toys, and both B2C and C2C sales models will be supported, i.e. both individuals and toy stores can become sellers. Buyers can easily search, compare products and access information of the toys they want, then buy toys at reasonable prices and avoid buying from speculators. Besides, the app’s structure and UIUX will be user-friendly and optimized for buying and selling toys. Lastly, it would be the most challenging part of this project, i.e. to create a personalised product recommendation engine with high accuracy.

2. Project Methodologies

2.1 System Architecture

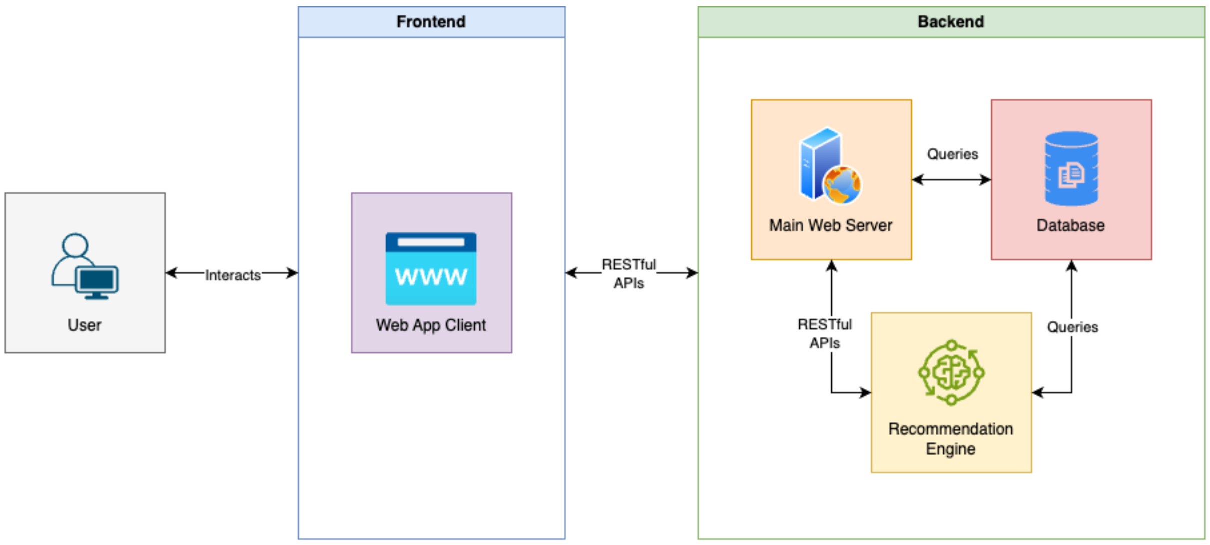


Fig 2: System architecture diagram

As shown in Fig 2, the Frontend of the web app consists of the web app client, which the user directly interacts with. It can be broken down to different webpages, e.g. Login page, Homepage, each product's detail page, Checkout page, etc., UI components on each webpage, e.g. Product Card, Product Section, navigation bar, search bar, etc.,

The Backend consists of three major components:

i) The main web server, which manipulates the data from the Frontend and other Backend components to operate the major features of the web app, e.g. product management, including storing and fetching product data from the database; User authentication including user account management and session management; and user interaction tracking, which will be mainly used for the recommendation system.

ii) The recommendation engine and its server, which generates personalized recommended products for each user and display them on the corresponding product sections on the web app client. Its workflow includes fetching user and product data from the database to train/retrain and test the recommendation models, then generating the recommended products and sending to the Frontend through the main web server. (Details for the recommendation engine's workflow will be discussed later)

iii) The database, which stores all data for the web app, including users, products, users' product reviews, user interactions (with the web app client), product recommendations.

2.2 Software Specs

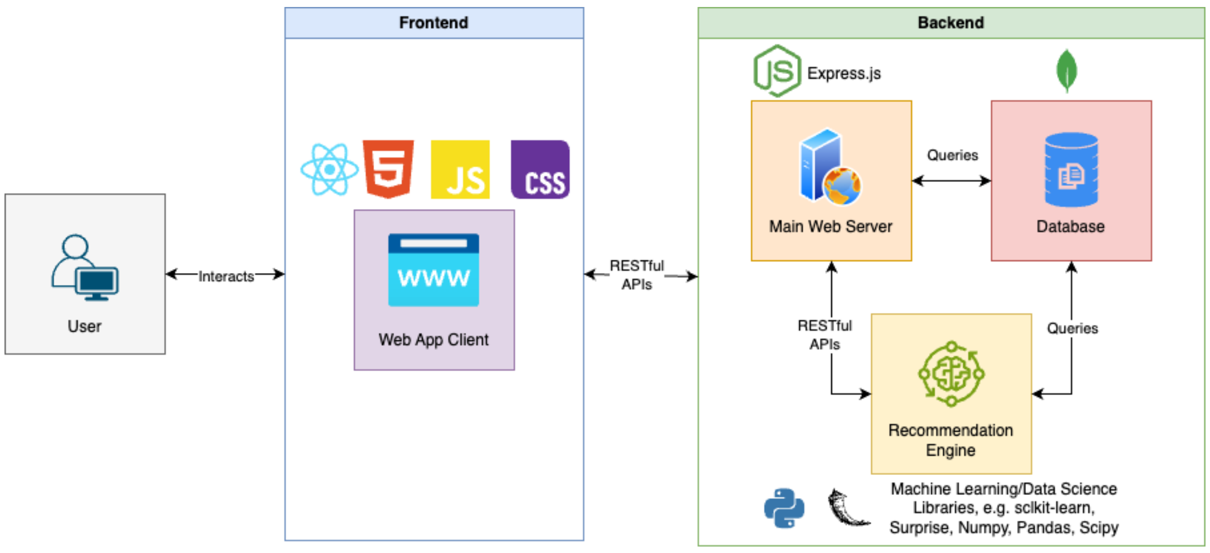


Fig. 3: System architecture diagram (with Software Specs)

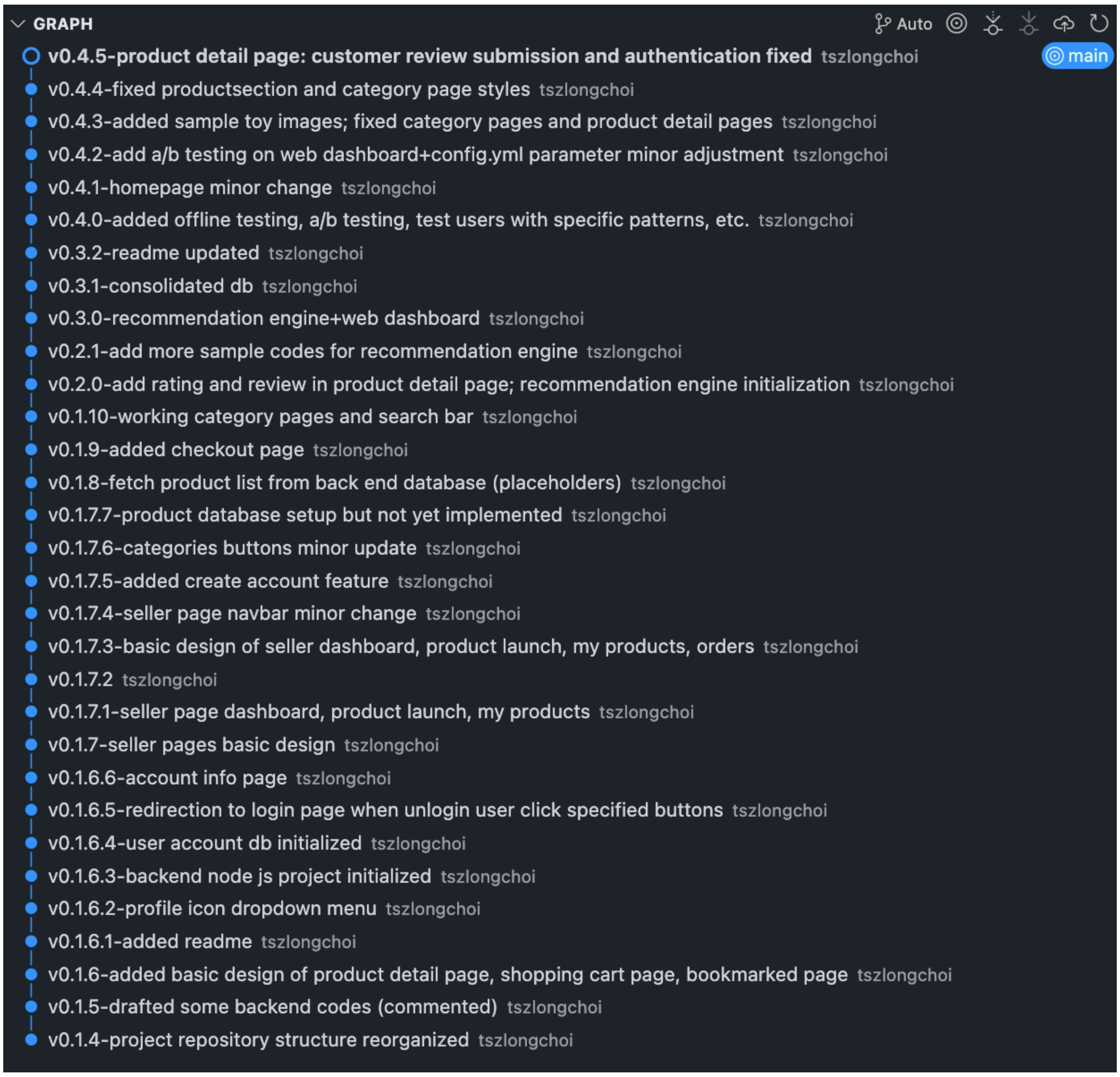
Examples of technologies used within the system are shown in Fig. 3. Typical and popular programming languages, libraries or frameworks for modern web development are used for easier development and support.

For the web app client at the Frontend, React JS which encapsulates HTML, CSS and JavaScript is used. Additionally, JSON Web Tokens are used for user validation and session management.

For the Backend, the main web server uses Node JS with Express JS and interacts with the Frontend and the recommendation engine's server with Restful APIs. The database uses the MongoDB database system. The recommendation engine is hosted by a Python Flask server and uses multiple Python Machine Learning and Data Science libraries to process user or product data collected from the database, then train the recommendation models, e.g. sclkit-learn, Surprise, Numpy, Pandas, Scipy

2.3 Project Management

A hybrid of the Agile model and the Waterfall model would be used for this project's management to take the benefits and avoid the drawbacks of each of the two models. The Agile model is adopted for the development and implementation stage, in which the whole stage is split into small sprints (e.g. only one or a part of a feature in each sprint (version), as shown in Fig. 4 below) then iteratively develop and deploy every small sprint throughout the semesters. Compared to the full Waterfall approach, i.e. developing the whole app at once, this approach allows more measurable progress and more flexible time management. Meanwhile, the other stages of the project's lifecycle (i.e. Identifying requirement, planning, designing) would adopt the Waterfall model, which has a linear and sequential timeline, as the project objectives are already fixed in early stage and do not change over time, hence the time for iterative planning or designing in the full Agile approach can be saved.


GRAPH

Auto

main

- v0.4.5-product detail page: customer review submission and authentication fixed tszlongchoi
- v0.4.4-fixed productsection and category page styles tszlongchoi
- v0.4.3-added sample toy images; fixed category pages and product detail pages tszlongchoi
- v0.4.2-add a/b testing on web dashboard+config.yml parameter minor adjustment tszlongchoi
- v0.4.1-homepage minor change tszlongchoi
- v0.4.0-added offline testing, a/b testing, test users with specific patterns, etc. tszlongchoi
- v0.3.2-readme updated tszlongchoi
- v0.3.1-consolidated db tszlongchoi
- v0.3.0-recommendation engine+web dashboard tszlongchoi
- v0.2.1-add more sample codes for recommendation engine tszlongchoi
- v0.2.0-add rating and review in product detail page; recommendation engine initialization tszlongchoi
- v0.1.10-working category pages and search bar tszlongchoi
- v0.1.9-added checkout page tszlongchoi
- v0.1.8-fetch product list from back end database (placeholders) tszlongchoi
- v0.1.7.7-product database setup but not yet implemented tszlongchoi
- v0.1.7.6-categories buttons minor update tszlongchoi
- v0.1.7.5-added create account feature tszlongchoi
- v0.1.7.4-seller page navbar minor change tszlongchoi
- v0.1.7.3-basic design of seller dashboard, product launch, my products, orders tszlongchoi
- v0.1.7.2 tszlongchoi
- v0.1.7.1-seller page dashboard, product launch, my products tszlongchoi
- v0.1.7-seller pages basic design tszlongchoi
- v0.1.6.6-account info page tszlongchoi
- v0.1.6.5-redirection to login page when unlogin user click specified buttons tszlongchoi
- v0.1.6.4-user account db initialized tszlongchoi
- v0.1.6.3-backend node js project initialized tszlongchoi
- v0.1.6.2-profile icon dropdown menu tszlongchoi
- v0.1.6.1-added readme tszlongchoi
- v0.1.6-added basic design of product detail page, shopping cart page, bookmarked page tszlongchoi
- v0.1.5-drafted some backend codes (commented) tszlongchoi
- v0.1.4-project repository structure reorganized tszlongchoi

