Report On

HOSPITAL MANAGEMENT SYSTEM

Bangladesh University of Business and Technology

Bangladesh University of Business and Technology

Department: CSE

Intake- 27

Section-1

Report on: Hospital Management System

Supervisor: Assistant Prof. Milon Biswas

Members of the Group:

Student ID	Student Name
16173203029	**Sharmin Akter**
16173203038	**Md. Alamin**

INDEX **TITLE** **PAGE**

DECLERATION

Declaration I hereby declare that this is my own work. Materials of work found by other researchers are mentioned by reference. This Thesis, neither in whole nor in part, has been previously submitted for any degree.

Sharmin Akter

16173203029

Md. Alamin

16173203038

ACKNOWLEDGEMENT

First of all, we are thankful and expressing our gratefulness to Almighty Allah who offers us His divine blessing, patient, mental and physical strength to complete this project work.

We are deeply indebted to our project supervisor Milon Biswas, Assistant Professor, Department of Computer Science and Engineering, CSE BUBT. His scholarly guidance, important suggestions, work for going through our drafts and correcting them, and generating courage from the beginning to the end of the research work has made the completion of this thesis possible. his wonderful guidance, inspiration, encouragement and also for through review and correction of this dissertation work that could not be nailed without his astute supervision. A very special gratitude goes out to all our friends for their support and help to implement our works .The discussions with them on various topics of our works have been very helpful for us to enrich our knowledge and conception regarding the work.

Last but not the least; Our parents are very much keen and hopeful in the best performance of the dissertation we are going to submit.

ABSTRACT

Our project Hospital Management system includes registration of Patients and Doctor's, storing their details into the system, and Our software has the facility to give a unique id for every patient as well as doctor's and stores the details of every patient and the doctor's. It includes a search facility to know the current status of the doctor's including their expertise. User can search availability of a doctor and the details of a patient using the id and also there's a section of searching any groups of blood.

We declare that this project and the work presented in it are our own and has been generated by us and hereby declare that the project entitled Hospital Management System. is our own work and that it contains no material which has been accepted for the award to the candidate(s) of any other degree or diploma, except where due reference is made in the text of the project. To the best of our knowledge, it contains no materials previously published or written by any other person except where due reference is made in the project.

APPROVAL

This Project “Hospital Management System" Submitted by Sharmin Akter ID NO: 16173203029, Md. Alamin ID NO: 16173203038, Department of Computer Science and Engineering (CSE), Bangladesh University of Business and Technology (BUBT) under the supervision of Milon Biswas, Assistant Professor, Department of Computer Science and Engineering has been accepted Science and approved as to its style and contents.

_____

Supervisor

Assistant Prof. Milon Biswas

Assistant Professor

Department of Computer Science and Engineering (CSE)

Bangladesh University of Business and Technology(BUBT)

_____

Chairman

Prof. Dr. Amir Ali

Professor and Chairman

Department of Computer Science and Engineering (CSE)

Bangladesh University of Business and Technology(BUBT)

1. Introduction

A unique cloud-based hospital management system for both patients and hospital stuffs (Doctors, Patient).
The primary target of this design is to make hospital experience better than we currently have. Hospital is a place where no one willingly wants to visit but there are times when we need to.

1.1. Problem Statement

- **Lack of immediate retrievals:** The information is very difficult to retrieve and to find particular information like- E.g. - To find out about the patient's history, the user has to go through various registers. This results in inconvenience and wastage of time.

- **Lack of immediate information storage:** The information generated by various transactions takes time and efforts to be stored at right place.

- **Lack of prompt updating:** Various changes to information like patient are difficult to make as paper work is involved.

1.2. Project Objectives

1. **Planned approach towards working:** - The working in the organization will be well planned and organized. The data will be stored properly in data stores, which will help in retrieval of information as well as its storage.

2. **Accuracy:** - The level of accuracy in the proposed system will be higher. All operation would be done correctly and it ensures that whatever information is coming from the center is accurate.

3. **Reliability:** - The reliability of the proposed system will be high due to the above stated reasons. The reason for the increased reliability of the system is that now there would be proper storage of information.

4. **Immediate retrieval of information:** - The main objective of proposed system is to provide for a quick and efficient retrieval of information. Any type of information would be available whenever the user requires.

5. **Immediate storage of information:** - In manual system there are many problems to store the largest amount of information.

6. **Easy to Operate:** - The system should be easy to operate and should be such that it can be developed within a short period of time and fit in the limited budget of the user.

1.3 Motivations of the Project

We do not usually visit hospitals unless we need to but this visit does not always give us good experience.
We face various problems there. There are automated hospital management system but they do not provide any functionality for us.

Even the doctors do not have access to their patient's data while they are at home.
It will be great for the patients to have an application that will keep them close to their personal data like: prescriptions, test reports and other important materials.

Patients will be able to browse through all the doctors available and ask for appointment for their desired doctor. A doctor will also experience the similar benefits. This will reduce a lot of paper works and make things easy for everyone associated with it.

People experience their worst nightmare while they need to visit the hospital.
No body visits the hospital unless it is extremely necessary. People get mad but they do not have anything to do then.

This gave me the perfect motivation to build something for them and offer them some help at their most needed time. I am confident that this product can and will help them

2. BACKGROND

2.1. Existing Systems

There is some other platform there which also based on hospital management system in terms of developing human being's healthcare.

2.1.1 https://www.bdtask.com

In here, there is a major section which name is Consultation System where there's a lot of medical facilities to have. In this section, for both Doctor and hospital e- prescription management system is the one of their biggest cloud-based application. This is based on internet totally.

2.1.2 https://www.saintfrancis.com

Saint Francis Health System offers a complete continuum of healthcare services. From primary care to emergency/trauma and advanced medical specialty care, Saint Francis Health System is dedicated to the healthcare needs of Oklahoma family.
Saint Francis Health System offers a wide variety of medical specialties at locations throughout eastern Oklahoma.

From the tiniest premature newborns to end-of-life support to all the needs in between, the physicians and staff of all Saint Francis Health System facilities are committed to treating each patient with dignity and integrity.

3. PROPOSED MODEL

3.1 Feasibility Analysis

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates.
During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Key considerations involved in the feasibility analysis are:

3.1.1 Economic Feasibility

This study is carried out to check the economic impact will have on the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified.

Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products have to be purchased.

3.1.2. Technical Feasibility

This study is carried out to check the technical feasibility, that is the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands being placed on the client.
The developed system must have a modest requirement, as only minimal or null changes for the implementing this system.

3.2. System Design

3.2.1. Methodology

A methodology is a formalized approach to implementing the SDLC (i.e., it is a list of steps and deliverables).
There are several system development methodologies, and each one is unique, based on the order and focus it places on each SDLC phase.

Some methodologies are formal standards used by government agencies, whereas others have been developed by consulting firms to sell to clients.

Many organizations have internal methodologies that have been honed over the years, and they explain exactly how each phase of the SDLC is to be performed in that company.
Historically, system development approaches have been progressing and continually new methodologies emerge yearly following a sequence is years.
Here is list of some methodologies: OOP, SCRUM, AGILE, WATERFALL, RAD, SDM2 etc.

