

Grocery Application Documentation

Overview

The Grocery App is a full-stack web application built using the MERN (MongoDB, Express.js, React.js, Node.js) stack. The application allows users to browse, add, update, and delete grocery items while interacting with a MongoDB database through a RESTful API.

Technology Stack

- **Frontend:** React.js
- **Backend:** Node.js with Express.js
- **Database:** MongoDB (Hosted on AWS)
- **Deployment:** AWS EC2 Ubuntu Instance

Instance and Backend setup

Launch an EC2 server with Ubuntu to deploy the server.

Database, Backend and Frontend all are installed and configured in the same instance, you can use multiple instances also.

NOTE: ALLOW PORT 5000 for backend server and 3000 for frontend server from security group

Launch an EC2 Ubuntu instance

```
0 updates can be applied immediately.
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-4-4:~$
```

Clone the repository in which the data of the project is there

```
git clone https://github.com/rajatpzade/GroceryApp.git
```

```
ubuntu@ip-172-31-4-4:~$ git clone https://github.com/rajatpzade/GroceryApp.git
Cloning into 'GroceryApp'...
remote: Enumerating objects: 1818, done.
remote: Counting objects: 100% (1818/1818), done.
remote: Compressing objects: 100% (1543/1543), done.
remote: Total 1818 (delta 200), reused 1814 (delta 199), pack-reused 0 (from 0)
Receiving objects: 100% (1818/1818), 3.13 MiB | 13.81 MiB/s, done.
Resolving deltas: 100% (200/200), done.
ubuntu@ip-172-31-4-4:~$
```

Install and configure the backend server and then integrate with database

For Backend we are installing node js and npm

Nodejs - Node.js is a **runtime environment** that allows you to run JavaScript outside of the browser.

NPM - npm is the **default package manager** for Node.js. It helps developers **install, manage, and share** JavaScript libraries (called **packages** or **modules**) easily.

- **Install node js**
- **curl -fsSL https://deb.nodesource.com/setup_lts.x | sudo -E bash -**
- **sudo apt install -y nodejs**

```
ubuntu@ip-172-31-4-4:~$ curl -fsSL https://deb.nodesource.com/setup_lts.x | sudo -E bash -
sudo apt install -y nodejs
2025-03-03 11:01:14 - Installing pre-requisites
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:5 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:10 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:11 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
```

```
after this operation, 223 MB of additional disk space will be used.
Get:1 https://deb.nodesource.com/node_22.x nodistro/main amd64 nodejs amd64 22.14.0-1nodesource1 [36.4 MB]
Fetched 36.4 MB in 1s (65.3 MB/s)
Selecting previously unselected package nodejs.
(Reading database ... 70614 files and directories currently installed.)
Preparing to unpack .../nodejs_22.14.0-1nodesource1_amd64.deb ...
Unpacking nodejs (22.14.0-1nodesource1) ...
Setting up nodejs (22.14.0-1nodesource1) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
```

Explanation of the command `curl -fsSL https://deb.nodesource.com/setup_lts.x | sudo -E bash -`
`sudo apt install -y nodejs`

- **curl** → Command-line tool to download files from a URL.
- **-fsSL** → These flags make curl more secure and reliable:
- **-f** → Fails silently on HTTP errors.
- **-s** → Runs in silent mode (hides progress output).
- **-S** → Shows errors (useful when combined with -s).
- **-L** → Follows redirects if the URL points to another location.
- **https://deb.nodesource.com/setup_lts.x** → This is the official NodeSource script that sets up the repository for **LTS (Long-Term Support) version** of Node.js.
- **| (pipe)** → Passes the output of the first command (curl) as input to the second command (bash).
- **sudo -E bash -**
- **sudo** → Runs the command with **root** privileges.

- -E → Preserves the user's environment variables.
- bash - → Executes the downloaded script.

Check the version of npm and nodejs

npm -v

nodejs -v

Now install and configure the database

MongoDB is a NoSQL database that stores data as JSON-like documents.

Install mongodb database from the official website

<https://www.mongodb.com/docs/manual/tutorial/install-mongodb-on-ubuntu/>

1. Import the public key.

From a terminal, install gnupg and curl if they are not already available:

```
sudo apt-get install gnupg curl
```

```
ubuntu@ip-172-31-4-4:~$ sudo apt-get install gnupg curl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
gnupg is already the newest version (2.4.4-2ubuntu17).
curl is already the newest version (8.5.0-2ubuntu10.6).
0 upgraded, 0 newly installed, 0 to remove and 134 not upgraded.
ubuntu@ip-172-31-4-4:~$
```