Fig. 4: Part of the web app's version history

2.4 Project Timeline

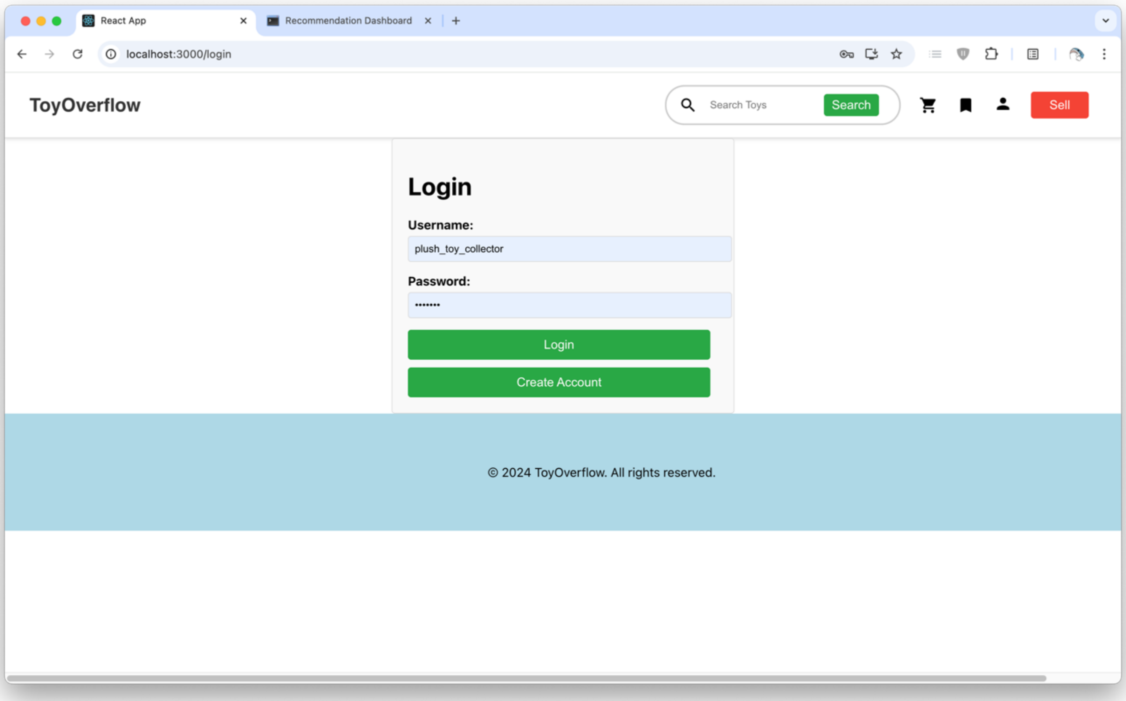
Schedule	**Tasks**
2024/08-2024/09	Identifying Requirements and Planning: ● Defining project topic, defining functional

	requirements and rough design of overall system architecture ● Preparation of Detail Project Plan
2024/10	Designing: ● Process flows/System flows diagrams ● UI design in Figma
2024/11-2024/12	Development and Implementation: ● Frontend: completed prototype for most pages of the web app client and most major features not involving backend Interim Report and Presentation
2025/01-2025/02	Development and Implementation: ● Backend: main web server
2025/03-2025/04	Development and Implementation ● Backend: recommendation engine ● Frontend: completed the remaining features involving the backend web server or the recommendation engine, e.g. fetch of recommended product lists Final Report, Presentation, Poster and Video

3. Project Results

3.1 Features of the Web App

3.1.1 User login/Create user account



Login

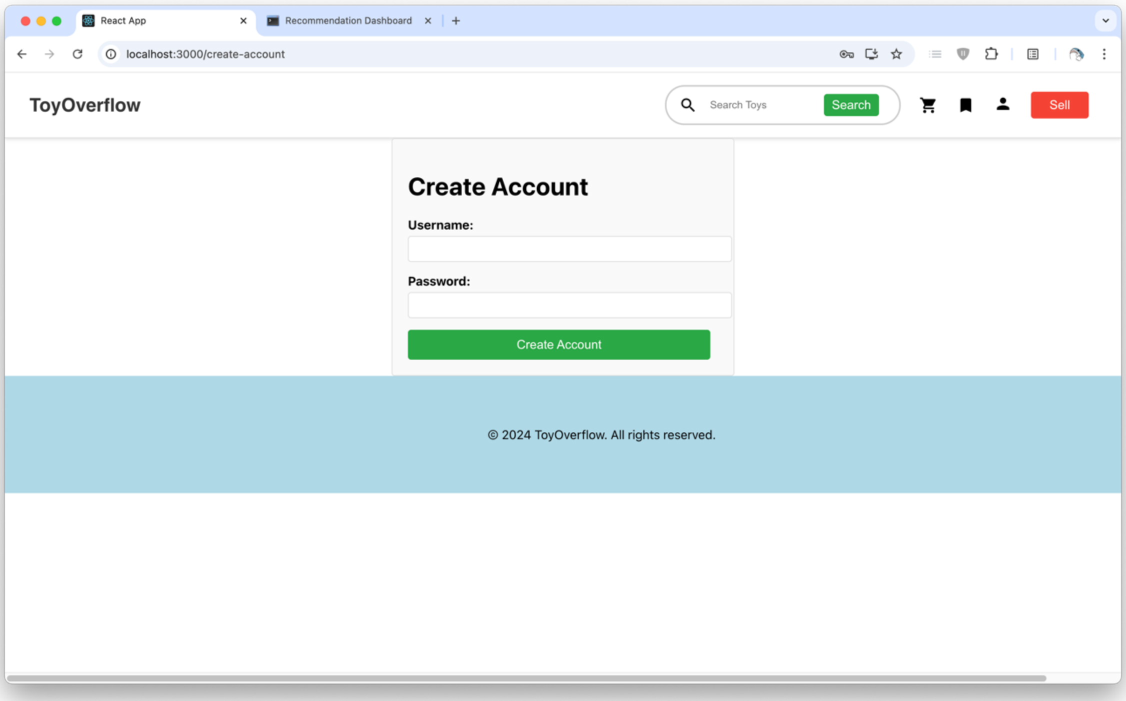
Username:

plush_toy_collector

Password:

•••••••

© 2024 ToyOverflow. All rights reserved.

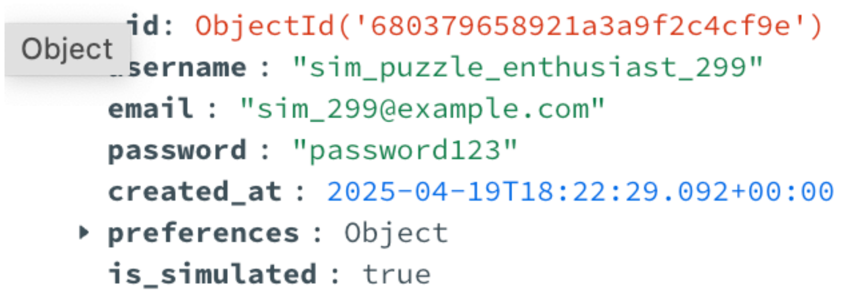


Create Account

Username:

Password:

© 2024 ToyOverflow. All rights reserved.



```
Object id: ObjectId('680379658921a3a9f2c4cf9e')
   sername : "sim_puzzle_enthusiast_299"
   email : "sim_299@example.com"
   password : "password123"
   created_at : 2025-04-19T18:22:29.092+00:00
 ▸ preferences : Object
   is_simulated : true
```

Fig. 5: Login page; Fig. 6: Create account; Fig. 7.: An entry in the Users table in the database

A user can login his account or create a new account in the login page. For the former, the user can successfully login his account if his login credentials entered in the ‘Username’ and ‘Password’ fields match with those in the users table in the database and receive a JWT token for authentication (Fig. 5). For the latter, a user can successfully create a new account if a valid username and password is entered. A new entry in the database is created and the user is given a authentication token as in the login process (Fig. 6).

3.1.2 Homepage

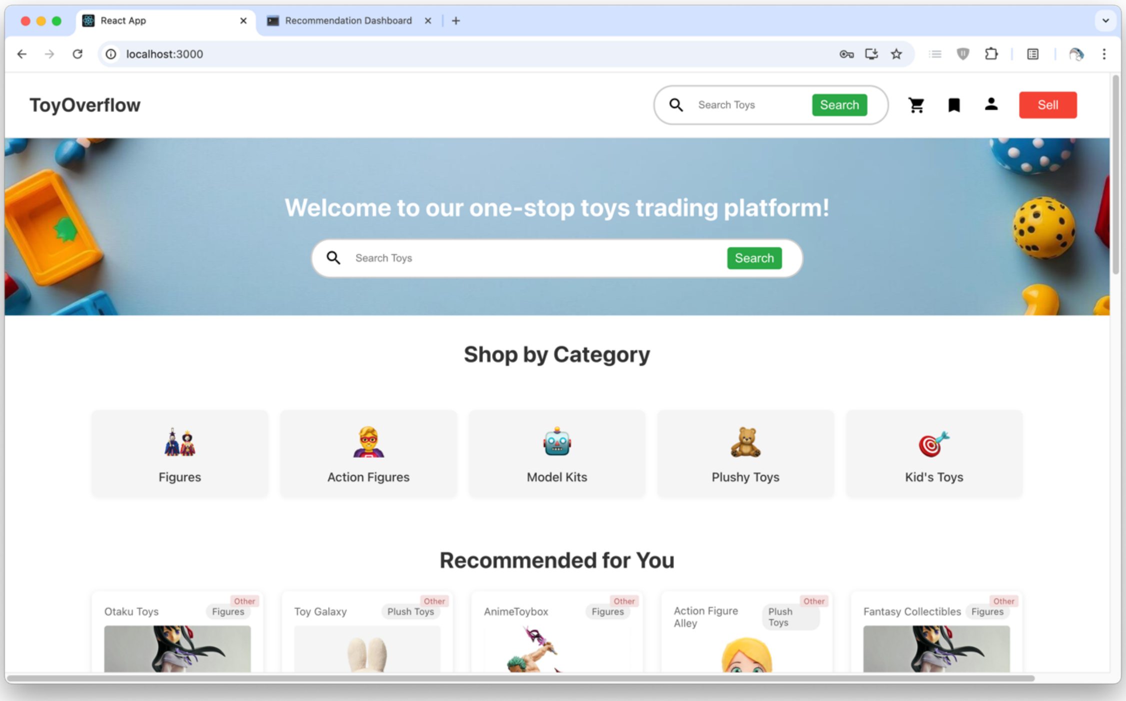


Fig 8: Homepage

The homepage is the default landing page of the web app client, regardless of whether the user has already login. On the top, there is a navigation bar with the web app's name, a search bar, buttons navigating to the shopping cart, bookmarks, account management and seller's pages. Below is the welcome banner with a search bar, then the Shop by Category section, which includes buttons for viewing products by categories.

3.1.3 Search Bar

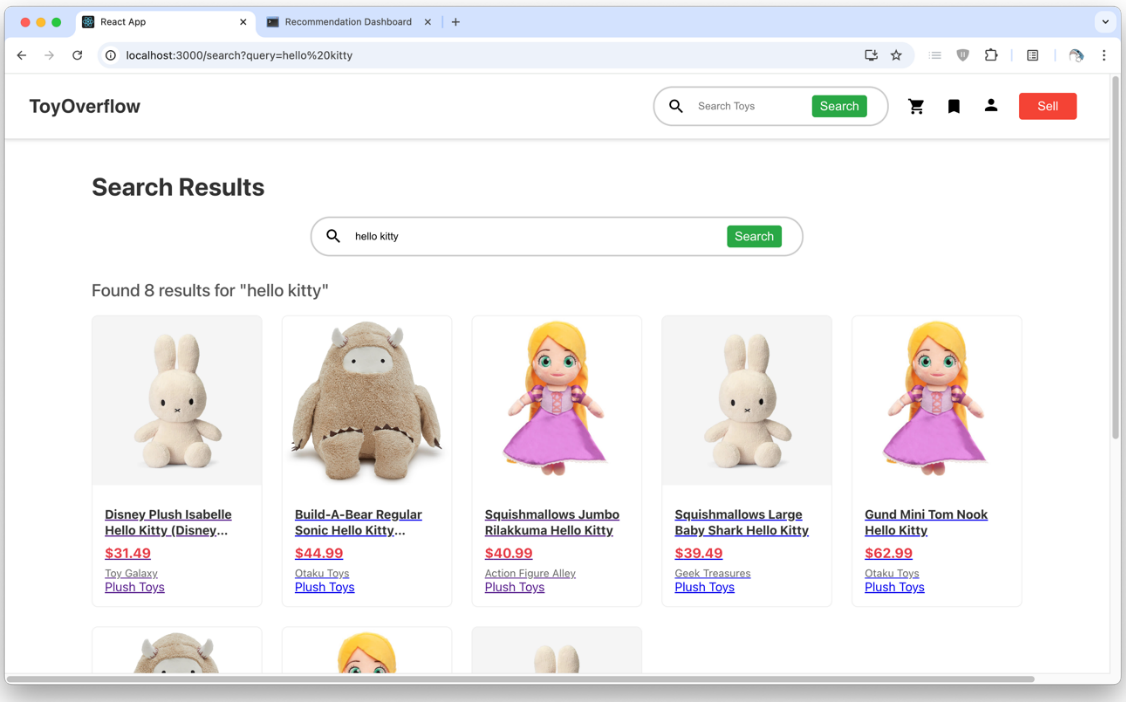

React App
Recommendation Dashboard
localhost:3000/search?query=hello%20kitty
ToyOverflow
Search Toys
Search
Sell
Search Results
hello kitty
Search
Found 8 results for "hello kitty"
Disney Plush Isabelle Hello Kitty (Disney...
$31.49
Toy Galaxy
Plush Toys
Build-A-Bear Regular Sonic Hello Kitty...
$44.99
Otaku Toys
Plush Toys
Squishmallows Jumbo Rilakkuma Hello Kitty
$40.99
Action Figure Alley
Plush Toys
Squishmallows Large Baby Shark Hello Kitty
$39.49
Geek Treasures
Plush Toys
Gund Mini Tom Nook Hello Kitty
$62.99
Otaku Toys
Plush Toys

Fig 9: Search Results

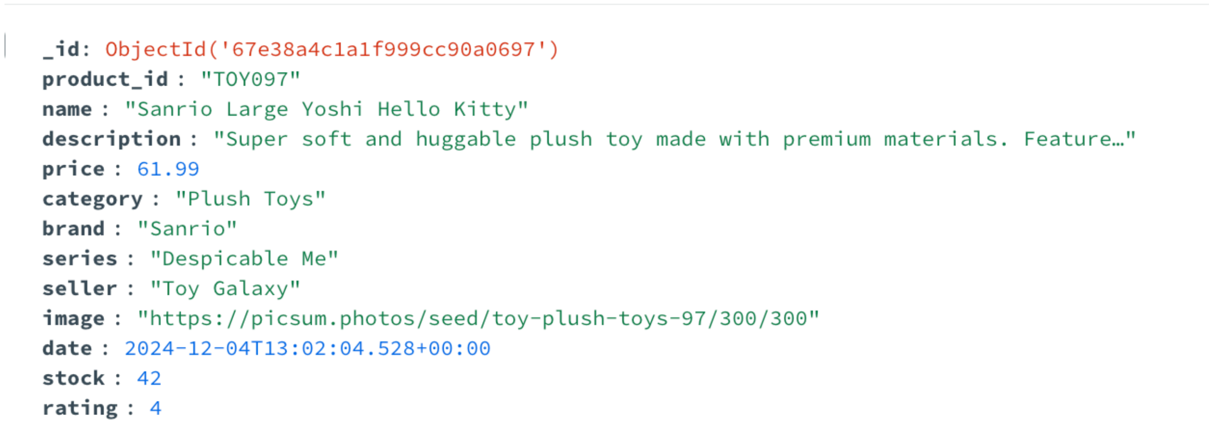

```
_id: ObjectId('67e38a4c1a1f999cc90a0697')
product_id : "TOY097"
name : "Sanrio Large Yoshi Hello Kitty"
description : "Super soft and huggable plush toy made with premium materials. Feature…"
price : 61.99
category : "Plush Toys"
brand : "Sanrio"
series : "Despicable Me"
seller : "Toy Galaxy"
image : "https://picsum.photos/seed/toy-plush-toys-97/300/300"
date : 2024-12-04T13:02:04.528+00:00
stock : 42
rating : 4
```

Fig 10: An entry in the products table in the database

Users can search for a product by its keyword using the search bar on the navigation bar or the welcome banner on the homepage. Users will be navigated to a Search Result page, where products with product name, brand or other relevant attributes matching the keyword or similar to the keyword are fetched from the products table of the database and displayed (Fig. 9, 10). Each product's seller, category, thumbnail, name, price is displayed.

