

Medical Counselling Platform (Web application)

Project report submitted in partial fulfillment
of the requirements for the degree of

Bachelor of Technology
in
Communication and Computer Engineering

by

Paarmita Bhargava - 16UCC065

Under Guidance of
Dr. Amit Neogi
Dr. Subrat Dash
Nirmal Sivaraman



Department of Communication and Computer Engineering
The LNM Institute of Information Technology, Jaipur

December 2019

Copyright © The LNMIIT 2019
All Rights Reserved

The LNM Institute of Information Technology
Jaipur, India

CERTIFICATE

This is to certify that the project entitled "**Medical Counselling Platform (Web application)**" , submitted by **Paarmita Bhargava (16ucc065)** in partial fulfillment of the requirement of degree in Bachelor of Technology (B. Tech), is a bonafide record of work carried out by them at the Department of Communication and Computer Engineering, The LNM Institute of Information Technology, Jaipur, (Rajasthan) India, during the academic session 2018-2019 under my supervision and guidance and the same has not been submitted elsewhere for award of any other degree. In my/our opinion, this report is of standard required for the award of the degree of Bachelor of Technology (B. Tech).

Date

Adviser: Dr. Amit Neogi

Acknowledgments

This project would not have been conceived without the kind support and help of many individuals. I would like to express our deepest gratitude to my supervisors Dr. Amit Neogi, Dr. Subrat Dash and Mr. Nirmal Sivaraman whose expertise, inspiring ideas, understanding, and patience added considerably to my ongoing B.Tech project experience. This project has helped me in enriching my experience and has allowed me to learn new techniques and apply them.

Abstract

The project deals with the collection of data from a large number of patients, through a social networking web application and makes suitable analysis and based on the analysis suggests few good options out of many options available. The motive of the project was to help people with the experience of each other.

Contents

Chapter	Page
1 Introduction	1
1.1 The Area of Work	1
1.2 Problem Addressed	1
1.3 Existing systems	2
1.3.1 Patients like me	2
1.3.2 Inspire.com	2
1.3.3 Problems with Existing systems	3
1.3.4 Features taken from these platforms	3
2 Proposed Work	4
2.1 Introduction	4
2.1.1 Project Goals	4
2.1.2 Software Scope	4
2.1.3 Limitations	4
2.2 Major Features	5
2.3 Technology Stack and Specs	5
2.3.1 Frontend	5
2.3.2 Backend	5
2.3.3 Database	5
2.3.4 NPM Packages	5
2.3.5 Version tools	5
2.3.6 Softwares	6
2.4 Problem Addressed	6
2.4.1 Use Cases	6
2.4.2 Component Diagram	7
2.4.3 Class diagram	7
2.4.4 Interface design	8
2.4.4.1 Wireframes	9
2.4.4.2 Site Map	10
3 Demonstrations	11
3.1 Frontend Components	11
3.2 Backend Components	16
3.3 Database Components	19

CONTENTS	vii
4 Conclusions and Future Work	23
4.1 Conclusion	23
4.2 Problem faced	23
4.3 Key Learning's	23
4.4 Scope of further work	24
Bibliography	25

Chapter 1

Introduction

The most common problem people suffer nowadays is to conclude while encountering a disease.

- Which doctor to consult?
- Which treatment should they pick up?
- Are they taking the right treatment?
- How many people are suffering from the same problem?
- How did they overcome this?
- Does the symptoms exactly match?
- How can they contact each other and seek each other's guidance.

To overcome all such problems faced by people, the idea of a Medical Counselling Application came into existence. It is a web-based system, aimed to provide trusted information regarding the various therapies used for the treatment of various diseases.

1.1 The Area of Work

Software Development - I worked on the platform to a large number of patients in India, who face problems to get suggestions on various types of pathy (Allopathy, Homeopathy, and Ayurveda) existing in various parts of the country.

1.2 Problem Addressed

To display information about diseases and various therapies (Homeopathy, Allopathic, Ayurveda, etc) to treat them. To collect and display Testimonials from users regarding their experience of any disease, treatment, therapy or medical product. Also, the viewers can comment and provide ratings to

the testimonial. These testimonials are categorized based on the therapy. These testimonials are taken as input via forms, testimonials can be verified or unverified depending upon the writer's choices and the contact information provided by the writer. To provide a catalog of doctors, their area of specialization, location of their practice and reviews regarding their work. To deliver a newsletter to all the registered users regarding the latest research and articles.

1.3 Existing systems

In the current technology world, whenever anybody has questions they search for answers about their questions on the internet. Some platforms are reliable for finding the solutions and ask suggestions. So, I researched some of these websites which provide medical information. Some of the main features:

1.3.1 Patients like me

- Member Stories
- News
- Patients(Age, sex, condition, treatment, Interests, Stars(rating))
- Conditions(overview, symptoms(severeness, therapies used by other patients to cure the symptom)), Conditions(Type, number of people opt for the particular treatment)
- Comparisons of several treatments(name, effectiveness, side effects, number of evaluations)
- Number of people having the disease, their age and the age at which they had their first symptom)
- Symptoms(total number of patients, number of patients at a different level of the severeness, treatments proposed)
- Treatments(name of the treatment and number of people opting for the treatment)

1.3.2 Inspire.com

- Community platform with Patients and Caregivers
- Discussion feed : Each member's homepage feed quickly displays the latest content being discussed in his/her community.
- Expert moderation: Inspire's moderators 24/7 vigilance maintains our supportive, caring, secure community.

1.3.3 Problems with Existing systems

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. These platforms only provide information internationally and do not focus on the methods that are prevalent in India like homeopathy, some medical centers etc.

1.3.4 Features taken from these platforms

Inspire.com

- Provide a list of disease when a user wants to add his/her experience and can check from them or add a new if not present in that list.
- Provide functionalities to users like posts I replied, posts I started, posts I follow, so that they could look other's comments in the post they are engaged.
- In this, there are groups, communities, friends with each disease, who could interact with messages which are not required in our website.
- Add a feature to share the blog of a user on social websites by a reader if that is useful.