3.2.2 Schema Diagram

Fig: Database schema diagram

3.2.3 Data Flow Diagram(DFD)

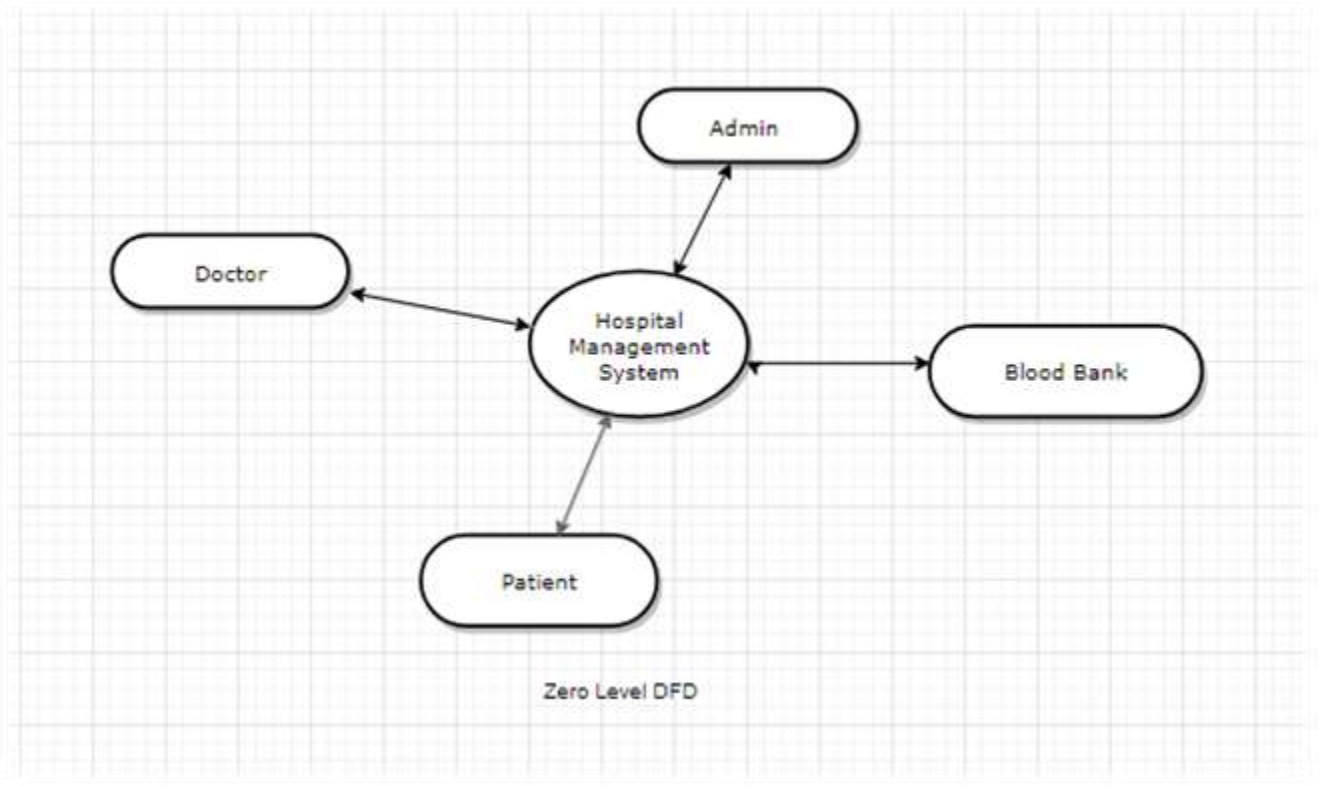


Fig: zero level data-flow-diagram

3.2.4. Use Case Diagram

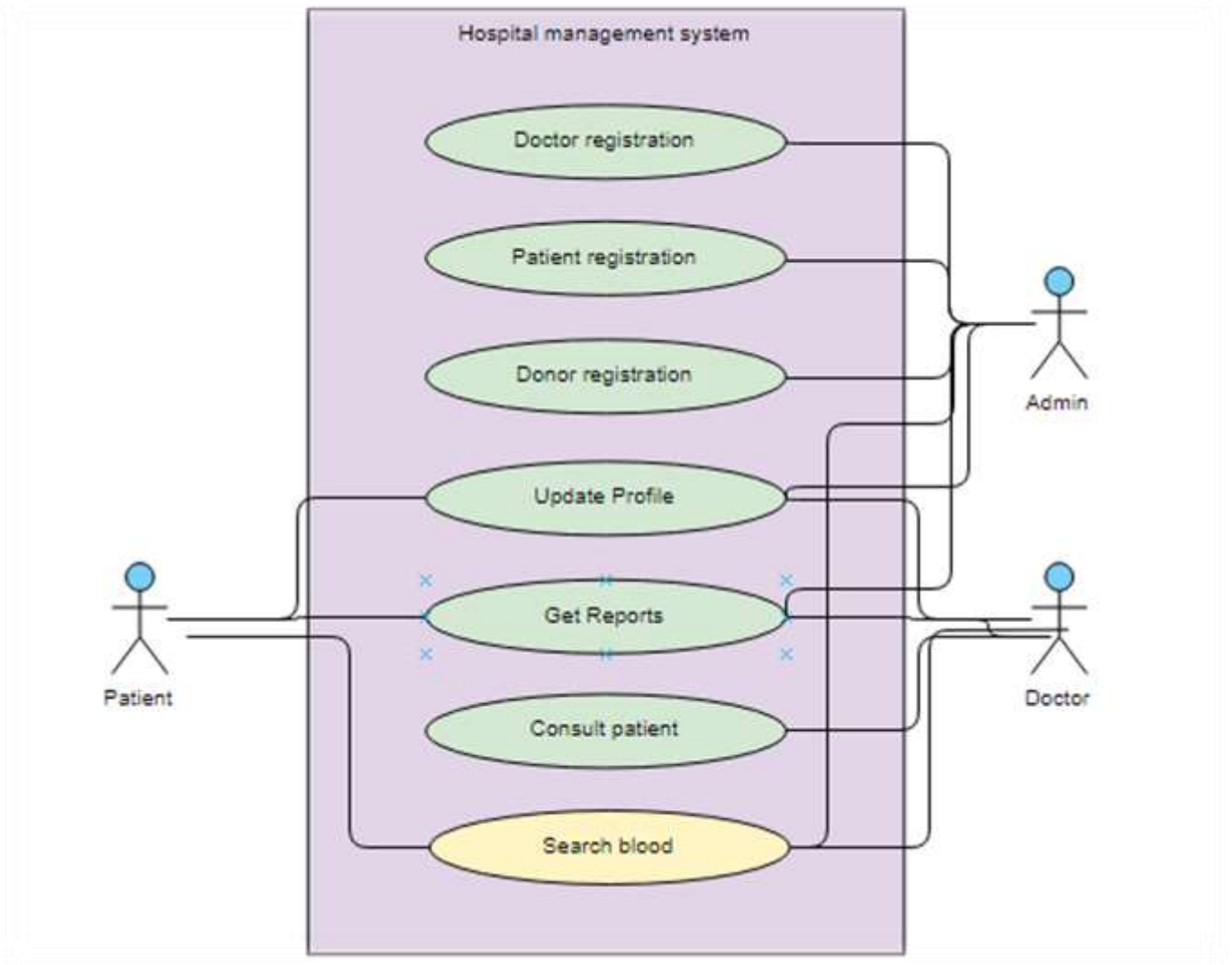
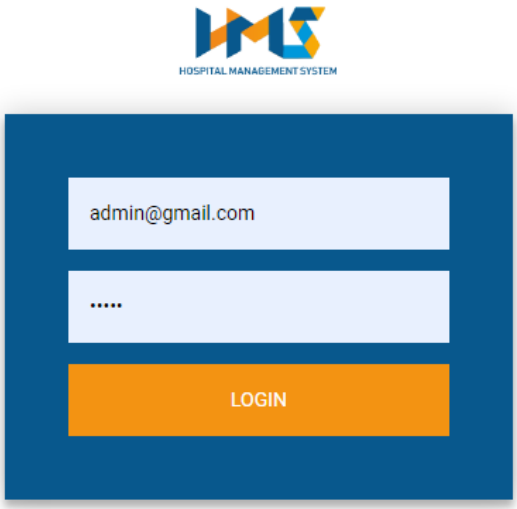


Fig: use case diagram

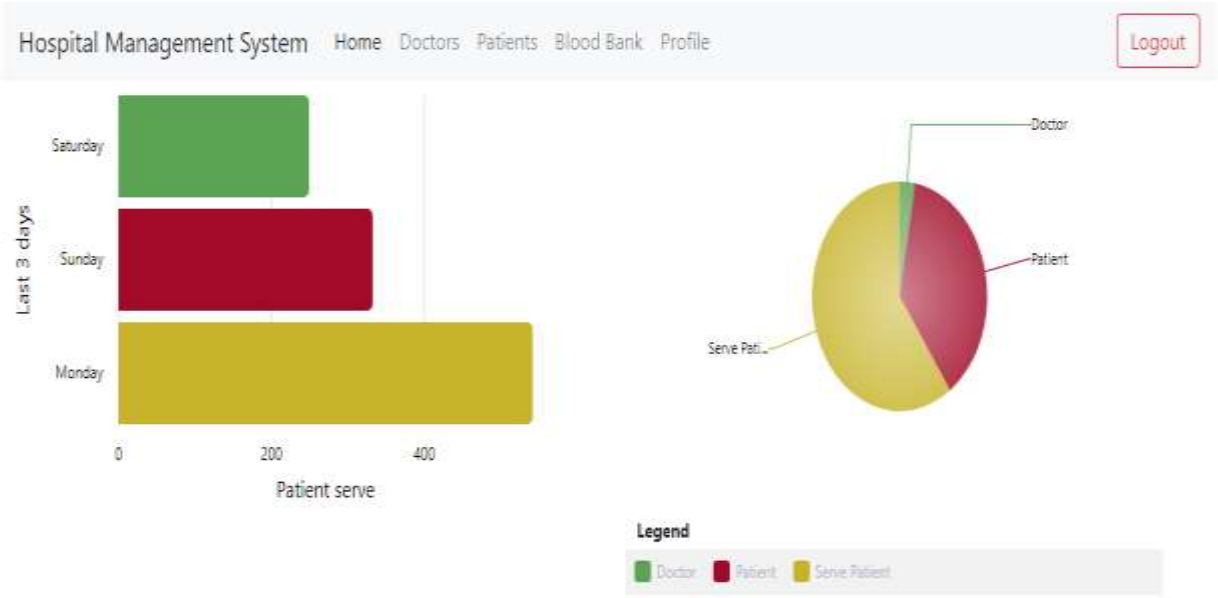
3.2.5 UI Design

Admin panel: Admin has the power to create user profile both for patients and doctors. They will monitor the patients and doctor's activities.