3.1.4 View products by categories

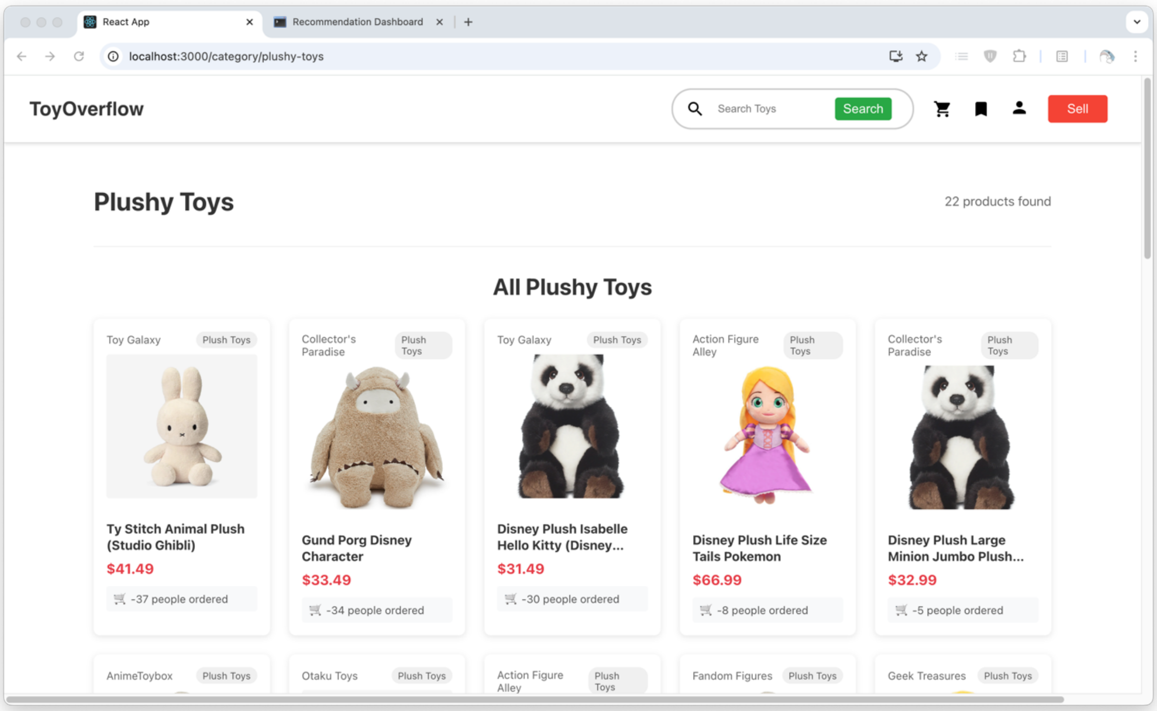
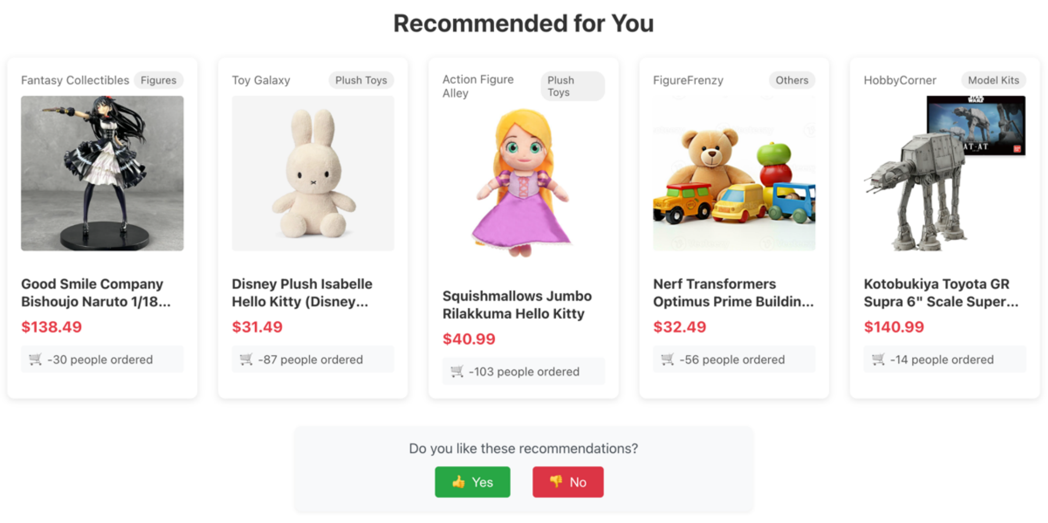


Fig 10. View products by categories

By clicking any of the 5 category buttons, i.e. Figures, Action Figures, Model Kits, Plushy Toys and Kid's Toys, in the Shop by Category section, users will be navigated to a page displaying all products of the clicked category. Each product's seller, category, thumbnail, name, price is displayed.

3.1.5 Product recommendations



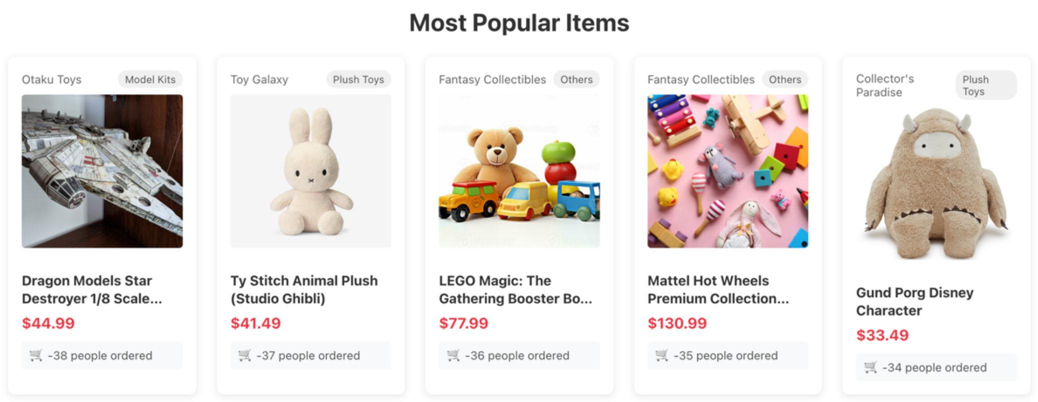


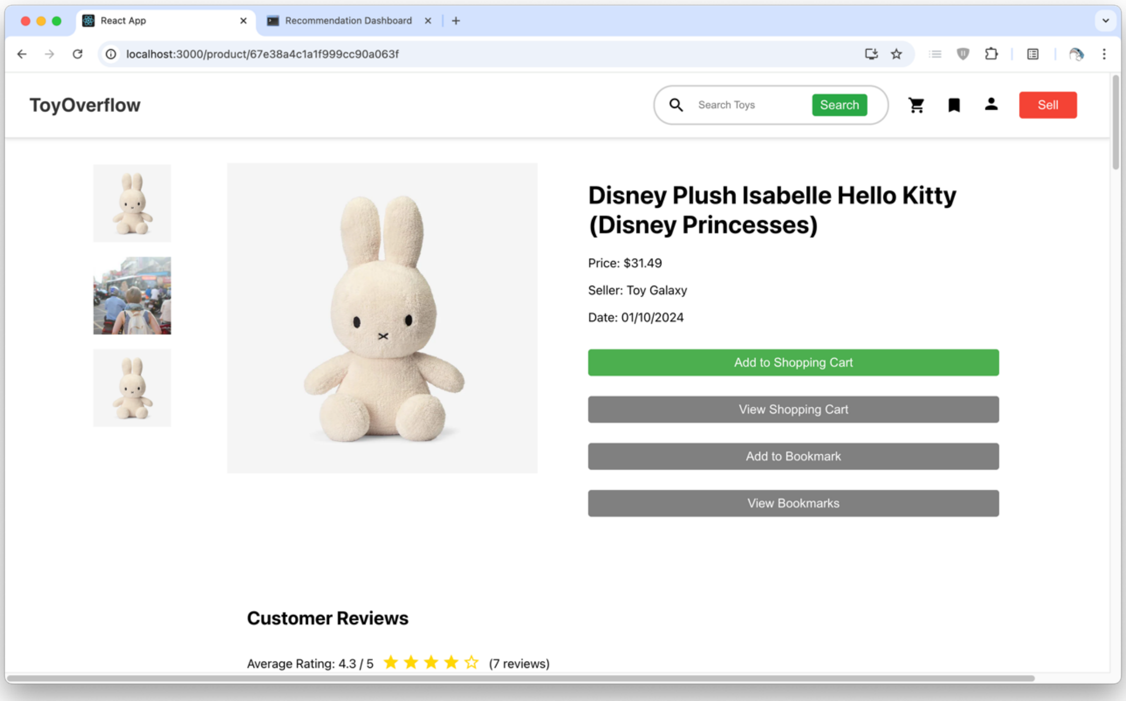
Fig. 11: Recommended for You product section; Fig. 12: Most Popular Items product section

Scrolling down the homepage, there are two product sections: Recommended for You and Most Popular Items. The former shows the top 5 personally recommended products generated by the recommendation engine (ranking calculation method to be discussed later), while the latter shows the top 5 most ordered products in the database.

For each product in the product section, the product's seller, category, thumbnail, name, price and no. of users ordered is displayed.

Below the Recommended for You section, there is a feedback section where the user can indicate whether he likes the personally recommended products by clicking either the Yes button or the No button, which is used as a reference for the recommendation model training.

3.1.6 Product detail pages



ToyOverflow

Disney Plush Isabelle Hello Kitty (Disney Princesses)

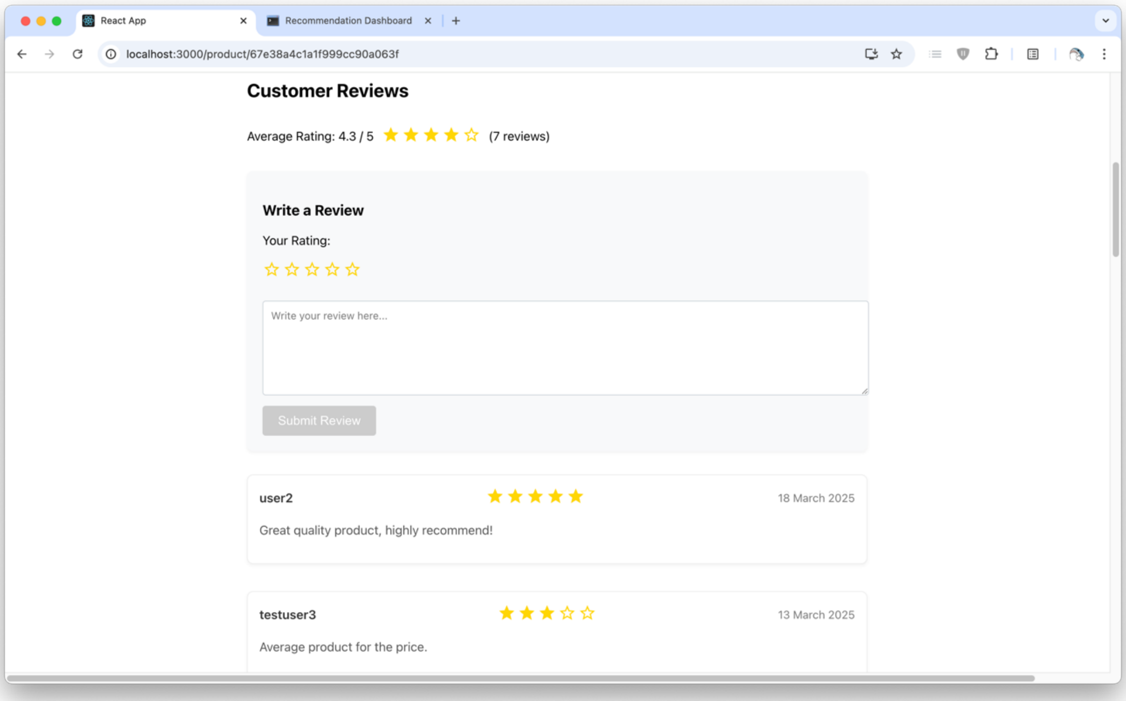
Price: $31.49

Seller: Toy Galaxy

Date: 01/10/2024

Customer Reviews

Average Rating: 4.3 / 5 ★★★★☆ (7 reviews)


React App
Recommendation Dashboard
localhost:3000/product/67e38a4c1a1f999cc90a063f
Write your review here...
Submit Review

Customer Reviews

Average Rating: 4.3 / 5 ★★★★☆ (7 reviews)

Write a Review

Your Rating: ☆☆☆☆☆

user2 ★★★★★ 18 March 2025

Great quality product, highly recommend!

testuser3 ★★★☆☆ 13 March 2025

Average product for the price.

