T2T MARKETPLACE

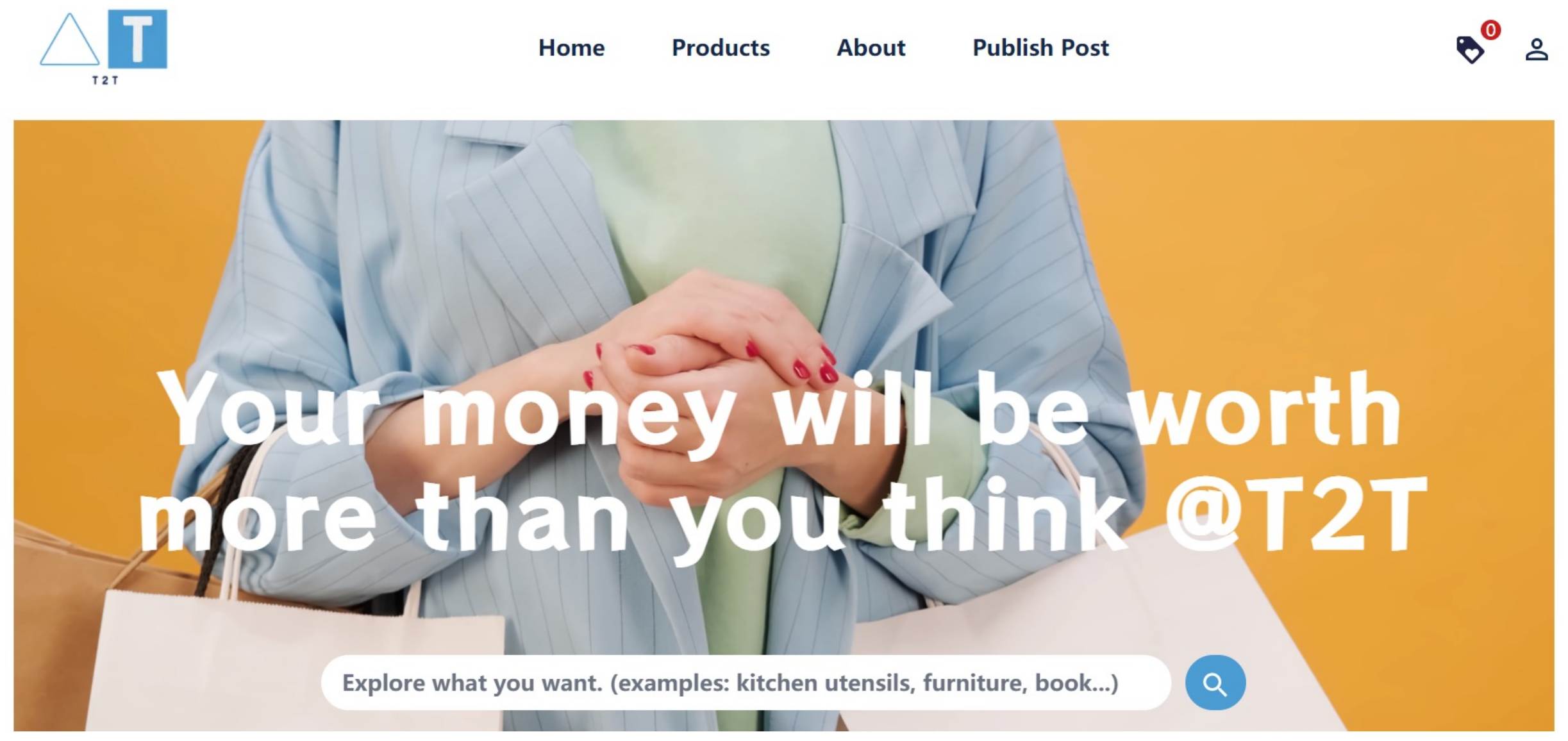
CAMPUS SECOND-HAND SHOPPING PLATFORM

Student: Yiran Chen, Zhidi Zhang, Ziqiao Ma

Sponsor: Antoine de Torcy

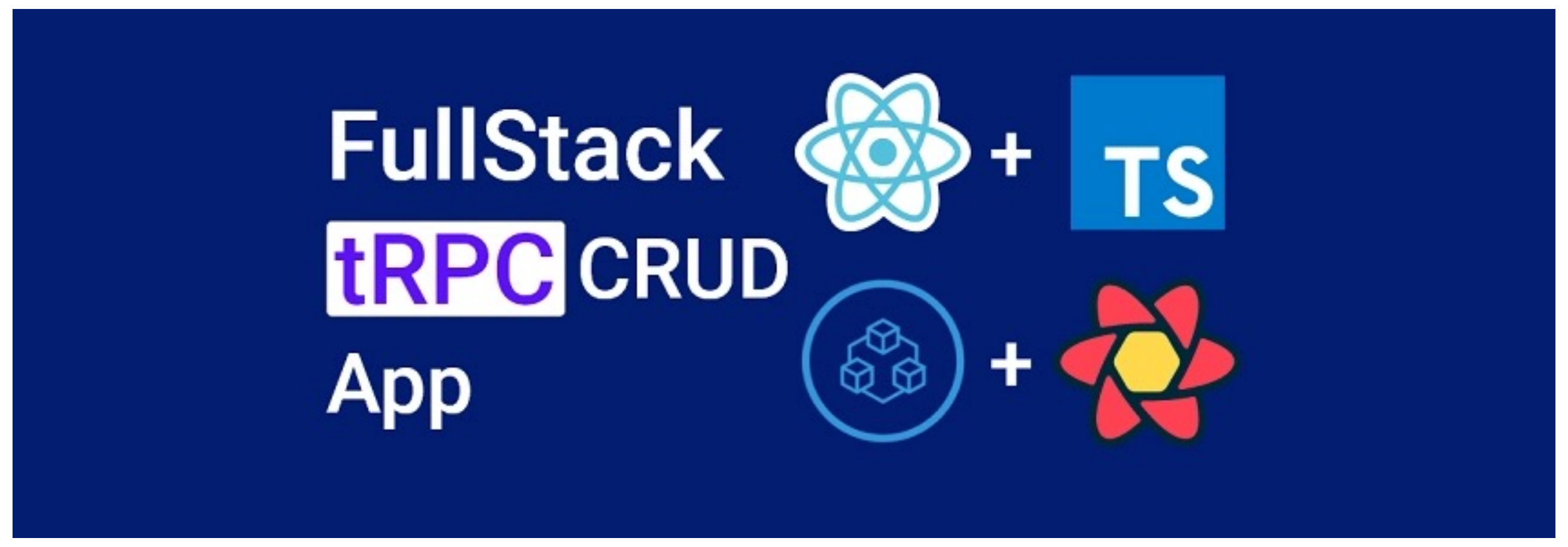
Objective

Our objective is to design and implement a cutting-edge, reliable online platform tailored to the unique needs of the University of North Carolina campus community, with a focus on creating a safe and intuitive user experience for second-hand shopping. To achieve this, we will be implementing stringent authentication protocols that require users to verify their identity through their verified .edu email addresses. In addition, we are committed to fostering a culture of trust within the campus community by promoting and prioritizing local transactions between UNC students, faculty, and staff. Our platform aims to revolutionize the way students buy and sell second-hand goods on campus, while upholding the highest standards of security and user-friendliness.


T2T
Home
Products
About
Publish Post
Your money will be worth more than you think @T2T
Explore what you want. (examples: kitchen utensils, furniture, book...)

Innovative Approach

This project is built using the T3 app, a powerful full-stack development platform that enables developers to create scalable, performant applications with ease. In addition to the T3 app, we have also employed the full-stack Next.js framework, which revolutionizes the traditional front-end and back-end connection paradigm. Leveraging the advanced features of Next.js, including API routes, server-side rendering, and static generation, our platform delivers unparalleled performance, scalability, and flexibility. With the T3 app and Next.js, we are able to provide dynamic content updates and enhance the overall user experience of our application, while maintaining a high level of developer flexibility and productivity.



