

Medical Counselling App

Paarmita Bhargava - 16UCC065

under supervision of

Dr. Amit Neogi

Dr. Subrat Dash

Introduction

The project deals with collection of data from large number of patients , through a website, make suitable analysis and based on the analysis suggest another patient of same type to choose from few good options out of many options available. The motive of the project was to help people with the experience of each other. The main points which needed to be considered are:

- Integrate solutions from all therapies
- Collection of therapies from all pathy
- Let the users know the best Doctors/clinics for the specific disease.

Importance of the project

We found that there was no platform in India where we can search specific disease which is mainly found in India and can get feedback from people for the cure of that particular disease.

So by this build platform we display all types of diseases that are common and mainly focusing on the Pathy(Homeopathy, Allopath, Ayurveda) we separate the cure for them and suggest good doctors/ clinics, best pathy and other summary details for it.

Features included in the application

Main features include:

- Ask Suggestions and Share Experience (A collaborative blog)
- Comments to the shared blog posts
- View details of Diseases and their cure in different Pathy
- Admin Dashboard to control all profiles and blog post

Technologies Used

NodeJs - Server Side API

React - JavaScript library for building user interfaces

MongoDB - Document and NoSQL database

Bootstrap - HTML, CSS, and JS library

Passport - Authentication middleware for Node.js



Tools Used

Postman - API Development Environment

Disqus - Comment plug-in

JSON Web Token (JWT) - Securely transmitting information between parties as a JSON object

Font Awesome - Icon set and toolkit



System Requirements

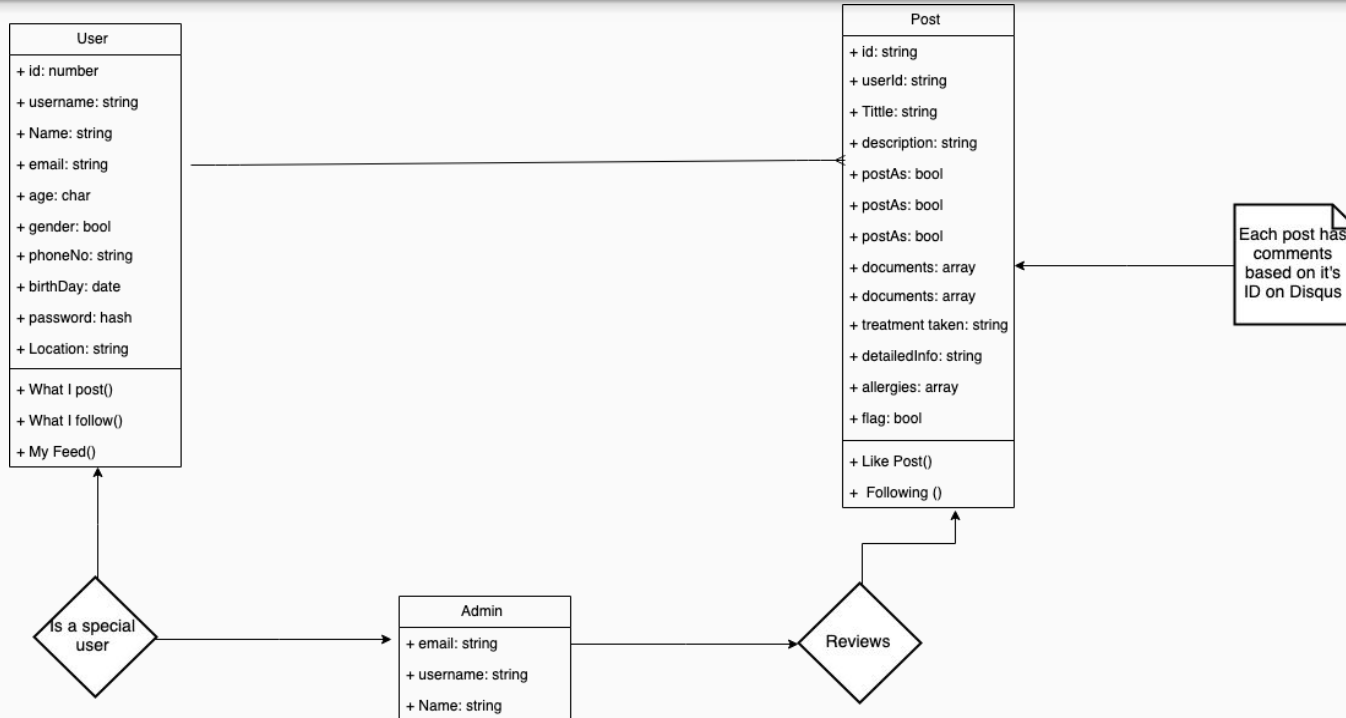
Final application will be compatible with the following specifications:

Browser - Chrome, Mozilla, Safari, etc.