Patientslikeme.com

- We can add filter search for therapy, disease, age, sex.
- They have done a detailed analysis of each disease and its therapy with different aged people.
- They have a section for symptoms as well which can also be done in by asking users to enter the symptoms and make them as tags.
- As a patient they ask them how they are feeling whenever they visit the site, and history of their previous thoughts. Patients provide symptoms, treatment, conditions, and new updates in different sections.

Chapter 2

Proposed Work

2.1 Introduction

2.1.1 Project Goals

The main goal of the system is to build a non-profit organization which aims at providing a better and effective way to share real world health experiences through user interaction and the integration of various therapies used for curing diseases. The website will also provide some blogs - guide to healthy living where articles on healthy living are published, campaign page which is static but emphasize the habits in the society which affects mental and physical health of an individual.

2.1.2 Software Scope

The scope of the ‘Medical counselling Application’ platform is:

- To provide information of the diseases with good doctors, best therapy and other analysis.
- To enable the admin to verify the posts, profiles and comments.
- To enable the users to create and edit their profiles.
- To enable the users to view posts of other users and their profiles.
- To enable the user to successfully post their experience.

2.1.3 Limitations

- The user can only add an image of up to 1 MB as the loading time for going too high for higher images. So the cloud can be used in the future to overcome this problem.
- As it is a web application, so it is highly recommended to use in Desktop view only.
- The highly sleek and intuitive interface was made to improve Human-Computer Interaction (HCI). However, this comes with challenges because lower versions of Internet Explorer (i.e. IE9 and

lower) do not support certain features such as column-fill, column-span, align-self, blackface-visibility, etc.

2.2 Major Features

- A user-friendly web-based application that tells what our website will be doing if you log in with the contact details.
- Feature to share an experience with a title and a full description of the disease and upload an image of the prescription.
- Allowing users to read the posts and if logged in, can comment as well.
- Users can search for a specific disease and read the best solution, and find advantages, disadvantages for a particular solution to the disease.
- Admin dashboard which can do CRUD operations, on comments, users and posts.
- Additional features- Users can follow other users and their posts will be shown on the top of the user dashboard like Facebook and other social networks.
- Users can also see all their posts which they have posted if they want to update/delete in the future.

2.3 Technology Stack and Specs

2.3.1 Frontend

HTML5 and CSS3, ReactJS, Bootstrap

2.3.2 Backend

ExpressJS, NodeJS, PassportJS

2.3.3 Database

MongoDB

2.3.4 NPM Packages

Body-parser, CORS, Jsonwebtoken, Lodash, Mongoose, Morgan, Session storage, Nodemon