Login screen:

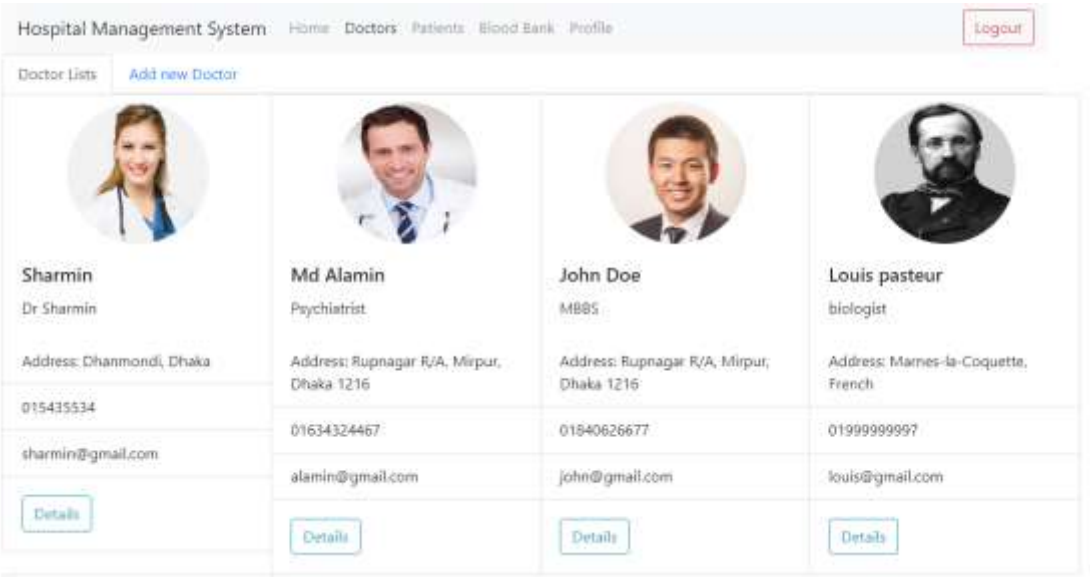


The dashboard screen:



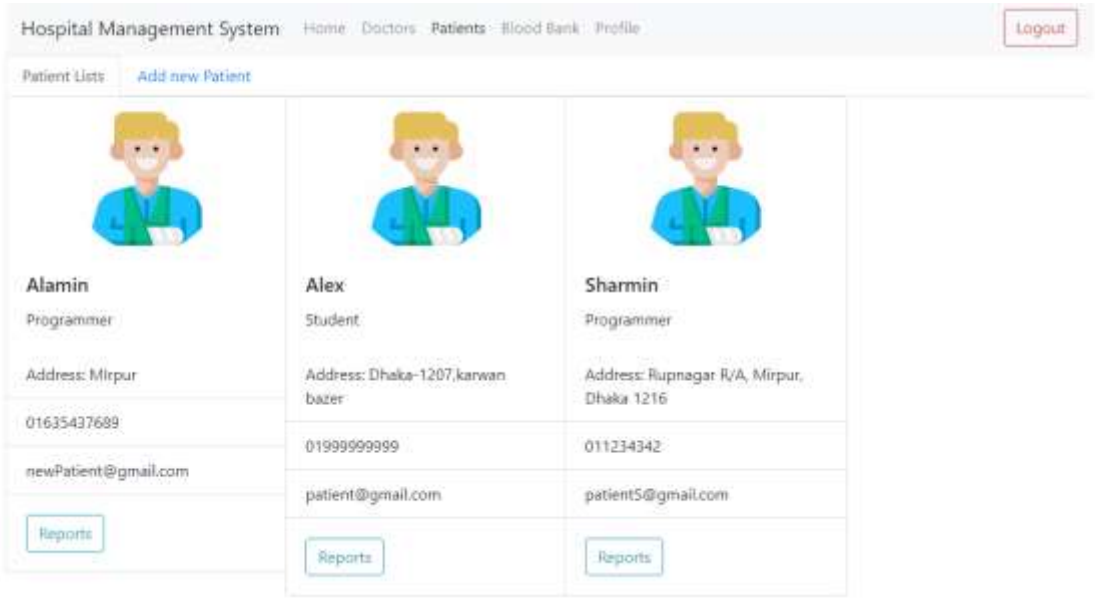
This dashboard is for admin panel. Admin can see how many patients serve from this hospital and how many doctors and patient in this system. And they can see last 3 days of data for serving patient.

Doctor list screen:



Here is the all doctor lists with their name, email, designation and others. And admin can go through this profile to see their details.

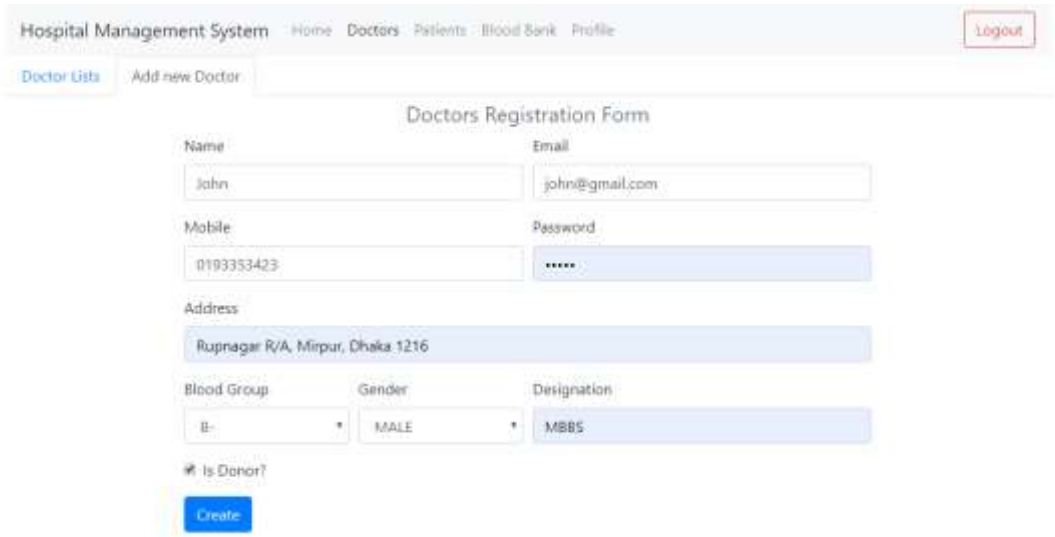
Patient list screen:



All patient lists are going there with patient details.

And admin can see details reports of every patients.

Add new doctor:



This is admin registration form, for register new doctor, they need some information.

- Name
- Email
- Mobile
- Password
- Address
- Gender
- Blood group
- Designations
- Is donor or not

Add new Patient:

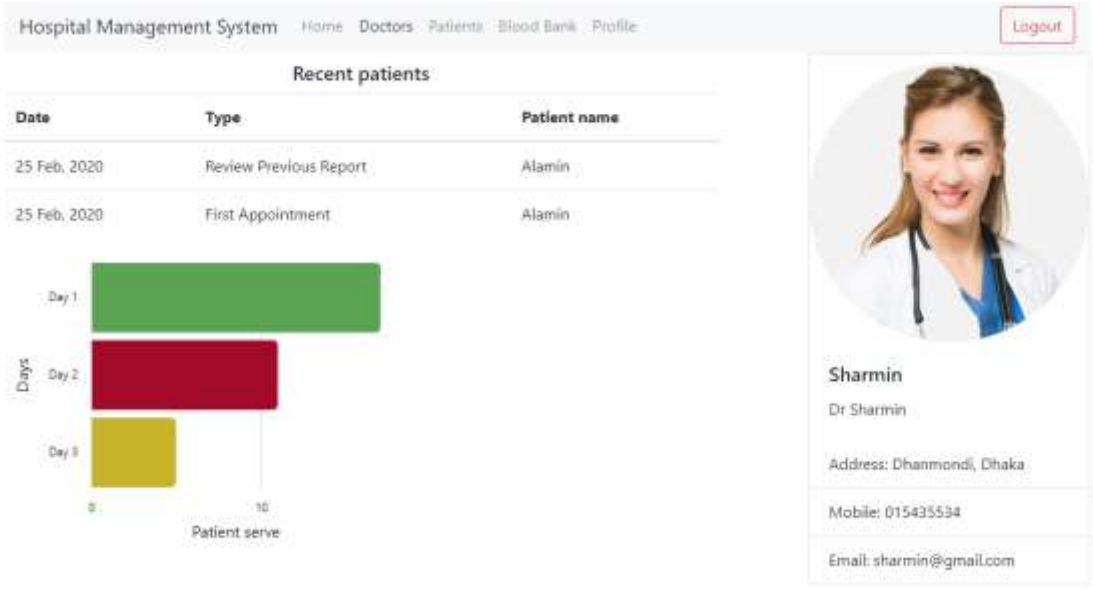
This is patient registration form, for register new patient, they need some information.

