Node.js

Frontend

HTML/JS/CSS

React.js
Angular
Vue.js

Request →
← **Response**

Backend

Node.js/Express
Django
Sprintboot
.Net Core

Middleware

ORM/ DB Connector →

Connection URL

DB

MySQL/MongoDB

Node.js is a runtime environment used for building JS Applications

React   Express

Node.js

MERN

MySQL /
MongoDB

DB

Package / Library / Module

NPM

Node
package
manager

Built-in

No need to install, but should be imported to use

Third Party/
External

install and then import to sue

Custom Librarie

just import to use that

Creating a Node.js Server

http
HyperText Transfer Protocol

Connection starts when we are sending request, and terminates right after the response has been given to the frontend

```
const http = require('http');
const app = http.createServer((req, res) => {
        res.writeHead(200);
    res.end('Happy Weekend!');
            });

    app.listen(4001, () => {
console.log('Server is running...');
            });
```

```
const express = require('express');
```

Express

```
npm init
npm install express
node index.js/ npm start / npm run dev
```

```javascript
const express = require('express');
const app = express();
app.get('/hello', (req, res) => {
    res.send('Hello World!')
});

app.listen(8080, () => {
console.log('App is running...')
})
```

HTTP Methods
GET/POST/PUT/DELETE

GET
To retrieve data

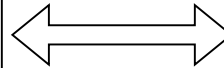POST
To insert data

PUT
To update data

DELETE
To delete data

Req/Res

```
http://localhost:4000
http://127.0.0.1:4000

45.65.12.45:5000 -> https://youtube.com
https://google.com
```

Req

header
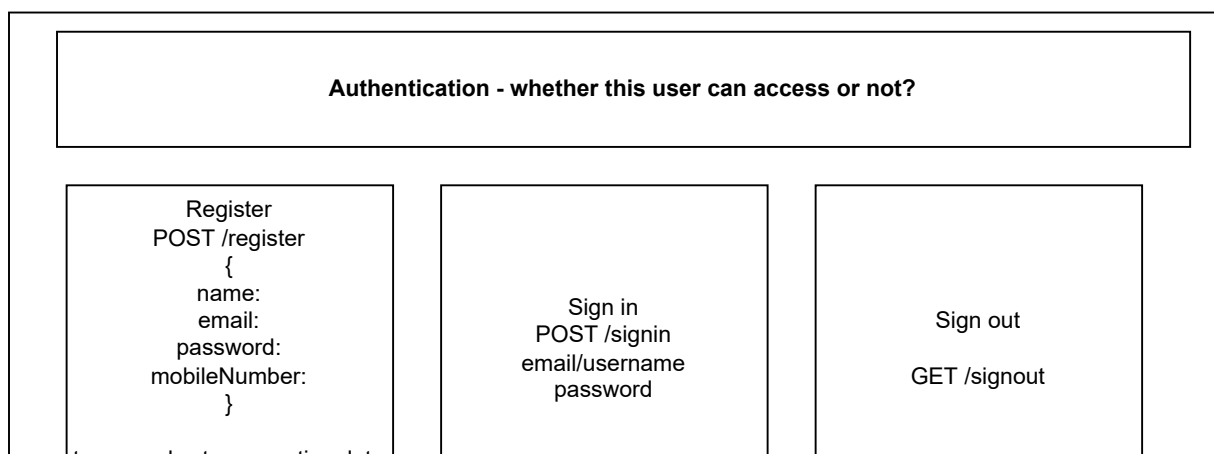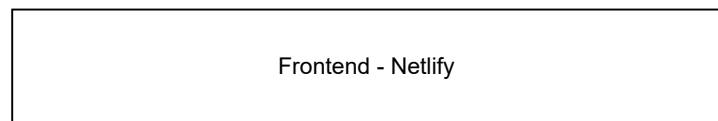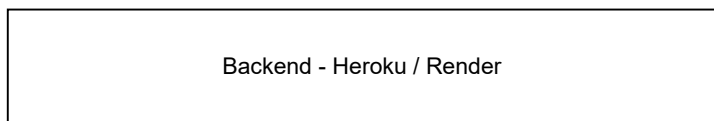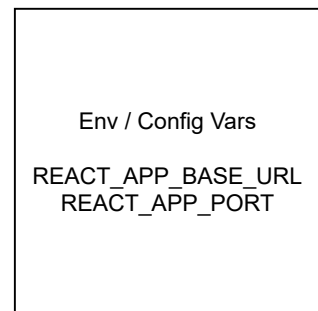parameters
body