To import the MongoDB public GPG key, run the following command:

```
curl -fsSL https://www.mongodb.org/static/pgp/server-8.0.asc | \
```

```
sudo gpg -o /usr/share/keyrings/mongodb-server-8.0.gpg \
```

```
--dearmor
```

```
ubuntu@ip-172-31-4-4:~$ curl -fsSL https://www.mongodb.org/static/pgp/server-8.0.asc | \
sudo gpg -o /usr/share/keyrings/mongodb-server-8.0.gpg \
--dearmor
ubuntu@ip-172-31-4-4:~$
```

2. Create the list file.

```
echo "deb [ arch=amd64,arm64 signed-by=/usr/share/keyrings/mongodb-server-8.0.gpg ]
https://repo.mongodb.org/apt/ubuntu noble/mongodb-org/8.0 multiverse" | sudo tee
/etc/apt/sources.list.d/mongodb-org-8.0.list
```

```
ubuntu@ip-172-31-4-4:~$ echo "deb [ arch=amd64,arm64 signed-by=/usr/share/keyrings/mongodb-server-8.0.gpg ] https://repo.mongodb.org/apt/ubuntu nobl
e/mongodb-org/8.0 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-8.0.list
deb [ arch=amd64,arm64 signed-by=/usr/share/keyrings/mongodb-server-8.0.gpg ] https://repo.mongodb.org/apt/ubuntu noble/mongodb-org/8.0 multiverse
ubuntu@ip-172-31-4-4:~$
```

3. Reload the package database.

sudo apt-get update

```
ubuntu@ip-172-31-4-4:~$ sudo apt-get update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 https://deb.nodesource.com/node_22.x nodistro InRelease
Hit:5 http://security.ubuntu.com/ubuntu noble-security InRelease
Get:6 https://repo.mongodb.org/apt/ubuntu noble/mongodb-org/8.0 InRelease [3005 B]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1028 kB]
Get:8 https://repo.mongodb.org/apt/ubuntu noble/mongodb-org/8.0/multiverse amd64 Packages [23.9 kB]
Get:9 https://repo.mongodb.org/apt/ubuntu noble/mongodb-org/8.0/multiverse arm64 Packages [23.9 kB]
Fetched 1205 kB in 2s (728 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-4-4:~$
```

4. Install MongoDB Community Server.

sudo apt-get install -y mongodb-org

```
ubuntu@ip-172-31-4-4:~$ sudo apt-get install gnupg curl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
gnupg is already the newest version (2.4.4-2ubuntu17).
curl is already the newest version (8.5.0-2ubuntu10.6).
0 upgraded, 0 newly installed, 0 to remove and 134 not upgraded.
ubuntu@ip-172-31-4-4:~$ curl -fsSL https://www.mongodb.org/static/pgp/server-8.0.asc | \
  sudo gpg -o /usr/share/keyrings/mongodb-server-8.0.gpg \
  --dearmor
ubuntu@ip-172-31-4-4:~$ echo "deb [ arch=amd64,arm64 signed-by=/usr/share/keyrings/mongodb-server-8.0.gpg ] https://repo.mongodb.org/apt/ubuntu nobl
e/mongodb-org/8.0 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-8.0.list
deb [ arch=amd64,arm64 signed-by=/usr/share/keyrings/mongodb-server-8.0.gpg ] https://repo.mongodb.org/apt/ubuntu noble/mongodb-org/8.0 multiverse
ubuntu@ip-172-31-4-4:~$ sudo apt-get update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 https://deb.nodesource.com/node_22.x nodistro InRelease
Hit:5 http://security.ubuntu.com/ubuntu noble-security InRelease
Get:6 https://repo.mongodb.org/apt/ubuntu noble/mongodb-org/8.0 InRelease [3005 B]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1028 kB]
Get:8 https://repo.mongodb.org/apt/ubuntu noble/mongodb-org/8.0/multiverse amd64 Packages [23.9 kB]
```

5. Check the version of the Mongoddb

mongod --version

```
ubuntu@ip-172-31-4-4:~$ mongod --version
db version v8.0.5
Build Info: {
  "version": "8.0.5",
  "gitVersion": "cb9e2e5e552ee39deale39d7859336456d0c9820",
  "opensslVersion": "OpenSSL 3.0.13 30 Jan 2024",
  "modules": [],
  "allocator": "tcmalloc-google",
  "environment": {
    "distmod": "ubuntu2404",
    "distarch": "x86_64",
    "target_arch": "x86_64"
  }
}
ubuntu@ip-172-31-4-4:~$
```