2.3.5 Version tools

Github, Heroku

2.3.6 Softwares

Disqus, Font-Awesome, Mlab, Postman- Testing tool

2.4 Problem Addressed

2.4.1 Use Cases

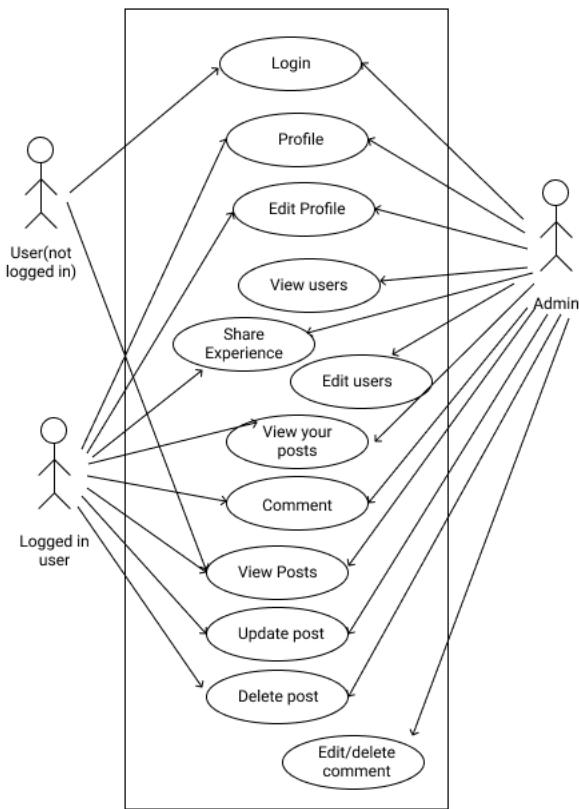


Figure 2.1 Use case Diagram

2.4.2 Component Diagram

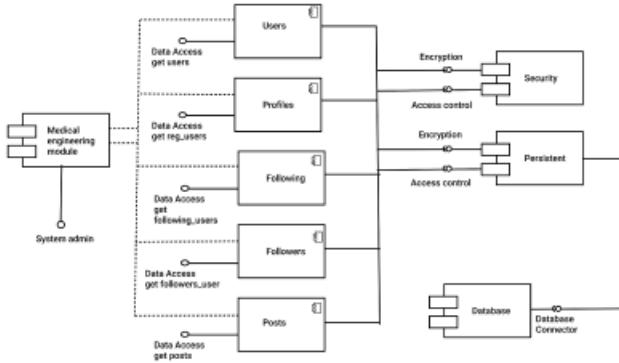


Figure 2.2 Component Diagram

2.4.3 Class diagram

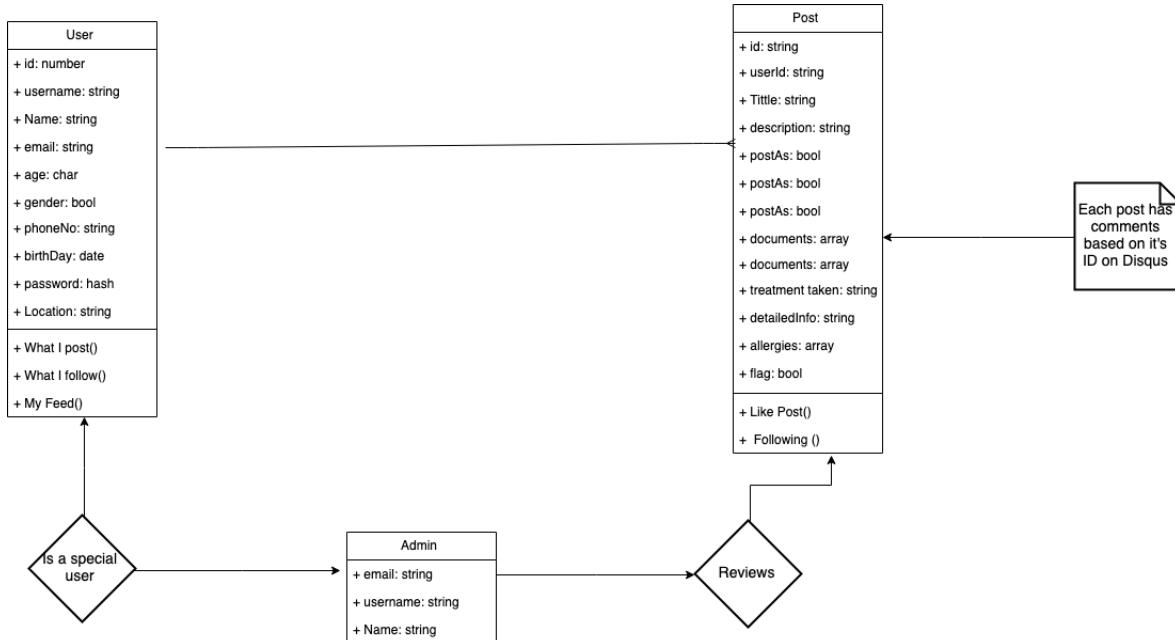


Figure 2.3 Class Diagram

2.4.4 Interface design

The web application was created with the following design considerations in mind:

- **Consistent-** The website should have a similar look and feel on every page.
- **Efficient and easy to maintain-** This refers to the fact that there is a need to separate content from the layout so that you can easily change your page design without editing every page on the site.
- **Layout-** The layout of each page should have a good contrast between the text and the background area. This helps considerably with visibility as it will be difficult to read the text if it is almost the same color as the background. Monitor size should also be taken into consideration.
- **Easy to navigate and use-** Users should not have a hard time trying to navigate the site. Navigation links should be consistent and clearly labeled. All navigation links should also be working properly and should point to the intended page/site.
- **Browser is compatible-** When designing the site consider different browser environments. Extensive testing should be done on each page in all the major browsers and the design changed appropriately to cater to all.
- **Visually appealing-** The use of color, text, fonts, and graphics should be carefully considered and used to ensure that the site is visually appealing to its visitors.
- **Speed-** The performance of a website is mostly rated by its up time and downtime. These terms refer to the amount of time it takes the site to respond to requests. Graphics should be kept to a minimum to allow the site to load faster. The pages on the site should load within an acceptable time e.g. under 10seconds.

