BOOK A DOCTOR USING MERN

TEAM MEMBERS

REGISTER NO.	**NAME**	**NM ID**	**EMAIL ID**
112821104015	DEEPAK RAJ N	A1B5255B51A30F68E274B94B254EE108	deepakraj12203@gmail.com
112821104017	DIVYABHARATHY G	99DC523077CB611A1844D3600F001ECB	divyabharathy0206@gmail.com
112821104028	HARINI R	43C547EC1780D6F85504737B22FEC1E1	rajasekerharini@gmail.com
112821104033	HEMNATH J	D9957099DD6AA58BE5E4E2126A414D57	hemnathhem88@gmail.com

Introduction of the Project Book A Doctor Using MERN:

The "Doctor Appointment System" has been developed to override the problems prevailing in the practicing manual system. This software is supported to eliminate and in some cases reduce the hardships faced by this existing system. Moreover this system is designed for the particular need of the company to carry out operations in a smooth and effective manner.

The application is reduced as much as possible to avoid errors while entering the data. It also provides error message while entering invalid data. No formal knowledge is needed for the user to use this system. Thus by this all it proves it is user-friendly. Doctor Appointment System, as described above, can lead to error free, secure, reliable and fast management system. It can assist the user to concentrate on their other activities rather to concentrate on the record keeping. Thus it will help organization in better utilization of resources.

Every organization, whether big or small, has challenges to overcome and managing the information of Appointment, Doctor, Booking, Doctor Fees, Doctor Schedule. Every Doctor Appointment System has different Doctor needs, therefore we design exclusive employee management systems that are adapted to your managerial requirements. This is designed to assist in strategic planning, and will help you ensure that your organization is equipped with the right level of information and details for your future goals. Also, for those busy executive who are always on the go, our systems come with remote access features, which will allow you to manage your workforce anytime, at all times. These systems will ultimately allow you to better manage resources.

Abstract of the Project Book A Doctor Using MERN:

The purpose of Doctor Appointment System is to automate the existing manual system by the help of computerized equipments and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. The required software and hardware are easily available and easy to work with.

Doctor Appointment System, as described above, can lead to error free, secure, reliable and fast management system. It can assist the user to concentrate on their other activities rather to concentrate on the record keeping. Thus it will help organization in better utilization of resources. The organization can maintain computerized records without redundant entries. That means that one need not be distracted by information that is not relevant, while being able to reach the information.

The aim is to automate its existing manual system by the help of computerized equipments and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. Basically the project describes how to manage for good performance and better services for the clients.

Objective of Project on Book A Doctor Using MERN:

The main objective of the Project on Doctor Appointment System is to manage the details of Doctor, Appointment, Patient, Booking, Doctor Schedule. It manages all the information about Doctor, Doctor Fees, Doctor Schedule, Doctor. The project is totally built at administrative end and thus only the administrator is guaranteed the access. The purpose of the project is to build an application program to reduce the manual work for managing the Doctor, Appointment, Doctor Fees, Patient. It tracks all the details about the Patient, Booking, Doctor Schedule.

Functionalities provided by Book A Doctor Using MERN are as follows:

- Provides the searching facilities based on various factors. Such as Doctor, Patient, Booking, Doctor Schedule
- Doctor Appointment System also manage the Doctor Fees details online for Booking details, Doctor Schedule details, Doctor.
- It tracks all the information of Appointment, Doctor Fees, Booking ect
- Manage the information of Appointment
- Shows the information and description of the Doctor, Patient
- To increase efficiency of managing the Doctor, Appointment
- It deals with monitoring the information and transactions of Booking.
- Manage the information of Doctor
- Editing, adding and updating of Records is improved which results in proper resource management of Doctor data.
- Manage the information of Booking
- Integration of all records of Doctor Schedule.

Scope of the project Book A Doctor Using MERN :

It may help collecting perfect management in details. In a very short time, the collection will be obvious, simple and sensible. It will help a person to know the management of passed year perfectly and vividly. It also helps in current all works relative to Doctor Appointment System. It will be also reduced the cost of collecting the management & collection procedure will go on smoothly.

Our project aims at Business process automation, i.e. we have tried to computerize various processes of Doctor Appointment System.

- In computer system the person has to fill the various forms & number of copies of the forms can be easily generated at a time.
- In computer system, it is not necessary to create the manifest but we can directly print it, which saves our time.
- To assist the staff in capturing the effort spent on their respective working areas.
- To utilize resources in an efficient manner by increasing their productivity through automation.
- The system generates types of information that can be used for various purposes.
- It satisfy the user requirement
- Be easy to understand by the user and operator
- Be easy to operate
- Have a good user interface
- Be expandable
- Delivered on schedule within the budget.

Reports of Book A Doctor Using MERN:

- It generates the report on Doctor, Appointment, Doctor Fees
- Provide filter reports on Patient, Booking, Doctor Schedule
- You can easily export PDF for the Doctor, Doctor Fees, Booking
- Application also provides excel export for Appointment, Patient, Doctor Schedule
- You can also export the report into csv format for Doctor, Appointment, Doctor Schedule

Modules of Book A Doctor Using MERN:

- Doctor Management Module: Used for managing the Doctor details.
- Doctor Schedule Module: Used for managing the details of Doctor Schedule
- Doctor Fees Module: Used for managing the details of Doctor Fee
- Appointment Management Module: Used for managing the information and details of the Appointment.
- Patient Module: Used for managing the Patient details
- Booking Module: Used for managing the Booking informations.
- Login Module: Used for managing the login details
- Users Module: Used for managing the users of the system

Input Data and Validation of Project on Book A Doctor Using MERN:

All the fields such as Doctor, Patient, Doctor Schedule are validated and does not take invalid values

Each form for Doctor, Appointment, Doctor Fees can not accept blank value fields

Avoiding errors in data

Controlling amount of input

Integration of all the modules/forms in the system.

Preparation of the test cases.

Preparation of the possible test data with all the validation checks.

Actual testing done manually.

Recording of all the reproduced errors.

Modifications done for the errors found during testing.

Prepared the test result scripts after rectification of the errors.

