Backend:

Databases:

* Categories.json – list of categories
* Products.json – list of products
* List.json – the shopping list

Folders :

* Databases
* Receipts – has subfolders of companies – osherad, yohananof
* backup

API Endpoints:

* POST /fetchReceipt
  + Gets url from body
  + Takes the raw url: <https://osher.pairzon.com/dba51f17-81ac-41f8-b76d-64e62fb13df4.html?id=fe4f9bcf-050e-48f5-af2b-bb5a7b81f001&p=1247> converts to that url - <https://osher.pairzon.com/v1.0/documents/fe4f9bcf-050e-48f5-af2b-bb5a7b81f001?p=1247>
  + Going to download the file and saves it in receipts folder as json file
  + Requesting to <https://pdf.pairzon.com/pdf/fe4f9bcf-050e-48f5-af2b-bb5a7b81f001/1247> to get the pdf and saves on backup folder
  + Triggers the processing flow
    - Processing flow:
      * Open receipt file

For each place on 'items'

* + - * + If barcode doesn’t exists on products.json Add to products.json the product name, category (empty),
* POST api/product/add – adds product name to list.json
* POST api/product/remove – removes product from list.json
* POST api/product/delete – deletes product from products.json
* POST api/product/quantity – update product quantity on list.json
* GET api/product/settings – returns the product with his settings
* POST api/product/settings – update settings of product
* GET api/receipt/{receipt\_name} – gets name and returns json of receipt
* GET api/receipt/download – downloads pdf of the receipt by transactionID from param
* GET api/list – returns the list.json
* GET api/receipts – returns json list of the Receipts folder
* GET api/products– returns json of products.json

Frontend:

General:

* Visually looks like SPA webapp
* Specially takes care of mobile responsiveness
* Sidebar menu that has the following with icons:
  + Shopping list
  + Products
  + Statistics
  + Receipts

API Endpoints:

* + /list – displays the shopping list
    - On body load Send req to /api/list and get the json from body
    - Send req to /api/products
    - Each product gets the following button that sends proper request to server: delete (trash icon), checkbox to sign as done. Quantity dropdown when selected updates quantity of product
    - If no products show centered text says that
    - Floating + button
  + /stats – shows overall shop stats (when loaded numbers should do counting animation)
  + /stats/{product\_barcode} – shows specific products stats by barcode
  + /receipts – displays list of receipts available by company and branch
  + /receipts/{ transactionID } – displays the receipt itself as designed html
  + /products
    - displays the products list with search filter by name, barcode (each item should has button to redirect /stats/{product\_barcode} and settings icon to redirect /product/{product\_barcode}/settings
    - has general button (camera icon) that triggers barcode scanner to extract barcode and search it.
  + /product/settings – sends req to GET /api/product/settings?product=name
    - * let the user update settings of product – name and category. Display the barcode as disabled