2.4.4.1 Wireframes

Earlier work

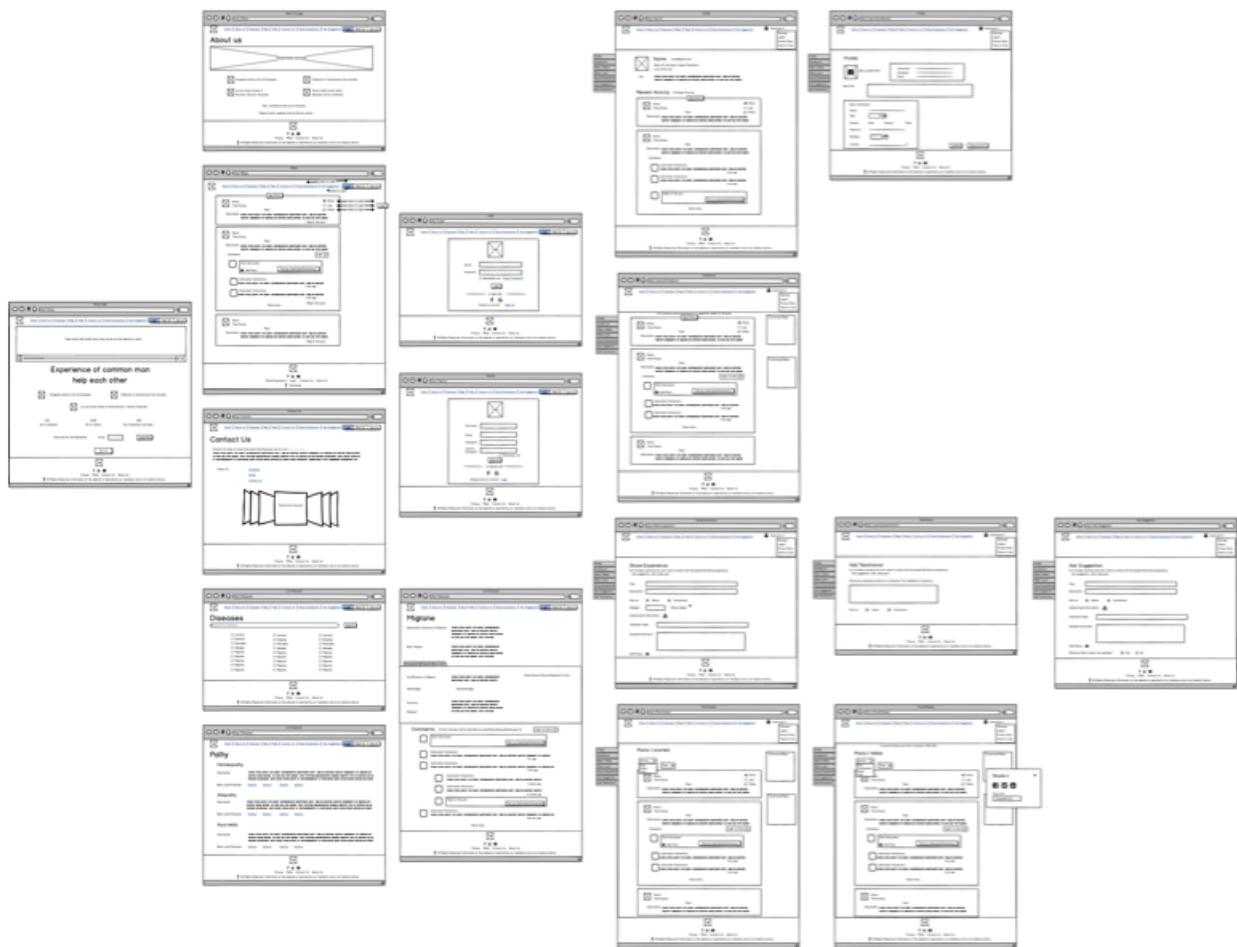


Figure 2.4 Wireframes idea for the app

2.4.4.2 Site Map

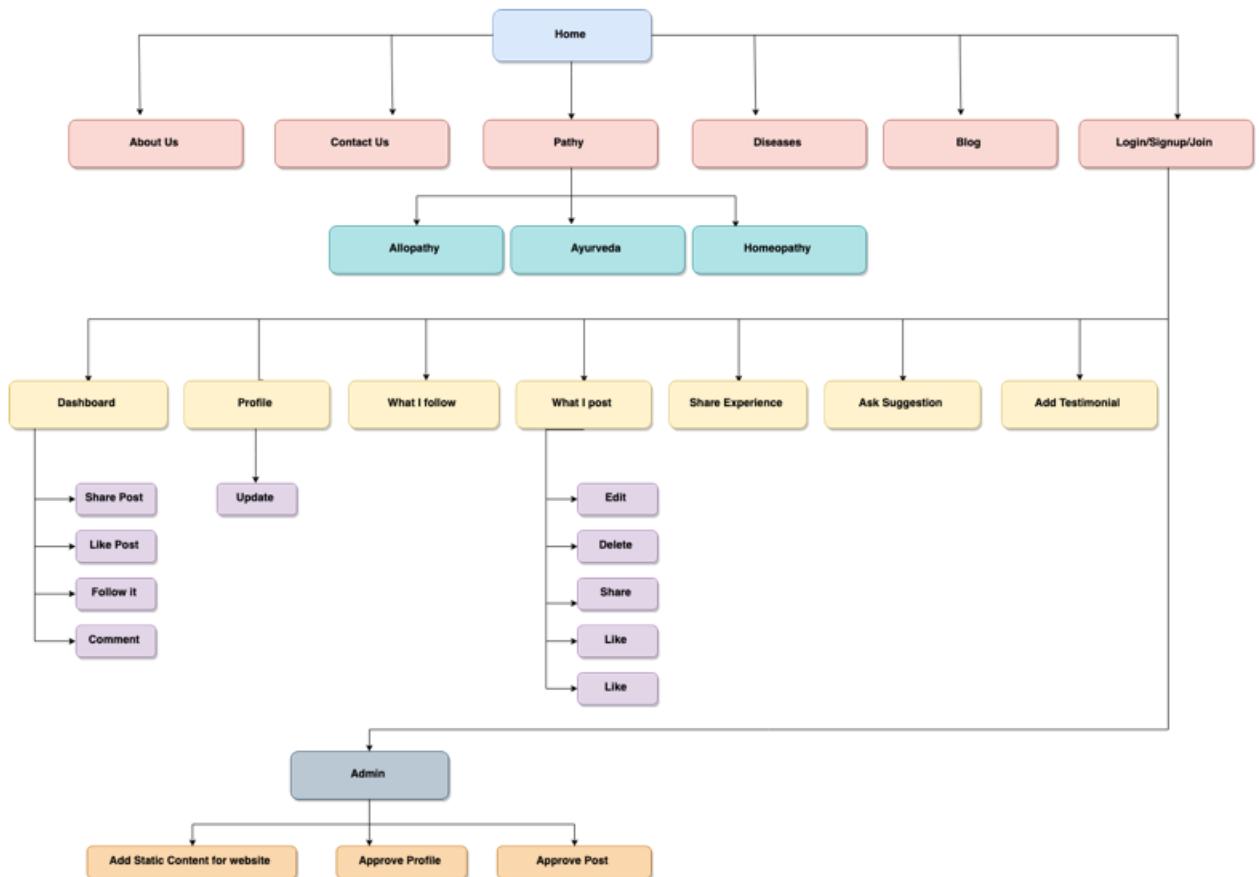


Figure 2.5 Site Map

Chapter 3

Demonstrations

3.1 Frontend Components

Can view the working platform here- <http://medical-platform.surge.sh/>

Dashboard

The screenshot shows the dashboard interface. At the top, there is a dark header bar with a logo, navigation links (Home, About Us, Diseases, Pathy, Posts, Contact us), and user information (Admin, master). On the left, a sidebar menu includes Dashboard, Profile, Users, My Posts, Share Experience, Settings, Logout, FAQ, Privacy Policy, and Terms of use. The main content area is titled 'Posts' and displays a single post by 'paarmita'. The post is about 'Diabetes' and describes homeopathy from a doctor nearby house. It mentions taking medicines from doctors and exercising once a day. A 'Read more' button is visible at the bottom of the post. A 'Next' button is located at the bottom of the sidebar.