Functionality of the entire module/forms.

Validations for user input.

Checking of the Coding standards to be maintained during coding

Testing the module with all the possible test data.

Testing of the functionality involving all type of calculations etc.

Commenting standard in the source files.

The software quality plan we will use the following SQA Strategy:

- In the first step, we will select the test factors and rank them. The selected test factors such as reliability, maintainability, portability or etc, will be placed in the matrix according to their ranks.
- The second step is for identifying the phases of the development process. The phase should be recorded in the matrix.

- The third step is that identifying the business risks of the software deliverables.
- The risks will be ranked into three ranks such as high, medium and low.

<u>Features of the project Book A Doctor Using MERN:</u>

- Product and Component based
- Creating & Changing Issues at ease
- Query Issue List to any depth
- Reporting & Charting in more comprehensive way
- User Accounts to control the access and maintain security
- Simple Status & Resolutions
- Multi-level Priorities & Severities
- Targets & Milestones for guiding the programmers
- Attachments & Additional Comments for more information
- Robust database back-end
- Various level of reports available with a lot of filter criteria's
- It contain better storage capacity.
- Accuracy in work.
- Easy & fast retrieval of information.
- Well designed reports.
- Decrease the load of the person involve in existing manual system.
- Access of any information individually.
- Work becomes very speedy.
- Easy to update information

Software Requirement Specification:

The Software Requirements Specification is produced at the culmination of the analysis task. The function and performance allocated to software as part of system engineering are refined by establishing a complete information description, a detailed functional and behavioral description, an indication of performance requirements and design constraints, appropriate validation criteria, and other data pertinent to requirements.

The proposed system has the following requirements:

- System needs store information about new entry of Doctor.
- System needs to help the internal staff to keep information of Appointment and
- find them as per various queries.
- System need to maintain quantity record.
- System need to keep the record of Patient.
- System need to update and delete the record.
- System also needs a search area.
- It also needs a security system to prevent data.

Identification of need:

The old manual system was suffering from a series of drawbacks. Since whole of the system was to be maintained with hands the process of keeping, maintaining and retrieving the information was very tedious and lengthy. The records were never used to be in a systematic order. there used to be lots of difficulties in associating any particular transaction with a particular context. If any information was to be found it was required to go through the different registers, documents there would never exist anything like report generation. There would always be unnecessary consumption of time while entering records and retrieving records. One more problem was that it was very difficult to find errors while entering the records. Once the records were entered it was very difficult to update these records.

The reason behind it is that there is lot of information to be maintained and have to be kept in mind while running the business For this reason we have provided features Present system is partially automated (computerized), actually existing system is quite laborious as one has to enter same information at three different places.

Following points should be well considered:

- Documents and reports that must be provided by the new system: there can also be few reports, which can help management in decision-making and cost controlling, but since these reports do not get required attention, such kind of reports and information were also identified and given required attention.
- Details of the information needed for each document and report.
- The required frequency and distribution for each document.
- Probable sources of information for each document and report.
- With the implementation of computerized system, the task of keeping records in an organized manner will be solved. The greatest of all is the retrieval of information, which will be at the click of the mouse. So the proposed system helps in saving the time in different operations and making information flow easy giving valuable reports.

Feasibility Study:

After doing the project Doctor Appointment System, study and analyzing all the existing or required functionalities of the system, the next task is to do the feasibility study for the project. All projects are feasible given unlimited resources and infinite time.

Feasibility study includes consideration of all the possible ways to provide a solution to the given problem. The proposed solution should satisfy all the user requirements and should be flexible enough so that future changes can be easily done based on the future upcoming requirements.

A. Economical Feasibility

This is a very important aspect to be considered while developing a project. We decided the technology based on minimum possible cost factor.

- All hardware and software cost has to be done by the organization.
- Overall we have estimated that the benefits the organization is going to receive from the proposed system will surely overcome the initial costs and the later on running cost for system.

B. Technical Feasibility

This included the study of function, performance and constraints that may affect the ability to achieve an acceptable system. For this feasibility study, we studied complete functionality to be provided in the system, as described in the System Requirement Specification (SRS), and checked if everything was possible using different type of frontend and backend platform.

C. Operational Feasibility

No doubt the proposed system is fully GUI based that is very user friendly and all inputs to be taken all self-explanatory even to a layman. Besides, a proper training has been conducted to let know the essence of the system to the users so that they feel comfortable with new system. As far our study is concerned the clients are comfortable and happy as the system has cut down their loads and doing.

<u>System Design of Doctor Appointment System:</u>

In this phase, a logical system is built which fulfils the given requirements. Design phase of software development deals with transforming the clients 's requirements into a logically working system. Normally, design is performed in the following in the following two steps:

1. Primary Design Phase:

In this phase, the system is designed at block level. The blocks are created on the basis of analysis done in the problem identification phase. Different blocks are created for different functions emphasis is put on minimising the information flow between blocks. Thus, all activities which require more interaction are kept in one block.

2. Secondary Design Phase:

In the secondary phase the detailed design of every block is performed.

<u>The general tasks involved in the design process are the following:</u>

- Design various blocks for overall system processes.
- Design smaller, compact and workable modules in each block.
- Design various database structures.
- Specify details of programs to achieve desired functionality.
- Design the form of inputs, and outputs of the system.
- Perform documentation of the design.
- System reviews.

User Interface Design:

User Interface Design is concerned with the dialogue between a user and the computer. It is concerned with everything from starting the system or logging into the system to the eventually presentation of desired inputs and outputs. The overall flow of screens and messages is called a dialogue.

The following steps are various guidelines for User Interface Design:

1. The system user should always be aware of what to do next.

2. The screen should be formatted so that various types of information, instructions and messages always appear in the same general display area.

3. Message, instructions or information should be displayed long enough to allow the system user to read them.

4. Use display attributes sparingly.

5. Default values for fields and answers to be entered by the user should be specified.

6. A user should not be allowed to proceed without correcting an error.

7. The system user should never get an operating system message or fatal error.

Preliminary Product Description:

The first step in the system development life cycle is the preliminary investigation to determine the feasibility of the system. The purpose of the preliminary investigation is to evaluate project requests. It is not a design study nor does it include the collection of details to describe the business system in all respect. Rather, it is the collecting of information that helps committee members to evaluate the merits of the project request and make an informed judgment about the feasibility of the proposed project.

