

Online Grocery Application prototype installation.

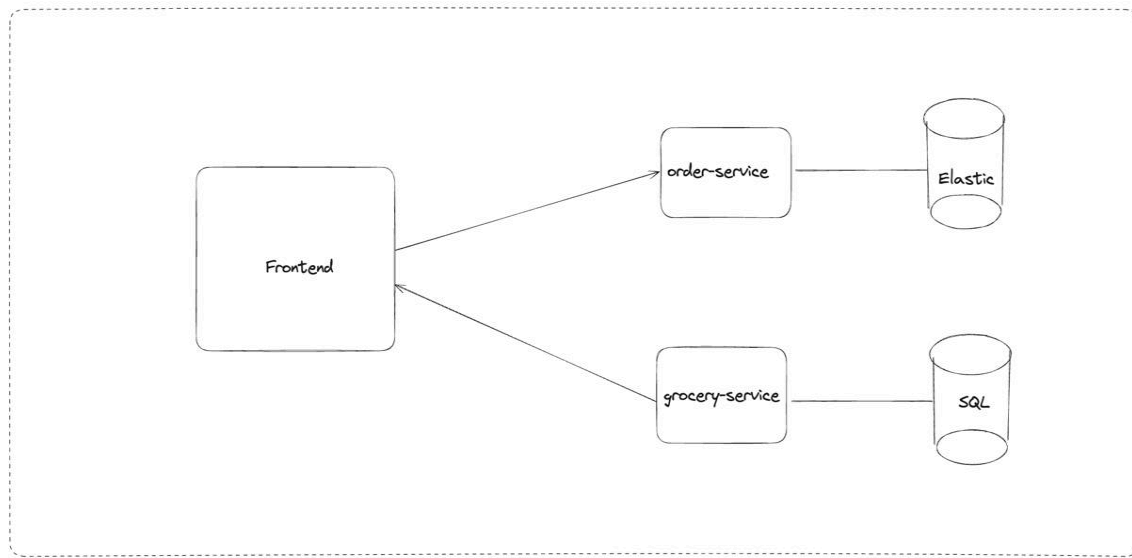
By Luba Yanovsky

Download from: <https://github.com/yanol/online-grocery-1.git>

OS: Linux. Dev Tool: VSC

Note : Online grocery was finished partially. It Fronted part was not completed.

The design of program :



1. DB running :

- For **elasticsearch docker**, please run command:
`sudo docker-compose -f elastic-docker-compose.yml up -d`
- For **SQL server docker**, please run command:
`sudo docker-compose -f sql-docker-compose.yml up -d`
- For creating sql tables run script : `product_details.sql` in sql server.

2. Download and unzip 3 files

- `React-dropdown-main.zip` - Code that starts the frontend part.
For running use commands : `npm run build`
`npm run dev`
- First backend service in file `grocery-list-dotnet.zip` - Was developed for handling products details and grocery categories.
Used in technology: Dotnet core 8.0, RestAPI (Port 5000). DB - SQL Server, EF

For running use commands : `dotnet build`

dotnet run

- Second backend service in file order-service-node.zip - Was developed for handling and saving final order details.

Used in technology: NodeJS, RestAPI (port 3008). DB - Elasticsearch

For running use commands : node app.js

All services were tested with postman application

Part 2:

- Frontend page 2 should be finished.
- Automation Tests should be added.
- For Logging could be used cloud observable programs : Datadog / logz.io / splunk.

For automated deployment kubernetes should be added.

In Production: User authentication could be managed by AWS Cognito / AUTH0.
Elastic and SQL DB should be managed in cloud.

Horizontal Scaling: In case of high traffic Gateway / load balancer/ application balancer should be added.