⟺

Res

data
json
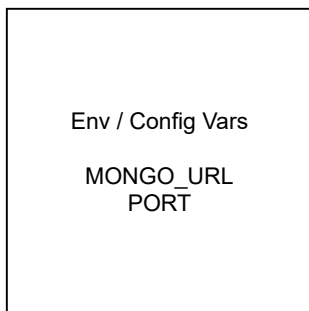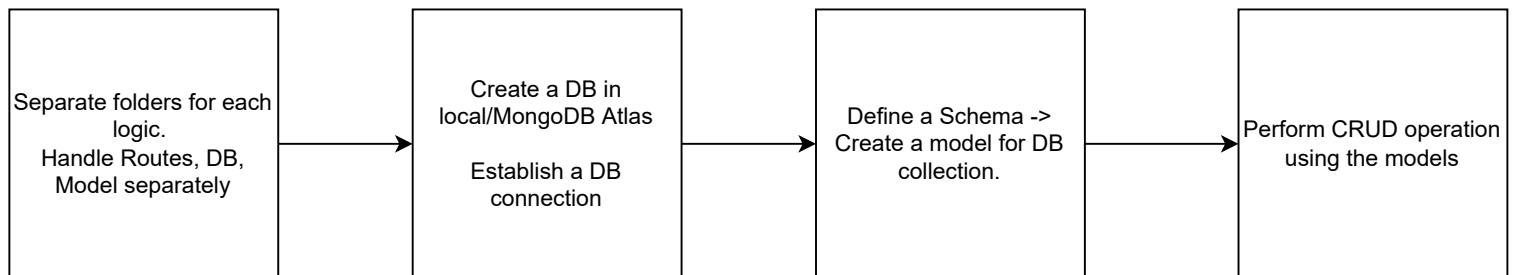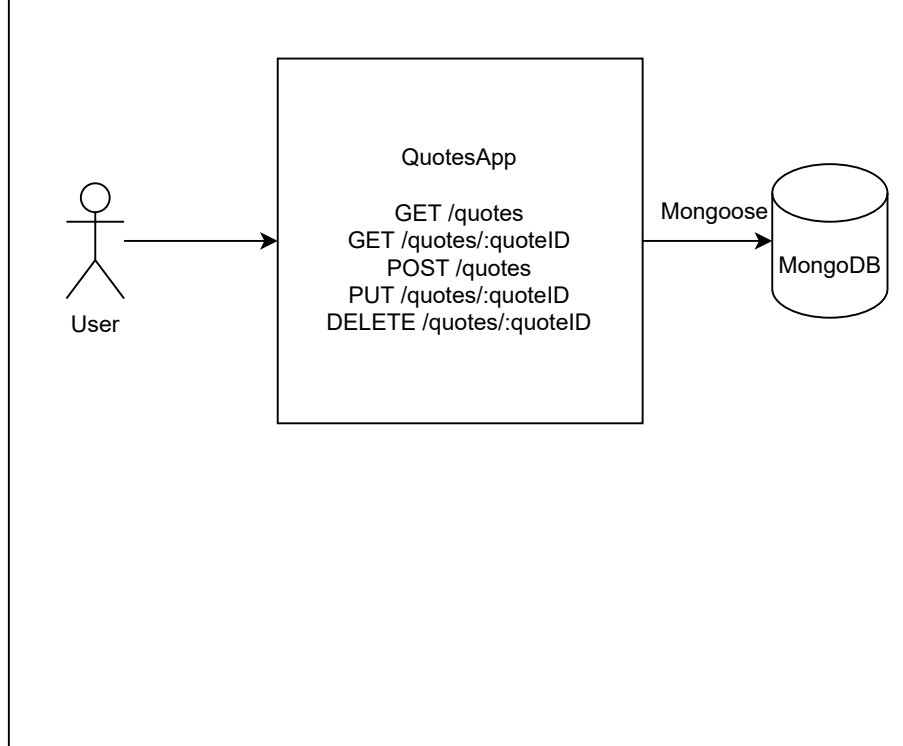
Parameters

PATH Parameters

www.example.com/users/1
www.example.com/users/100
www.example.com/products/1001

QUERY Parameters

www.example.com/users?name=farook&city=Chennai

QuotesApp

GET /quotes
GET /quotes/:quoteID
POST /quotes
PUT /quotes/:quoteID
DELETE /quotes/:quoteID

User

Mongoose

MongoDB

Separate folders for each logic.
Handle Routes, DB, Model separately

Create a DB in local/MongoDB Atlas

Establish a DB connection

Define a Schema ->
Create a model for DB collection.

Perform CRUD operation using the models

Env / Config Vars

MONGO_URL
PORT

Env / Config Vars

REACT_APP_BASE_URL
REACT_APP_PORT

Backend - Heroku / Render

Frontend - Netlify

Authentication - whether this user can access or not?

Register
POST /register
{
name:
email:
password:
mobileNumber:
}

Sign in
POST /signin
email/username
password

Sign out

GET /signout

to securely store senstive data
-> Hashing/Encyptions

Hashing
Orginal form -> Random Form

Unidirectional

npm i bcrypt

Encryption/Decryption
Orginal form -> Random Form

Bidirectional

npm i jsonwebtoken

Authorization -> who can access what?