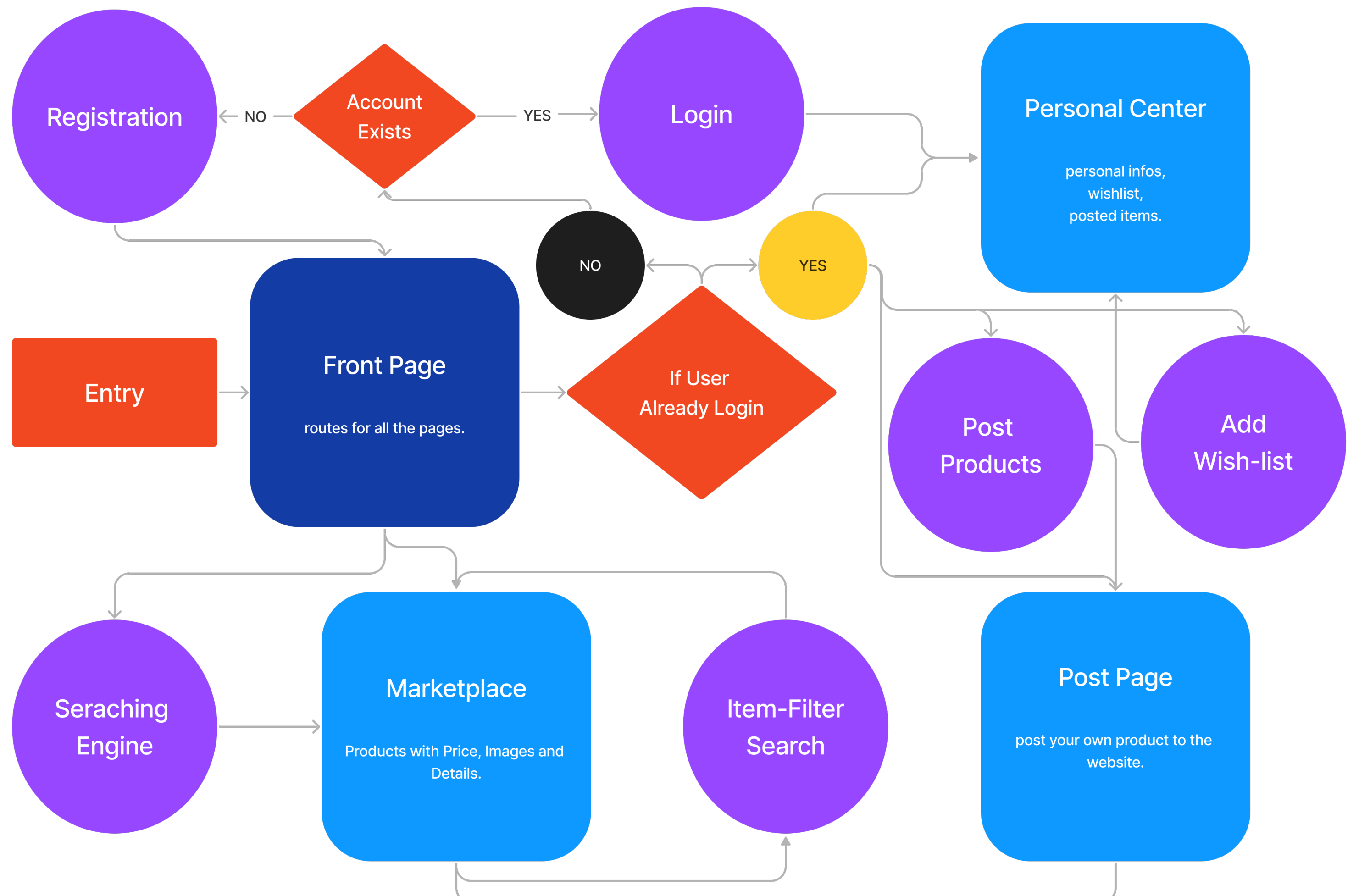
UI Design & Fronted-end

We have integrated Tailwind CSS and Redux Toolkit into our frontend development process to create an efficient and streamlined ecommerce platform.