- Name
- Email
- Mobile
- Password
- Address
- Gender
- Blood group
- Designations
- Is donor or not

Doctor Panel:

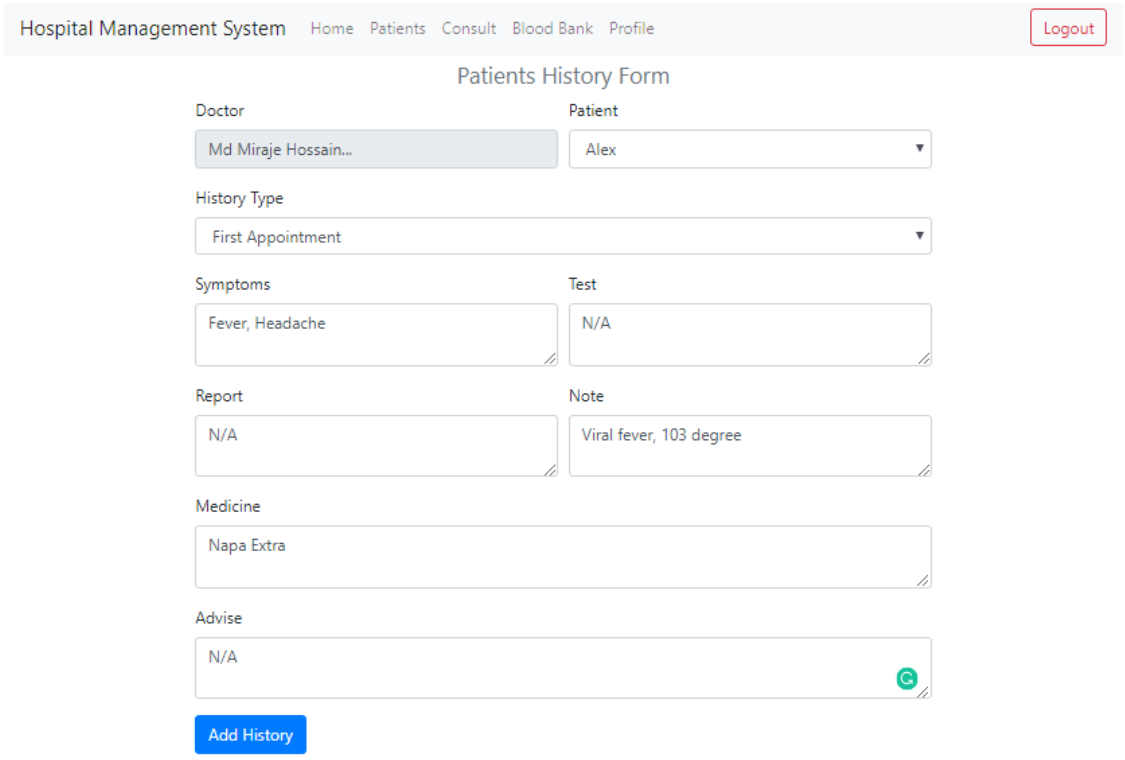
Doctors can start prescribing their patients by clicking the Consult tab. It will directly lean them for the prescription page. The prescription page has a lot of automated features for the doctor to set things and assign to their patients. Every field that is necessary for a doctor to make a perfect prescription is available. Which will obviously reduce the workload of doctors and they can give proper time to patients. There will be few clicks for them to make a proper prescription

Doctor details report:



This is doctors details report, doctors can see their recent patients and their recent graph.

Consult patient:

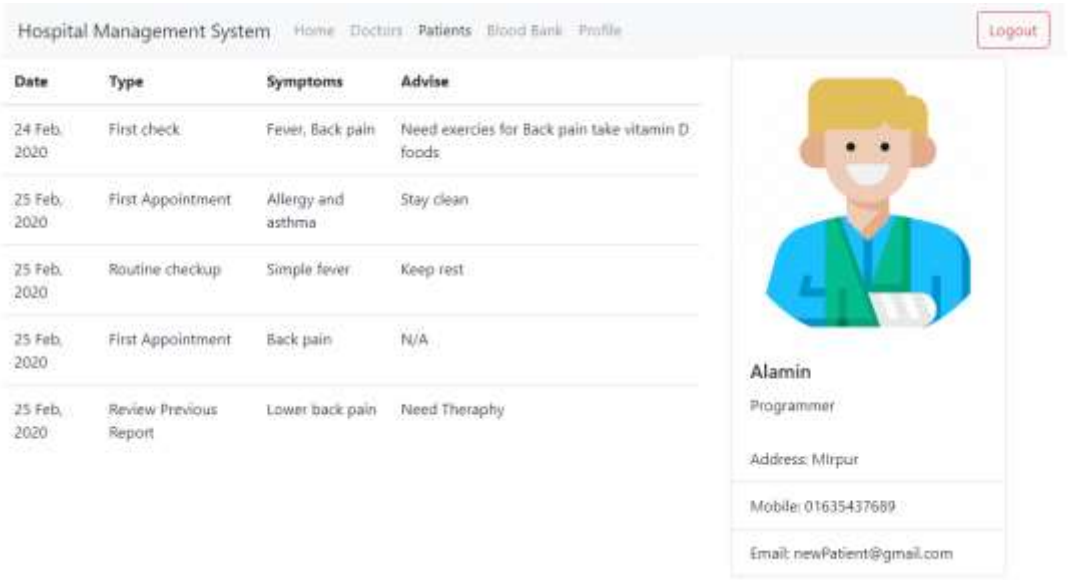


The Consult page has some input options for the doctor to set things and assign to their patients. Every field that is necessary for a doctor to make a perfect prescription. Which will obviously reduce the workload of doctors and they can give proper time to patients. There will be few clicks for them to make a proper prescription

Patient Panel:

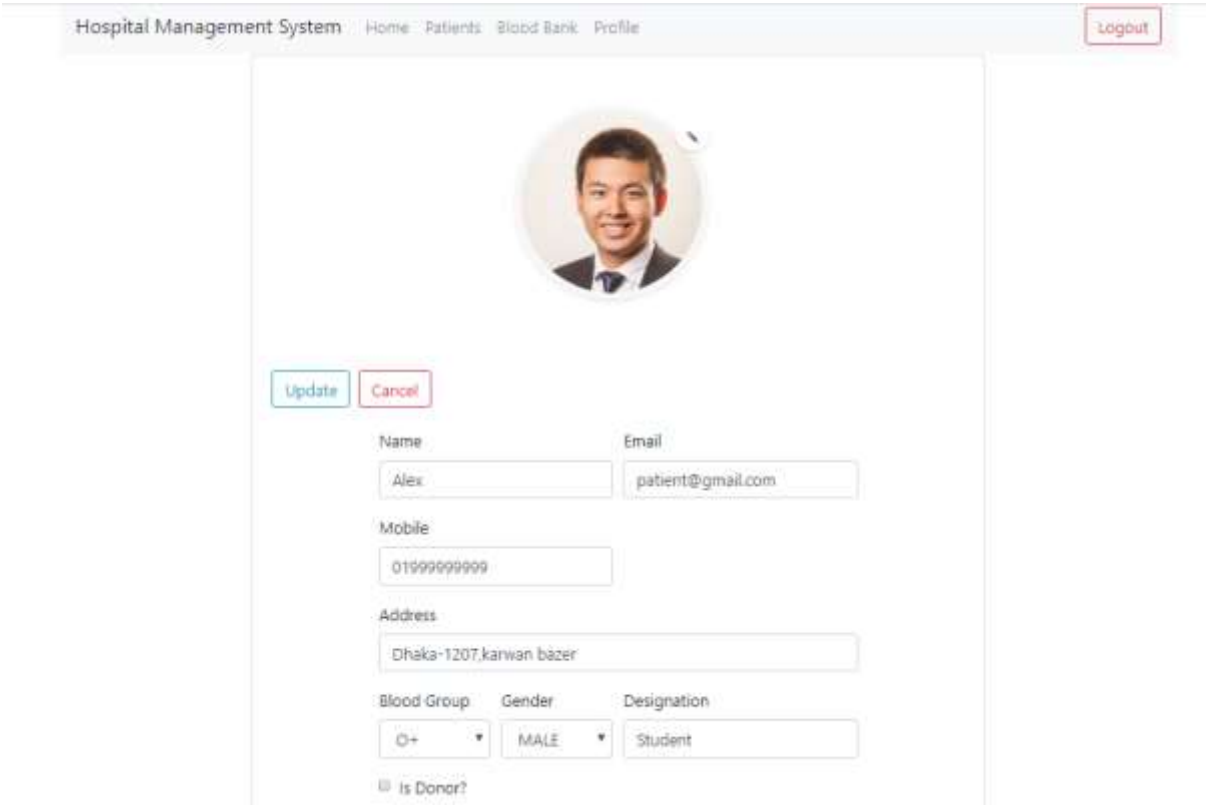
Patients are normal users like us. Patients can also check their previous history. The patient's functionality is limited based on their need. They will only be able to change their personal information and other things.

