GROUP
fyp24112

E-commerce Web App for Toys

Student: Choi Tsz Long

Supervisor: Dr. Tam, Anthony T.C.

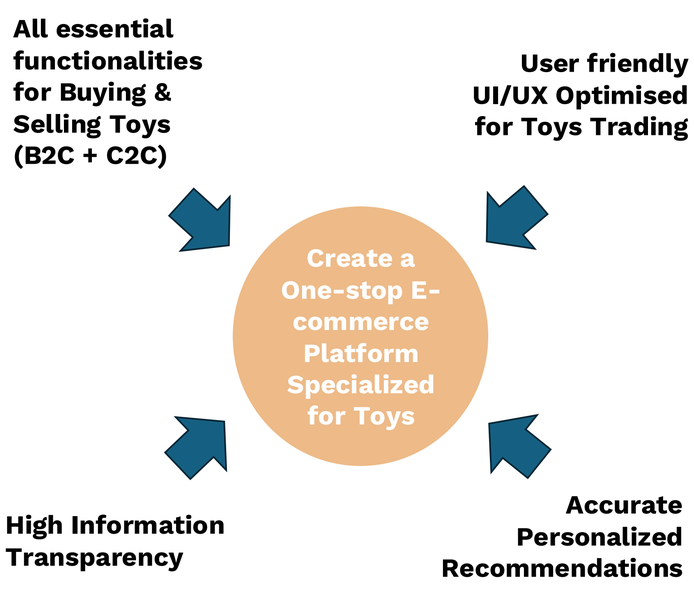
Background

The toy market in Hong Kong has a high but undeveloped potential. HK has been one of the world's toy importers (7th in 2019), showing the business volume and influence of the toy market, its growth is limited by:

(1) **Insufficient online channels to buy/sell toys** - Most toy stores do not have a online store/provide e-commerce services, which lowers the market's exposure and limits the market information available to consumers. Such information bias encourages speculation, which damages the market ecology and reduces consumers' buying incentives.

(2) **Lack of specialization and localization for existing major e-commerce platforms** – popular e-commerce platforms for buying/selling toys, e.g. Carousell, Taobao and Amazon, are either specialized for toys, hence the user experience provided is not optimized, e.g. product recommendations are often irrelevant due to high product variety, or only provide limited services or support to users in HK, e.g. shipping and return policies

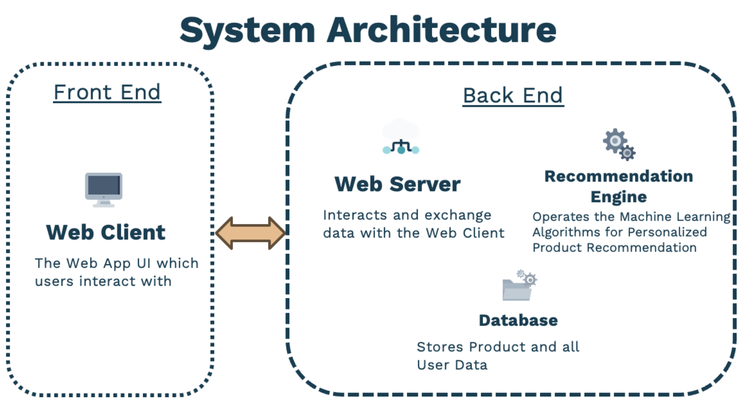
Objectives



Methodologies

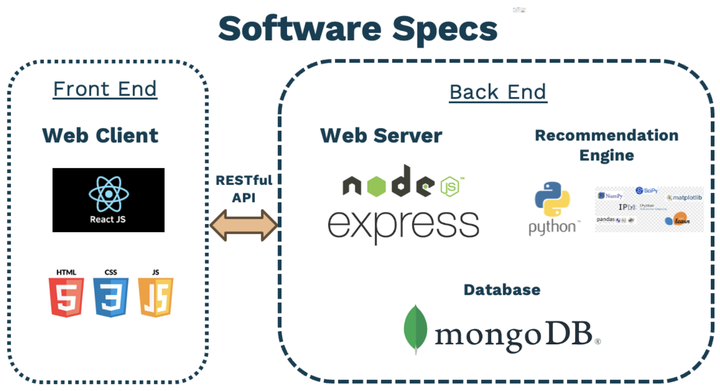
System Architecture:

