PHARMADOC

A PHARMACEUTICAL PRODUCT MANAGEMENT REST API

BY AHMED DAMMAK

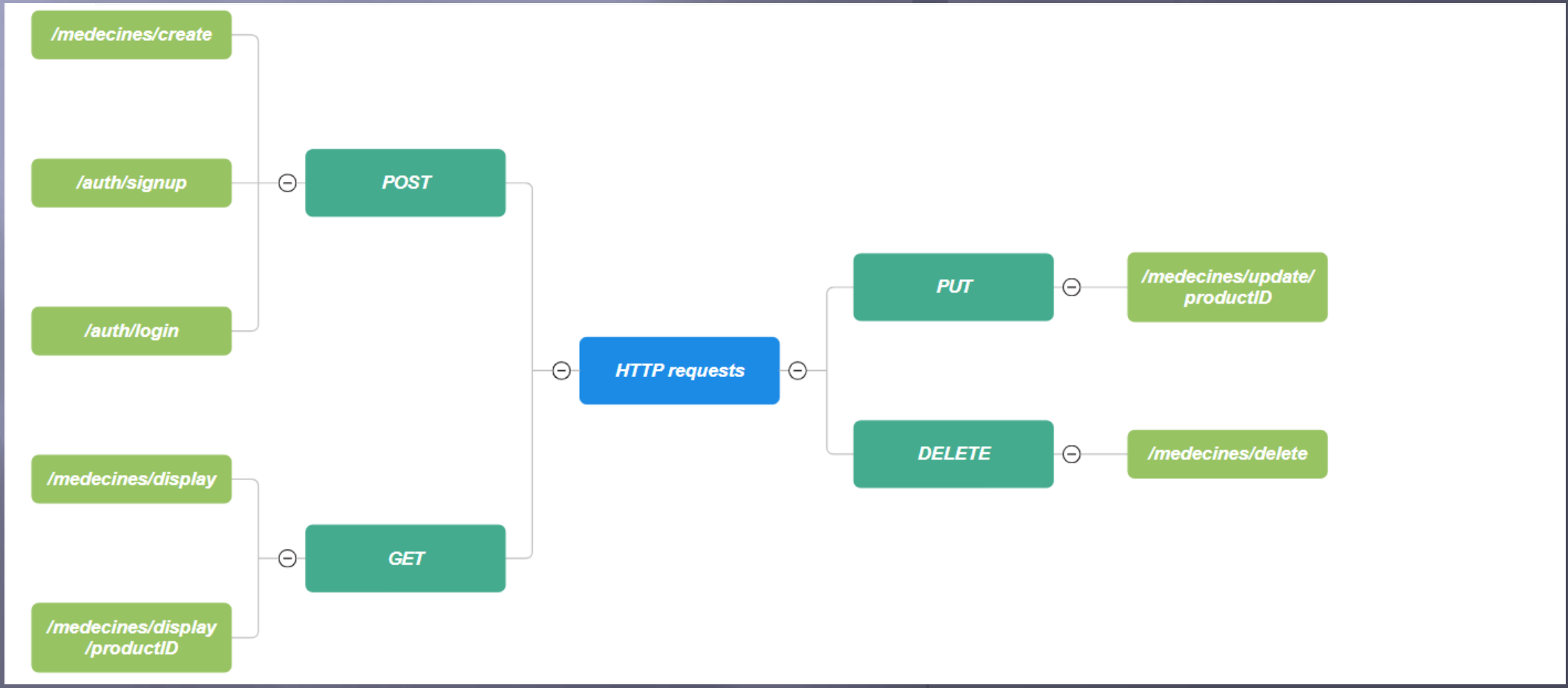
JANUARY 2024

Overview

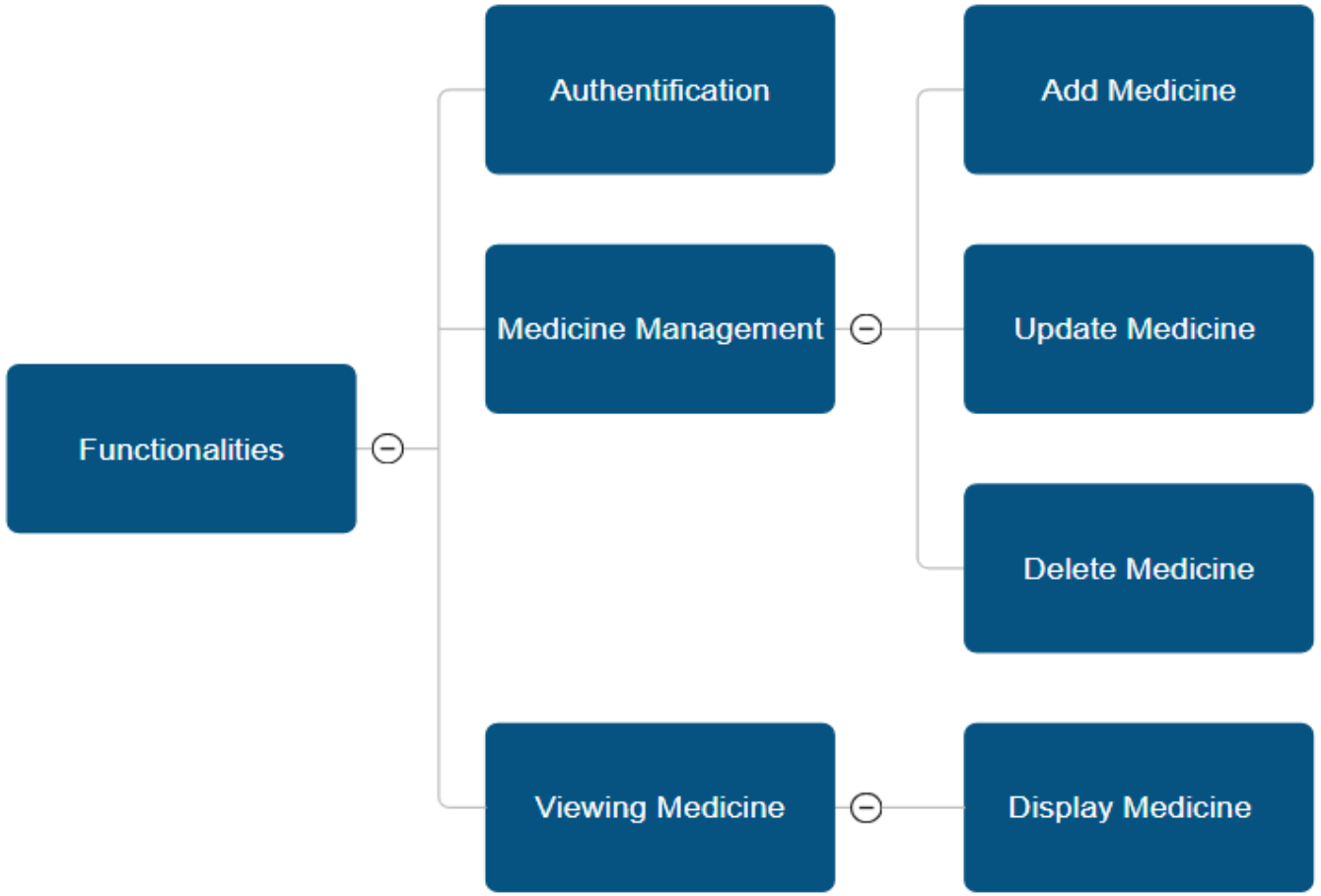
- What is PharmaDoc ?
- Main Functionalities
- Technologies used
- Possible enhancements
- Appendix
- Conclusion

What is PharmaDoc ?

PharmaDoc is a product management REST API, it enables its users to send the following requests:



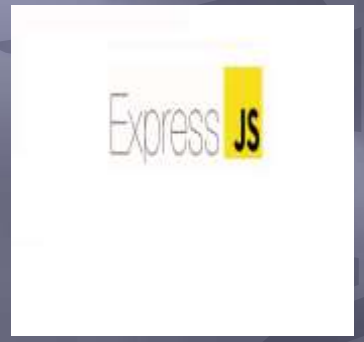
Main Functionalities



Technologies used

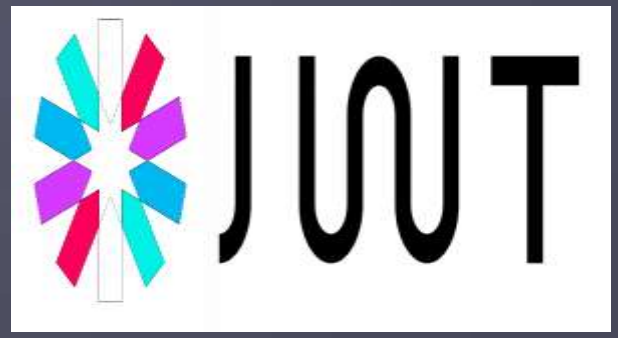
API testing

Backend framework

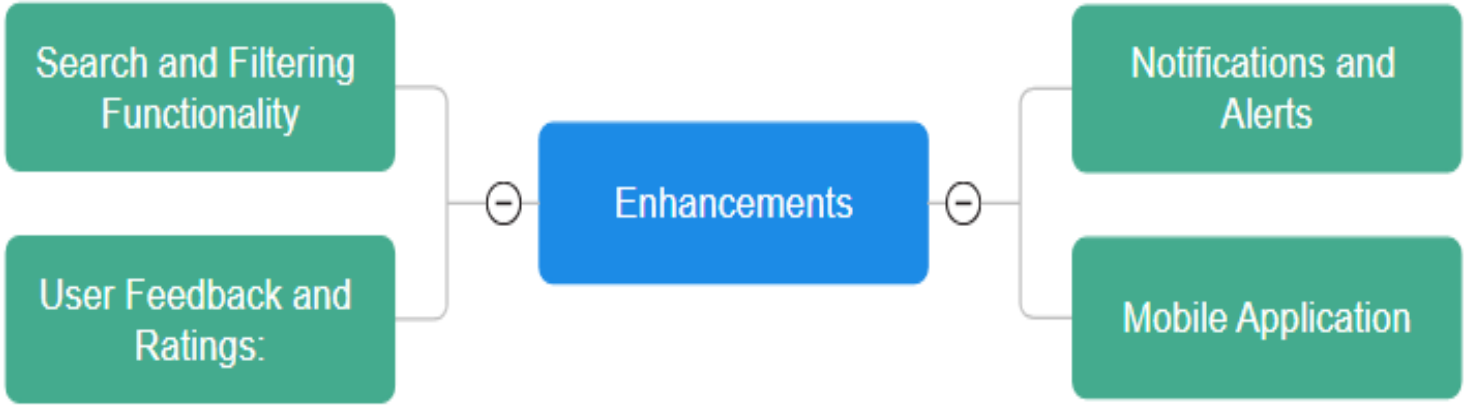


Database hosting

Authentication



Possible Enhancements



APPENDIX
APP.JS

app.js > ...

```
const express = require('express'); //framework
const app = express(); //app used in the express framework
const cors = require('cors'); // we use corse so that the frontend can consume API's
const bodyparser = require('body-parser') // modify the data to json type
const authRoute = require('./routes/auth.route') // service of authentfication located in the routes directory
const medRoute = require('./routes/meds.route') // service of books located in the routes directory
require('dotenv').config(); // where we store our sensitive data
const mongoose = require('mongoose'); // ODM to connect to our MongoDB database
const checkAuth = require('./middleware/checkAuth'); // Where we used JWT (json web token) to verify authentification

app.use(cors({
  origin: '*'
}));
app.use((req, res, next) => {
  res.header('Access-Control-Allow-Origin', '*');
  res.header('Access-Control-Allow-Methods', 'GET, POST, PUT, DELETE, OPTIONS');
  next();
});

app.use(bodyparser.urlencoded({ extended: false }))
app.use(bodyparser.json())

mongoose.connect( process.env.BD_LINK_TO_CONNECT)
  .then(() => console.log('Connected to MongoDB'))
  .catch((err) => console.error('Failed to connect to MongoDB', err));


app.use('/auth', authRoute)
app.use('/medecines', checkAuth, medRoute)

module.exports = app
```

USER.JS

models > JS user.js > ...

```
1  const mongoose = require('mongoose');
2  const userSchema = new mongoose.Schema({
3    name: { type: String, required: true },
4    email: { type: String, required: true, unique: true },
5    password: { type: String, required: true },
6    createdAt: { type: Date, default: Date.now },
7    isConncted:{type:Boolean,default:0},
8  })
9  const User=mongoose.model('User',userSchema);
10 module.exports=User;
```

MEDS.JS

```
1  const mongoose = require('mongoose');
2  const MedecinesSchema = new mongoose.Schema({
3    name: { type: String, required: true },
4    description: { type: String, required: true },
5    createdAt: { type: Date, default: Date.now },
6    category : {type:String,required:true},
7    price: {  type: Number, required: true }
8  })
9  const Medecines = mongoose.model('Medecines', MedecinesSchema);
10 module.exports = Medecines;
```

CONCLUSION

- PharmaDoc is meticulously designed to provide a seamless and secure experience for managing pharmaceutical products.
- The RESTful API architecture ensures scalability, allowing the system to handle varying loads efficiently.

THANK YOU
FOR YOUR ATTENTION