COMPUTER SYSTEM DESIGN

PROJECT PROPOSAL
CAMPUS HUB

Design Doc

Introduction

The primary objective of Campus Hub is to provide a centralised platform for catering to all individuals on campus. From Daily essentials to pharmacy to services like laundry, the goal is to streamline access to a wide range of services and amenities, enhancing convenience and fostering a vibrant campus culture.

Our solution aims to simplify the navigation of the myriad services available on campus, providing a user-friendly hub for students to access college services effortlessly, stay informed about emerging offers, and conveniently buy and sell products. "CampusHub" envisions a connected ecosystem that transforms the campus experience, fostering convenience, efficiency, and opportunities for all stakeholders.

Features Implemented

1: Centralised Hub for Campus Services:

2: Efficient Products, Stores and Services Discovery:

3: Trending recommendations:

4: User-Friendly Cart and Wishlist Management:

5: Empowering Sellers:

6: Secure Transaction.

7: Responsive Portal

8: CI/CD Pipeline

Tech Stack

Front-end

- Typescript (Type safety)
- React (framework) Shadcn (UI)
- Tailwind (CSS)
- Eslint and Prettier (linting and formatting)
- npm (package management)
- Jest (Testing)

Back-end

- Python Flask (framework)
- Pydantic (Data validation)
- Pymongo (Mongo driver)
- Poetry (Package management)
- Ruff (linting and formatting)
- Pytest (testing)
- gunicorn (WSGI)
- Nginx (reverse proxy, load balancing, DDOS firewall)
- Celery and RabbitMQ (Message Queue)
- OpenAPI3 (API Documentation)

Database

- MongoDB

Features to be implemented

Seller Inventory Management System

Based on the feedback from the first phase of the project, we have planned to revamp the seller portal. Statistics based on user demand and offer trends will be added. Moreover, excel sheets are being exploited to upload products in bulk keeping in mind the effort it would take to upload each product at a time. But having json fields in excel may make this system more prone to errors, leading to bad UX. Hence in this phase, we will provide forms as well as excel upload options for more flexibility. The fields and constraints in the excel will be revamped keeping the seller UX in mind.

Feedback mechanism

Reviews and ratings will help the user get authentic and reliable information about the products sold on campus. Moreover,the freedom to express thoughts on one's service and making the sellers aware of users' expectations and satisfaction in turn will lead to better services.

Community Store

The USP of our project happens to be the provision of selling or reselling items. Despite implementing the "become a seller" option (which allows one to become a seller, create multiple stores and add products), we as students, feel the need to have a dedicated community store where one can add products on the portal without having to own a store before selling the same. This will facilitate the campus community including staff, professors and students to re-sell old books, cycles, mirrors or other items more easily than ever before.

Filters

Filter and sorting by price, ratings, availability, etc will be implemented to ease product searches, improving the UX.

Push notification

Create a dashboard section for push notification using web sockets for persistent connection, this will be used to immediately create on coming orders for sellers as well as for users who would want to check in the details of their orders.

OTP Delivery

Create a mechanism for to verify delivery using a diluted concept of asymmetric cryptography, but in this case rather than having two key, one key will be generate on the fly using the hash of the order of the user, this key will then be securely shared with the user and sellers, the delivery will have to verify the key using the barcode generated by the hashed key, this maintains the brevity and authenticity of the order as well as trust that the delivery happened.

Resolving the Double Purchase Problem

Implement a distributed locking mechanism using techniques like optimistic locking at server level to ensure atomicity and consistency during the purchase transactions. Utilise unique identifiers and timestamps to prevent race conditions and enforce single purchase per product instance. Implement a transactional rollback mechanism to handle failures gracefully and maintain data integrity across concurrent transactions.

Pagination

Implement a server-side pagination mechanism by configuring APi endpoints to accept parameters like page number and page size. Retrieve and serve data in chunks, reducing server load and optimising response times. Implement caching strategies to further enhance performance and scalability, ensuring smooth navigation through paginated results.

Campus Admin Portal

Addition of a superuser for each campus where CampusHub is deployed, in order to manage and verify the sellers listed on the platform. This authority would likely be provided to a trusted campus official and would be used in order to ensure enhanced seller vetting as well as clearer dispute resolution. It will help regulate services and products sold, and the sellers would be accountable for their own sale towards the admin, which would lead to further trust and transparency among the customers.

Recommendation System

Implementation of a Recommendation System that would be focused on customer behaviour analysis. This would ideally include a model-based collaborative filtering system based on customer's purchase history and ratings provided by other users who bought items similar to that user. While this would require the implementation of another server for the deployment of the ML model, it would lead to a great boost in user experience as well as sales for the sellers and stores.

Task Division

Task	Assigned to
Seller Inventory Management System, Filters, Feedback mechanism, Community Store	**Aishika Nandi**
Push Notification, OTP Delivery	**Javed Habib**
Double purchase problem, Pagination	**Arnav Prakash**
Campus Admin Portal, Recommendation system	**Aayush Kataria**