With Tailwind's comprehensive CSS framework and pre-built ecommerce components, such as Wishlist, product listings, and detail pages, developers can save time and effort by not having to develop these features from scratch. Additionally, Tailwind's customization options allow for easy modifications to the design, and its pre-defined classes for breakpoints ensure that the design is responsive across different screen sizes.

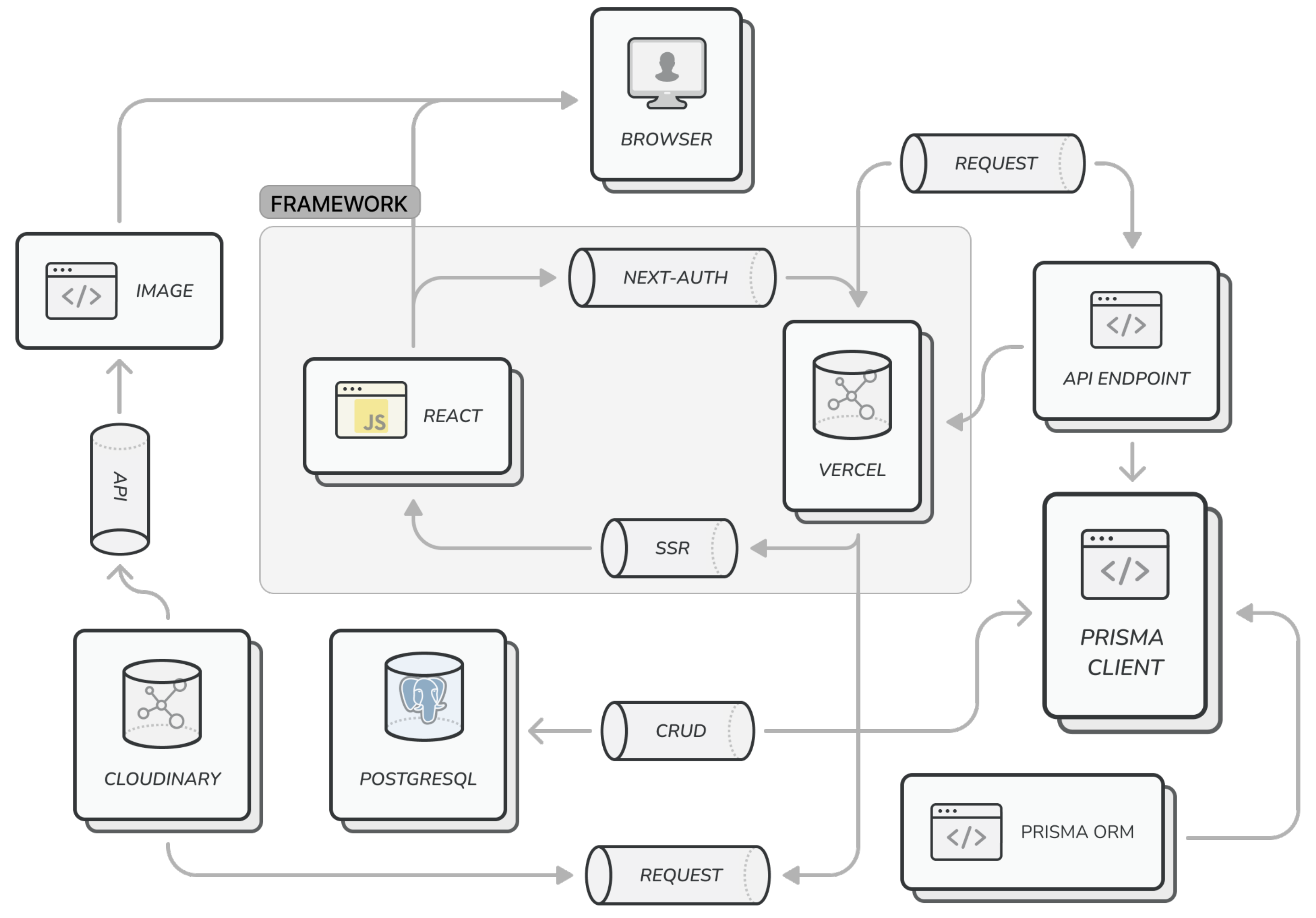
Furthermore, Redux Toolkit simplifies the process of writing Redux code by providing opinionated APIs for common tasks, such as creating reducers, defining actions, and configuring the Redux store. This allows developers to write more maintainable and concise code while also benefiting from the scalability and performance of the Redux architecture. The integration of these two tools allows our development team to create a responsive, customizable, and efficient ecommerce platform for our T2T Marketplace.