- **Frontend**: the **Web App Client** which the users interact with.
- **Backend**:
  - **The Web Server**: interacts and exchanges data between the web app client and other Backend components;
  - **The Database**: stores all products, users, orders, user interaction/preference data, etc.
  - **The Recommendation Engine** and Server: builds and trains machine learning recommendation models using user data to generate personalized product recommendations



Software Specs:

- **Frontend Web App Client:** React JS (encapsulates HTML, CSS and JavaScript)
- **Backend**:
  - **Web Server**: Node JS, Express JS; Interaction with Front End: RESTful API
  - **Database**: MongoDB
  - **Recommendation Engine**: Python
    - Libraries for training recommendation model: Surprise, Scikit-learn, Pandas, NumPy, PyMongo, Flask, etc.



Model Training

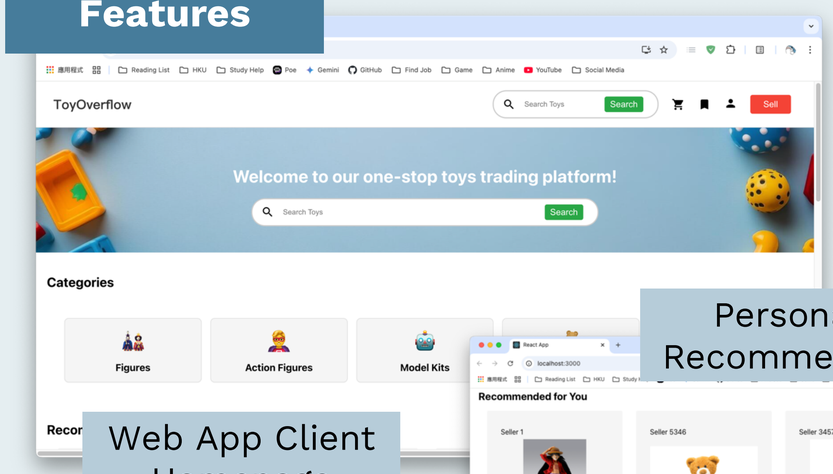
The implementation of the personalized recommendation engine is one of the most unique and challenging parts of the project. Its operation workflow is as follows:

1. **Data Collection** – Collect product data, e.g. product categories, brand, price, description and user interaction data, e.g. view/purchase/order/review
2. **Model Training** - (1) Collaborative Filtering: Build a user-item matrix from product ratings to Predict a user's rating to a unseen product; (2) Content-based Filtering: Build feature vectors from product attributes and similarity matrix to find relevant product (3) Hybrid: use both filtering. A weighted score is calculated using these 3 models which decide which products to recommend and their rankings.
3. **Model Evaluation** – Performance and accuracy of the recommendation engine are evaluated using metrics e.g. Mean Absolute Error, Root Mean Square Error (user preference prediction accuracy), Hit Rate (Percentage of users receiving relevant recommendations), etc.
4. **Model Serving** – Outputs recommended product list generated and other relevant data to web server then to web client through APIs
5. **Continuous improvement** – Models are automatically retrained timely or when certain amount of user interaction events are accumulated to ensure accuracy

