**E-Commerce Project – IT E-Commerce Platform**

*Kinshuk Vasisht, M.Sc. Semester 3, R.N.: 19*

*Dated:* *2022-10-06*

## 1. URL (GitHub Repository)

<https://github.com/kinshuk-h/EComProject>

## 2. Project Proposal

* This project aims to implement an E-Commerce website for selling and purchasing of hardware and software components related to devices such as computers, mobiles, tablets, etc.
* The website will provide a common platform for sellers and interested buyers of technical devices and computing devices to connect.
* Sellers can register items they wish to sell over the platform for purchase by customers.
* Customers may search and choose items that they desire based on best features.
* Items offered over the platform include computing devices such as computers, tablets or smartwatches, and/or accessories & spare-parts related to these devices.

## 3. Features

* Must-have Features:
  + Authentication: Login, register as seller, register as buyer
  + Registration of items for selling (name, category, sub-category, quantity, expected price)
  + Order management: view, confirm orders
  + Cart management: Add, remove, view items
  + Item management: view items with details (name, category, price, rating) (add item to cart) (rate purchased item)
  + Item search and filter: (specific, category/sub-category wise, filters over properties)
* Good-to-have Features:
  + Real-Time Order tracking
  + Recommendations based on product purchase and frequent user purchases
  + Product comparison for best deals / best features
  + Fuzzy Filtering

## 4. Technical Stack

* Frontend
  + HTML + CSS + JavaScript for dynamic front-end content
* Backend
  + Flask (Python) (for route handling and dispatch)
  + SQLAlchemy (Python) (for interaction with an SQL-based DBMS)
  + socket.io (Python, JS) for real-time updates
* Database
  + MySQL (Relational DBMS, data to be maintained present in structural form, with identifiable relationships)

*Dated:* *2022-10-14*

## 5. Schema Description (for RDBMS)

The following schema is proposed for the project, for use with an RDBMS software.

Entities:

* User
  + Name
  + Username (unique)
  + E-Mail Address (optional) (unique)
  + Profile Image (optional)
  + Password (composite)
    - Hash (unique?)
    - Salt (unique?)
  + ID (primary)
  + Contact
* Customer (IS A User)
  + Date of Registration
  + Shipping Address (composite) (multi-valued)
    - Label (unique)
    - Description
  + Payment Mode (composite) (multi-valued)
    - Label (unique)
    - Type
    - Details
* Seller (IS A User)
  + Company Name
  + Business Contact (multi-valued)
* Item
  + Name
  + Type
  + Category (multi-valued)
  + Description
  + Features (multi-valued) (composite)
    - Key
    - Value

Relationships:

(Note: Nested bullets denote relationship-specific attributes)

* *Item* OFFERED BY / ASSOCIATED WITH *Seller*  - **M:N**
  + Quantity
  + Expected Price
* *Customer* GIVES RATING TO *Seller* - **M:N**
  + Rating
  + Feedback
* *Customer* ADDS *Item* PROVIDED BY *Seller* TO CART - **M:N:P**
  + Quantity
* *Customer* PURCHASES *Item* PROVIDED BY *Seller* FOR ORDER - **M:N:P**
  + Order ID
  + Shipping Address Label
  + Payment Mode Label
  + Date of Shipment

Relations:

* User
* Customer
* Seller
* Item
* Cart
* Order
* Transactions
* User
* Customer\_Addresses
* Customer\_Payment\_Modes
* Item\_Associations
* Item\_Categories
* Item\_Assigned\_Categories
* Item\_Feature\_Keys
* Item\_Assigned\_Feature\_Keys

## 6. API Description:

The following API endpoints are provided, categorized as follows:

[**Authentication**](https://vscode-vfs+github.vscode-resource.vscode-cdn.net/kinshuk-h/EComProject/docs/api/auth.md):

* POST /api/auth/login: Login using an existing customer or seller account, init session
  + Parameters:
    - Customer ID | Customer Username | Customer E-Mail (One of any unique Identifiers)
    - Customer Account Password
  + Returns: Customer Account ID, Access Token, Expiration Duration (JSON)
* POST /api/auth/refresh: Refresh login state, update login token
  + Headers:
    - Authentication: Bearer "<account\_id>:<access\_token>"
  + Returns: Customer Account ID, New Access Token, Updated Expiration Duration (JSON)

[**Customers**](https://vscode-vfs+github.vscode-resource.vscode-cdn.net/kinshuk-h/EComProject/docs/api/customer.md):

* PUT /api/customer/register: Register a new customer
  + Parameters:
    - Customer Details (Name, etc.)
    - Customer Account Password (plaintext, over HTTPS, stored in DB as hash+salt)
* GET /api/customer/<customer\_id?>:
  + Headers:
    - Authentication: Bearer "<account\_id>:<access\_token>"
  + Parameters:
    - customer\_id (optional): ID of customer to access, if empty, access logged in customer
  + Returns:
    - Customer Profile Details (JSON)

[**Sellers**](https://vscode-vfs+github.vscode-resource.vscode-cdn.net/kinshuk-h/EComProject/docs/api/seller.md):

* PUT /api/seller/register: Register a new seller
  + Parameters:
    - Seller Details (Representative Name, Company, etc.)
    - Seller Address(es)
    - Seller Contact(s)
    - Seller Profile Image (optional)
    - Seller Account Password (plaintext, over HTTPS, stored in DB as hash+salt)