6. Start the Mongoddb

sudo systemctl start mongod

7. Status of the Mongoddb

sudo systemctl status mongod

```
ubuntu@ip-172-31-4-4:~$ sudo systemctl start mongod
ubuntu@ip-172-31-4-4:~$ sudo systemctl status mongod
● mongod.service - MongoDB Database Server
   Loaded: loaded (/usr/lib/systemd/system/mongod.service; disabled; preset: enabled)
   Active: active (running) since Mon 2025-03-03 11:12:41 UTC; 13s ago
     Docs: https://docs.mongodb.org/manual
    Main PID: 3166 (mongod)
      Memory: 96.1M (peak: 96.2M)
        CPU: 780ms
    CGroup: /system.slice/mongod.service
            └─3166 /usr/bin/mongod --config /etc/mongod.conf

Mar 03 11:12:41 ip-172-31-4-4 systemd[1]: Started mongod.service - MongoDB Database Server.
Mar 03 11:12:41 ip-172-31-4-4 mongod[3166]: {"t":{"$date":"2025-03-03T11:12:41.444Z"},"s":"i", "c":"CONTROL", "id":7484500, "ctx":"main","ms
ubuntu@ip-172-31-4-4:~$
```

Connect to the mongodb database

mongosh

```
ubuntu@ip-172-31-4-4:~$ mongosh
Current Mongoosh Log ID: 67c58f2ee13aafff8451e943
Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.4.0
Using MongoDB:      8.0.5
Using Mongosh:      2.4.0

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/

To help improve our products, anonymous usage data is collected and sent to MongoDB periodically (https://www.mongodb.com/legal/privacy-policy).
You can opt-out by running the disableTelemetry() command.

-----
  The server generated these startup warnings when booting
  2025-03-03T11:12:41.791+00:00: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
  2025-03-03T11:12:42.476+00:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
  2025-03-03T11:12:42.476+00:00: For customers running the current memory allocator, we suggest changing the contents of the following sysfsFile
  2025-03-03T11:12:42.476+00:00: For customers running the current memory allocator, we suggest changing the contents of the following sysfsFile
  2025-03-03T11:12:42.476+00:00: We suggest setting the contents of sysfsFile to 0.
  -----

test>
```

Access the database

use grocerydb

```
test> use grocerydb
switched to db grocerydb
```

View the data inside the database

db.groceries.find()

No data will shown, as not data is recorded.

```
grocerydb> db.groceries.find()
grocerydb> exit
```

PM2 is a **production process manager** for Node.js applications. It helps manage, monitor, and keep applications running continuously in the background, even after system restarts.

Install pm2

```
sudo npm install -g pm2
```

```
ubuntu@ip-172-31-4-4:~$ sudo npm install -g pm2
added 137 packages in 9s
13 packages are looking for funding
  run 'npm fund' for details
npm notice
npm notice New major version of npm available! 10.9.2 -> 11.1.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v11.1.0
npm notice To update run: npm install -g npm@11.1.0
npm notice
```

Switch to BE location to start the server and fetched the grocery-backend data from the repository.

cd /Grocery-app/BE

pm2 start server.js --name grocery-backend

```
ubuntu@ip-172-31-4-4:~$ cd GroceryApp/BE/
ubuntu@ip-172-31-4-4:~/GroceryApp/BE$ pm2 start server.js --name grocery-backend
[PM2] Starting /home/ubuntu/GroceryApp/BE/server.js in fork_mode (1 instance)
[PM2] Done.
```

id	name	namespace	version	mode	pid	uptime	U	status	cpu	mem	user	watching
0	grocery-backend	default	1.0.0	Fork	3311	0s	0	online	0%	50.5mb	ubuntu	disabled

```
ubuntu@ip-172-31-4-4:~/GroceryApp/BE$
```

? **pm2 start server.js** → Starts the **server.js** file in the background.

? **--name grocery-backend** → Assigns a custom name (grocery-backend) to the process, making it easier to manage.

Save the fetched data

pm2 save

```
ubuntu@ip-172-31-4-4:~/GroceryApp/BE$ pm2 save
[PM2] Saving current process list...
[PM2] Successfully saved in /home/ubuntu/.pm2/dump.pm2
ubuntu@ip-172-31-4-4:~/GroceryApp/BE$
```

pm2 startup

```
ubuntu@ip-172-31-4-4:~/GroceryApp/BE$ pm2 startup
[PM2] Init System found: systemd
[PM2] To setup the Startup Script, copy/paste the following command:
sudo env PATH=$PATH:/usr/bin:/usr/lib/node_modules/pm2/bin/pm2 startup systemd -u ubuntu --hp /home/ubuntu
```

pm2 list

```
ubuntu@ip-172-31-4-4:~/GroceryApp/BE$ pm2 list
```

id	name	namespace	version	mode	pid	uptime	U	status	cpu	mem	user	watching
0	grocery-backend	default	1.0.0	Fork	3311	106s	0	online	0%	75.7mb	ubuntu	disabled

✦ **pm2 start server.js --name grocery-backend** → Runs your app in the background with a custom name.

✦ **pm2 list** → View running processes.

✦ **pm2 logs grocery-backend** → Check logs.

✦ **pm2 restart grocery-backend** → Restart the app.

✦ **pm2 save && pm2 startup** → Enable auto-restart on server reboot.

