[1. Overview 2](#_Toc69572969)

[1.1 Introduction 2](#_Toc69572970)

[1.2 Our Product 2](#_Toc69572971)

[1.3 System Architecture 3](#_Toc69572972)

[2. Third Party APIs and Frameworks 5](#_Toc69572973)

[2.1 Frontend 5](#_Toc69572974)

[2.1.1 Material UI 5](#_Toc69572975)

[2.2 Backend 5](#_Toc69572976)

[2.2.1 Werkzeug 5](#_Toc69572977)

[2.2.2 Numpy 5](#_Toc69572978)

[2.2.3 Dialogflow 5](#_Toc69572979)

[3. Functionalities 6](#_Toc69572980)

[3.1 Frontend 6](#_Toc69572981)

[3.1.2 Admin Interfaces 6](#_Toc69572982)

[3.1.3 User Interfaces 6](#_Toc69572983)

[3.2 Backend 6](#_Toc69572984)

[3.2.1 Database Management 6](#_Toc69572985)

[3.2.2 Admin Operations 6](#_Toc69572986)

[3.2.3 User Operations 6](#_Toc69572987)

[3.2.4 Product Promotion 6](#_Toc69572988)

[3.2.5 Chatbot AI 6](#_Toc69572989)

[4. Implementation Challenges 7](#_Toc69572990)

[4.1 Database Management – Atomic operations 7](#_Toc69572991)

[4.2 Chatbot AI implementation 7](#_Toc69572992)

[4.3 Frontend Framework 7](#_Toc69572993)

[5. User Documentation and Manual 8](#_Toc69572994)

[5.1 Windows 10 8](#_Toc69572995)

[5.2 MAC OS 8](#_Toc69572996)

# 1. Overview

## 1.1 Introduction

E-commerce earned its first glance from the world at the end of last century (Cohen, 2003; Stone, 2013). Although it was much insignificant compared to its scale today, the pioneers, Amazon and Auction Web (predecessor of eBay until 1997 (Mullen)), made their first attempt in 1995. Until today, the form of e-commerce has become much broader and adaptive. Almost every commercial good from dairy products to industrial materials, and even infrastructures have been covered by those famous enterprises and brands, including Amazon, eBay and Alibaba. However, for some of less populated products, small e-commerce companies could still earn trust from potential, but distant, customers.

Regardless the type of good is selling, working e-commerce companies still need websites to pair with their warehouses and delivery systems. And a good website for e-commerce provides the user with pages of different products available for purchase in the online store, as well as suitable recommendation systems, which extends the shopping window for each customer. In the long-run, extended shopping windows will not only raise profitability, but also helps in analysing customer personal preferences for further precise recommendations.

For e-commerce companies, the web operator, an admin-friendly website is also valuable for both web maintenance and development. Firstly, being able to import product information both individually and in batches from external files could massively improve web setup efficiency. Then, being able to batch editing existing product details allows a clear big picture view, together with convenient product management. Last but not least, order history from customers can be collected in the database, and ready for exporting or future analysis.

## 1.2 Our Product

Our Product e-commerce website provides various functions for user and admin. Standard utilities, including posting products and buying products, are polished for smoother experiences comparing to modern websites. Moreover, our product offers more than standard functionalities. Several additional features are provided for users and admins, to ensure higher management efficiency and better user-product match making.

For admins, a standard overview of existing product, past orders are collected for further analysis, together with product editing functions, which allows modification of product information without clearing past user ratings and past orders. Advanced functions, including batch product export and import makes database rebasing much less labour intensive.

For users, standard product searching and traversing experiences are provided, while advanced user-product match making is specialized for individual user already logged in. Upon user login, personalized products recommendations are automatically pushed to individual user in homepage. Additionally, upon each successful purchase, future recommendations will shift towards dynamic interest of individual user. Such features would significantly increase the chance of purchasing corresponding products.

A chat bot AI is also implemented as the novelty feature of our product …… (ChAtBoT pArT)

## 1.3 System Architecture

Our product e-commerce website has a following system architecture (Figure 1). Users and admins gain access to their web page from a computer, through a browser which supports JavaScript. Their actions, including searching for products and organizing orders, are passed to backend by components implemented in frontend. Then, the results requested by caller shall be passed back to and displayed by frontend.

Frontend balabala

The communication between frontend and backend is achieved through Flask API by binding a series of backend functions with a URL route. For example, an admin can receive all products list from [http://*\*domain\**/product/get\_all](http://*domain*/product/get_all). Therefore, the load would be shift from frontend webpage to backend server. By only letting backend server to process large amount of data request can greatly reduced the response time of webpage and reduce memory usage of browser. Meanwhile, backend server can isolate individual requests to maintain the integrity of database at any moment, to make database access being exclusive while writing action happens.

The backend focuses on processing requests, manages access to database and communicate with online API services. Python is used for faster feature implementation, broader API access and better compatibility with flask. Several features utilize complex mathematical equation and theory. Using APIs, such as Numpy, greatly reduces the difficulty in calculation. Other APIs, like DialogFlow, allow our server to uses google cloud services for Chatbot AI. They ease the effort in building natural language processing algorithm, and help our product focusing on selling products to our customers. The backend server also has frequent access to the database, to reduce the complexity of database updating, a \*.json file is used to store the changes we made.

Our products have been tested on Windows 10 and Mac OS, it has shown an excellent stability and a fast average response time. That is to say, our product can provide users with fabulous experiences, while advertising the most fit products to the right potential individuals.



Figure 1: System Architecture

# 2. Third Party APIs and Frameworks

## 2.1 Frontend

### 2.1.1 Material UI

## 2.2 Backend

### 2.2.1 Werkzeug

### 2.2.2 Numpy

### 2.2.3 Dialogflow

# 3. Functionalities

## 3.1 Frontend

### 3.1.2 Admin Interfaces

### 3.1.3 User Interfaces

## 3.2 Backend

### 3.2.1 Database Management

### 3.2.2 Admin Operations

### 3.2.3 User Operations

### 3.2.4 Product Promotion

### 3.2.5 Chatbot AI

# 4. Implementation Challenges

## 4.1 Database Management – Atomic operations

## 4.2 Chatbot AI implementation

## 4.3 Frontend Framework

# 5. User Documentation and Manual

## 5.1 Windows 10

## 5.2 MAC OS