Patient history details:



This is patient history, patient can see their reports by the doctors.

Patient profile update:



Patient can update their profile, the can change their name, designation address and so on.

4. IMPLEENTATION AND TESTING

4.1 Front-end (Admin/Doctor/Patient) panel:

Technologies: We use Angular as a front end framework to implement this application.

So Why Angular?

- **Angular** is a frontend **framework** which remains one of the best solutions for web development. Discover its characteristics, bottlenecks, and best use cases. Angular is latest most popular front-end framework which is used to build Single Page Application(SPA).

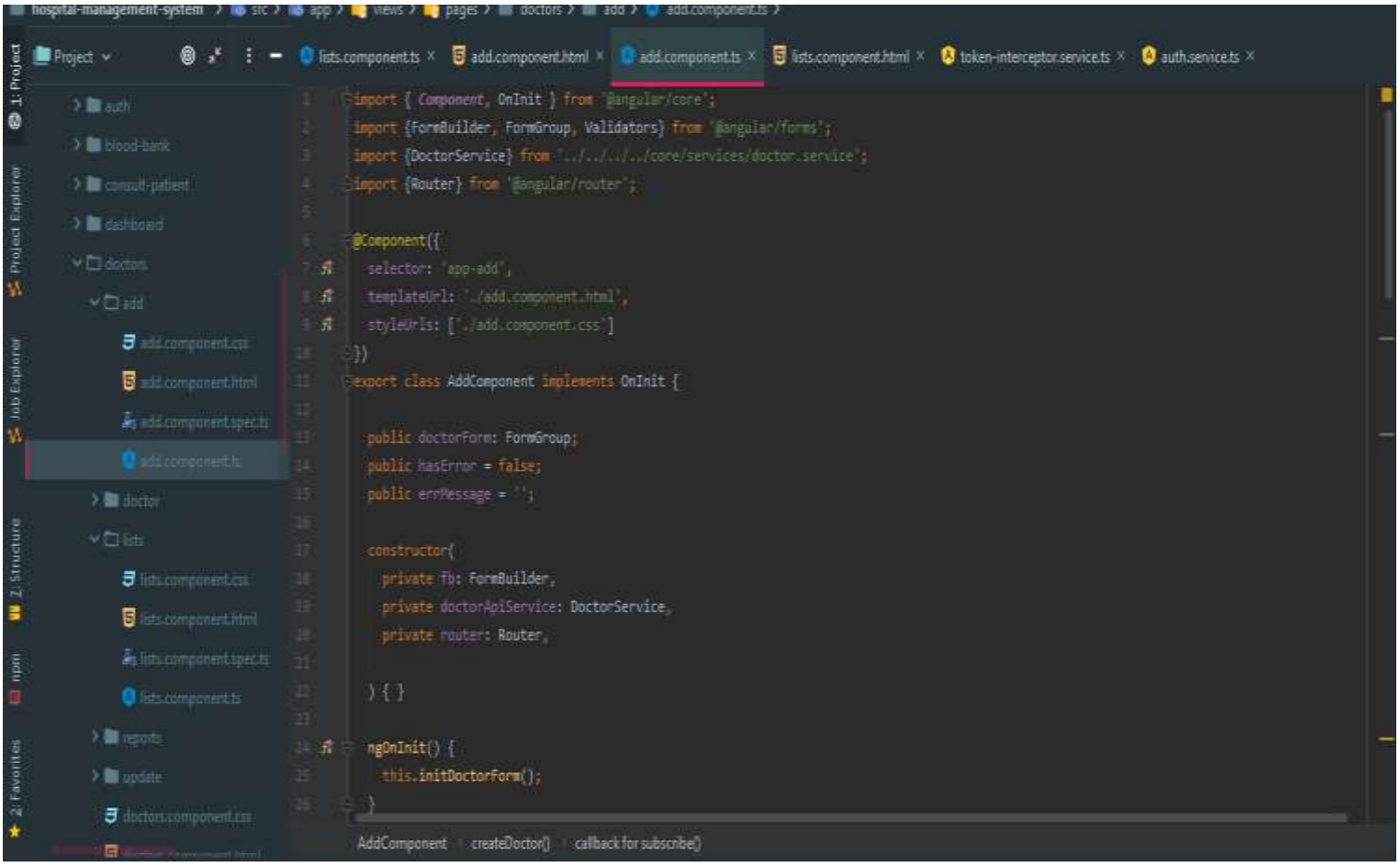
- **Detailed documentation.** Angular boasts detailed documentation where developers can find all necessary information without asking their colleagues. As a result, developers can quickly come up with technical solutions and resolve emerging issues.

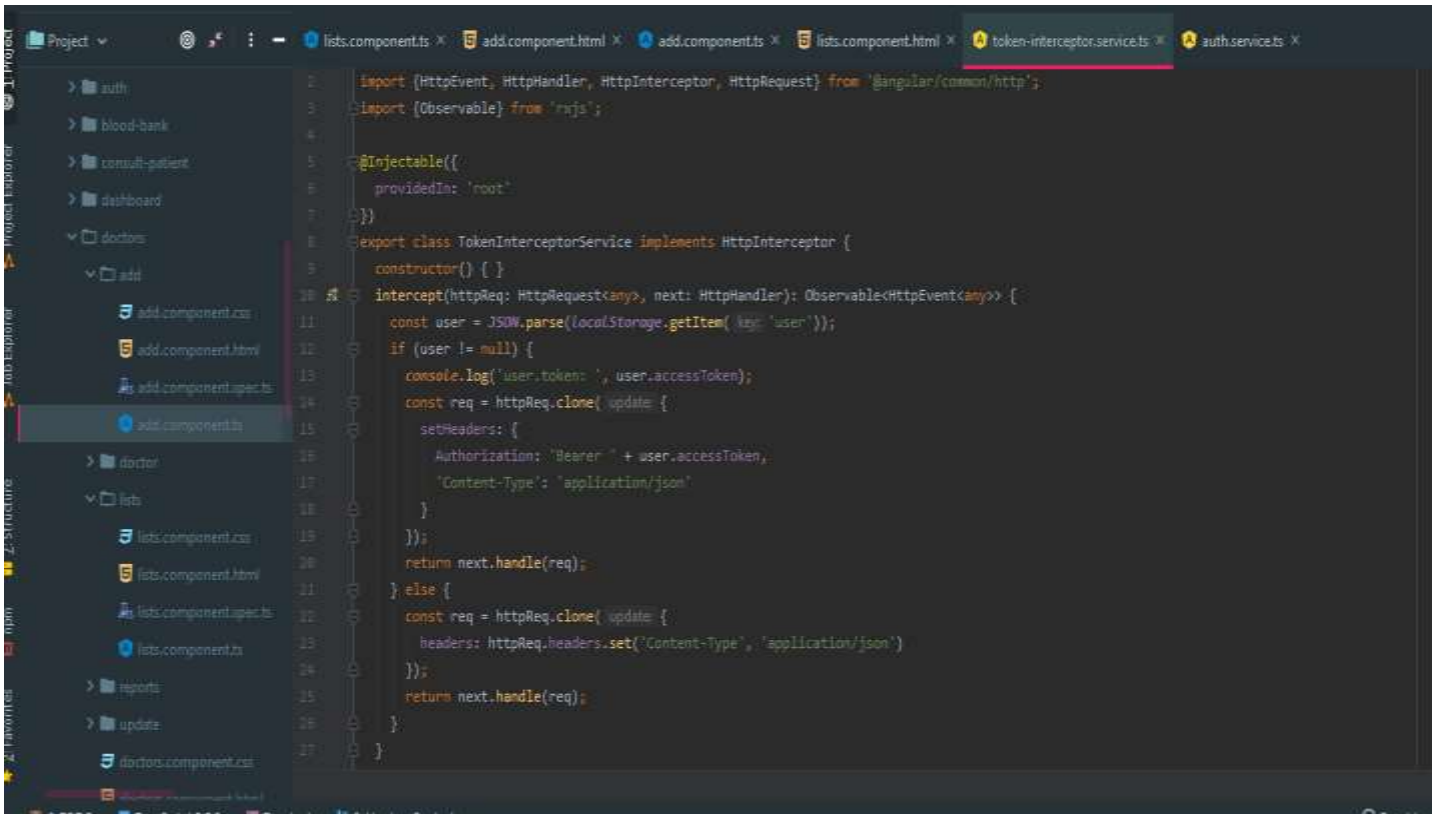
- **Support by Google.** A lot of developers consider Google support another benefit of Angular, making the platform trustworthy. At ng-conf 2017, the developers of Angular confirmed that Google will support Angular on a long-term basis.