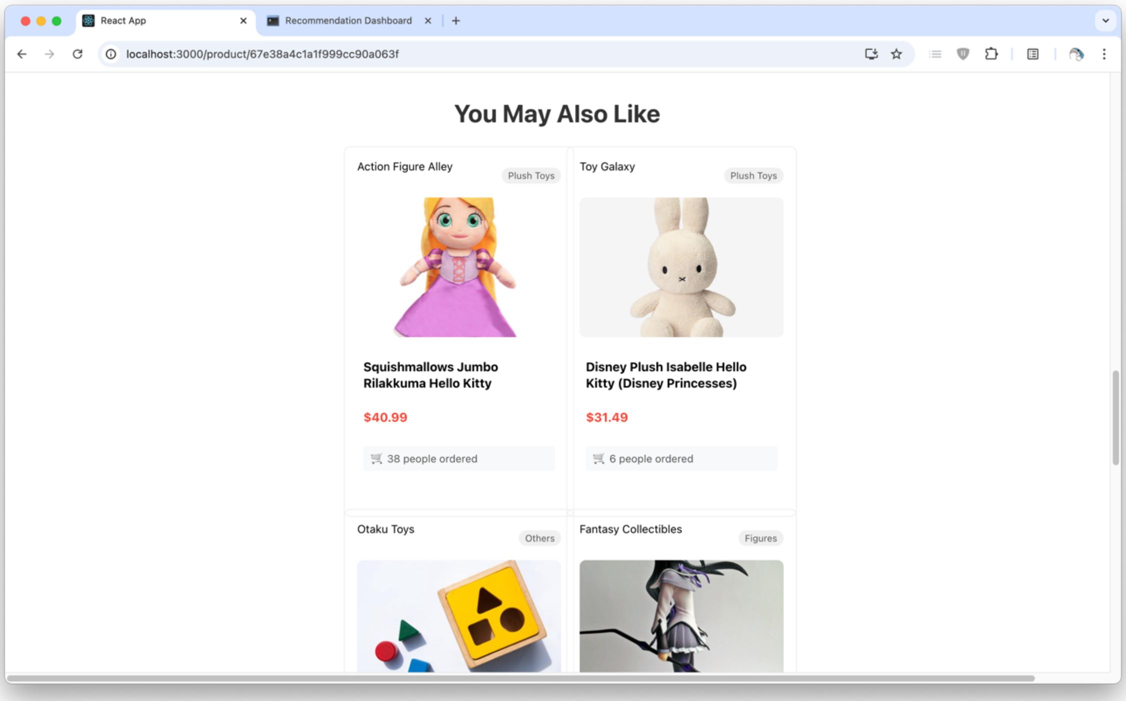
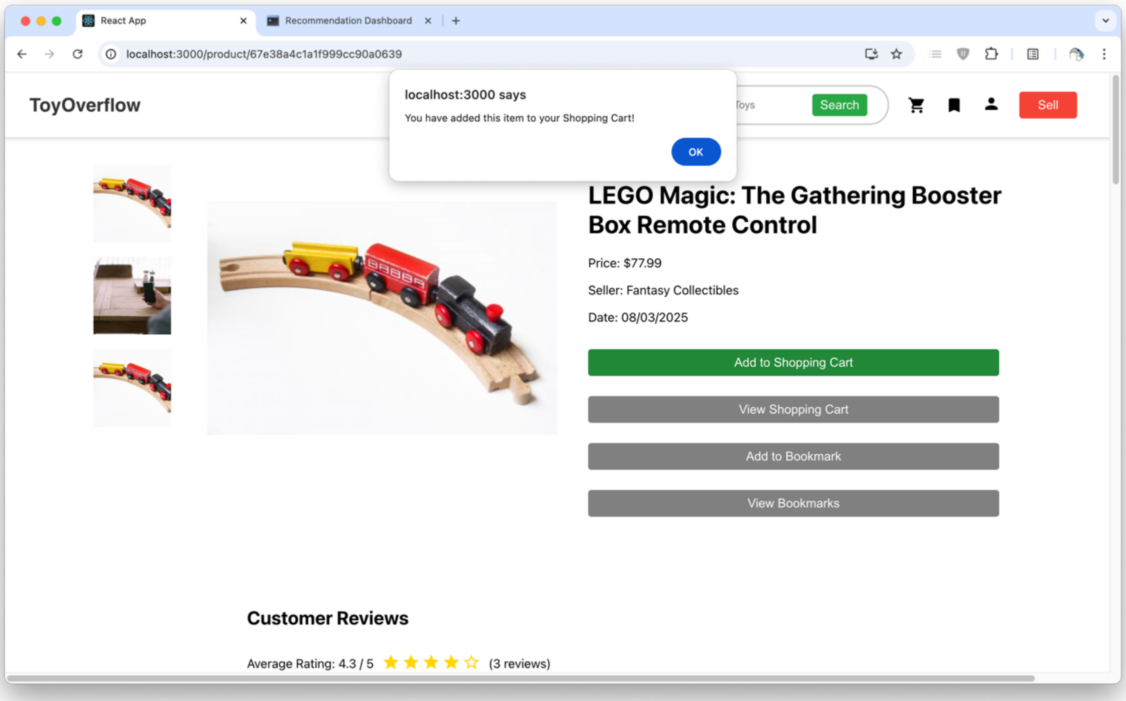


Fig. 13: Product detail section; Fig. 14: Customer Reviews section; Fig. 15: You May Also Like product section

When clicking a product on any product sections, the search result page or the shop by category pages, the user is navigated to the clicked product's detail page. The page contains i) a product detail section, which displays some thumbnails for the product, the product's detail information and buttons for adding the product to the shopping cart, adding the product to bookmarks, viewing the shopping cart and viewing the bookmarks (Fig 13); ii) a Customers Reviews section where a login user can submit his review to the product (including a rating out of 5 star and review in texts) or view other users' reviews; iii) a You May Also Like product section which displays top n personally recommended products generated by the recommendation engine, similar to the Recommended for You section in the homepage.

3.1.7 Shopping Cart and Bookmarks



localhost:3000 says

You have added this item to your Shopping Cart!

LEGO Magic: The Gathering Booster Box Remote Control

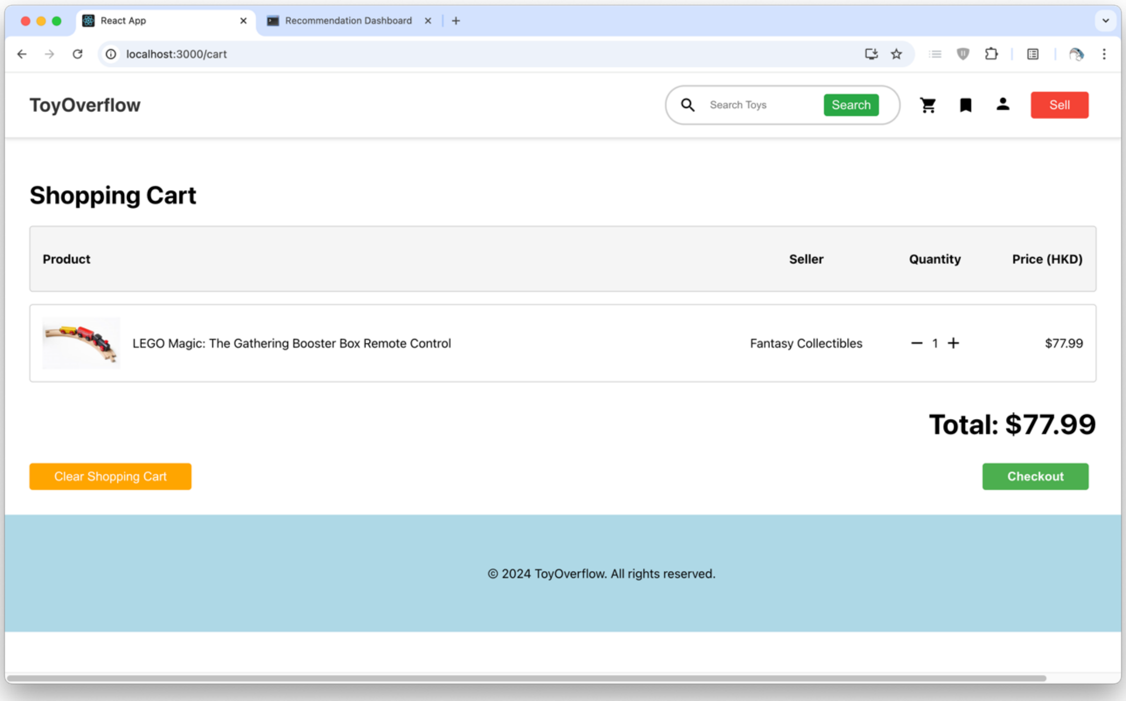
Price: $77.99

Seller: Fantasy Collectibles

Date: 08/03/2025

Customer Reviews

Average Rating: 4.3 / 5 ★★★★☆ (3 reviews)

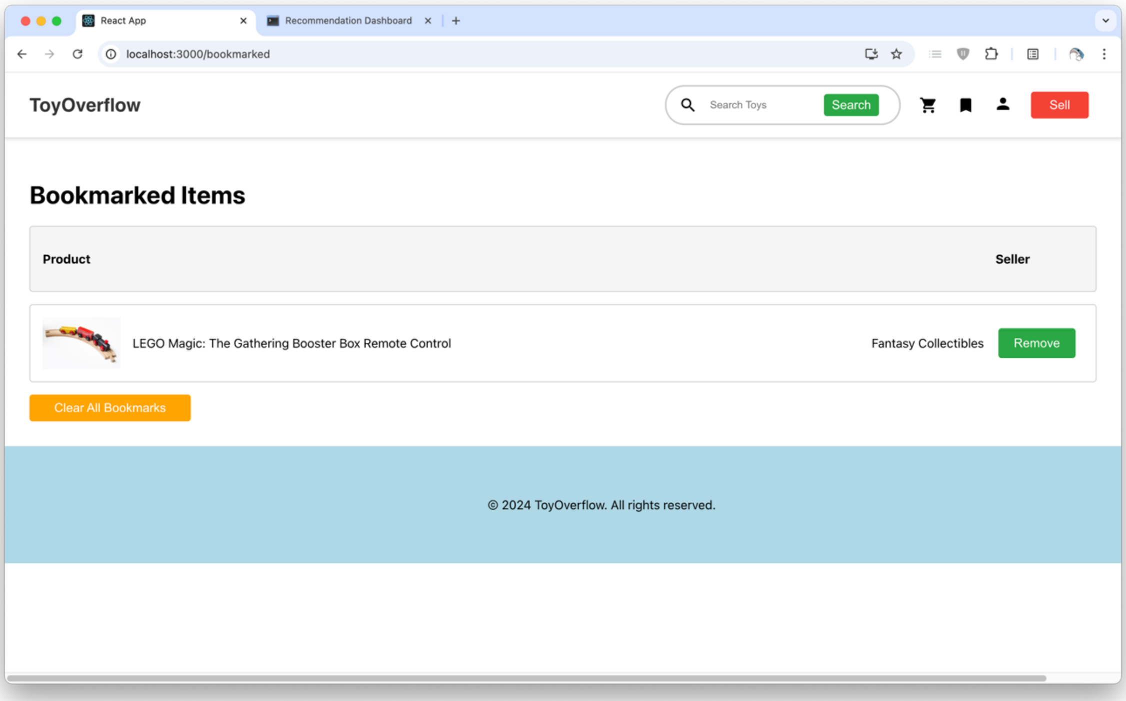


Shopping Cart

Product	Seller	Quantity	Price (HKD)
LEGO Magic: The Gathering Booster Box Remote Control	Fantasy Collectibles	− 1 +	$77.99

Total: $77.99

© 2024 ToyOverflow. All rights reserved.



Bookmarked Items

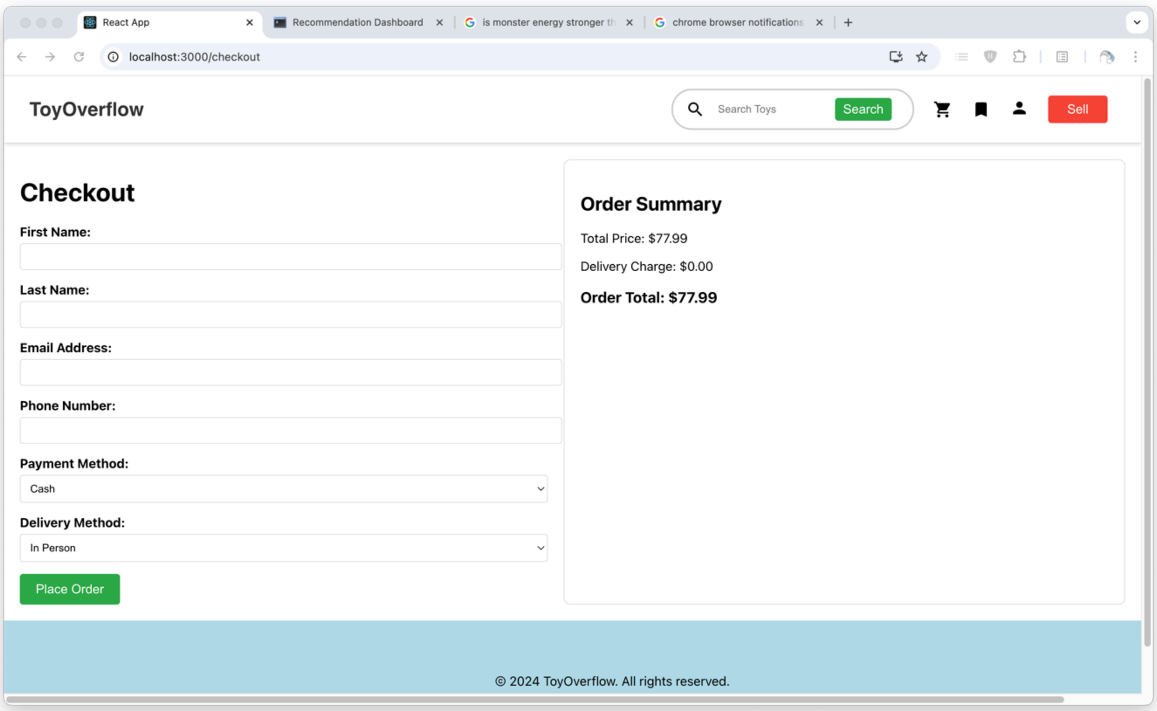
Product	Seller
LEGO Magic: The Gathering Booster Box Remote Control	Fantasy Collectibles

© 2024 ToyOverflow. All rights reserved.