• Other pm2 commands (which we not used)

#pm2 stop grocery-backend will stop your pm2 backend api

#[PM2] Freeze a process list on reboot via: few additional commands

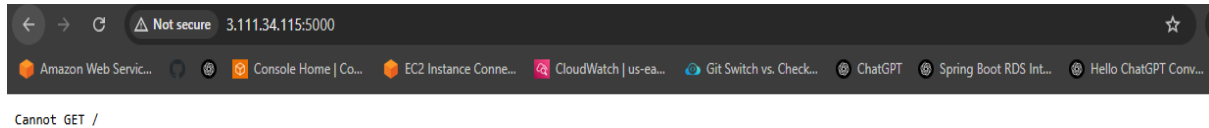
#\$ pm2 save

#[PM2] Remove init script via:

#\$ pm2 unstartup system

Now, check the backend is running or not

<public ip :5000>



This output means - Backend is perfectly running

Frontend configuration

Make changes in .env file in /FE

```
cd GroceryApp/FE/
ls -a
ubuntu@ip-172-31-4-4:~/GroceryApp$ cd FE/
ubuntu@ip-172-31-4-4:~/GroceryApp/FE$ ls -a
.  ..  .DS_Store  .env  README.md  package-lock.json  package.json  public  src
ubuntu@ip-172-31-4-4:~/GroceryApp/FE$ vim .env
ubuntu@ip-172-31-4-4:~/GroceryApp/FE$
```

```
vim .env
REACT_APP_API_URL=http://3.111.34.115:5000
```

Install npm and start in the /GroceryApp/FE directory

```
npm install
ubuntu@ip-172-31-4-4:~/GroceryApp/FE$ npm install
npm warn deprecated w3c-hr-time@1.0.2: Use your platform's native performance.now() and performance.timeOrigin.
npm warn deprecated stable@0.1.8: Modern JS already guarantees Array#sort() is a stable sort, so this library is deprecated. See the compatibility table on MDN: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/sort#browser_compatibility
npm warn deprecated source-map-codec@1.4.8: Please use @jridgewell/source-map-codec instead
npm warn deprecated rimraf@3.0.2: Rimraf versions prior to v4 are no longer supported
npm warn deprecated rollup-plugin-terser@7.0.2: This package has been deprecated and is no longer maintained. Please use @rollup/plugin-terser
npm warn deprecated q@1.5.1: You or someone you depend on is using Q, the JavaScript Promise library that gave JavaScript developers strong feelings about promises. They can almost certainly migrate to the native JavaScript promise now. Thank you literally everyone for joining me in this bet against the odds. Be excellent to each other.
npm warn deprecated
npm warn deprecated (For a CapTP with native promises, see @endo/eventual-send and @endo/captp)
npm warn deprecated workbox-cacheable-response@6.6.0: workbox-background-sync@6.6.0
npm warn deprecated workbox-google-analytics@6.6.0: It is not compatible with newer versions of GA starting with v4, as long as you are using GAv3 it should be ok, but the package is not longer being maintained
npm warn deprecated inflight@1.0.6: This module is not supported, and leaks memory. Do not use it. Check out lru-cache if you want a good and tested way to coalesce async requests by a key value, which is much more comprehensive and powerful.
npm warn deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
npm warn deprecated domexception@2.0.1: Use your platform's native DOMException instead
npm warn deprecated abab@2.0.6: Use your platform's native atob() and btoa() methods instead
npm warn deprecated svgo@1.3.2: This SVGO version is no longer supported. Upgrade to v2.x.x.
npm warn deprecated @humanwhocodes/object-schema@2.0.3: Use @eslint/object-schema instead
npm warn deprecated @humanwhocodes/config-array@0.13.0: Use @eslint/config-array instead
```

npm install (or npm i for short) is a command used in Node.js projects to install dependencies listed in the package.json file.

Key Files Created

node_modules/ → Directory containing installed packages.

package.json → Lists project dependencies and metadata.

package-lock.json → Locks dependency versions for consistency.