Analysts working on the preliminary investigation should accomplish the following objectives:

- Clarify and understand the project request
- Determine the size of the project.
- Assess costs and benefits of alternative approaches.
- Determine the technical and operational feasibility of alternative approaches.
- Report the findings to management, with recommendations outlining the acceptance or rejection of the proposal.

- **Benefit to Organization**

  The organization will obviously be able to gain benefits such as savings in operating cost, reduction in paperwork, better utilization of human resources and more presentable image increasing goodwill.

- **The Initial Cost**

  The initial cost of setting up the system will include the cost of hardware software (OS, add-on software, utilities) & labour (setup & maintenance). The same has to bear by the organization.
- **Running Cost**

Besides, the initial cost the long term cost will include the running cost for the system including the AMC, stationary charges, cost for human resources, cost for update/renewal of various related software.

- **Need for Training**

The users along with the administrator need to be trained at the time of implementation of the system for smooth running of the system. The client will provide the training site.

We talked to the management people who were managing a the financial issues of the center, the staff who were keeping the records in lots of registers and the reporting manager regarding their existing system, their requirements and their expectations from the new proposed system. Then, we did the system study of the entire system based on their requirements and the additional features they wanted to incorporate in this system.

Reliable, accurate and secure data was also considered to be a complex task without this proposed system. Because there was no such record for keeping track of all the activities, which was done by the Doctor Appointment System on the daily basis.

The new system proposed and then developed by me will ease the task of the organization in consideration. It will be helpful in generating the required reports by the staff, which will help them to track their progress and services.

Thus, it will ease the task of Management to a great extent as all the major activities to be performed, are computerized through this system.

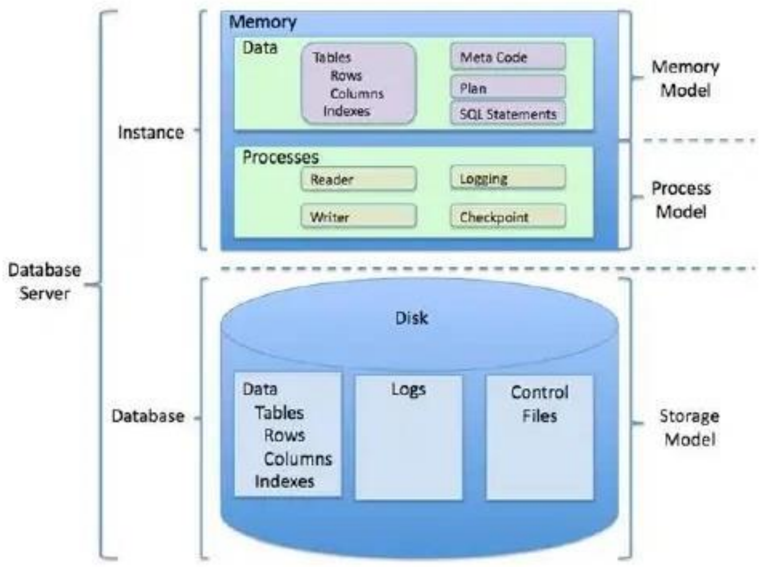
Project Category:

Relational Database Management System (RDBMS): This is an RDBMS based project which is currently using MySQL for all the transaction statements. MySQL is an opensource RDBMS System.

Brief Introduction about RDBSM:

A relational database management system (RDBMS) is a database management system (DBMS) that is based on the relational model as invented by E. F. Codd, of IBM's San Jose Research Laboratory. Many popular databases currently in use are based on the relational database model.

RDBMSs have become a predominant choice for the storage of information in new databases used for financial records, manufacturing and logistical information, personnel data, and much more since the 1980s. Relational databases have often replaced legacy hierarchical databases and network databases because they are easier to understand and use. However, relational databases have been challenged by object databases, which were introduced in an attempt to address the object-relational impedance mismatch in relational database, and XML databases.



Implementation Methodology:

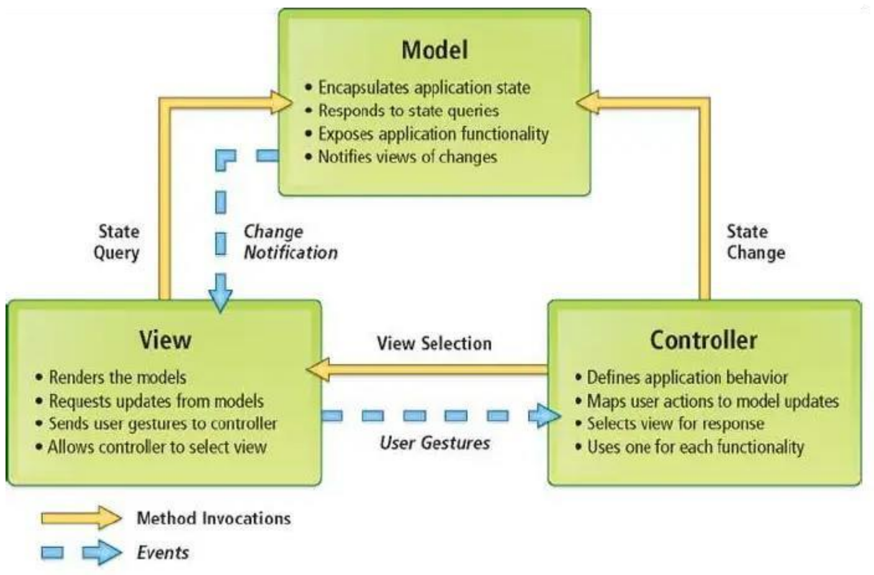
Model View Controller or MVC as it is popularly called, is a software design pattern for developing web applications. A Model View Controller pattern is made up of the following three parts:

- Model - The lowest level of the pattern which is responsible for maintaining data.
- View - This is responsible for displaying all org portion of the data to the user.
- Controller - Software Code that controls the interactions between the Model and View.