- **Great ecosystem of third-party components.** The popularity of Angular has resulted in the appearance of thousands of additional tools and components that can be used in Angular apps. As a result, you can get additional functionality and productivity improvements.

- **Component-based architecture**. In the second version, Angular shifted from an MVC to a component-based architecture. According to this architecture, an app is divided into independent logical and functional components. These components can easily be replaced and decoupled as well as reused in other parts of an app. In addition, component independence makes it easy to test a web app and ensure that every component works seamlessly.

  Here I attach some snipped of code of front-end:




Project
lists.component.ts
add.component.html
add.component.ts
lists.component.html
token-interceptor.service.ts
auth.service.ts
auth
blood-bank
consult-patient
dashboard
doctors
add
add.component.css
add.component.html
add.component.spec.ts
add.component.ts
doctor
lists
lists.component.css
lists.component.html
lists.component.spec.ts
lists.component.ts
reports
update
doctors.component.css
import {HttpEvent, HttpHandler, HttpInterceptor, HttpRequest} from '@angular/common/http';
import {Observable} from 'rxjs';
@Injectable({
providedIn: 'root'
})
export class TokenInterceptorService implements HttpInterceptor {
constructor() { }
intercept(httpReq: HttpRequest<any>, next: HttpHandler): Observable<HttpEvent<any>> {
const user = JSON.parse(localStorage.getItem(key: 'user'));
if (user != null) {
console.log('user.token: ', user.accessToken);
const req = httpReq.clone(update: {
setHeaders: {
Authorization: 'Bearer ' + user.accessToken,
'Content-Type': 'application/json'
}
});
return next.handle(req);
} else {
const req = httpReq.clone(update: {
headers: httpReq.headers.set('Content-Type', 'application/json')
});
return next.handle(req);
}
}

4.2 Backend:

For backend we use latest **Node.js** latest LTS version for server side with **express framework**.
For database we use MongoDB as document based databased called NoSQL.

What is Node.js: Node.js is a platform built on Chrome's JavaScript runtime for easily building fast and scalable network applications.
Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.

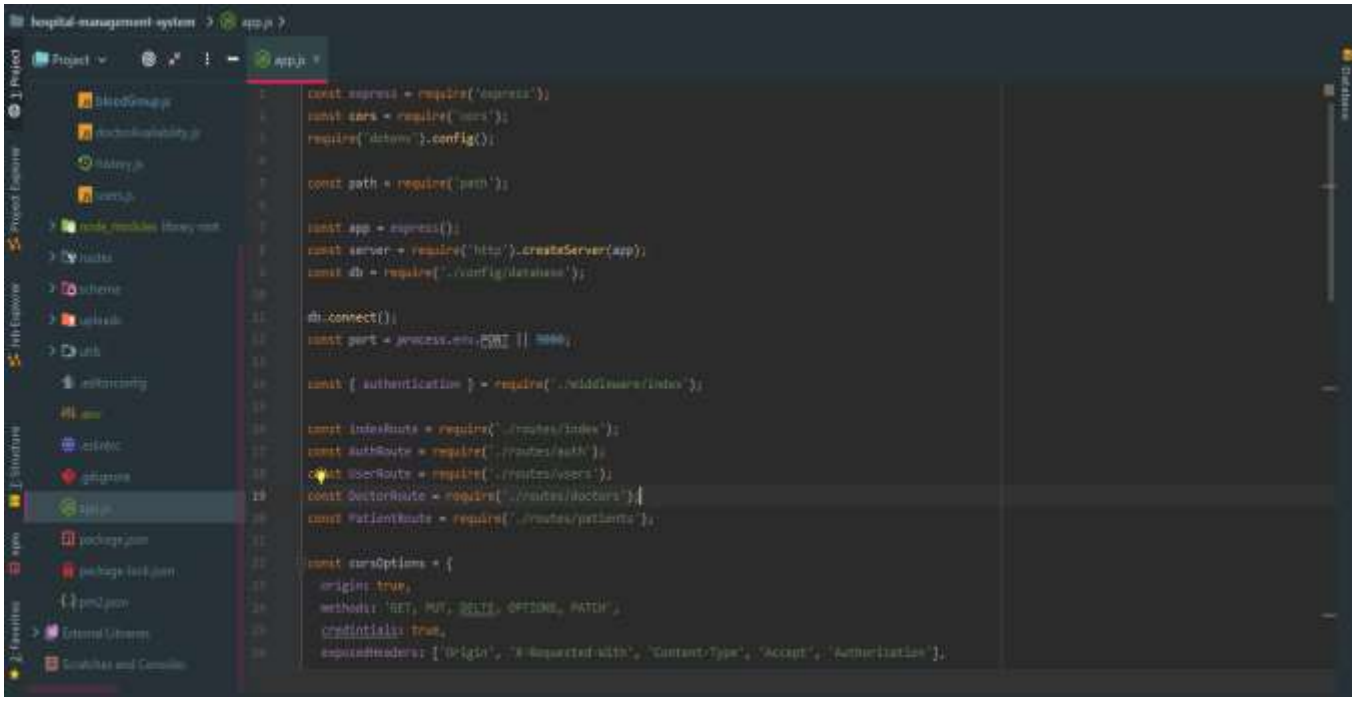
Why Node.js: The major advantage, is that this JavaScript language doesn't block I/O – meaning input/output communication method.
Here, though, the developer community has two views. Some argue, that applications with many CPU cycles can crash then. Others say it's not a big deal at all, as Node code works in small processes.

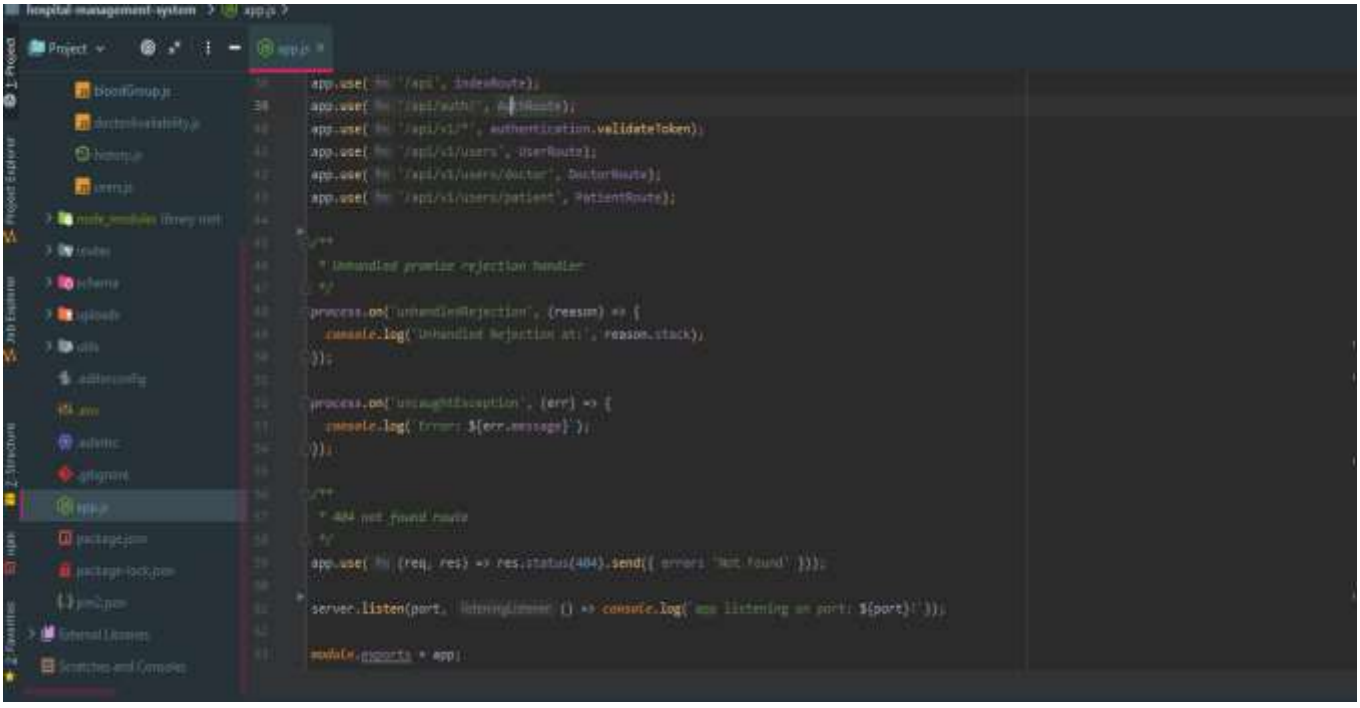
Another benefit is **single-threaded event loop**, that is responsible for abstracting I/O from external requests. Speaking plainly, this means that Node initiates the event loop at the start, processes the input, and begins the order of operations.