Fig. 16: Browser alert when clicking add to shopping cart/bookmark button; Fig 17: Shopping Cart page; Fig 18: Bookmarked Items

When clicking the Add to Shopping Cart button or Add to Bookmarks button in the product details section, a browser alert will be displayed to confirm to the user the product has been added to the shopping cart or bookmarks (Fig. 16). Then, the user can click the View Shopping Cart button or the View Bookmarks button, or alternatively, click the shopping cart icons or the bookmark icons on the navigation bar on the top to navigate to the Shopping Cart page (Fig 17) and the Bookmarked Items page (Fig 18) respectively to view, remove or edit the quantity (only for shopping cart) of the product added.

3.1.8 Checkout


React App
Recommendation Dashboard
is monster energy stronger th
chrome browser notifications
localhost:3000/checkout

ToyOverflow

Search Toys

Search

Sell

Checkout

First Name:

Last Name:

Email Address:

Phone Number:

Payment Method:

Cash

Delivery Method:

In Person

Place Order

Order Summary

Total Price: $77.99

Delivery Charge: $0.00

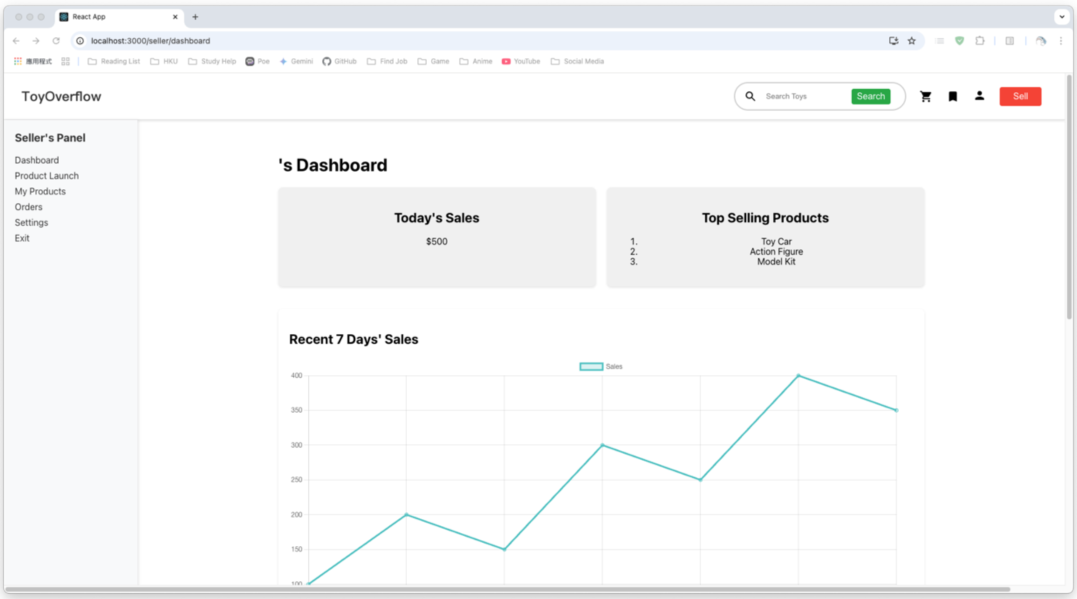
Order Total: $77.99

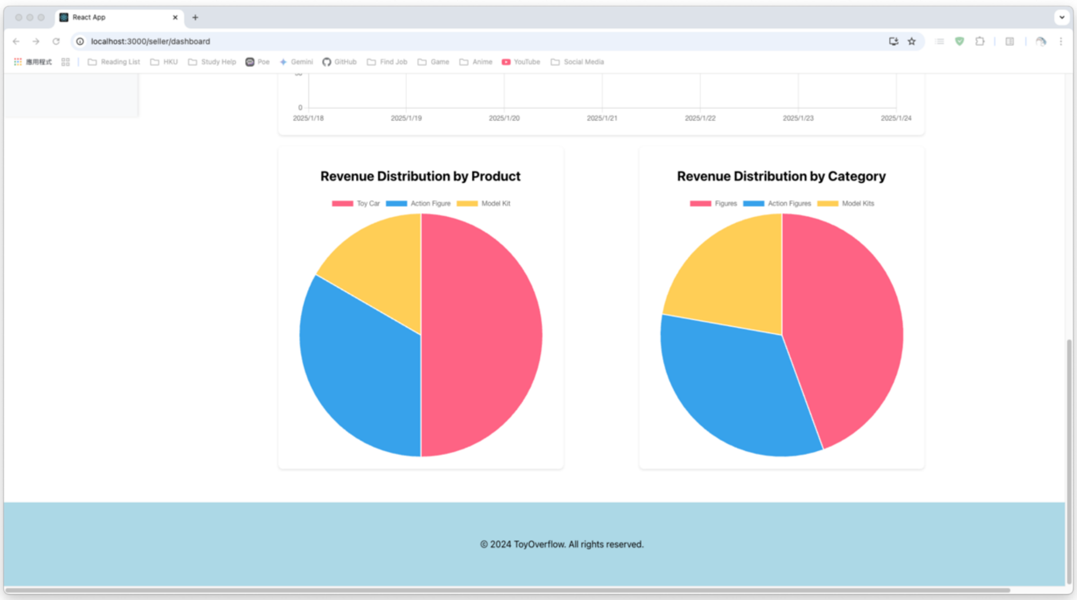
© 2024 ToyOverflow. All rights reserved.

Fig 17: Checkout Page

When a user wants to place an order, he can click the Checkout button at the bottom of the Shopping Cart page to navigate to the checkout page, then enter the required personal information and select the payment method and delivery method. He can also verify the order's total price including the delivery charge from the Order Summary section on the right.

3.1.9 Seller's Dashboard





React App
localhost:3000/seller/dashboard
2025/1/18
2025/1/19
2025/1/20
2025/1/21
2025/1/22
2025/1/23
2025/1/24
Revenue Distribution by Product
Toy Car
Action Figure
Model Kit
Revenue Distribution by Category
Figures
Action Figures
Model Kits
© 2024 ToyOverflow. All rights reserved.

Fig 18, 19 Seller's Dashboard – default page.

If users want to start selling their products, by clicking the Sell button on the navigation bar, users can navigate to the seller's panel. Different pages of the seller's panel can be navigated through the navigation column on the left. The Dashboard page, which is the default displayed page for the seller's panel, displays statistics and graphs related to product sales and revenue distribution.

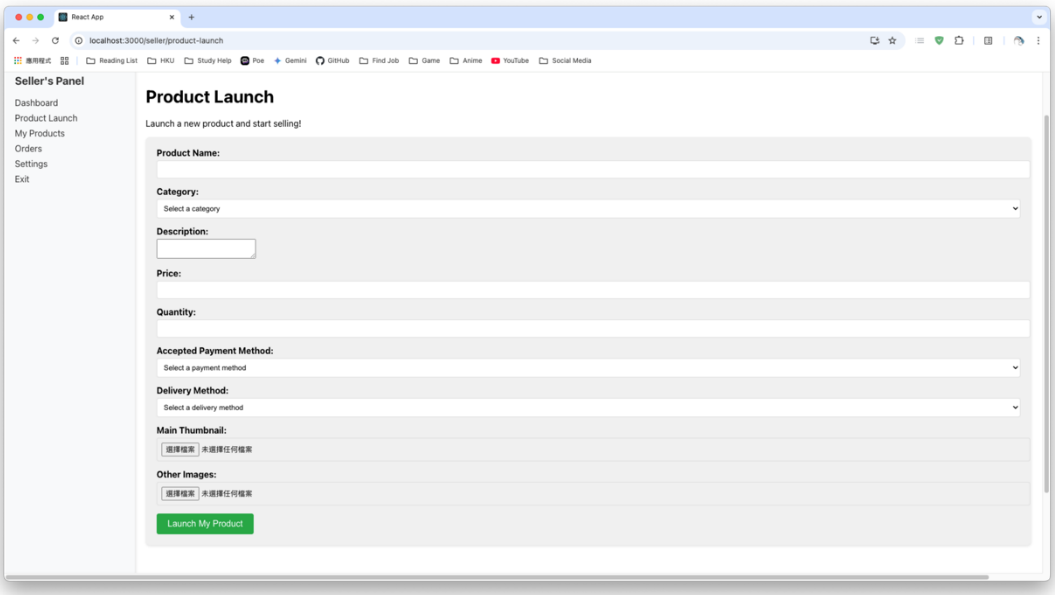
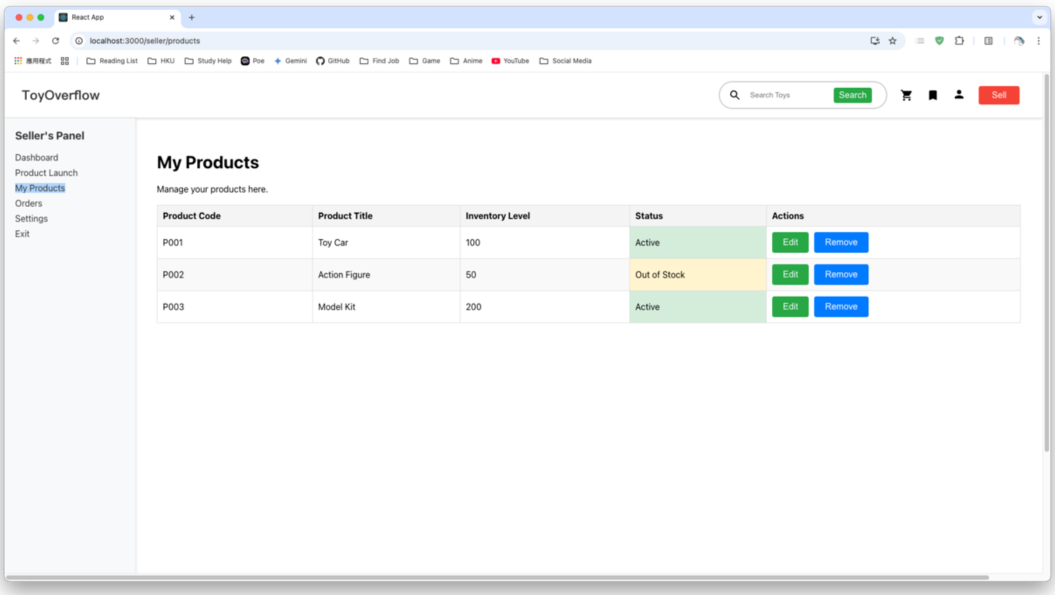


Fig. 20: Seller's Dashboard - Product Launch page

On the Product Launch page, the user can launch a new product by entering product details and uploading thumbnails and images.



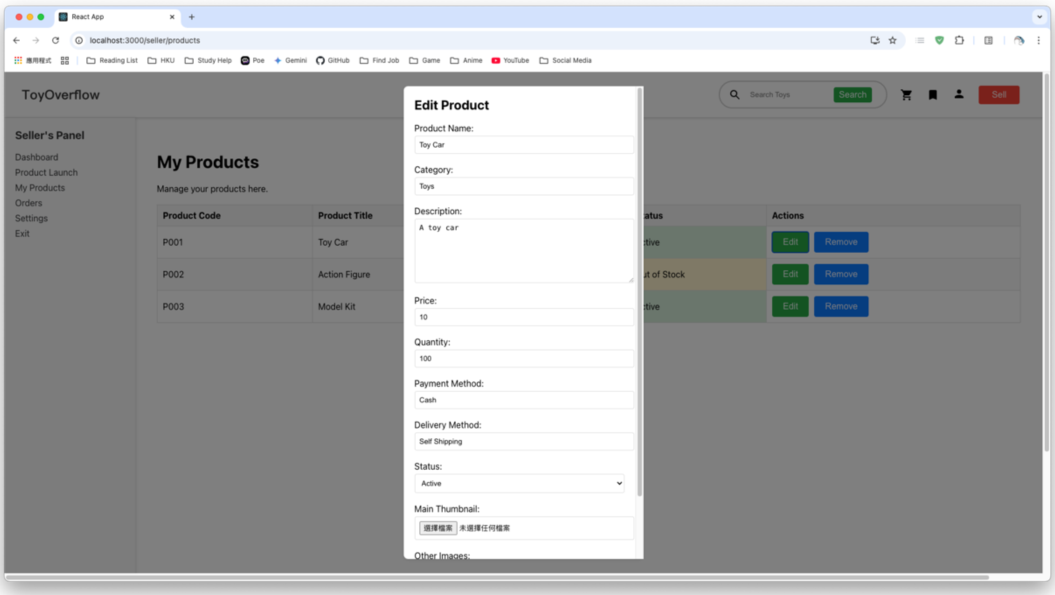
Seller's Panel

- Dashboard
- Product Launch
- My Products
- Orders
- Settings
- Exit

My Products

Manage your products here.