Figure 3.1 Dashboard for all type of users(logged in or not)

Authentication and Authorization with Password and Username

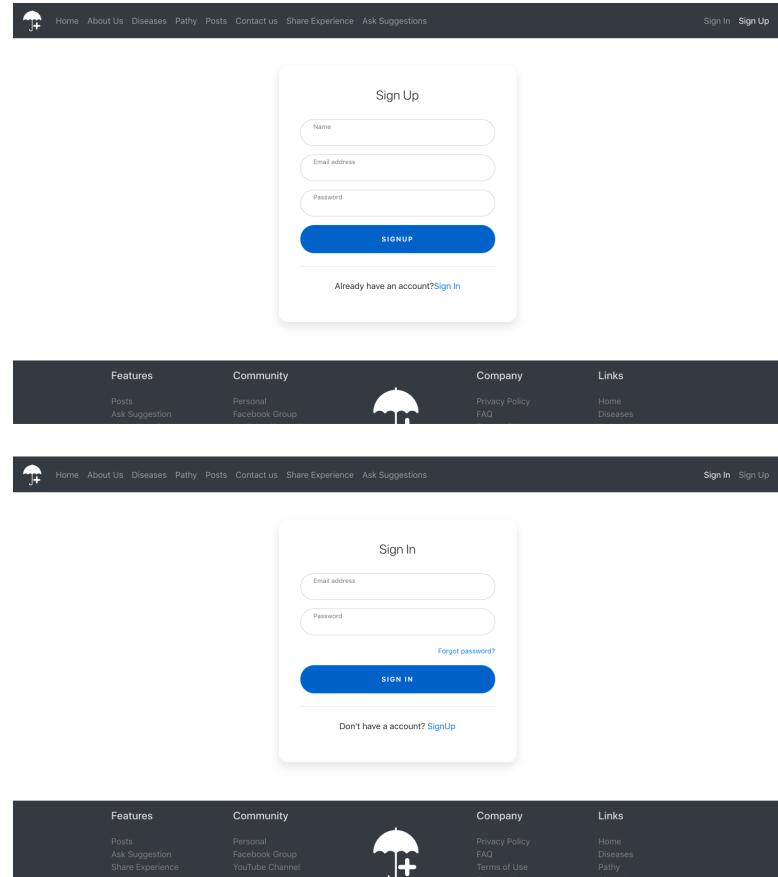


Figure 3.2 Signup and login screens

Diseases

The screenshot shows a search interface for diseases. The search bar at the top contains the letter 'a'. Below it, a section titled 'Arthritis' lists categories: 'Eye, Ear, Nose and Throat', 'Digestive and Intestinal', and 'Cancer'. Under 'Arthritis', there are three boxes: 'Arthritis' (with sub-options like Childhood Arthritis, Fibromyalgia, Gout, Osteoarthritis (OA), Rheumatoid Arthritis (RA)), 'Eye, Ear, Nose and Throat' (with sub-options like Hearing Loss, Glaucoma, Macular Degeneration), and 'Digestive and Intestinal' (with sub-options like Crohn's Disease, Ulcerative Colitis, IBS).

Figure 3.3 Diseases with search

Profile

The screenshot shows a profile page for a user named '@paarmita'. The profile picture is a cartoon illustration of a person with long brown hair and glasses. The bio reads 'Hi, I'm a developer and designer'. The sidebar on the left includes links for Dashboard, Profile, Users, My Posts, Share Experience, Settings, Logout, FAQ, Privacy Policy, and Terms of use. The main profile area shows 0 followers and 0 following. It also displays the user's email (paarmita1998@gmail.com) and the date they joined (Fri Jul 19 2019). There are tabs for Followers and Following.

Figure 3.4 Profile UI

My posts

View all the posts that you have posted, so you can delete, edit the post later and view other info.

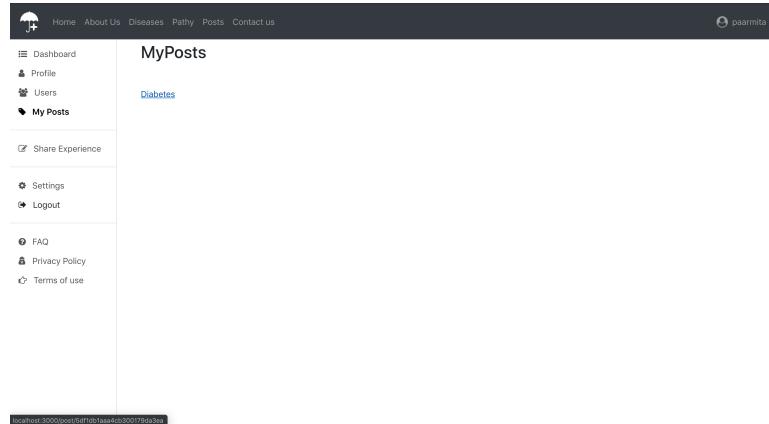


Figure 3.5 Posts posts by a user

Post detail

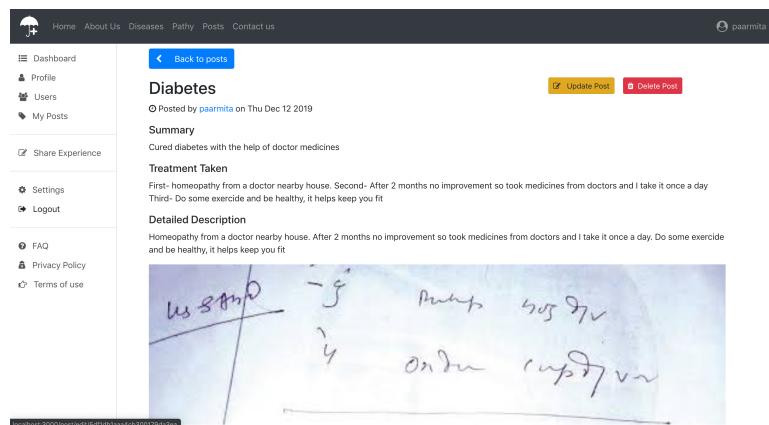


Figure 3.6 Posts posts by a user

List of all the users

The screenshot shows a web application interface for managing users. At the top, there is a navigation bar with links: Home, About Us, Diseases, Pathy, Posts, Contact us, and a user profile icon for 'paarmita'. On the left, a sidebar menu includes: Dashboard, Profile, Users (selected), My Posts, Share Experience, Settings, Logout, FAQ, Privacy Policy, and Terms of use. The main content area is titled 'Users' and displays four user profiles in cards:

- paarmita** (Profile picture) - Email: paarmita1998@gmail.com - [View Profile](#)
- master** (Profile picture) - Email: master@gmail.com - [View Profile](#)
- divyanshu.r46956** (Profile picture) - Email: divyanshu.r46956@gmail.com - [View Profile](#)
- divyanshu** (Profile picture) - Email: divyanshu@gmail.com - [View Profile](#)

Figure 3.7 Users

Disqus for comments

Comment plug-in to add in websites, which help the users to engage in chat and provide the admin user to analyse analytics and update/delete the comments.

The screenshot shows the Disqus comment plugin integrated into a website. At the top, it displays '0 Comments' and the topic 'medicalcounselling'. Below this are social sharing buttons for 'Recommend', 'Tweet', and 'Share'. To the right, there is a user profile for 'Paarmita Bhargava' and a 'Sort by Best' dropdown. The main area features a text input field with placeholder text 'Start the discussion...'. At the bottom, there is a note 'Be the first to comment.' and links for 'Subscribe', 'Add Disqus to your site', and 'Disqus' Privacy Policy'. The Disqus logo is also present.

Figure 3.8 Disqus for comments

Admin Access Access the master admin dashboard using the following details username- master@gmail.com password- master1234 Admin has the authorization to update/delete any post and profile from the users.

The screenshot shows a web-based admin dashboard. At the top, there's a navigation bar with links for Home, About Us, Diseases, Pathy, Posts, and Contact us. On the far right, it shows 'Admin' and 'master'. The main content area has a sidebar on the left with links for Dashboard, Profile, Users, My Posts, Share Experience, Settings, Logout, FAQ, Privacy Policy, and Terms of use. The main content area displays a post titled 'Diabetes' by paarmita, posted on Thu Dec 12 2019. The post includes a 'Summary' section with the text 'Cured diabetes with the help of doctor medicines', a 'Treatment Taken' section with the text 'First- homeopathy from a doctor nearby house. Second- After 2 months no improvement so took medicines from doctors and I take it once a day. Third- Do some exercise and be healthy, it helps keep you fit', and a 'Detailed Description' section with the text 'Homeopathy from a doctor nearby house. After 2 months no improvement so took medicines from doctors and I take it once a day. Do some exercise and be healthy, it helps keep you fit'. Below the post is a handwritten image of text. At the bottom of the main content area, there are buttons for 'Update Post' (green) and 'Delete Post' (red).

Figure 3.9 Admin Dashboard

Deploy React frontend on Surge

3.2 Backend Components

• Server-side form validation

I did the server-side validation because for unaccountable browser they could have javascript turned off and whenever a client will submit a form it's gonna bypass any of the JavaScript that we have written on the client-side and it's gonna go straight to your server and then we need to validate it and if it's not then that will return an error a malicious user

```

34
35 exports.userSignupValidator = (req, res, next) => {
36   // name is not null and between 4-10 characters
37   req.check("name", "Name is required").notEmpty();
38   // email is not null, valid and normalized
39   req.check("email", "Email must be between 3 to 32 characters")
40     .matches(/.+@\..+\.+/)
41     .withMessage("Email must contain @")
42     .isLength({
43       min: 4,
44       max: 2000
45     });
46   // check for password
47   req.check("password", "Password is required").notEmpty();
48   req.check("password")
49     .isLength({ min: 6 })
50     .withMessage("Password must contain at least 6 characters")
51     .matches(/\d/)
52     .withMessage("Password must contain a number");
53   // check for errors
54   const errors = req.validationErrors();
55   // if error show the first one as they happen
56   if (errors) {
57     const firstError = errors.map(error => error.msg)[0];
58     return res.status(400).json({ error: firstError });
59   }
60   // proceed to next middleware
61   next();
62 };

```

Figure 3.10 Docs endpoints

```

1  exports.createPostValidator = (req, res, next) => {
2    // title
3    req.check("title", "Write a title").notEmpty();
4    req.check("title", "Title must be between 4 to 150 characters").isLength({
5      min: 4,
6      max: 150
7    });
8    // body
9    req.check("description", "Write a summary of the experience").notEmpty();
10   req.check("description", "Summary must be between 4 to 100 characters").isLength({
11     min: 4,
12     max: 100
13   });
14   req.check("treatmentTaken", "Write the detail about the treatment taken").notEmpty();
15   req.check("treatmentTaken", "Treatment taken must be between 4 to 2000 characters").isLength({
16     min: 4,
17     max: 2000
18   });
19   req.check("body", "Write a body").notEmpty();
20   req.check("body", "Body must be between 20 to 3000 characters").isLength({
21     min: 50,
22     max: 3000
23   });
24   // check for errors
25   const errors = req.validationErrors();
26   // if error show the first one as they happen
27   if (errors) {
28     const firstError = errors.map(error => error.msg)[0];
29     return res.status(400).json([ { error: firstError } ]);
30   }
31   // proceed to next middleware
32   next();
33 };