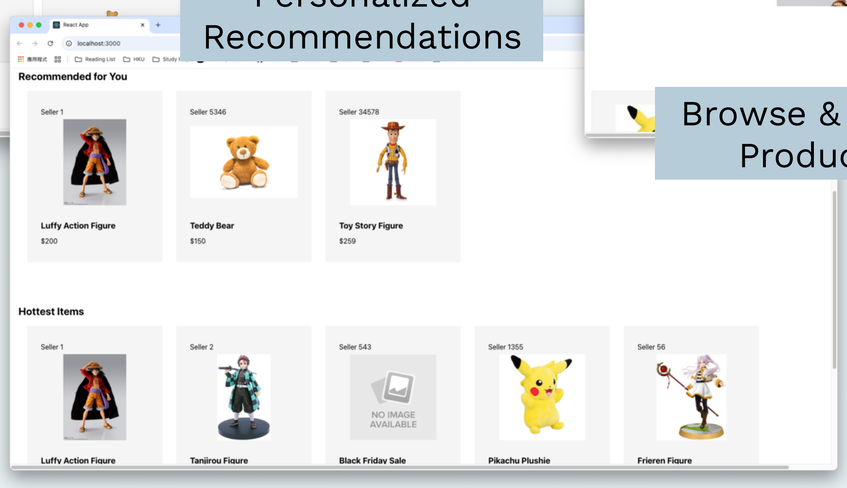
Future Enhancements

Various aspects of future enhancements could be done for the project, making it more suitable to release for production e.g. implementation on cloud and other infrastructure expansion, security enhancement, algorithm enhancements and for recommendation models, better data processing pipeline, etc.

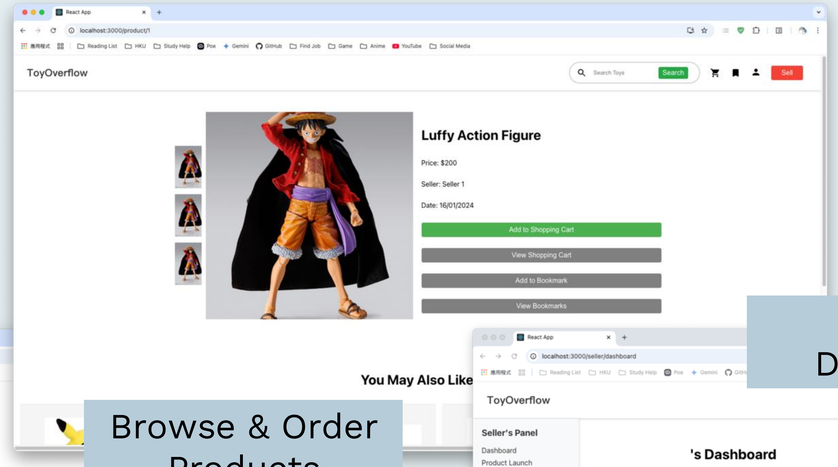
Features



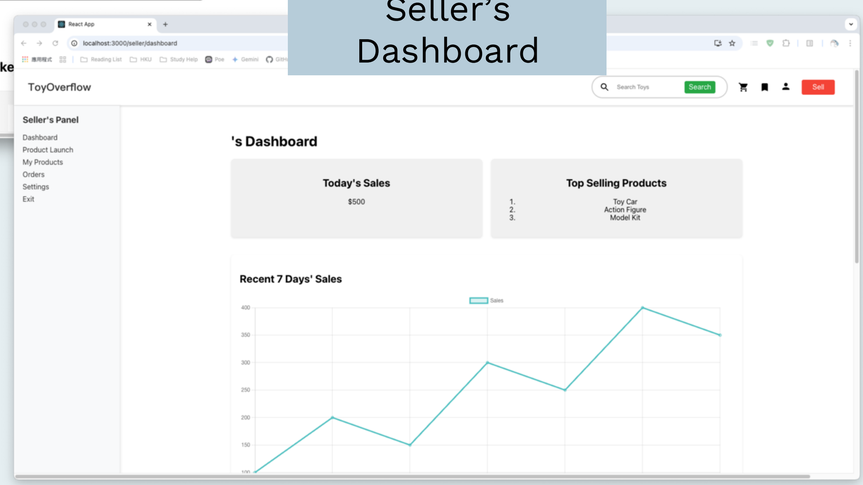
Web App Client Homepage



Personalized Recommendations



Browse & Order Products



Seller's Dashboard