Framework



Once enter the website, users should create an account on the website by providing their email address, name, and password, and log in to the website after their registration.
On the front page and shopping page, the website displays a list of products that match the user's search criteria, including product name, image, price, and seller information. They can search for products by using the search bar, browsing categories, or filtering by price range or location.
During the browsing, users can add products to their wish list by clicking on the heart icon on the product page. They can also access their wish list from their personal center.
Users who want to sell their own items can post them for sale on the post page, and filling in product details such as title, description, category, and price.
All these information can be accessed through personal center to view their wish-list, purchase history, and other useful information. Users can also update their account information and preferences.

Backend Implementation



We deployed our website on Vercel and used Next.js as our framework. The server used dynamic rendering (SSR) to render the pages and used next-auth to verify users' authentication. Once verification succeeded, the server fetched data from our database. Data fetching is implemented by an API endpoint between the server-side and the Prisma client, where Prisma provides a type-safe and intuitive API that allows developers to easily interact with databases using JavaScript. And we deployed our database on a private cloud server using PostgreSQL and mapped the entities and tables into Prisma schema. As for product images, we upload the image onto Cloudinary through Cloudinary-API and save the public link in the database.

Database Design

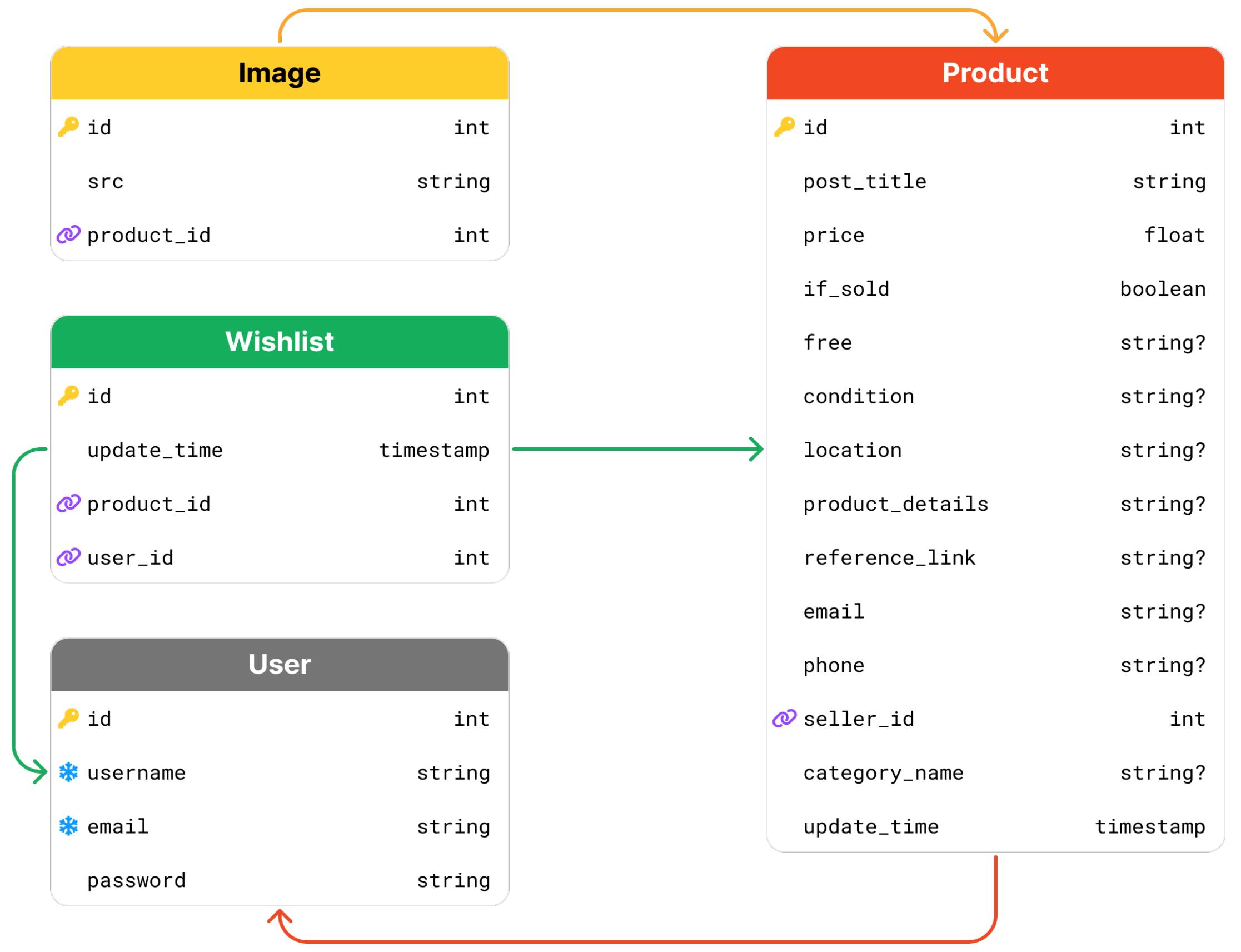


Image	
id	int
src	string
product_id	int

Wishlist	
id	int
update_time	timestamp
product_id	int
user_id	int

User	
id	int
username	string
email	string
password	string

Product	
id	int
post_title	string
price	float
if_sold	boolean
free	string?
condition	string?
location	string?
product_details	string?
reference_link	string?
email	string?
phone	string?
seller_id	int
category_name	string?
update_time	timestamp

Go-live / Server-Maintenance