Product Code	Product Title	Inventory Level	Status	Actions
P001	Toy Car	100	Active	Edit Remove
P002	Action Figure	50	Out of Stock	Edit Remove
P003	Model Kit	200	Active	Edit Remove



Edit Product

Product Name: Toy Car

Category: Toys

Description: A toy car

Price: 10

Quantity: 100

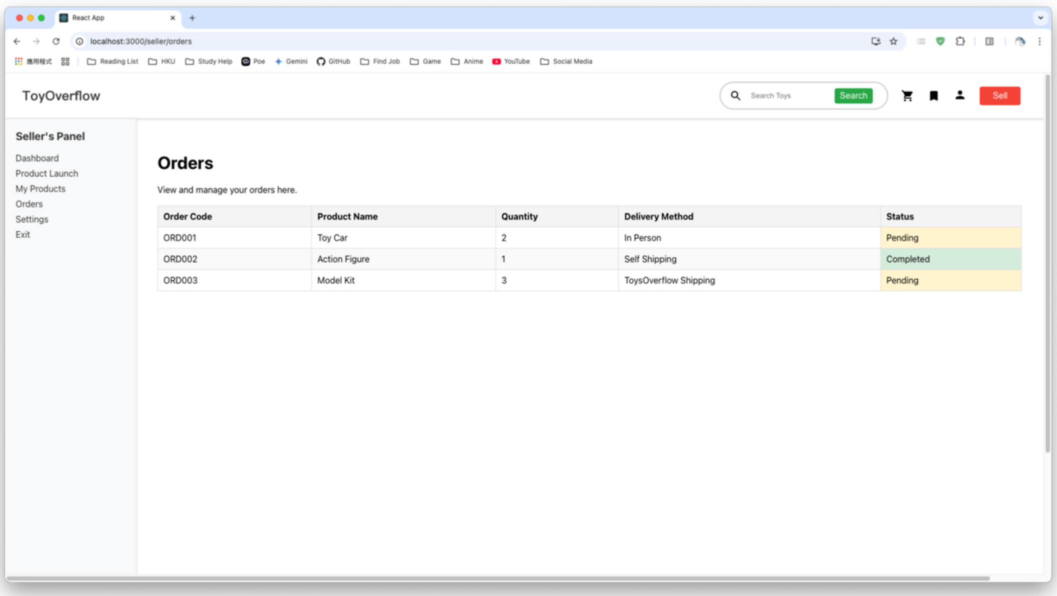
Payment Method: Cash

Delivery Method: Self Shipping

Status: Active

Main Thumbnail: 選擇檔案 未選擇任何檔案

Other Images:



Orders

View and manage your orders here.

Order Code	Product Name	Quantity	Delivery Method	Status
ORD001	Toy Car	2	In Person	Pending
ORD002	Action Figure	1	Self Shipping	Completed
ORD003	Model Kit	3	ToysOverflow Shipping	Pending

Fig. 21, 22: Seller's Dashboard – My Products page; Fig. 23: Seller's Dashboard – Orders page

On the My Products pages, the user can view the inventory levels and status of products launched (Fig. 21) or edit their information (Fig. 22). On the Orders page, the user can view the status of all orders received (Fig. 23).

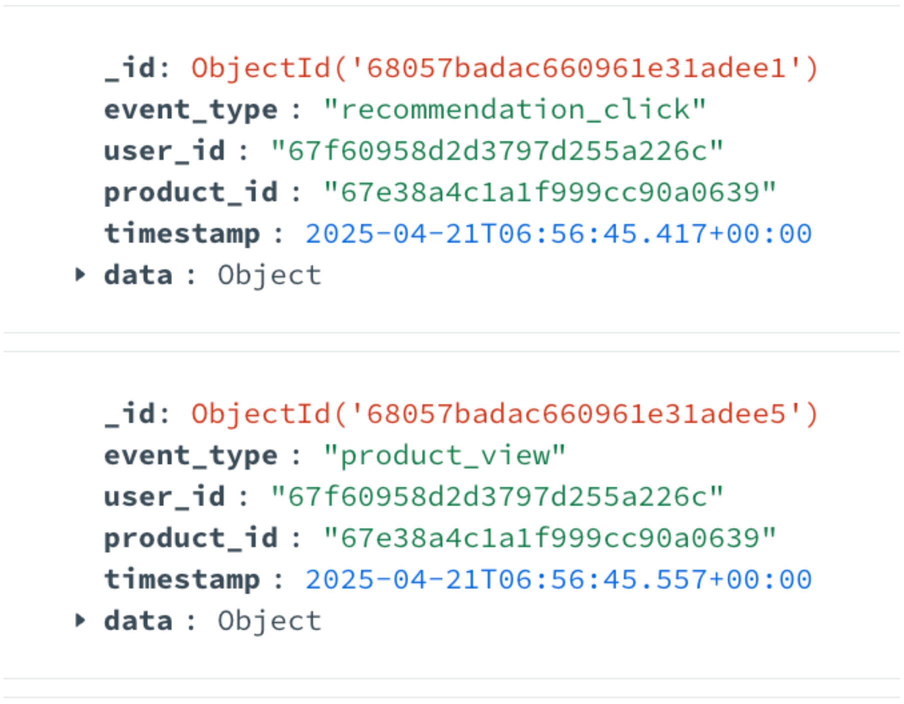
3.2 Recommendation Engine

The implementation of the personalized product recommendation engine is one of the most unique features of the app and the most challenging part of the project.

3.2.1 Workflows of the recommendation engine

i) Data Collection and Storage

The recommendation engine's workflow starts by tracking each user's interaction with the web app client, including viewing a product, purchasing a product, adding a review, clicking a recommended product, adding a product to bookmarks, etc. Each user interaction is considered as an 'event' and is stored in user_events table in the database and later retrieved by the recommendation engine.


```
_id: ObjectId('68057badac660961e31adee1')
event_type : "recommendation_click"
user_id : "67f60958d2d3797d255a226c"
product_id : "67e38a4c1a1f999cc90a0639"
timestamp : 2025-04-21T06:56:45.417+00:00
▸ data : Object
```

```
_id: ObjectId('68057badac660961e31adee5')
event_type : "product_view"
user_id : "67f60958d2d3797d255a226c"
product_id : "67e38a4c1a1f999cc90a0639"
timestamp : 2025-04-21T06:56:45.557+00:00
▸ data : Object
```

Fig. 24: Examples of entries of user events in the database

ii) Data Preprocessing and Model Training

Before the model training starts, required raw data such as users, products, reviews, user events are fetched from the database, cleaned, e.g. handling missing/null values, duplicate values and extreme values (outliers), then extract required data or transform data to specific formats so the data can be received by the model to be trained as input. In this recommendation engine, content-based filtering, which recommends products with similar properties as products the user previously interacted with (focus on item properties), and collaborative filtering models, which recommend products interacted by similar users (focus on user behaviour patterns) (Banerjee, 2020). Below are their rough training workflow:

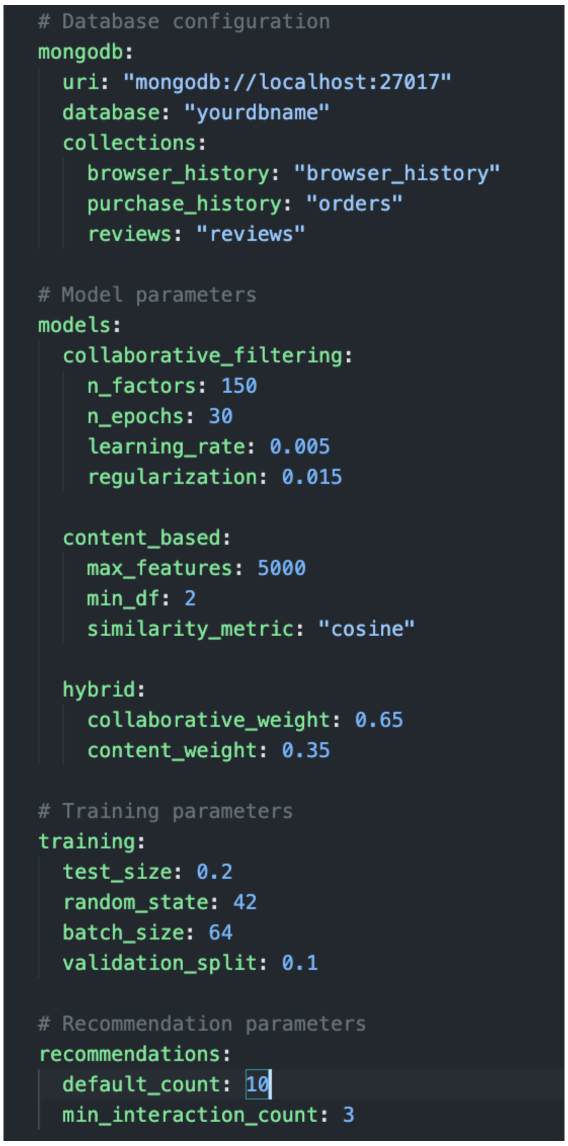
- Content-based filtering: receive product dataframes containing product metadata (descriptions, categories, features), normalize product descriptions and remove stop words, then do feature extraction by TF-IDF vectorization (a method to find out a term's importance in a text by calculating no. of times the term appears in the text and total no.

of terms in the text) (GeeksforGeeks, 2025). After extracting features, a similarity matrix is computed using cosine similarity, which represents how similar each product is to every other product.

- Collaborative filtering: receive reviews dataframes which forms a user-item interaction matrix (user IDs, product IDs, and ratings), decompose the matrix with SVD algorithm to find out the user latent vectors (represents user preferences across latent dimensions), item latent vectors (represents product characteristics across same dimensions) and bias terms (Global average, user bias, and item bias), then calculate the dot product of user and item latent factors plus bias terms to make product predictions. (GeeksforGeeks, 2024).

A hybrid model combines a weighted output from both content-based and collaborative filtering is also implemented.

To ensure the accuracy of recommendations, the models are automatically retrained timely or when certain amount of user interaction events are accumulated. Additionally, the models' parameters can be adjusted from the configuration file depending on their performance:



```yaml
# Database configuration
mongodb:
  uri: "mongodb://localhost:27017"
  database: "yourdbname"
  collections:
    browser_history: "browser_history"
    purchase_history: "orders"
    reviews: "reviews"

# Model parameters
models:
  collaborative_filtering:
    n_factors: 150
    n_epochs: 30
    learning_rate: 0.005
    regularization: 0.015

  content_based:
    max_features: 5000
    min_df: 2
    similarity_metric: "cosine"

  hybrid:
    collaborative_weight: 0.65
    content_weight: 0.35

# Training parameters
training:
  test_size: 0.2
  random_state: 42
  batch_size: 64
  validation_split: 0.1

# Recommendation parameters
recommendations:
  default_count: 10
  min_interaction_count: 3
```

Fig. 25: current parameter configurations

Content-based parameters:

- max_features: Maximum number of terms to be included in the vocabulary (TF-IDF formula). Typical range: 3000-10000. Setting it too low may lose important descriptive terms for products, too high may introduce noise and computational overhead
- min_df: Minimum document frequency for a term to be included in the vocabulary (TF-IDF formula). Typical range: 1-5. Too low may includes rare terms, preserves unique descriptors but may introduce noise

Collaborative parameters:

- n_factors: The number of latent factors (dimensions) used to model user and item relationships. Typical range: 50-300. Too low: the model cannot capture complex patterns. Too high: model overfit, increase noise and computational resources required
- n_epochs: The number of complete passes through the training dataset. Typical range: 10-50. Too low may cause model underfit or not learning enough from the data; Too high may cause model overfit and longer training time.
- learning_rate: Controls the step size during gradient descent when updating model parameters. Typical range: 0.001-0.01
- regularization: Penalizes model complexity to prevent overfitting. Typical range: 0.01-0.1. Too low may cause model overfit and more noise; Too high may cause underfit and unable to capture patterns

Hybrid model parameters:

- Collaborative_weight, content_weight: weighting given to each model. Typical range: 0.5-0.8, 0.2-0.5 respectively. When user behaviour data is sufficient, it is suggested to set collaborative_weight higher; When having diverse product catalogs, it is suggested to set content_weight higher.

Training parameters:

- Test_size: Proportion of data reserved for testing model performance. (train/test split ratio). Typical range: 0.1-0.3. Too low: less reliable evaluation metrics; Too high: less data for training
- Random_state: Seed value for random number generation to ensure reproducibility

- Batch_size: Number of training examples processed before updating model parameters. Typical range: 32-256. Smaller batch size: better exploration but more noise: larger batch: more stable updates but less exploration
- Validation_split: Proportion of training data used for validation during training. Typical range: 0.1-0.2. Too low: Less reliable validation signals; Too high: less data for training

Recommendation parameters

- Default_count: Default number of recommendations to generate per request. Typical range: 5-20. Too high: less precise; Too low: may miss relevant items
- Min_interaction_count: Minimum number of interactions required for a user/item to be included. Typical range: 1-10. Too high: exclude too many users especially new users; Too low: May include unreliable preference signals.

(Hug, 2015; scikit-learn, 2025)

iii) Testing and Evaluation

Two types of testing, offline testing and simulated online A/B testing are mainly used to evaluate the performance of the recommendation models.

The offline testing includes evaluation metrics that do not require user's feedback/subsequent action after generatin recommendations. First, prepare the data by splitting the user-item interactions into train and test sets (80/20 split by default), then train models on the training set and use the test set for evaluation. Metrics are calculated comparing predicted vs. actual ratings or other user interactions:

- Mean Absolute Error (MAE): calculated using actual product ratings and predicted product ratings (Formula: Fig. 26), which can measure the average absolute difference actual product rating and predicted product ratings. Lower values are better.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \widehat{y_i}|$$

where,

n: *number of observation*

y_i: *the actual value of the i^{th} observation*

-

$\widehat{y_i}$: *the predicted value of the i^{th} observation*

Fig. 26: MAE formula. (Source: Shiksha)

- Root Mean Square Error: another formula calculated using actual product ratings and predicted product ratings (Fig. 27). Similar to MAE, but penalises larger errors heavier. Lower values are better.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (Predicted_i - Actual_i)^2}{N}}$$

Fig. 27: RMSE Formula. (Source: Medium)

- Hit Rate: The proportion of users who received at least one relevant product recommendations among the list of recommended products he receives. Generally, the longer the list, the higher the hit rate.

(EVIDENTLY AI, 2025; Deutschman. 2023).

As for the online A/B testing, although the project has not released to public and hence do not have real users to observe and compare feedback from, we can still setup experiments to simulate user feedback. For example, create a pool of synthetic users, then divide them into Group A and Group B. Users in Group A will receive recommendations generated by collaborative filtering model and users in Group B will receive recommendations generated by content-based filtering model. Then, simulate each group's users' feedback by assuming the probability for an event of user feedback to happen and include noises or bias in the assumption(e.g. probability for a user to click on a recommended product) and observe which group has a better feedback, hence it can be predicted that the recommendation model used in that group has a better performance (Kumar, 2024).