```

Figure 3.11 Docs endpoints

- **Full Source Code open source**

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

- **Docs for documenting the API endpoints**

```
1      [
2        "/api": "api docs",
3        "/signup": "signup",
4        "/signin": "signin",
5        "/signout": "signout",
6        "/users": "get all users",
7        "/user/:userId": "get/update/delete user",
8        "/user/photo/:userId": "get user photo",
9        "/user/findpeople/:userId": "get list of users to follow",
10       "/user/follow": "update following & follower",
11       "/posts": "get all posts",
12       "/post/new/:userId": "create new post",
13       "/posts/by/:userId": "get posts by user",
14       "/post/:postId": "get/update/delete post",
15       "/post/photo/:postId": "get post photo",
16       "url": "https://medical-umbrella.herokuapp.com"
17     ]
18
```

Figure 3.12 Docs endpoints

- **Deploy Node API to Heroku**

The screenshot shows the Heroku dashboard for the app 'medical-umbrella'. At the top, there's a navigation bar with links for Personal, medical-umbrella, Overview, Resources, Deploy, Metrics, Activity, Access, and Settings. Below the navigation, there are sections for Installed add-ons (\$0.00/month) and Dyno formation (\$0.00/month). The Dyno formation section indicates the app is using free dynos and has a command 'web npm start' with 'ON' status. On the right side, there's a 'Latest activity' section showing deployment logs for paarmita1998@gmail.com, including build successes and log entries from Oct 13 at 8:26 PM.

Figure 3.13 Heroku dashboard

3.3 Database Components

Mlab connection for Mongo Database

```
mongoose
  .connect(
    config.mongoUrl,
    { useNewUrlParser: true }
  )
  .then(() => console.log("DB Connected"));

mongoose.connection.on("error", err => {
  console.log(`DB connection error: ${err.message}`);
});
```

Figure 3.14 DB connection

MongoDB Schema Design The Application has user and posts schemas and these schemas are written using Mongoose the popular MongoDB ODM for NodeJS.

Posts Schema-

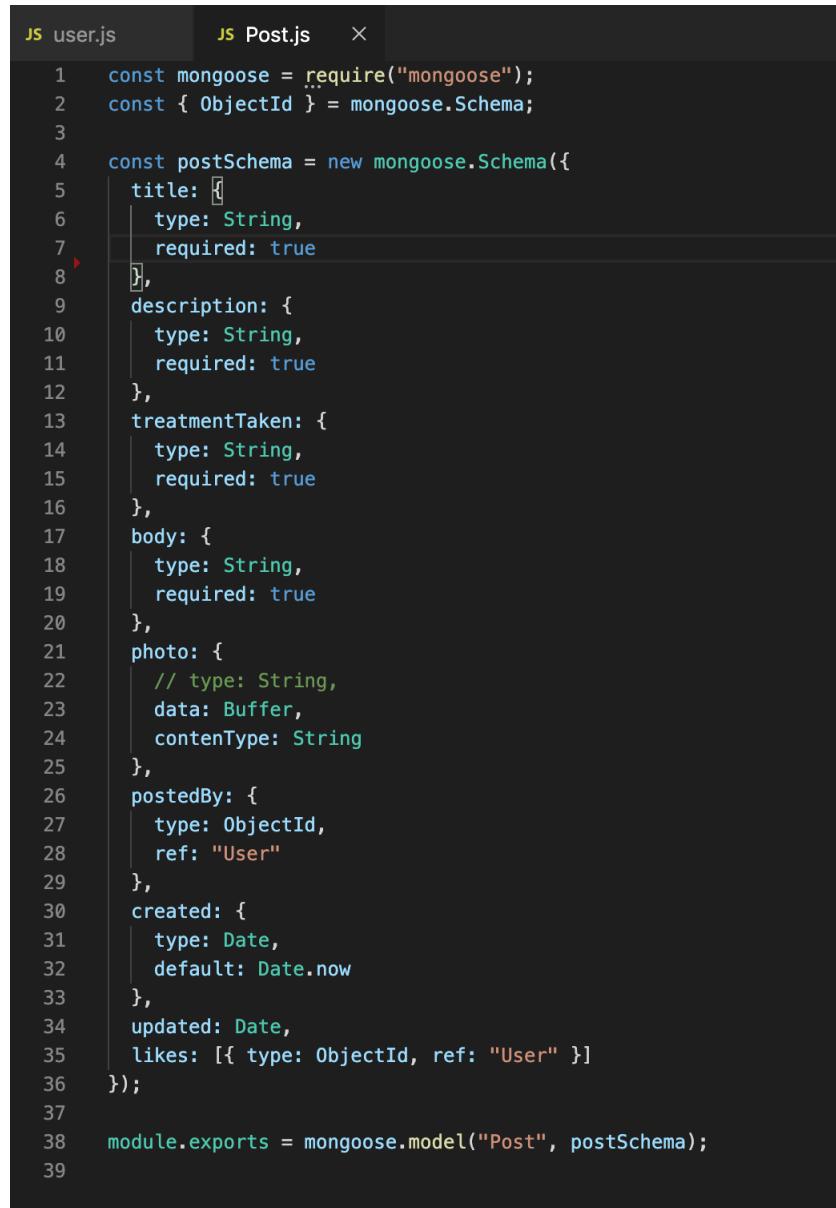
Posts schema defines the share experience posts by the user, which contains the user, post- created, posted by, etc details.

```

1 {
2   "_id": {
3     "$oid": "5dcffba3c4d562001730ef50"
4   },
5   "photo": {
6     "data": "<Binary Data>"
7   },
8   "likes": [],
9   "title": "Diabetes exprience",
10  "description": "Homeopathy",
11  "treatmentTaken": "Consulted with doctors and also took some Homeopathy medicines",
12  "body": "Initially I consulted many people and found a homeopathy doctor near my house, which people said is very effective and reduce the sugar level at a high rate. But I couldn't see improvement and went to a doctor and take medicines 1 a day before breakfast and do yoga and exercise as well, and is really effective for me",
13  "created": {
14    "date": "2019-11-16T13:37:39.172Z"
15  },
16  "postedBy": {
17    "$oid": "5dce4b3c1bfec50017b023f3"
18  },
19  "__v": 0
20 }