To ensure the smooth operation of our online shopping platform, we are implementing a reliable and cost-effective Go-live and Server-maintenance system. Our platform is currently hosted on VERCEL, and we believe that this system will allow us to minimize downtime and provide our customers with a seamless online shopping experience.

To ensure the reliability and availability of our platform, we will implement a Go-live strategy that includes rolling deployment and A/B testing. We will also set up an automated Server-maintenance system that includes performance monitoring, log analysis, and alerts for potential issues.

Challenge

The first challenge we are facing with our website is its loading time. It takes a long time for the website to load, which affects user experience. One possible solution to address the loading time issue is to cache frequently used content. Next.js has a built-in caching feature that can significantly improve page loading times.
The second challenge is the adaptability for the phone device, the current user interface is optimized for PC view, but may not work perfectly on mobile device.
The last challenge is scalability, the current backend setting may not be able to handle large user traffic in the future.

Future Work

In future work, we plan to add the following features to our website:

1. **AI-powered recommendation system**
   To implement an AI-powered recommendation system in our ecommerce website, we would like to use machine learning algorithms to analyze user data such as browsing history, purchase history, and product interactions to provide personalized recommendations to users.
2. **Image-based search engine**
   Our goal is to enhance our ecommerce website by adding an image-based search engine. This can be achieved through the use of computer vision and machine learning to analyze product images, extract relevant features, and use these features to search for visually similar products in our catalog.
3. **GPT-powered chatbot**
   A GPT-powered chatbot can offer several benefits to our website, including providing 24/7 customer support, handling customer inquiries more efficiently, and offering personalized product recommendations, ultimately leading to improved customer satisfaction and increased sales.