Then the following A/B test metrics can be observed:

- Click-Through Rate (CTR): no. of clicks/no. of impressions (no. of users who got recommended the product).
- Conversion Rate: no. of orders placed/no. of impressions.
- Purchase per users

(EVIDENTLY AI, 2025)

However, such simulation cannot fully replace a real online A/B test participated by actual users as it cannot reassemble unexpected situations like unexpected/extreme user preferences or behavior, hence the result is less reliable than a real A/B test.

iv) Data Visualization (Recommendation generation)

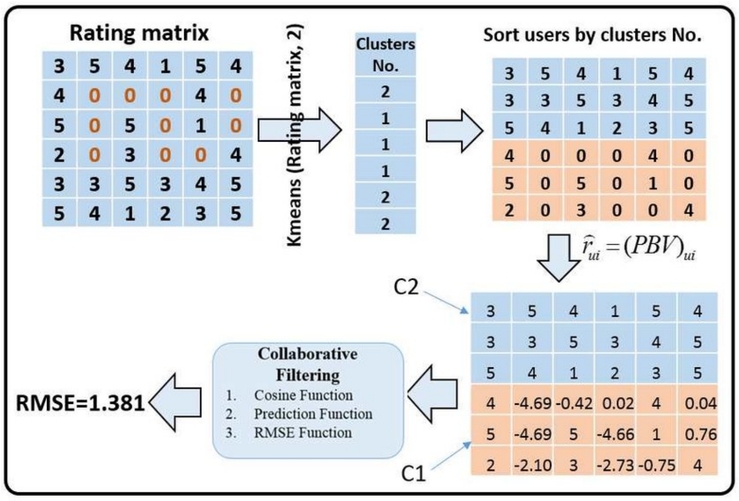
After the model training, each product is given a score which decides its ranking on the recommended product list. Products ranked top n will then be sent to the Frontend and displayed to the user (n's value depends on the product section's settings, e.g. n=5 for the Recommended for You section on the web app client's homepage). The scoring method depends on the model:

- Content-based: score = cosine similarity (user vector, product vector)

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum\limits_{i=1}^{n} A_i B_i}{\sqrt{\sum\limits_{i=1}^{n} A_i^2} \sqrt{\sum\limits_{i=1}^{n} B_i^2}}$$

Fig. 28: Cosine Similarity formula (Source: Medium)

- Collaborative: score = predicted rating calculated by SVD Algorithm



Rating matrix

3	5	4	1	5	4
4	0	0	0	4	0
5	0	5	0	1	0
2	0	3	0	0	4
3	3	5	3	4	5
5	4	1	2	3	5

Clusters No.
2
1
1
1
2
2

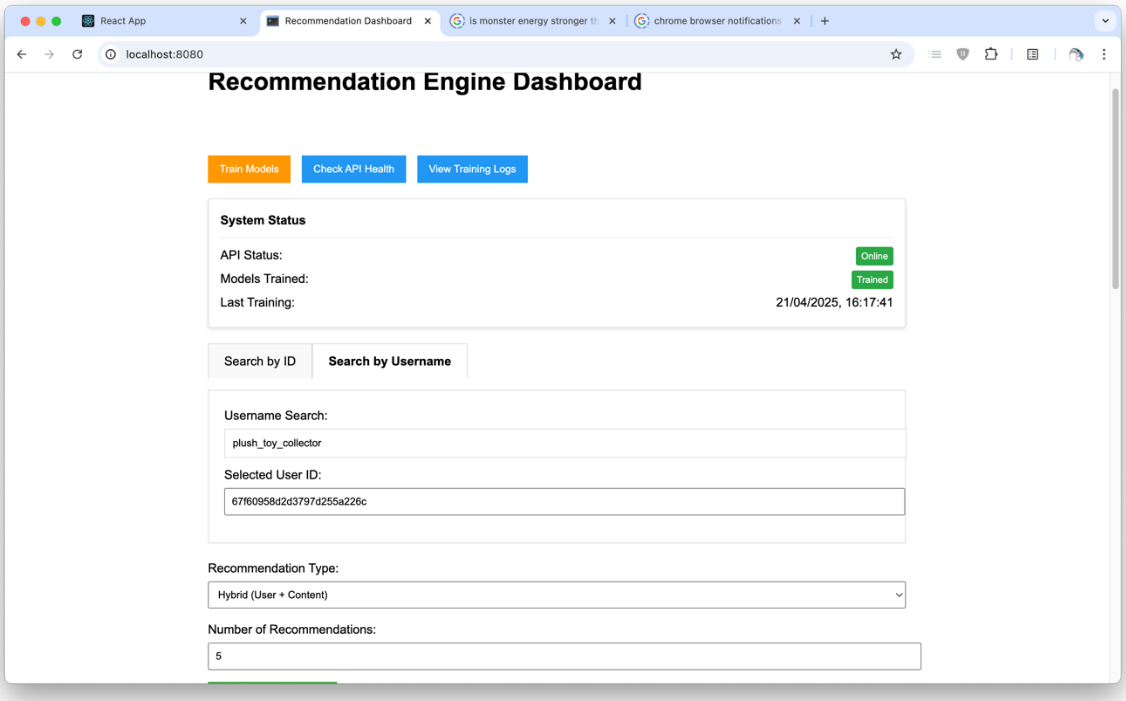
Sort users by clusters No.

3	5	4	1	5	4
3	3	5	3	4	5
5	4	1	2	3	5
4	0	0	0	4	0
5	0	5	0	1	0
2	0	3	0	0	4

3	5	4	1	5	4
3	3	5	3	4	5
5	4	1	2	3	5
4	-4.69	-0.42	0.02	4	0.04
5	-4.69	5	-4.66	1	0.76
2	-2.10	3	-2.73	-0.75	4

Fig. 29: SVD Predicted Rating (Source: ResearchGate)

3.2.2 Dashboard demo



Recommendation Engine Dashboard

Train Models | Check API Health | View Training Logs

System Status

API Status: Online

Models Trained: Trained

Last Training: 21/04/2025, 16:17:41

Search by ID | **Search by Username**

Username Search: plush_toy_collector

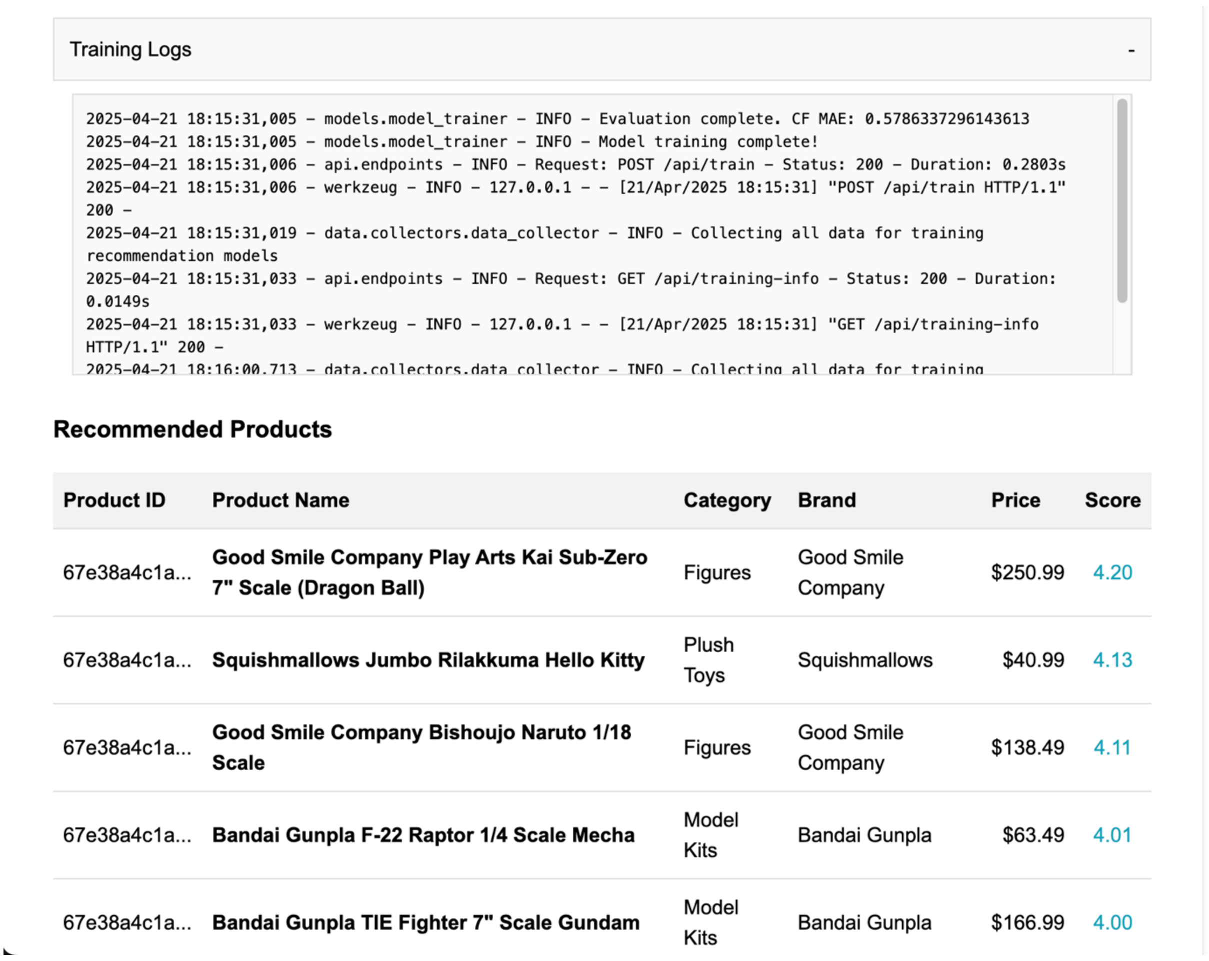
Selected User ID: 67f60958d2d3797d255a226c

Recommendation Type: Hybrid (User + Content)

Number of Recommendations: 5

Fig. 30: developer’s dashboard

A developer’s dashboard is implemented to better visualise the workflow for the recommendation engine and easier debugging or troubleshooting.


Training Logs -

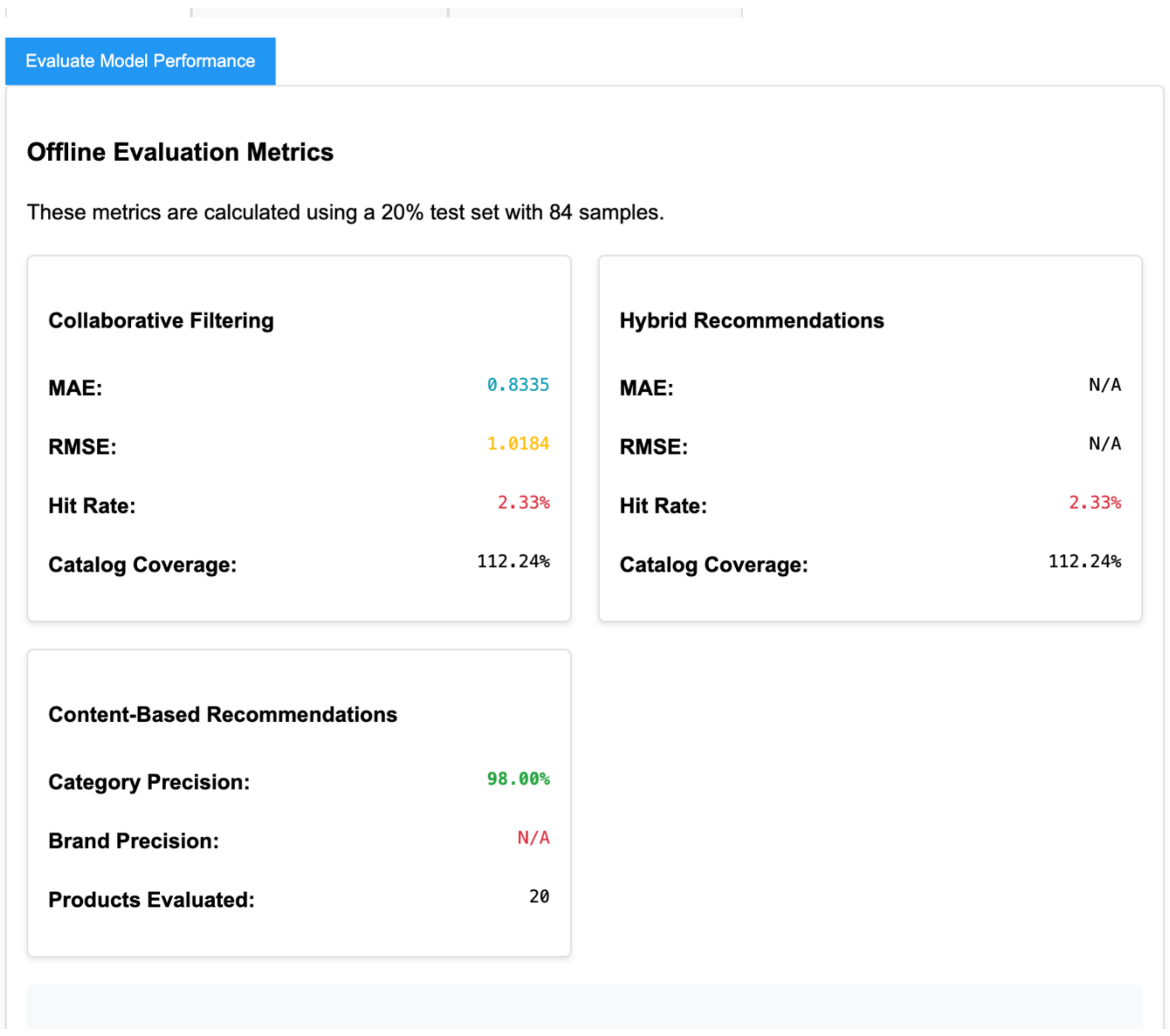
```
2025-04-21 18:15:31,005 - models.model_trainer - INFO - Evaluation complete. CF MAE: 0.5786337296143613
2025-04-21 18:15:31,005 - models.model_trainer - INFO - Model training complete!
2025-04-21 18:15:31,006 - api.endpoints - INFO - Request: POST /api/train - Status: 200 - Duration: 0.2803s
2025-04-21 18:15:31,006 - werkzeug - INFO - 127.0.0.1 - - [21/Apr/2025 18:15:31] "POST /api/train HTTP/1.1"
200 -
2025-04-21 18:15:31,019 - data.collectors.data_collector - INFO - Collecting all data for training
recommendation models
2025-04-21 18:15:31,033 - api.endpoints - INFO - Request: GET /api/training-info - Status: 200 - Duration:
0.0149s
2025-04-21 18:15:31,033 - werkzeug - INFO - 127.0.0.1 - - [21/Apr/2025 18:15:31] "GET /api/training-info
HTTP/1.1" 200 -
2025-04-21 18:16:00,713 - data.collectors.data_collector - INFO - Collecting all data for training
```

Recommended Products

Product ID	Product Name	Category	Brand	Price	Score
67e38a4c1a...	**Good Smile Company Play Arts Kai Sub-Zero 7" Scale (Dragon Ball)**	Figures	Good Smile Company	$250.99	4.20
67e38a4c1a...	**Squishmallows Jumbo Rilakkuma Hello Kitty**	Plush Toys	Squishmallows	$40.99	4.13
67e38a4c1a...	**Good Smile Company Bishoujo Naruto 1/18 Scale**	Figures	Good Smile Company	$138.49	4.11
67e38a4c1a...	**Bandai Gunpla F-22 Raptor 1/4 Scale Mecha**	Model Kits	Bandai Gunpla	$63.49	4.01
67e38a4c1a...	**Bandai Gunpla TIE Fighter 7" Scale Gundam**	Model Kits	Bandai Gunpla	$166.99	4.00

Fig. 31 Training Logs and Recommended Products list

Using the dashboard, we can train the models, check the training logs, and check the recommendations generated to a specific user when using different models (content-based/collaborative/hybrid).