MVC is popular as it isolates the application logic from the user interface layer and supports separation of concerns. Here the Controller receives all requests for the application and then works with the Model to prepare any data needed by the View. The View then uses the data prepared by the Controller to generate a final presentable response. The MVC abstraction can be graphically represented as follows.

MVC (Model View Controller Flow) Diagram

DATA FLOW DIAGRAMS


Model
• Encapsulates application state
• Responds to state queries
• Exposes application functionality
• Notifies views of changes
State Query
Change Notification
State Change
View
• Renders the models
• Requests updates from models
• Sends user gestures to controller
• Allows controller to select view
View Selection
User Gestures
Controller
• Defines application behavior
• Maps user actions to model updates
• Selects view for response
• Uses one for each functionality
Method Invocations
Events

Project Planning:

Software project plan can be viewed as the following:

1) Within the organization: How the project is to be implemented? What are various constraints (time, cost, staff)? What is market strategy?

2) With respect to the customer: Weekly or timely meetings with the customer with presentation on status reports. Customers feedback is also taken and further modification and developments are done. Project milestones and deliverables are also presented to the customer.

For a successful software project, the following steps can be followed:

- Select a project
  - Identifying project's aims and objectives
  - Understanding requirements and specification
  - Methods of analysis, design and implementation
  - Testing techniques
  - Documentation
- Project milestones and deliverables
- Budget allocation
  - Exceeding limits within control
- Project Estimates
  - Cost
  - Time
  - Size of code
  - Duration
- Resource Allocation
  - Hardware
  - Software
  - Previous relevant project information
  - Digital Library
- Risk Management
  - Risk avoidance
  - Risk detection

Project Profile

There has been continuous effort to develop tools, which can ease the process of software development. But, with the evolving trend of different programming paradigms today's software developers are really challenged to deal with the changing technology. Among other issues, software re-engineering is being regarded as an important process in the software development industry. One of the major tasks here is to understand software systems that are already developed and to transform them to a different software environment. Generally, this requires a lot of manual effort in going through a program that might have been developed by another programmer. This project makes a novel attempt to address the issued of program analysis and generation of diagrams, which can depict the structure of a program in a better way. Today, UML is being considered as an industrial standard for software engineering design process. It essential provides several diagramming tools that can express different aspects/ characteristics of program such as

Use cases: Elicit requirement from users in meaningful chunks. Construction planning is built around delivering some use cases in each interaction basis for system testing.

Class diagrams: shows static structure of concepts types and class. Concepts how users think about the world: type shows interface of software components; classes shows implementation of software component

Interaction diagrams: shows how several objects collaborate in single use case.

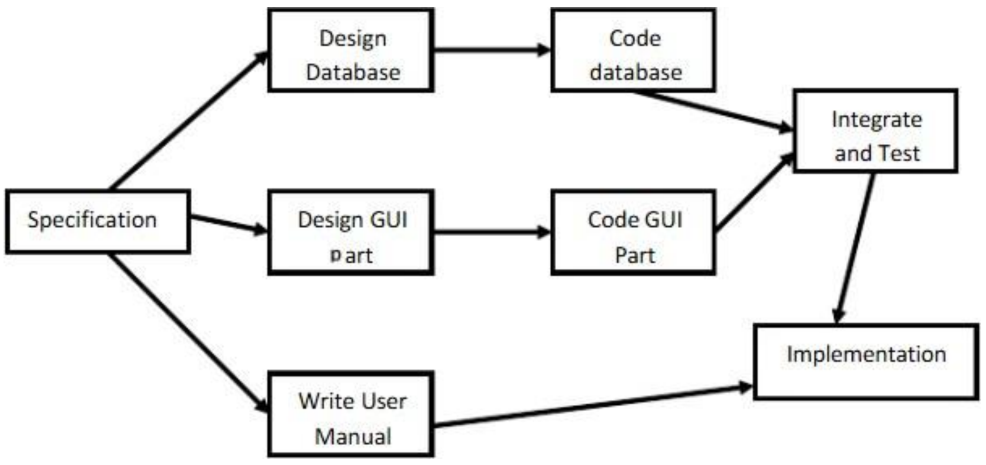
Package diagram: show group of classes and dependencies among them.

State diagram: show how single object behaves across many use cases

Activity diagram: shows behaviour with control structure. Can show many objects over many uses, many object in single use case, or implementations methods encourage parallel behaviour, etc.