npm start

```
Compiled successfully!

You can now view my-grocery-app in the browser.

Local:            http://localhost:3000
On Your Network:  http://172.31.4.4:3000

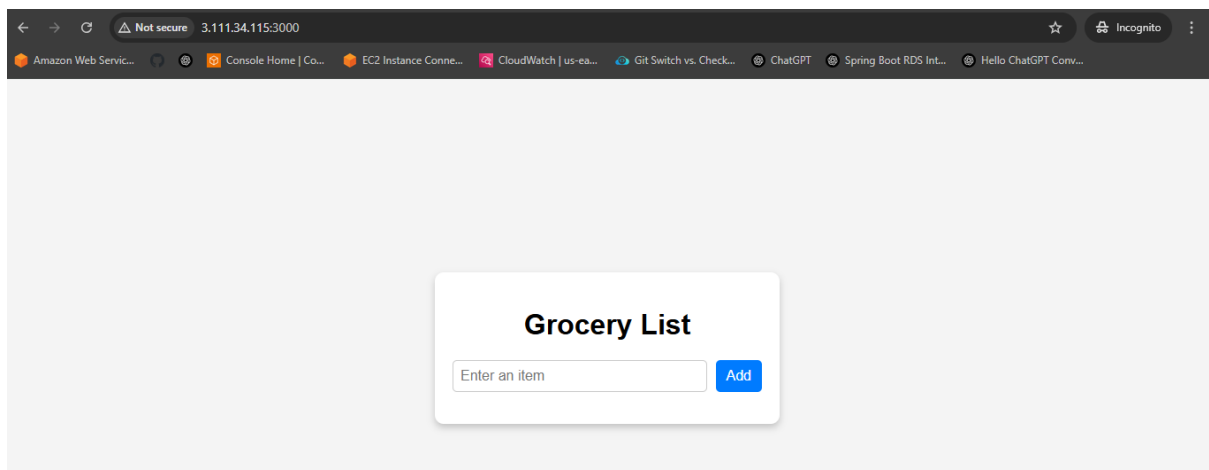
Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

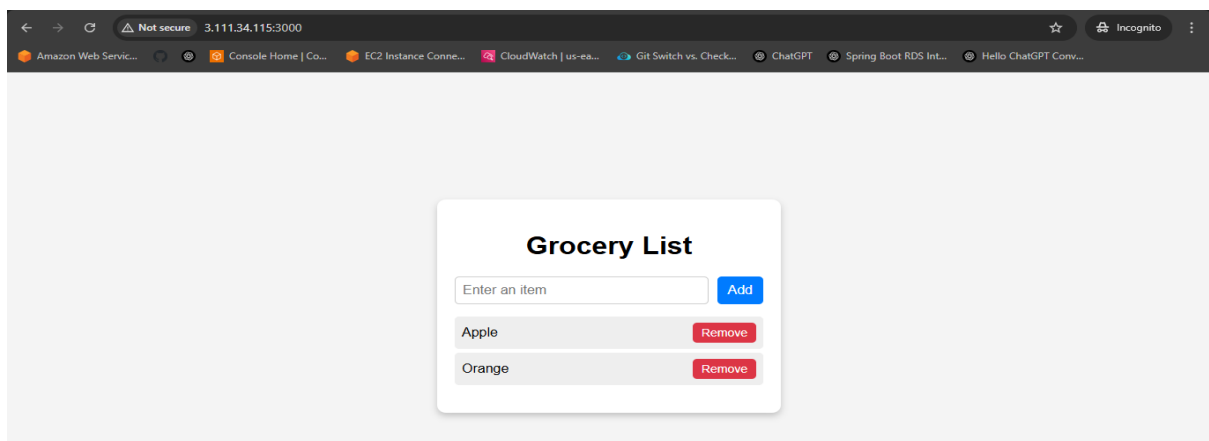
npm start is a command used to run the start script defined in the package.json file of a Node.js project.

On CLI Frontend application is started

Check with the server, Hit the IP in the server <public ip>:3000



Add data in the box and click on Add button



Now check with CLI database whether the data is saved and can we make changes in the database.

Go to CLI

cd

Come to home directory Ubuntuip - <ip> \$

Now access the database with

```
mongosh
ubuntu@ip-172-31-4-4:~/GroceryApp/FE$ cd
cd
ubuntu@ip-172-31-4-4:~$ mongosh
Current Mongosh Log ID: 67c595beefa73eec7951e943
Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.4.0
Using MongoDB:      8.0.5
Using Mongosh:       2.4.0