```

Figure 3.15 Posts Schema stored data



```

JS user.js          JS Post.js      X
1  const mongoose = require("mongoose");
2  const { ObjectId } = mongoose.Schema;
3
4  const postSchema = new mongoose.Schema({
5    title: {
6      type: String,
7      required: true
8    },
9    description: {
10      type: String,
11      required: true
12    },
13    treatmentTaken: {
14      type: String,
15      required: true
16    },
17    body: {
18      type: String,
19      required: true
20    },
21    photo: {
22      // type: String,
23      data: Buffer,
24      conteNtType: String
25    },
26    postedBy: {
27      type: ObjectId,
28      ref: "User"
29    },
30    created: {
31      type: Date,
32      default: Date.now
33    },
34    updated: Date,
35    likes: [{ type: ObjectId, ref: "User" }]
36  });
37
38  module.exports = mongoose.model("Post", postSchema);
39

```

Figure 3.16 Posts Schema

User Schema- This schema defines the data used to store individual users. The user document will contain the user's username, profile picture, etc.

```
1 {
2     "_id": {
3         "$oid": "5d31a75062b22731c3df4d42"
4     },
5     "following": [
6         {
7             "$oid": "5d3300f524b6225d3517d171"
8         }
9     ],
10    "followers": [
11        {
12            "$oid": "5d3300f524b6225d3517d171"
13        }
14    ],
15    "name": "paarmita",
16    "email": "paarmita1998@gmail.com",
17    "salt": "1c233570-aa17-11e9-89b3-39c0fb40e72e",
18    "hashed_password": "784559f8cf2d918c8a5e8bf2265444838699277e",
19    "created": {
20        "$date": "2019-07-19T11:19:44.458Z"
21    },
22    "__y": 0,
23    "about": "Hi, I'm a developer and designer",
24    "updated": {
25        "$date": "2019-07-19T11:28:09.749Z"
26    }
27}
```

Figure 3.17 User Schema stored data

```
JS user.js    ×
 3  const crypto = require("crypto");
 4  const { ObjectId } = mongoose.Schema;
 5
 6  const userSchema = new mongoose.Schema({
 7    username: {
 8      type: String,
 9      trim: true,
10      required: true
11    },
12    name: {
13      type: String,
14      trim: true,
15      required: true
16    },
17    email: {
18      type: String,
19      trim: true,
20      required: true
21    },
22    hashed_password: {
23      type: String,
24      required: true
25    },
26    salt: String,
27    created: {
28      type: Date,
29      default: Date.now
30    },
31    updated: Date,
32    photo: {
33      data: Buffer,
34      contentType: String
35    },
36    about: {
37      type: String,
38      trim: true
39    },
40    following: [{ type: ObjectId, ref: "User" }],
41    followers: [{ type: ObjectId, ref: "User" }],
42    role: {
43      type: String,
44      default: "subscriber"
45    }
46  });

```

Figure 3.18 User schema

Chapter 4

Conclusions and Future Work

We encourage more and more ideas and people to join us accomplishing our mission of making the world a better place.

4.1 Conclusion

The project comprises of all the required features. It can be customized in terms of performance by using cloud services for all the documents and using efficient data structure algorithms on the database to fasten the retrieving time of data. The project delivers a working **Medical web application**. It provides a platform for users to interact with the community and view disease summary and post their experiences. Nothing is truly perfect and the same is the case for this project.

There is a room for improvements which can make this platform, even more, better, such improvements are discussed in the next section.

4.2 Problem faced

The structure for the diseases was getting too big and nested and the efficiency of the website became poor, so after some research, I created an ID and divided the object and accessed the two using an ID. Also, the application faced some serious glitches for different browsers so, it required me to build all the things from scratch. The responsiveness is also not been able to fix because the sidebar is creating issues, so I discussed it with the Advisor and we decided to make it web-based only.

4.3 Key Learning's

This project assisted me to gain practical experience and apply the knowledge assimilated from the previous courses undertook like Database, Software engineering, language courses, etc. Applying the gained knowledge gained and different techniques were interesting and challenging. After implementation in real-world scenarios, I better understood certain concepts, tools and techniques learned earlier.

It was extremely challenging at times but it has been a great and worthwhile learning experience. So, hope this platform would be really helpful for people looking for advice in the medical field.

4.4 Scope of further work

The project can be extended to a very high level, so based on my research I have created issues on **Github issues**, for adding the features which will make this platform much more efficient. The features which are yet to implement in the website are-

- Upload assets to the cloud
- Add better text editor for forms in Share experience to enhance the user experience.
- Healthy living blog- A blog that will provide information around: Daily Nutrition, Keys to maintaining fitness and hygiene. This data can be pulled from third-party APIs like Yahoo Health.
- Allowing testimonials to be edited- Logged in users would be able to edit their testimonials added by them.
- The feature of commenting and rating testimonial- The testimonials can be rated by users logged into the platform. Facebook, Google +, Twitter sharing abilities can be integrated with plugins.
- Newsletters
- Visitors Count- Once integrated to the server-side and hosted on a domain, we would be able to fetch the analytics for the site. Google Analytics, Omniture and CMS tools like SiteCatalyst.
- Verification/email confirmation to be added in future work. and furthermore.

Bibliography

1. **Bootstrap**- <https://getbootstrap.com/>
2. **React**- <https://reactjs.org/>
3. <http://www.w3schools.com/>
4. **PatientsLikeMe's** mission is to improve the lives of patients through new knowledge derived from shared real-world experience and outcomes.
5. **Inspire.com** - Leading social network for health that connects patients and caregivers in a safe permission-based manner
6. **Centre for disease control and prevention** - Diseases and Conditions data for diseases sample data