Evaluate Model Performance

Offline Evaluation Metrics

These metrics are calculated using a 20% test set with 84 samples.

Collaborative Filtering

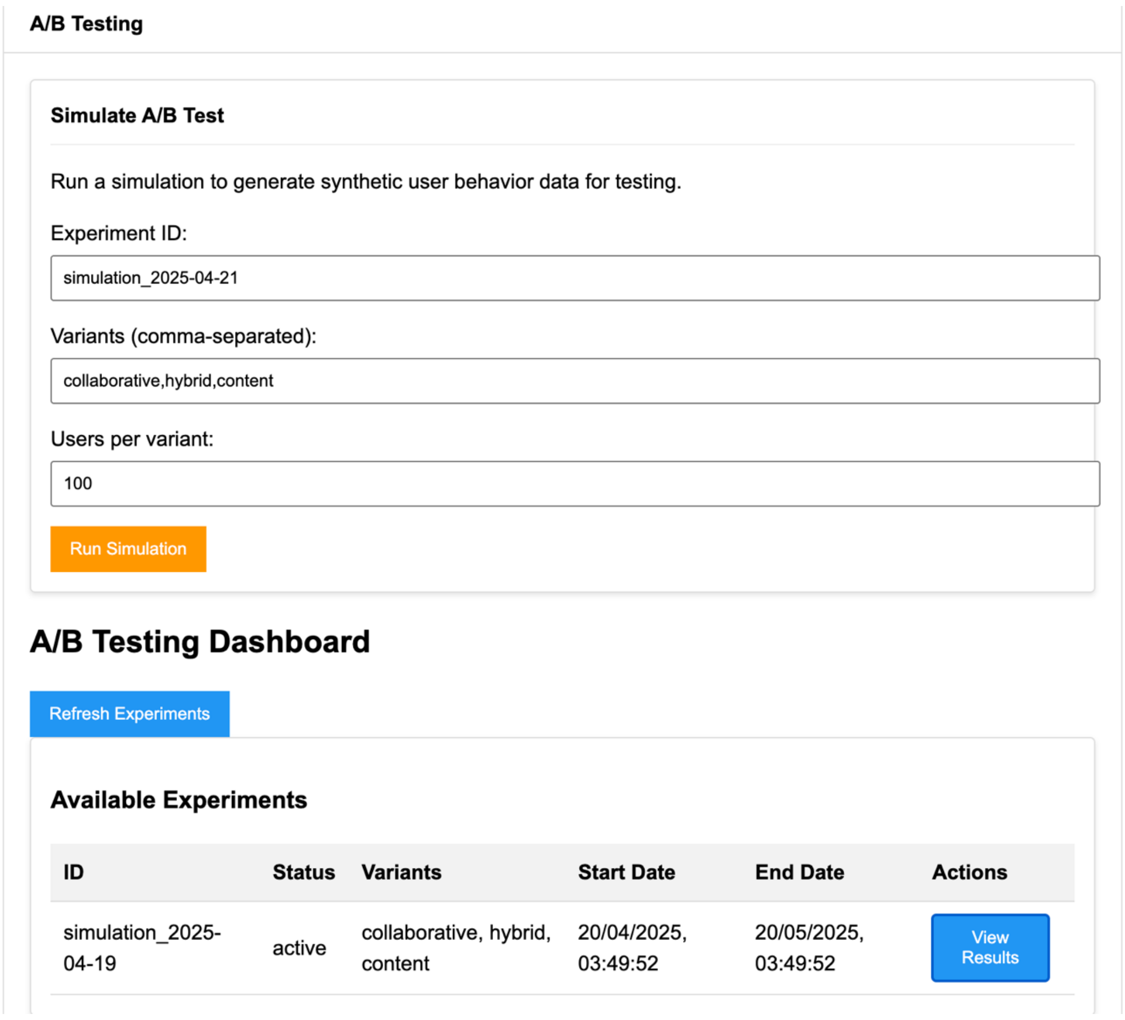
MAE:	0.8335
RMSE:	1.0184
Hit Rate:	2.33%
Catalog Coverage:	112.24%

Hybrid Recommendations

MAE:	N/A
RMSE:	N/A
Hit Rate:	2.33%
Catalog Coverage:	112.24%

Content-Based Recommendations

Category Precision:	98.00%
Brand Precision:	N/A
Products Evaluated:	20



A/B Testing

Simulate A/B Test

Run a simulation to generate synthetic user behavior data for testing.

Experiment ID: simulation_2025-04-21

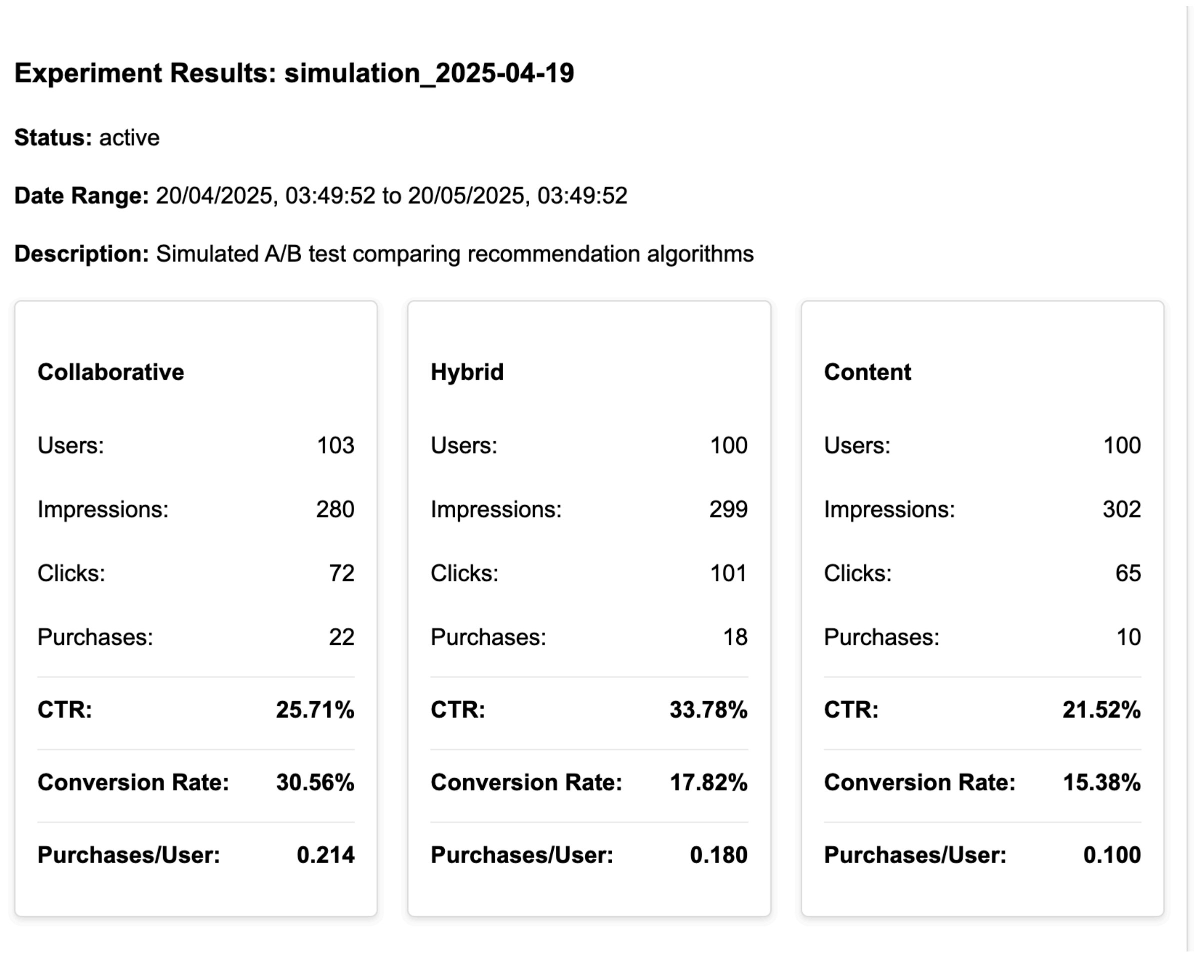
Variants (comma-separated): collaborative,hybrid,content

Users per variant: 100

A/B Testing Dashboard

Available Experiments

ID	Status	Variants	Start Date	End Date	Actions
simulation_2025-04-19	active	collaborative, hybrid, content	20/04/2025, 03:49:52	20/05/2025, 03:49:52	View Results


Experiment Results: simulation_2025-04-19

Status: active

Date Range: 20/04/2025, 03:49:52 to 20/05/2025, 03:49:52

Description: Simulated A/B test comparing recommendation algorithms

	Collaborative	Hybrid	Content
Users:	103	100	100
Impressions:	280	299	302
Clicks:	72	101	65
Purchases:	22	18	10
CTR:	**25.71%**	**33.78%**	**21.52%**
Conversion Rate:	**30.56%**	**17.82%**	**15.38%**
Purchases/User:	**0.214**	**0.180**	**0.100**

Fig. 32: Offline testing; Fig. 33, 34: Simulated A/B testing

Offline testing and Simulated A/B testing can also be done and observed on the developer's dashboard.

4. Conclusion and Future Works

Due to time and manpower constraints, the project is only implemented to a limited extent and has not reached production level, hence it has a large potential for future developments. For example:

i) Deployment on cloud: Due to time constraint, the project has not been deployed on a cloud host and still on localhost. Deploying the project on cloud would allow the implementation of features that require multiple actual users' feedback or interaction, such as the real-life A/B testing for the recommendation engine mentioned previously or real-time chatting between buyer and seller/customer service, or other production-level features.

ii) Security: Implement production-level security features like improved user authentication, API security, and server security

iii) Infrastructure: Streamline project deployment by containerization (Docker) and CI/CD Pipeline

iv) UI/UX: Implement mobile app version for the web app/implement responsive design to existing web app

v) Recommendation engine: implement more extensive and online testing, import external datasets to improve cold start problem

References

Banerjee, P. (2020). Recommender Systems in Python. Retrieved from: https://www.kaggle.com/code/prashant111/recommender-systems-in-python

Deutschman, Z., (2023). Recommender Systems: Machine Learning Metrics and Business Metrics. Retrieved from: https://neptune.ai/blog/recommender-systems-metrics

EVIDENTLY AI (2025). 10 metrics to evaluate recommender and ranking systems. Retreived from: https://www.evidentlyai.com/ranking-metrics/evaluating-recommender-systems

GeeksforGeeks (2024). SVD in Recommendation Systems. Retrieved from: https://www.geeksforgeeks.org/svd-in-recommendation-systems/

GeeksforGeeks (2025). Understanding TF-IDF (Term Frequency-Inverse Document Frequency). Retrieved from: https://www.geeksforgeeks.org/understanding-tf-idf-term-frequency-inverse-document-frequency/

HK01 (2022). 高達模型炒風不斷｜MG 大魔$300 炒到$900　轉售 Gundam 模型月入 10 萬原文網址: 高達模型炒風不斷｜MG 大魔$300 炒到$900　轉售 Gundam 模型月入 10 萬 | 香港 01. Retrieved from: https://www.hk01.com/%E9%81%8A%E6%88%B2%E5%8B%95%E6%BC%AB/775830/%E9%AB%98%E9%81%94%E6%A8%A1%E5%9E%8B%E7%82%92%E9%A2%A8%E4%

B8%8D%E6%96%B7-mg%E5%A4%A7%E9%AD%94-300%E7%82%92%E5%88%B0-900-%E8%BD%89%E5%94%AEgundam%E6%A8%A1%E5%9E%8B%E6%9C%88%E5%85%A510%E8%90%AC

HK Ulifestyle (2024). Retrieved from: https://www.facebook.com/photo.php?fbid=911125491054751&id=100064719244865&set=a.631855555648414

Hug (2015). Welcome to Surprise' documentation! Retrieved from: https://surprise.readthedocs.io/en/stable/

Kumar, S., (2024). A Guide to User Behavior Modeling. Retrieved from: https://blog.reachsumit.com/posts/2024/01/user-behavior-modeling-recsys/

Milankovich, M. (2015). The Cold Start Problem for Recommender Systems. Medium. Retrieved from: https://medium.com/@markmilankovich/the-cold-start-problem-for-recommender-systems-89a76505a7

Scikit-learn (2025). Scikit-learn. Retrieved from: https://scikit-learn.org/stable/index.html

Shaw, A. (2019). Product Recommendation System for e-commerce. Kaggle. Retrieved from: https://www.kaggle.com/code/shawamar/product-recommendation-system-for-e-commerce

World Integrated Trade Solution (2019). Toys nes imports by country in 2019. Retrieved from: https://wits.worldbank.org/trade/comtrade/en/country/ALL/year/2019/tradeflow/Imports/partner/WLD/product/950390