For mongosh info see: https://www.mongodb.com/docs/mongosh-shell/

-----
The server generated these startup warnings when booting
2025-03-03T11:12:41.791+00:00: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
2025-03-03T11:12:42.476+00:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
2025-03-03T11:12:42.476+00:00: For customers running the current memory allocator, we suggest changing the contents of the following sysfsFile
2025-03-03T11:12:42.476+00:00: For customers running the current memory allocator, we suggest changing the contents of the following sysfsFile
2025-03-03T11:12:42.476+00:00: We suggest setting the contents of sysfsFile to 0.
-----
```

Enter the database name with use

use grocerydb

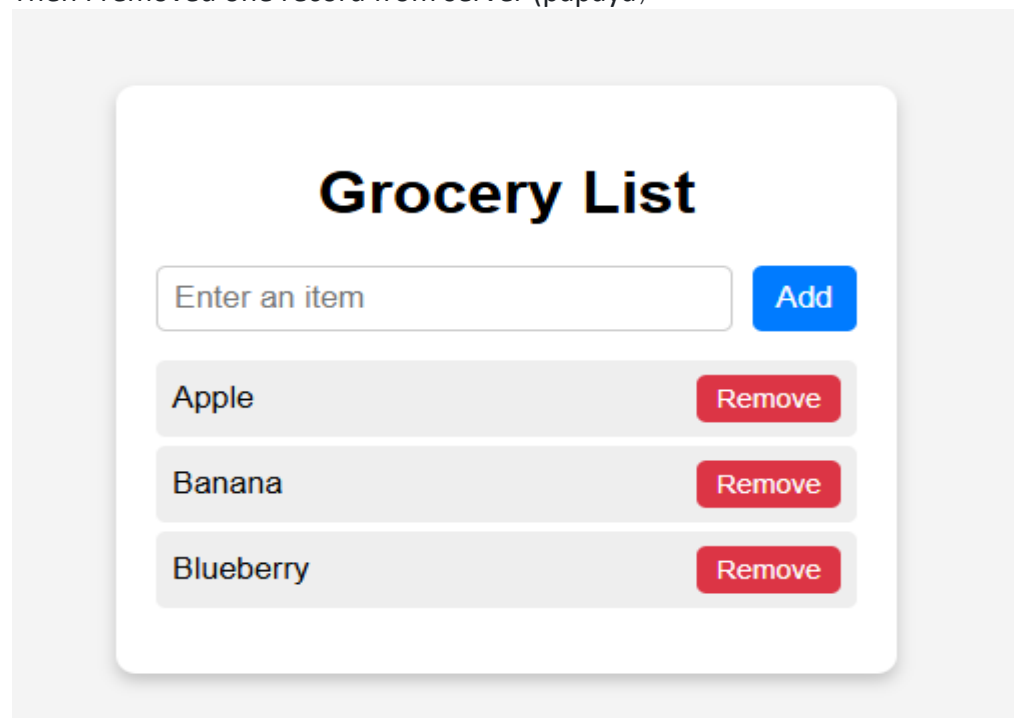
To list the all stored data in the database

db.groceries.find()

```
test> use grocerydb
switched to db grocerydb
grocerydb> db.groceries.find()
[
  { _id: ObjectId('67c596722dbd3db334584832'), name: 'Apple', __v: 0 },
  { _id: ObjectId('67c5967a2dbd3db334584834'), name: 'Banana', __v: 0 },
  { _id: ObjectId('67c5967f2dbd3db334584836'), name: 'Papaya', __v: 0 },
  {
    _id: ObjectId('67c596842dbd3db334584838'),
    name: 'Blueberry',
    __v: 0
  }
]
grocerydb>
```

Earlier I have 4 records so it is showing 4 records,

Then I removed one record from server (papaya)



Now I run the command `db.groceries.find()` to see the list of records in the database

```
grocerydb> db.groceries.find()
[
  {
    _id: ObjectId('67c596722dbd3db334584832'), name: 'Apple', __v: 0 },
  {
    _id: ObjectId('67c5967a2dbd3db334584834'), name: 'Banana', __v: 0 },
  {
    _id: ObjectId('67c596842dbd3db334584838'),
    name: 'Blueberry',
    __v: 0
  }
]
```

As I can see it is showing 3 records of fruits, this means the deleted record (papaya) has been deleted from the database

Now I will Deploy frontend – Groceryapp on nginx

Nginx Installation

```
sudo apt update
```

```
sudo apt install nginx -y
```

```
ubuntu@ip-172-31-4-4:~$ sudo apt update
sudo apt install nginx -y

Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 https://deb.nodesource.com/node_22.x nodistro InRelease
Hit:5 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:6 https://repo.mongodb.org/apt/ubuntu noble/mongodb-org/8.0 InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
134 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  nginx-common
Suggested packages:
  fcgiwrap nginx-doc ssl-cert
  nginx nginx-common
0 upgraded, 2 newly installed, 0 to remove and 134 not upgraded.
Need to get 552 kB of archives.
After this operation, 1596 kB of additional disk space will be used.
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 nginx-common all 1.24.0-2ubuntu7.1 [31.2 kB]
```

These two commands are used to install **Nginx**, a popular web server, on a Debian-based Linux system (like Ubuntu).

❓ `sudo` → Runs the command as a superuser (admin privileges).

❓ `apt` → The package manager for Debian-based systems.

❓ `update` → Refreshes the local package index, ensuring you get the latest versions from the repositories.

❓ `install nginx` → Installs the Nginx web server.

❓ `-y` → Automatically confirms installation (otherwise, you'd have to type "Y" manually).

To check nginx is installed or not, for this we can check the version of the server

nginx -version

```
ubuntu@ip-172-31-4-4:~$ nginx -version
nginx version: nginx/1.24.0 (Ubuntu)
```

This command shows the version of the server which we installed

Deploy a Frontend Build

cd /Groceryapp/FE

run the build command - **npm run build**

```
ubuntu@ip-172-31-4-4:~/GroceryApp/FE$ npm run build
> my-grocery-app@0.1.0 build
> react-scripts build

Creating an optimized production build...
Compiled successfully.