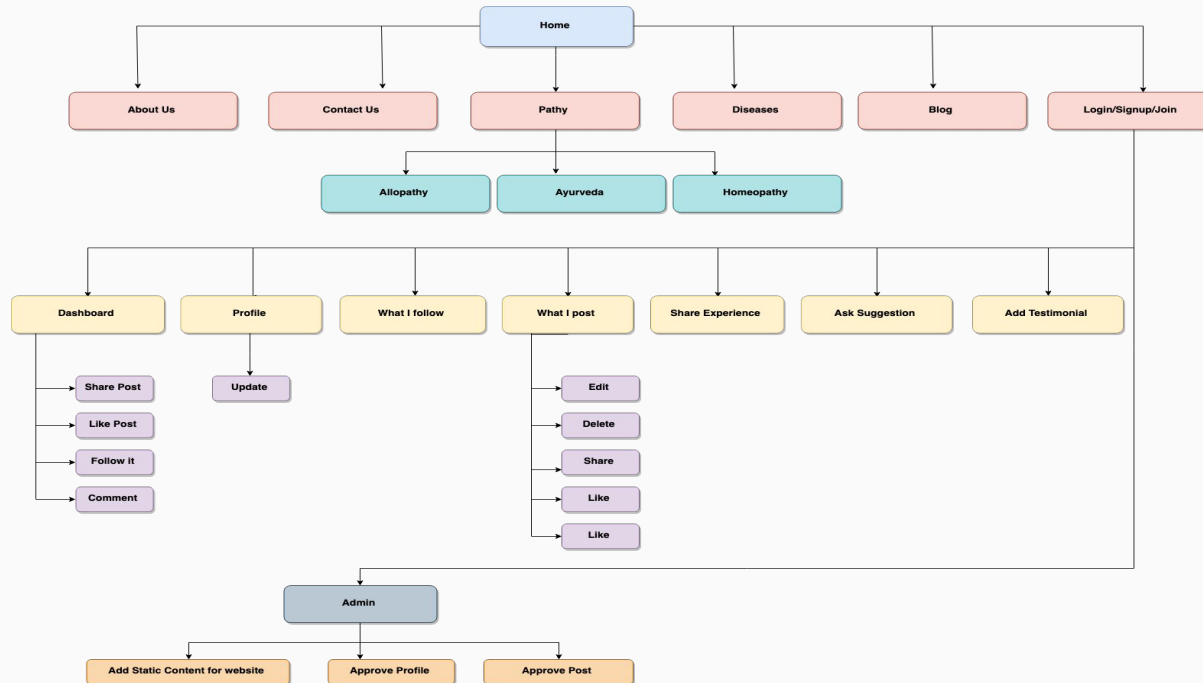
OS - Linux, Windows, MacOS, etc

Screens - Desktop, Tablets, Mobile

Use Case Diagrams



Site Map



API Authentication

GET localhost:4000/api/user/test

localhost:4000/api/user/test

GET localhost:4000/api/user/test

Send Save

Params Authorization Headers Body Pre-request Script Tests

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (10) Test Results

Status: 200 OK Time: 13 ms Size: 435 B Download

Pretty Raw Preview JSON

```
1 {  
2   "msg": "Users work"  
3 }
```

GET localhost:4000/api/user/test

localhost:4000/api/

GET localhost:4000/api/

Send Save

Params Authorization Headers Body Pre-request Script Tests

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (10) Test Results

Status: 200 OK Time: 13 ms Size: 461 B Download

Pretty Raw Preview JSON

```
1 {  
2   "Message": "Welcome To Medical Counselling:"  
3 }
```

GET localhost:4000/api/user/test

POST localhost:4000/api/

localhost:4000/api/user/register

POST localhost:4000/api/user/register

Send Save

Params Authorization Headers (1) Body Pre-request Script Tests

none form-data x-www-form-urlencoded raw binary

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> username	test	
<input checked="" type="checkbox"/> email	test@test.com	
<input checked="" type="checkbox"/> password	test	
Key	Value	Description

Body Cookies Headers (10) Test Results

Status: 200 OK Time: 902 ms Size: 462 B Download

Pretty Raw Preview JSON

```
1 {  
2   "success": false,  
3   "error": "User Already Exists"  
4 }
```