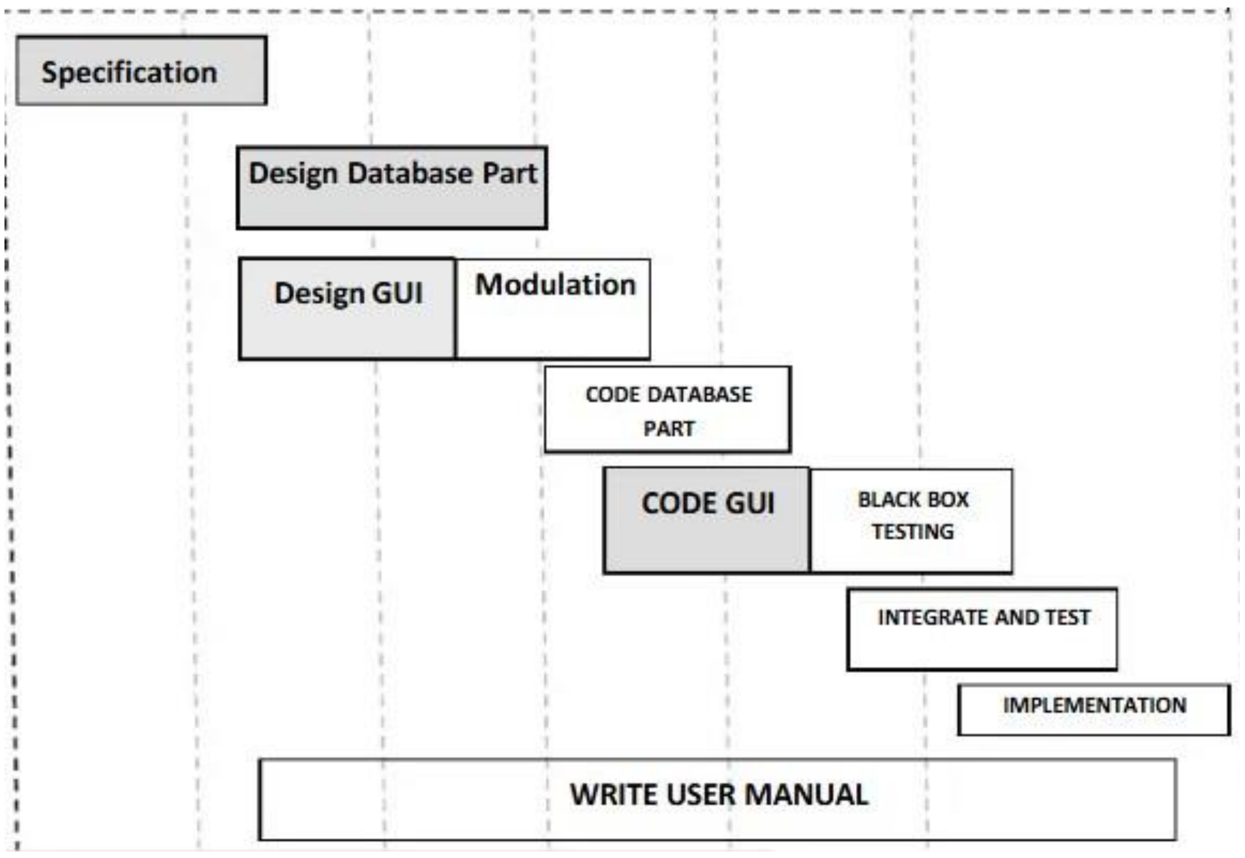
PERT CHART (Program Evaluation Review Technique)

PERT chart is organized for events, activities or tasks. It is a scheduling device that shows graphically the order of the tasks to be performed. It enables the calculation of the critical path. The time and cost associated along a path is calculated and the path requires the greatest amount of elapsed time in critical path.



GANTT CHART

It is also known as Bar chart is used exclusively for scheduling purpose. It is a project controlling technique. It is used for scheduling. Budgeting and resourcing planning. A Gantt is a bar chart with each bar representing activity. The bars are drawn against a time line. The length of time planned for the activity. The Gantt chart in the figure shows the Gray parts is slack time that is the latest by which a task has been finished.



<u>THE STEPS IN THE SOFTWARE TESTING</u>

The steps involved during Unit testing are as follows:

a. Preparation of the test cases.

b. Preparation of the possible test data with all the validation checks.

c. Complete code review of the module.

d. Actual testing done manually.

e. Modifications done for the errors found during testing.

f. Prepared the test result scripts.

The unit testing done included the testing of the following items:

1. Functionality of the entire module/forms.
2. Validations for user input
3. Checking of the Coding standards to be maintained during coding.
4. Testing the module with all the possible test data.
5. Testing of the functionality involving all type of calculations etc.
6. Commenting standard in the source files.

After completing the Unit testing of all the modules, the whole system is integrated with all its dependencies in that module. While System Integration, We integrated the modules one by one and tested the system at each step. This helped in reduction of errors at the time of the system testing.

The steps involved during System testing are as follows:

- Integration of all the modules/forms in the system.
- Preparation of the test cases.
- Preparation of the possible test data with all the validation checks
- Actual testing done manually
- Recording of all the reproduced errors.
- Modifications done for the errors found during testing.
- Prepared the test result scripts after rectification of the errors.

The System Testing done included the testing of the following items:

1. Functionality of the entire system as a whole.
2. User Interface of the system.

3. Testing the dependent modules together with all the possible test data scripts
4. Verification and Validation testing.
5. Testing the reports with all its functionality.

After the completion of system testing, the next following phase was the Acceptance Testing. Clients at their end did this and accepted the system with appreciation. Thus, we reached the final phase of the project delivery.

<u>There are other six tests, which fall under special category. They are described below:</u>

- Peak Load Test: It determines whether the system will handle the volume of activities that occur when the system is at the peak of its processing demand. For example, test the system by activating all terminals at the same time
- Storage Testing: It determines the capacity of the system to store transaction data on a disk or in other files.
- Performance Time Testing: it determines the length of time system used by the system to process transaction data. This test is conducted prior to implementation to determine how long it takes to get a response to an inquiry, make a backup copy of a file, or send a transmission and get a response.
- Recovery Testing: This testing determines the ability of user to recover data or re-start system after failure. For example, load backup copy of data and resume processing without data or integrity loss
- Procedure Tasting: it determines the clarity of documentation on operation and uses of system by having users do exactly what manuals request. For example, powering down system at the end of week or responding to paper-out light on printer.
- Human Factors Testing: It determines how users will use the system when processing data or preparing reports.

System Analysis:

System analysis is a process of gathering and interpreting facts, diagnosing problems and the information about the Doctor Appointment System to recommend Improvements on the system. It is a problem solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system is studied to the minutest detail and analyzed. The system analyst plays the role of the interrogator and dwells deep into the working of the present system. The system is viewed as a whole and the input to the system are identified. The outputs from the organizations are traced to the various processes. System analysis is concerned with becoming aware of the problem, identifying the relevant and decisional variables, analyzing and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action. A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now the existing system is subjected to close study and problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal. Preliminary study is the process of gathering and interpreting facts, using the information for further studies on the system. Preliminary study is problem solving activity that requires intensive communication between the system users and system developers. It does various feasibility studies. In these studies rough figure of the system activities can be obtained, from which the decision about the strategies to be followed for effective system study and analysis can be taken.

Existing System of Book A Doctor Using MERN:

In the existing system the exams are done only manually but in proposed system we have to computerize the exams using this application.

- Lack of security of data.
- More man power.
- Time consuming.
- Consumes large volume of pare work
- Needs manual calculations
- No direct role for the higher officials

Proposed System of Book A Doctor Using MERN:

The aim of proposed system is to develop a system of improved facilities. The proposed system can overcome all the limitations of the existing system. The system provides proper security and reduces the manual work.