File sizes after gzip:

  59.62 kB  build/static/js/main.27681343.js
  1.78 kB   build/static/js/453.be86f43f.chunk.js
  263 B     build/static/css/main.e6c13ad2.css

The project was built assuming it is hosted at /.
You can control this with the homepage field in your package.json.

The build folder is ready to be deployed.
You may serve it with a static server:

  npm install -g serve
  serve -s build
```

npm run build is a command used in Node.js projects to create an optimized, production-ready version of the application.

As the command is executed successfully build version is created of application, now I can deploy on the server

We can see the created build directory after complete execution of npm run build command

```
ubuntu@ip-172-31-4-4:~/GroceryApp/FE$ ls
README.md  build  node_modules  package-lock.json  package.json  public  src
ubuntu@ip-172-31-4-4:~/GroceryApp/FE$
```

Inside the Build directory the file packages and dependencies are stored (asset-manifest.json favicon.ico index.html logo192.png logo512.png manifest.json robots.txt static)

And the files inside the build directory cannot be executed from the current directory for which we have to copy the all files to /var/www/grocery-app before which we have to create a directory in the /var directory as mkdir -p /var/www/grocery-app

Now I create the directory in the /var

```
mkdir -p /var/www/grocery-app
```

```
ubuntu@ip-172-31-4-4:~$ sudo mkdir -p /var/www/grocery-app
```

Then I can execute the cp for copy the files to the source destination

```
cp -r build/* /var/www/grocery-app
```

```
ubuntu@ip-172-31-4-4:~$ cd GroceryApp/FE/
ubuntu@ip-172-31-4-4:~/GroceryApp/FE$ ls
README.md  build  node_modules  package-lock.json  package.json  public  src
ubuntu@ip-172-31-4-4:~/GroceryApp/FE$ sudo cp -r build/* /var/www/grocery-app/
```

After executing the cp command without any error, I will check the files in the /var, whether all are properly copied.

```
ubuntu@ip-172-31-4-4:~/GroceryApp/FE$ cd /var/www/grocery-app/
ubuntu@ip-172-31-4-4:/var/www/grocery-app$ ls
asset-manifest.json  favicon.ico  index.html  logo192.png  logo512.png  manifest.json  robots.txt  static
ubuntu@ip-172-31-4-4:/var/www/grocery-app$
```

Now I have to configure the nginx then only the application will run on the server

Configuration file are in /etc/nginx/sites-available

Command to create the grocery-app inside the /etc directory **sudo vim /etc/nginx/sites-available/grocery-app**

Store the command inside the /etc/nginx/sites-available/grocery-app to set the reverse proxy

```
server {
    listen 80;
    server_name your_domain_or_ip; #- Replace `your_domain_or_ip` with your server
public IP or domain name

    root /var/www/grocery-app;
    index index.html;

    location / {
        try_files $uri /index.html;
    }
}
```

In this line in script server_name your_domain_or_ip; I have entered my public ip.

```
server {
    listen 80;
    server_name 3.111.34.115; #- Replace `your_domain_or_ip` with your server public IP or domain name

    root /var/www/grocery-app;
    index index.html;

    location / {
        try_files $uri /index.html;
    }
}
```

Now I have to Configure and restart the nginx server

```
sudo ln -s /etc/nginx/sites-available/grocery-app /etc/nginx/sites-enabled/
```