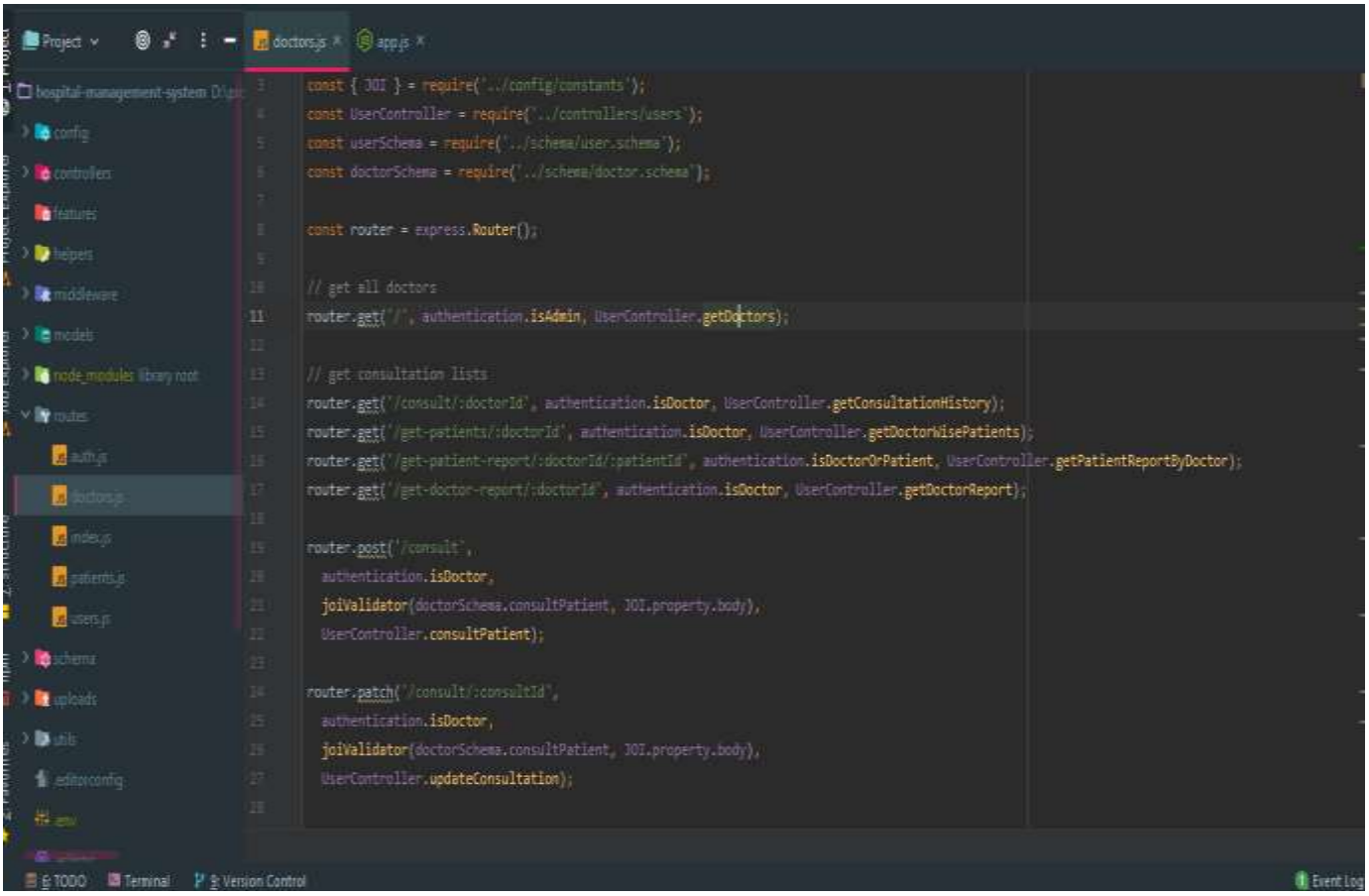
What is Express: Express.js, or simply Express, is a web application framework for Node.js, released as free and open-source software under the MIT License.
It is designed for building web applications and APIs. It has been called the de facto standard server framework for Node.js

MongoDB: MongoDB is a cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with schema. MongoDB is developed by MongoDB Inc. and licensed under the Server Side Public License.

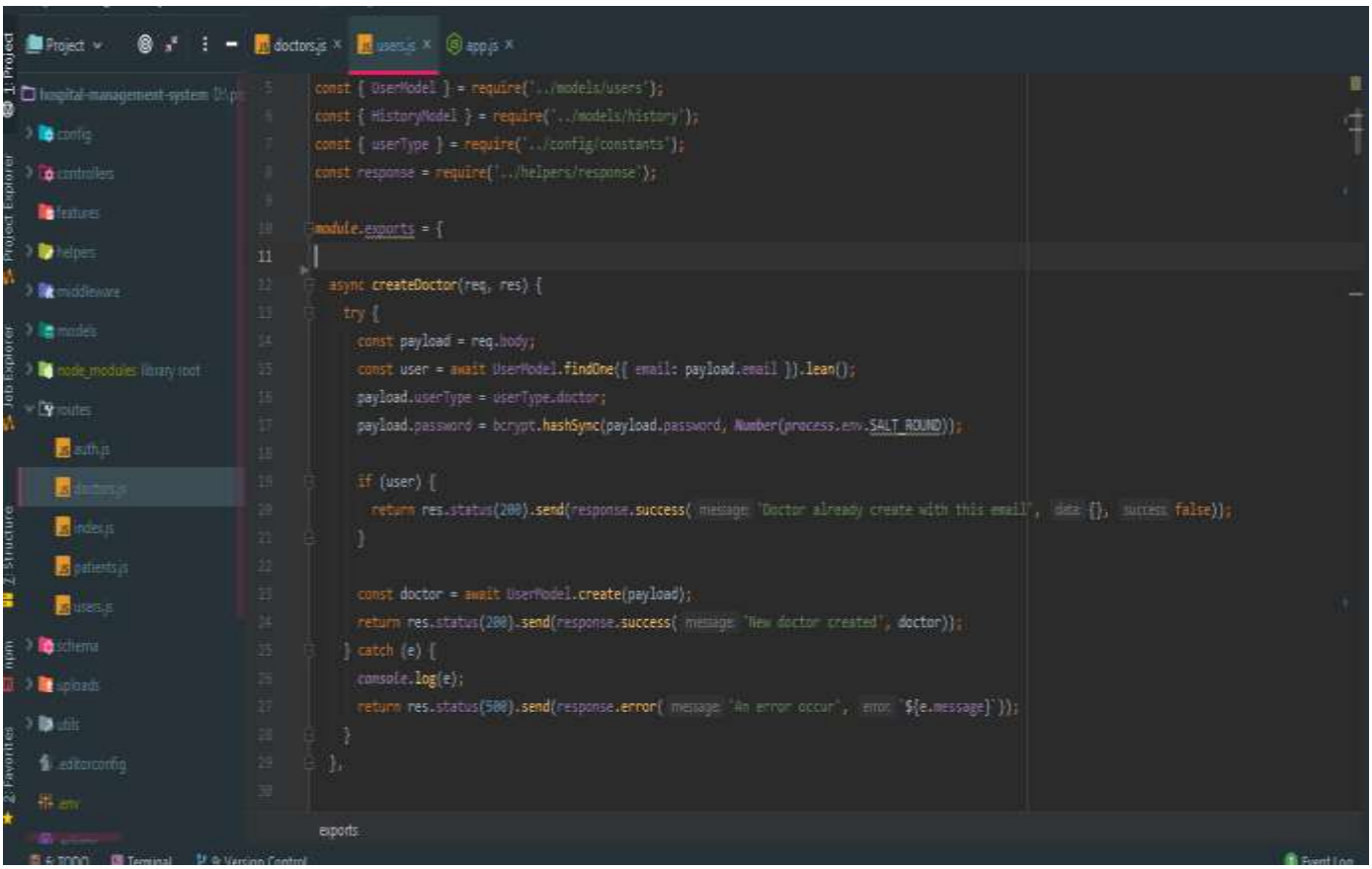
Here I attach some server side code:



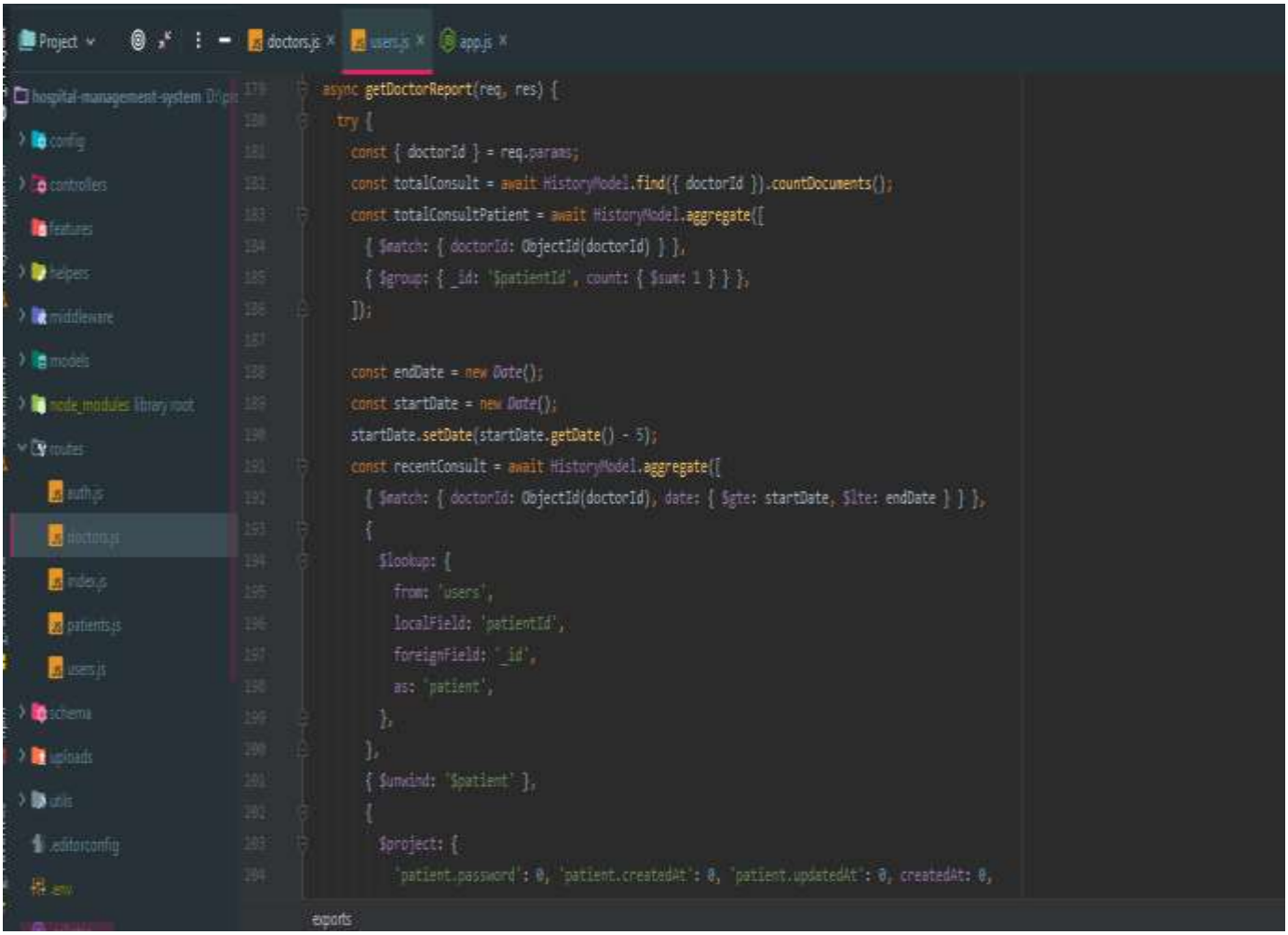




```
const { JOI } = require('../config/constants');
const UserController = require('../controllers/users');
const userSchema = require('../schema/user.schema');
const doctorSchema = require('../schema/doctor.schema');

const router = express.Router();

// get all doctors
router.get('/', authentication.isAdmin, UserController.getDoctors);

// get consultation lists
router.get('/consult/:doctorId', authentication.isDoctor, UserController.getConsultationHistory);
router.get('/get-patients/:doctorId', authentication.isDoctor, UserController.getDoctorWisePatients);
router.get('/get-patient-report/:doctorId/:patientId', authentication.isDoctorOrPatient, UserController.getPatientReportByDoctor);
router.get('/get-doctor-report/:doctorId', authentication.isDoctor, UserController.getDoctorReport);

router.post('/consult',
  authentication.isDoctor,
  joiValidator(doctorSchema.consultPatient, JOI.property.body),
  UserController.consultPatient);

router.patch('/consult/:consultId',
  authentication.isDoctor,
  joiValidator(doctorSchema.consultPatient, JOI.property.body),
  UserController.updateConsultation);
```



```
const { UserModel } = require('../models/users');
const { HistoryModel } = require('../models/history');
const { userType } = require('../config/constants');
const response = require('../helpers/response');

module.exports = {

  async createDoctor(req, res) {
    try {
      const payload = req.body;
      const user = await UserModel.findOne({ email: payload.email }).lean();
      payload.userType = userType.doctor;
      payload.password = bcrypt.hashSync(payload.password, Number(process.env.SALT_ROUND));

      if (user) {
        return res.status(200).send(response.success( message: 'Doctor already create with this email', data: {}, sucess: false));
      }

      const doctor = await UserModel.create(payload);
      return res.status(200).send(response.success( message: 'New doctor created', doctor));
    } catch (e) {
      console.log(e);
      return res.status(500).send(response.error( message: 'An error occur', error: `${e.message}`));
    }
  },
```


Project
doctors.js
users.js
app.js
hospital-management-system
config
controllers
features
helpers
middleware
models
node_modules library root
routes
auth.js
doctors.js
index.js
patients.js
users.js
schema
uploads
utils
.editorconfig
.env
async getDoctorReport(req, res) {
try {
const { doctorId } = req.params;
const totalConsult = await HistoryModel.find({ doctorId }).countDocuments();
const totalConsultPatient = await HistoryModel.aggregate([
{ $match: { doctorId: ObjectId(doctorId) } },
{ $group: { _id: '$patientId', count: { $sum: 1 } } },
]);
const endDate = new Date();
const startDate = new Date();
startDate.setDate(startDate.getDate() - 5);
const recentConsult = await HistoryModel.aggregate([
{ $match: { doctorId: ObjectId(doctorId), date: { $gte: startDate, $lte: endDate } } },
{
$lookup: {
from: 'users',
localField: 'patientId',
foreignField: '_id',
as: 'patient',
},
},
{ $unwind: '$patient' },
{
$project: {
'patient.password': 0, 'patient.createdAt': 0, 'patient.updatedAt': 0, createdAt: 0,
exports

Testing:

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product.
It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner.
There are various types of test. Each test type addresses a specific testing requirement.

4.3 System Setup

This is a web based application system. For setup this system, need a server where the application and run from that server.

What is server:

A server is a computer equipped with specific programs and/or hardware that enables it to offer services to other computers (clients) on its network.

There are different types and capabilities of servers. Think about transportation. We can think of transportation as anything that can move something or someone from one location to the other.

A bicycle can move one person, a car can move four people, a bus can move 50 people, and a plane can move 500 people.

They are all modes of transport, but each has a different capacity. The same applies to servers.

Types of Servers

A computer may need many services to work. As such, there are different types of servers available.

File Server

A **file server** is a server that contains files which are made accessible to other clients on the network. A file server has the sole responsibility for storing and managing a set of files, which are made accessible to other computers.

These files are shared among clients in the network by allowing access without having to physically transfer the accessed files to their local systems.

Print Server

A **print server** is a server which has a dedicated printer connected to it which is accessible by other clients through it on the same network. Other clients on the network can print work to this printer through this print server.

Web Server

A **web server** is a server equipped with HTTP (Hypertext Transfer Protocol) that serves web pages in response to requests submitted by clients. For example, if you type www.ismellgood.com/homepage on your browser as a client, you are in effect requesting a web page stored on a server with a domain named ismellgood.com, called a homepage.

In response to your request, the respective web server locates the homepage page in its system and displays it to you. If you erroneously type www.ismellgood.com/homewage, the server will return an error message saying - web page not found! Well, that's familiar!

Application Server

An **application server** stores and manages all applications between an organization's users and its databases or backend business applications. If you've visited a bank to withdraw money, then you've accessed the bank's application server through the services of the attending teller.

The teller's machine through the banking application accesses the bank's application server to retrieve your bank account details and facilitate your transaction.

After selecting the server, we need a domain for this system.
After buying the domain we can host our application into the server.

5. Constraints

Constraints	Meanings
DB	Database
SDLC	Software Development Life Cycle
OOP	Object Oriented Programming
DFD	Data Flow Diagram

6. Conclusion

This was an excellent project to work on and we have learned a lot of things to complete this project. A lot of new idea and several problems that may happen to a system like this is quite clear for me now. We are not going to claim that this program offers you 100% functionality.

It has mind blowing features but while working we have figured out how we can make this more beneficial for us. we have enjoyed this project and we want to keep working on it to give it a better shape.

We believe that we can do make this different in a way that people will love the final output. Our creativity makes us different from every other species on this planet. We build tools and machines to make our lives easy and more comfortable. This has begun at a very early age and since then we have never stopped to build or design new things.

Future works:

- We will build mobile application for this system.
- Will implement lot more features like ambulance services
- Implement doctors quality control system.
- Implement doctor schedule system
- Cash and billing
- Pharmacy and medicine service
- Payroll management etc.