# App Description:

## 1. Introduction

Our team chose the E-commerce theme because we are all interested in working with online shopping websites and apps in the future. E-commerce is a growing field, and learning how to manage product data, customer information, and orders will be very useful for our careers.

This document explains our e-commerce database and how to set up and use our application.

## 2. Dataset Overview

We chose the ecommerce\_db that stores information for an online store. It has three collections:

2.1 Users Collection (Customer Information)

This stores information about people who shop on the website:

* name: Customer's full name
* email: Their email address
* address: Where they live (street, city, zip code, country)
* order\_history: List of orders they've placed
* payment\_methods: How they pay (credit cards, etc.)

2.2 Products Collection (Items for Sale)

This stores information about products available for purchase:

* name: Product name
* description: What the product is
* price: How much it costs
* category: Type of product (Electronics, Books, Clothing, etc.)
* tags: Keywords to help with search
* features: Special characteristics of the product
* reviews: What customers think about the product
* variants: Different colors or sizes available

2.3 Orders Collection (Customer Purchases)

This stores information about customer orders:

* user\_id: Which customer placed the order
* order\_date: When they ordered
* total\_amount: Total cost of the order
* items: What products they bought and how many
* shipping\_address: Where to send the order
* status\_history: Order progress (Pending, Shipped, etc.)

# Group Findings:

NoSQL felt amazing for the project as had we used relational databases as we were always changing our code and affecting the database. Due to performing CRUD as well as function that change and/or affect our data NoSQLs flexibility and easy scalability helped as we had issues where we writing to a new collection by accident, this happened so seamlessly that it allowed us to see the issue. Although we were not exactly doing complex large volumes changes, we still were performing our fair share of changes so NOSQL databases high-speed access and scalability allowed for a smooth project experience. The project was working with real-time terminal application so NOSQL benefit of allowing this helped greatly as it allowed us to update and see immediately if what we coded worked extremely well or not so this was one of the best parts of this type of database. The strengths of NoSQL are truly amazing in our group’s opinion, it allowed us to work with large datasets with no issues as well as manipulate the data quite easily with no issues, truly a great experience. NoSQls flexible schema was an added bonus so there really was not many issues in working with the queries. Although we did run it this issue since we all used python but the lacking of a universal query language we have been an eventual issue. The eventual lack of data consistency that may come from NoSQL is an issue that we may have run into unintentionally but it was not a large issue.

# Challenges

Array manipulation was an issue as the grasping of the concept was truly a pain to deal with things like accessing the correct array, manipulating it correctly and removing/adding to it was a challenge. Referring to the notes as well as going to external resources as well helped me to overcome these issues. The data aggregation started of as an issue, understanding the stages proved to start of as a hassle however once I read through the notes more it showed hope easy these aggregation functionalities can be and just how much they expanded the scope of what I can do when it comes to querying. The processing of these data became more powerful and allowed a better way to goo beyond the limit of the lack of a universal NoSQL querying language.