```
sudo nginx -t
```

```
ubuntu@ip-172-31-4-4:~$ sudo ln -s /etc/nginx/sites-available/grocery-app /etc/nginx/sites-enabled/
ubuntu@ip-172-31-4-4:~$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
ubuntu@ip-172-31-4-4:~$
```

sudo → Runs the command with superuser (admin) privileges.

In -s → Creates a **symbolic (soft) link**.

/etc/nginx/sites-available/grocery-app → The actual configuration file for your **Grocery App**.
/etc/nginx/sites-enabled/ → Directory where Nginx looks for active site configurations.

Why is This Needed?

In **Nginx**, site configuration files are usually stored in /etc/nginx/sites-available/ but are **not active** by default.

By creating a **symbolic link** in /etc/nginx/sites-enabled/, Nginx recognizes the site configuration and serves it.

sudo nginx -t

sudo → Runs the command with superuser (admin) privileges.

nginx -t → Tests the Nginx configuration files.

Expected Output (Success)

If your configuration is correct, you'll see:

swift

CopyEdit

nginx: the configuration file /etc/nginx/nginx.conf syntax is ok

nginx: configuration file /etc/nginx/nginx.conf test is successful

This means Nginx will run without issues.

Now the restart the server

sudo systemctl restart nginx

and check the status of the server

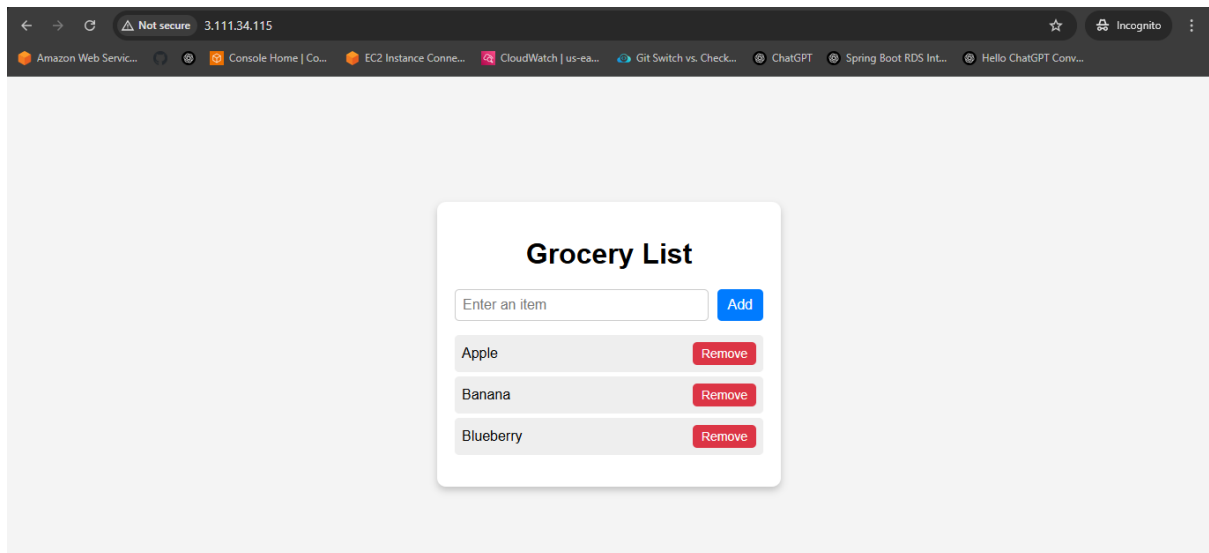
sudo systemctl status nginx

```
ubuntu@ip-172-31-4-4:~$ sudo systemctl restart nginx
ubuntu@ip-172-31-4-4:~$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: enabled)
   Active: active (running) since Mon 2025-03-03 12:18:58 UTC; 33s ago
     Docs: man:nginx(8)
   Process: 4506 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master process on; (code=exited, status=0/SUCCESS)
   Process: 4511 ExecStart=/usr/sbin/nginx -g daemon on; master process on; (code=exited, status=0/SUCCESS)
   Main PID: 4512 (nginx)
    Tasks: 2 (limit: 1130)
   Memory: 1.7M (peak: 2.3M)
      CPU: 12ms
   CGroup: /system.slice/nginx.service
           └─4512 "nginx: master process /usr/sbin/nginx -g daemon on; master process on;"
             └─4513 "nginx: worker process"

Mar 03 12:18:58 ip-172-31-4-4 systemd[1]: Starting nginx.service - A high performance web server and a reverse proxy server...
Mar 03 12:18:58 ip-172-31-4-4 systemd[1]: Started nginx.service - A high performance web server and a reverse proxy server.
ubuntu@ip-172-31-4-4:~$
```

As the server is running

- Your frontend is now deployed on Nginx...
- open your browser and then put your server public IP and check, our application is deployed on nginx



connect to database with CLI and check, it is updated correctly .
by the steps

mongosh

use grocerydb

db.groceries.find()

Data is reflected in the database.

```
ubuntu@ip-172-31-4-4:~$ mongosh
Current Mongosh Log ID: 67c59f1d3878288cd451e943
Connecting to:  mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.4.0
Using MongoDB:  8.0.5
Using Mongosh:  2.4.0

For mongosh info see: https://www.mongodb.com/docs/mongosh-shell/

-----
  The server generated these startup warnings when booting
  2025-03-03T11:12:41.791+00:00: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
  2025-03-03T11:12:42.476+00:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
  2025-03-03T11:12:42.476+00:00: For customers running the current memory allocator, we suggest changing the contents of the following sysfsFile
  2025-03-03T11:12:42.476+00:00: For customers running the current memory allocator, we suggest changing the contents of the following sysfsFile
  2025-03-03T11:12:42.476+00:00: We suggest setting the contents of sysfsFile to 0.
  -----

test> use grocerydb
switched to db grocerydb
grocerydb> db.groceries.find()
[
  { _id: ObjectId('67c596722dbd3db334584832'), name: 'Apple', __v: 0 },
  { _id: ObjectId('67c5967a2dbd3db334584834'), name: 'Banana', __v: 0 },
  {
    _id: ObjectId('67c596842dbd3db334584838'),
    name: 'Blueberry',
  }
]
```