- Security of data
- Ensure data accuracy's
- Proper control of the higher officials.
- Minimize manual data entry.
- Minimum time needed for the various processing.
- Greater efficiency.
- Better service.
- User friendliness and interactive.
- Minimum time required.

CODING:

1. Database Design

1.1Doctor Schema

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

const doctorSchema = new mongoose.Schema({
  name: String,
  specialization: String,
  availability: Array  // e.g., ['Monday', 'Wednesday', 'Friday']
});

module.exports = mongoose.model('Doctor', doctorSchema);
```

1.2Appointment Schema

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

const appointmentSchema = new mongoose.Schema({
  doctorId: { type: mongoose.Schema.Types.ObjectId, ref: 'Doctor' },
  patientName: String,
  date: Date,
  timeSlot: String
});

module.exports = mongoose.model('Appointment', appointmentSchema);
```

2. Frontend Design (React)

The frontend is designed using React.js with the following components:

2.1 Doctor List Component

- Fetches the list of doctors and displays them to the user.
- Each doctor has a link to book an appointment.

```
import React, { useEffect, useState } from 'react';
import axios from 'axios';
import { Link } from 'react-router-dom';

const DoctorList = () => {
  const [doctors, setDoctors] = useState([]);

  useEffect(() => {
    axios.get('http://localhost:5000/doctors')
      .then(response => setDoctors(response.data))
      .catch(error => console.log(error));
  }, []);

  return (
    <div>
      <h1>Doctors</h1>
      <ul>
        {doctors.map(doctor => (
          <li key={doctor._id}>
            {doctor.name} ({doctor.specialization}) - <Link
to={`/book/${doctor._id}`}>Book Appointment</Link>
          </li>
        ))}
      </ul>
    </div>
  );
};

export default DoctorList;
```

2.2Book Appointment Component

- Allows the user to book an appointment with a selected doctor.

```
import React, { useState } from 'react';
import axios from 'axios';

const BookAppointment = ({ match }) => {
  const doctorId = match.params.doctorId;
  const [patientName, setPatientName] = useState('');
  const [date, setDate] = useState('');
  const [timeSlot, setTimeSlot] = useState('');

  const handleSubmit = () => {
    axios.post('http://localhost:5000/appointments', {
    doctorId, patientName, date, timeSlot
    }).then(() => alert('Appointment booked!'))
      .catch(error => console.log(error));
  };

  return (
    <div>
      <h2>Book Appointment</h2>
      <input type="text" placeholder="Patient Name" onChange={e =>
setPatientName(e.target.value)} />
      <input type="date" onChange={e => setDate(e.target.value)} />
      <input type="text" placeholder="Time Slot" onChange={e =>
setTimeSlot(e.target.value)} />
      <button   onClick={handleSubmit}>Book</button>
    </div>
  );
};

export default BookAppointment;
```

3. Backend Design (Express + Node.js)

3.1 Server Setup

The server is set up using Express.js and provides RESTful API endpoints to manage doctors and appointments.

3.2 Routes

- **GET /doctors:** Fetch the list of available doctors.
- **POST /appointments:** Book an appointment.
- **GET /appointments/:doctorId:** View all appointments for a doctor.

```
const express = require('express');
const mongoose = require('mongoose');
const Doctor = require('./models/doctor');
const Appointment = require('./models/appointment');
const bodyParser = require('body-parser');
const cors = require('cors');

const app = express();
```

```
app.use(bodyParser.json());

app.use(cors());


// MongoDB Connection

mongoose.connect('mongodb://localhost:27017/doctor-booking', {
useNewUrlParser: true, useUnifiedTopology: true });


// Get list of doctors

app.get('/doctors', async (req, res) => {

  const doctors = await Doctor.find();

  res.json(doctors);

});


// Book an appointment

app.post('/appointments', async (req, res) => {

  const { doctorId, patientName, date, timeSlot } = req.body;

  const newAppointment = new Appointment({ doctorId, patientName, date,
timeSlot });

  await newAppointment.save();

  res.json({ message: 'Appointment booked successfully!' });

});


// View appointments for a doctor

app.get('/appointments/:doctorId', async (req, res) => {

  const appointments = await Appointment.find({ doctorId:
req.params.doctorId });

  res.json(appointments);
});


app.listen(5000, () => console.log('Server running on port 5000'));
```

INDEX.HTML:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Doctor Booking System</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <header>
    <div class="container">
      <h1>Doctor Booking System</h1>
      <nav>
        <a href="#home">Home</a>
        <a href="#doctors">Doctors</a>
        <a href="#booking">Book Appointment</a>
      </nav>
    </div>
  </header>

  <main>
    <!-- Home Section -->
    <section id="home">
      <div class="container">
        <h2>Welcome</h2>
        <p>Find the best doctors and book an appointment with ease. We are here to help you
stay healthy!</p>
      </div>
    </section>

    <!-- Doctors Section -->
    <section id="doctors">
      <div class="container">
        <h2>Our Doctors</h2>
        <div id="doctor-list" class="grid"></div>
      </div>
    </section>
    <!-- Booking Section -->
    <section id="booking">
      <div class="container">
        <h2>Book an Appointment</h2>
        <form id="booking-form">
          <label for="name">Your Name:</label>
          <input type="text" id="name" placeholder="Enter your name" required>

          <label for="doctor">Select Doctor:</label>
          <select id="doctor" required>
            <option value="">--Choose a doctor--</option>
          </select>

          <label for="date">Date:</label>
          <input type="date" id="date" required>

          <label for="time">Time:</label>
          <input type="time" id="time" required>

          <button type="submit">Book Now</button>
        </form>
      </div>
    </section>
  </main>

  <footer>
    <div class="container">
      <p>Doctor Booking System &copy; 2024. All rights reserved.</p>
    </div>
  </footer>

  <script src="script.js"></script>
</body>
</html>
```

STYLE.CSS:

```
/* Global Styles */
body {
    margin: 0;
    font-family: 'Arial', sans-serif;
    line-height: 1.6;
    background-color: #f9f9f9;
    color: #333;
```

```css
}

.container {
  max-width: 1100px;
  margin: 0 auto;
  padding: 20px;
}

/* Header Styles */
header {
  background: #34a09b;
  color: white;
  padding: 20px 0;
  text-align: center;
}

header nav a {
  color: white;
  text-decoration: none;
  margin: 0 15px;
  font-size: 18px;
  font-weight: bold;
}

header nav a:hover {
  text-decoration: underline;
}

/* Section Styles */
section {
  padding: 40px 0;
  text-align: center;
}

section h2 {
  margin-bottom: 20px;
  font-size: 28px;
}

.grid {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
  gap: 20px;
}
```

```
.card {
  background: white;
  border: 1px solid #ddd;
  border-radius: 8px;
  box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
  padding: 20px;
  text-align: center;
}

.card h3 {
  margin-bottom: 10px;
}

.card p {
  color: #666;
}

.card:hover {
  transform: scale(1.05);
  transition: 0.3s;
}

/* Form Styles */
form {
  background: white;
  border: 1px solid #ddd;
  border-radius: 8px;
  padding: 20px;
  box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
  max-width: 400px;
  margin: 0 auto;
}

form label {
  display: block;
  margin-bottom: 5px;
  font-weight: bold;
}

form input, form select, form button {
  width: 100%;
  padding: 10px;
  margin: 10px 0;
  border: 1px solid #ddd;
  border-radius: 4px;
 }

 form button {
  background: #1faeb0;
  color: white;
  border: none;
  cursor: pointer;
  font-size: 16px;
 }

 form button:hover {
  background: #45a049;
 }

 /* Footer Styles */
 footer {
  background: #333;
  color: white;
  text-align: center;
  padding: 10px 0;
 }
```

SCRIPT.JS:

```
// Sample doctors data
const doctors = [
  { id: 1, name: "Dr. Divyabharathy", specialization: "Cardiologist" },
  { id: 2, name: "Dr. Harini", specialization: "Dermatologist" },
  { id: 3, name: "Dr. Hemnath", specialization: "Neurologist" },
  { id: 4, name: "Dr. Deepak", specialization: "Pediatrician" },
 ];

 // Populate the doctor list
 const doctorList = document.getElementById("doctor-list");
 doctors.forEach((doctor) => {
  const card = document.createElement("div");
  card.classList.add("card");
  card.innerHTML = `
   <h3>${doctor.name}</h3>
   <p>${doctor.specialization}</p>
  `;
  doctorList.appendChild(card);
 });
```

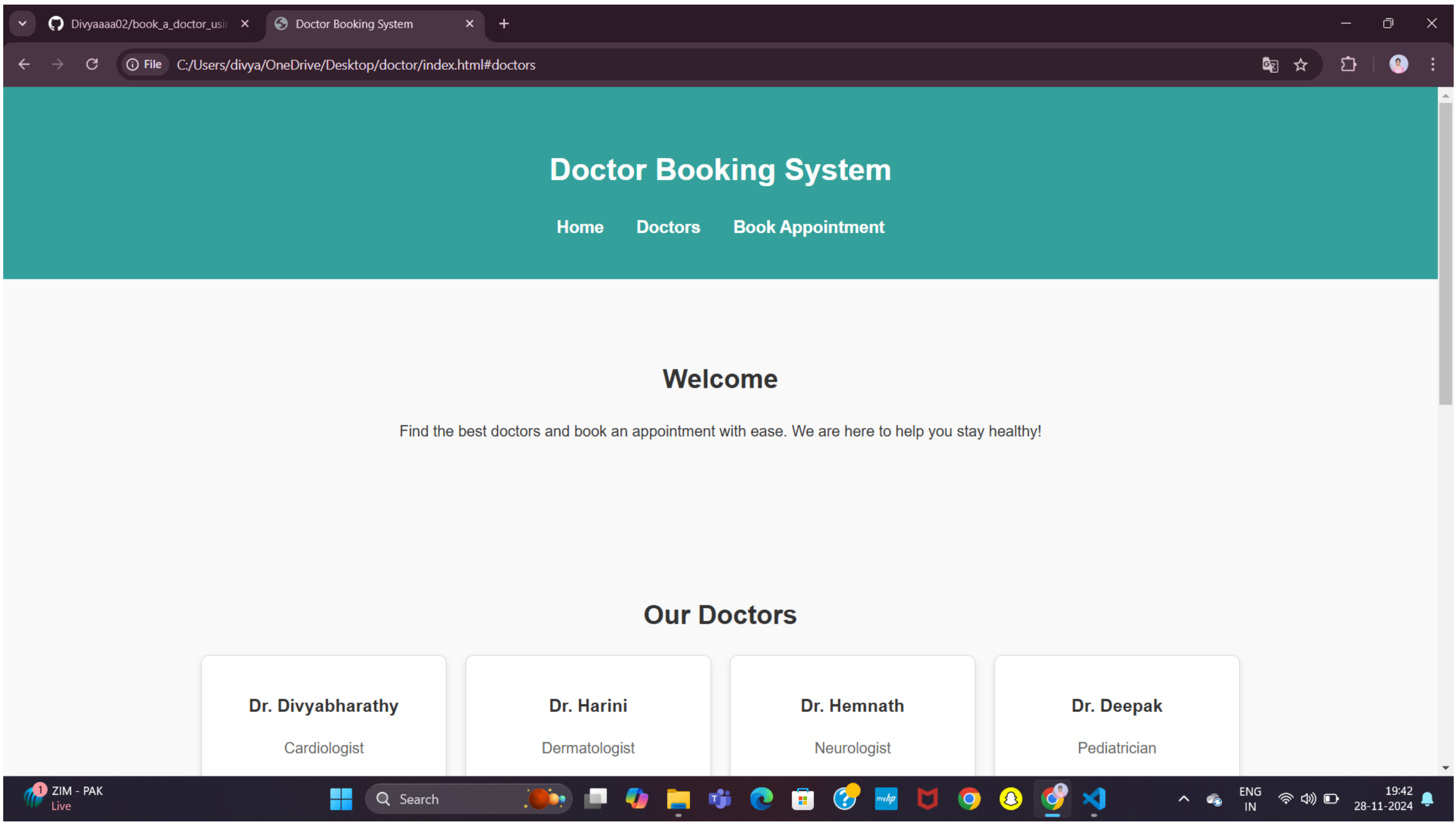
```
// Populate the doctor dropdown in the booking form
const doctorDropdown = document.getElementById("doctor");
doctors.forEach((doctor) => {
  const option = document.createElement("option");
  option.value = doctor.id;
  option.textContent = `${doctor.name} (${doctor.specialization})`;
  doctorDropdown.appendChild(option);
});

