

### Mern App

- Frontend: React js ./tucshop
- Backend: Node js with Express ./tucshop-product & ./tucshop-order
- Databases: MongoDB for both product and order databases
- Data Streaming: kafka with zookeeper and control-center
- Authentication System: Keycloak

### Execution

docker compose up --build

*go to your browser and open localhost:3000*

docker compose down to stop execution

*change localhost to if running on gcp*

- port 8182: Keycloak Console
- port 9021: Control Center, watch the pub-sub system
- MongoDB Compass to watch the database records. The urls are in .env file in ./tucshop-product & ./tucshop-order

### Explore the Eshop

When the app starts running, the user can login as a *Customer* or a *Seller* or *continue as guest*

Customer page: it is the main page which contains the products that are available. The user can search for a product, add it to the cart and browse through the site.

- **Show Orders** , shows all the orders

- **Cart** , the products that are added to the cart are also saved in the sessionStorage, inside this page the user can modify the amount of the cart products before purchasing. After the *Purchase* button is pressed, the order receipt is generated in a pdf file.

Seller page: - **Seller Tools** , it is the *my products* page the user can create a new product and update the data of an existing product

### Migration Guide

If you want to migrate the eshop to a different machine follow these steps:

1. Change the vm's ip inside the code. Go to docker-compose.yml: in keycloak\_authentication\_service: change KC\_HOSTNAME: <vm\_ip>. Go to tucshop.config.js change localhost = <vm\_ip>

2. Change the keycloak configurations. Run `docker compose up --build`. Go to `http://vm_ip:8182` and import the realm file `realm-export.json`. Then import the client file `frontend-app.json`. Finally go to TUCSHOP realm settings, then credentials copy the rs 256 secret to the `.env` file in `./tucshop-product` & `./tucshop-order`
3. Stop the app. Run `docker compose down`
4. Start the app again!