GET localhost:4000/api/user/test

POST localhost:4000/api/

localhost:4000/api/user/register

POST localhost:4000/api/user/register

Send Save

Params Authorization Headers (1) Body Pre-request Script Tests

none form-data x-www-form-urlencoded raw binary

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> username	test	
<input checked="" type="checkbox"/> email	test@test.com	
<input checked="" type="checkbox"/> password	test	
Key	Value	Description

Body Cookies Headers (10) Test Results

Status: 200 OK Time: 820 ms Size: 431 B Download

Pretty Raw Preview JSON

```
1 {  
2   "success": true  
3 }
```

Document: 5cd3c83c135daf61c8480bcf

Edit document ([view keyboard shortcuts](#))

```
1 {
2   "id": {
3     "oid": "5cd3c83c135daf61c8480bcf"
4   },
5   "admin": false,
6   "username": "test",
7   "email": "test@test.com",
8   "password": "$2b$10$K1Jk4LykWXObKvkWz2NGruA4/yVG6kKjNbXf6FurhLMUV7fH03.",
9   "v": 0
10 }
```

GET localhost:4000/api/user/test

GET localhost:4000/api/

+

...

No Environment

localhost:4000/api/celeb_jokes/celebrity

GET

localhost:4000/api/celeb_jokes/celebrity

Send

Save

Params

Authorization

Headers (1)

Body

Pre-request Script

Tests

Cookies

Code

Comments

none

form-data

x-www-form-urlencoded

raw

binary

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input type="checkbox"/>	username	test			
<input type="checkbox"/>	email	test@test.com			
<input type="checkbox"/>	password	test			
<input checked="" type="checkbox"/>	token	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJl...			
	Key	Value	Description		

Body

Cookies

Headers (10)

Test Results

Status: 200 OK

Time: 15 ms

Size: 1.04 KB

Download

Pretty

Raw

Preview

JSON

```

1  [
2  {
3    "id": 88881,
4    "joke": "As President Roosevelt said: \"We have nothing to fear but fear itself. And Chuck Norris
5    },
6  {
7    "id": 88882,
8    "joke": "Chuck Norris only lets Charlie Sheen think he is winning. Chuck won a long time ago."
9  },
10 {
11   "id": 88883,
```

GET localhost:4000/api/user/test

POST localhost:4000/api/

+ ...

No Environment

POST

localhost:4000/api/user/authenticate

Send

Save

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input type="checkbox"/> username	test			
<input checked="" type="checkbox"/> email	test@test.com			
<input checked="" type="checkbox"/> password	test			
Key	Value	Description		

Body

Cookies

Headers (10)

Test Results

Status: 200 OK

Time: 763 ms

Size: 888 B

Download

Pretty

Raw

Preview

JSON

≡

```

1 {
2   "user": {
3     "admin": false,
4     "_id": "5cd3c83c135daf61c8480bcf",
5     "username": "test",
6     "email": "test@test.com",
7     "password": "$2b$10$K1Jk4LykWX0bKvkWz2NGruA4/yVG06kKjNbXf16FurhLMUV7FH03.",
8     "__v": 0
9   },
10  "authenticate": true,
11  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1bWVhbGciOiR3AdGVzdC5jb20iLCJwYXNzd29yZCI6IyQyYiQxMCRLSUpnEx5a1dT2JldmtXejJROR3J1QTQveVZhbzZrS2p0YlVmbDZlGdXJoTE1VjdmS0ZAZLIiIsImh0dC1uZm4zM4U2NSwiZXhwIjoxNTU3NDY5OTY1fQ.w6aGNO-IhdhVUeSMwaDZJe3SiLSF0eFeUQL5iHsrk"
12 }

```

Conclusion

Work till so far in this semester :

1. Worked on the basic structure of the application and workflow.
2. Client Side of the application using static data.
3. Authentication of user in database using JSON web tokens and Passport
4. Setup Public, Private and Protected Routes in Client

Next Semester Work

I will be working on the following points for the next semester :

1. Backend - API endpoints
 - i. Blog posts - Add, create, edit, delete
 - ii. Disqus - Integrating the comments API
 - iii. Profile - Edit, create, view
 - iv. Admin - confirm, edit static data
2. Frontend - Connecting the API endpoints with the Client, replacing the static data
3. Create documentation

Reference Links

- <https://www.techhighness.com/post/how-to-draw-no-sql-data-model-diagram/>
- <https://getbootstrap.com/docs/4.3/getting-started/introduction/>
- <https://jasonwatmore.com/post/2018/11/28/nodejs-role-based-authorization-tutorial-with-example-api>
- <https://scotch.io/@devGson/api-authentication-with-json-web-tokensjwt-and-passport>

Github repo -

<https://github.com/Paarmita/MedicalProject>

Thank You!