// Handle form submission
const bookingForm = document.getElementById("booking-form");
bookingForm.addEventListener("submit", (event) => {
  event.preventDefault();

  const name = document.getElementById("name").value;
  const doctorId = document.getElementById("doctor").value;
  const date = document.getElementById("date").value;
  const time = document.getElementById("time").value;

  if (name && doctorId && date && time) {
    alert(
      `Appointment confirmed for ${name} with ${
        doctors.find((doc) => doc.id === parseInt(doctorId)).name
      } on ${date} at ${time}.`
    );
    bookingForm.reset();
  } else {
    alert("Please fill in all the fields.");
  }
});
```

screenshot of the project:



Divyaaaa02/book_a_doctor_usi
Doctor Booking System
File C:/Users/divya/OneDrive/Desktop/doctor/index.html#doctors

Doctor Booking System

Home Doctors Book Appointment

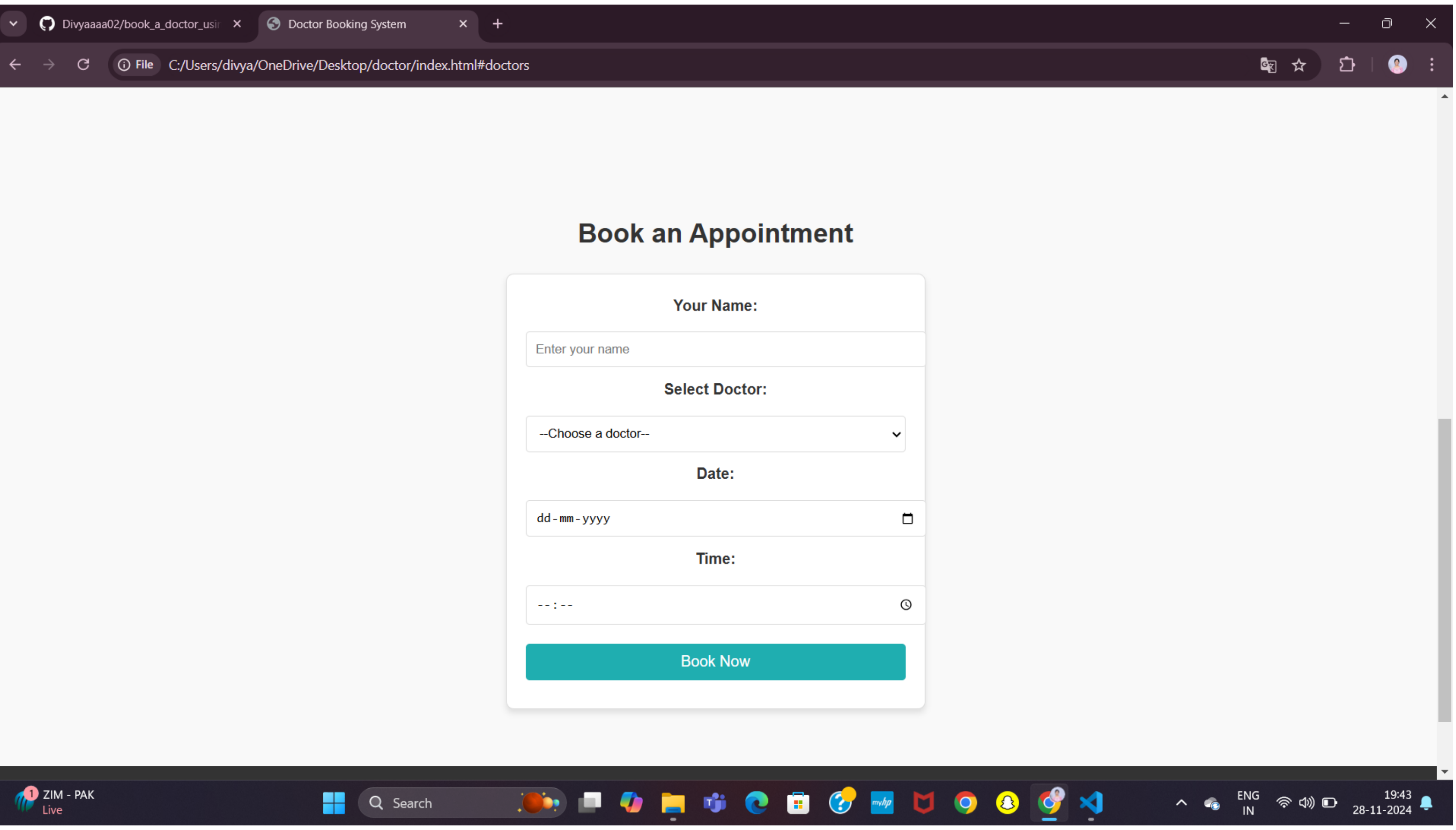
Welcome

Find the best doctors and book an appointment with ease. We are here to help you stay healthy!

Our Doctors

Dr. Divyabharathy	Dr. Harini	Dr. Hemnath	Dr. Deepak
Cardiologist	Dermatologist	Neurologist	Pediatrician

ZIM - PAK
Live
Search
ENG
IN
19:42
28-11-2024



Divyaaaa02/book_a_doctor_usi

Doctor Booking System

C:/Users/divya/OneDrive/Desktop/doctor/index.html#doctors

Book an Appointment

Your Name:

Enter your name

Select Doctor:

--Choose a doctor--

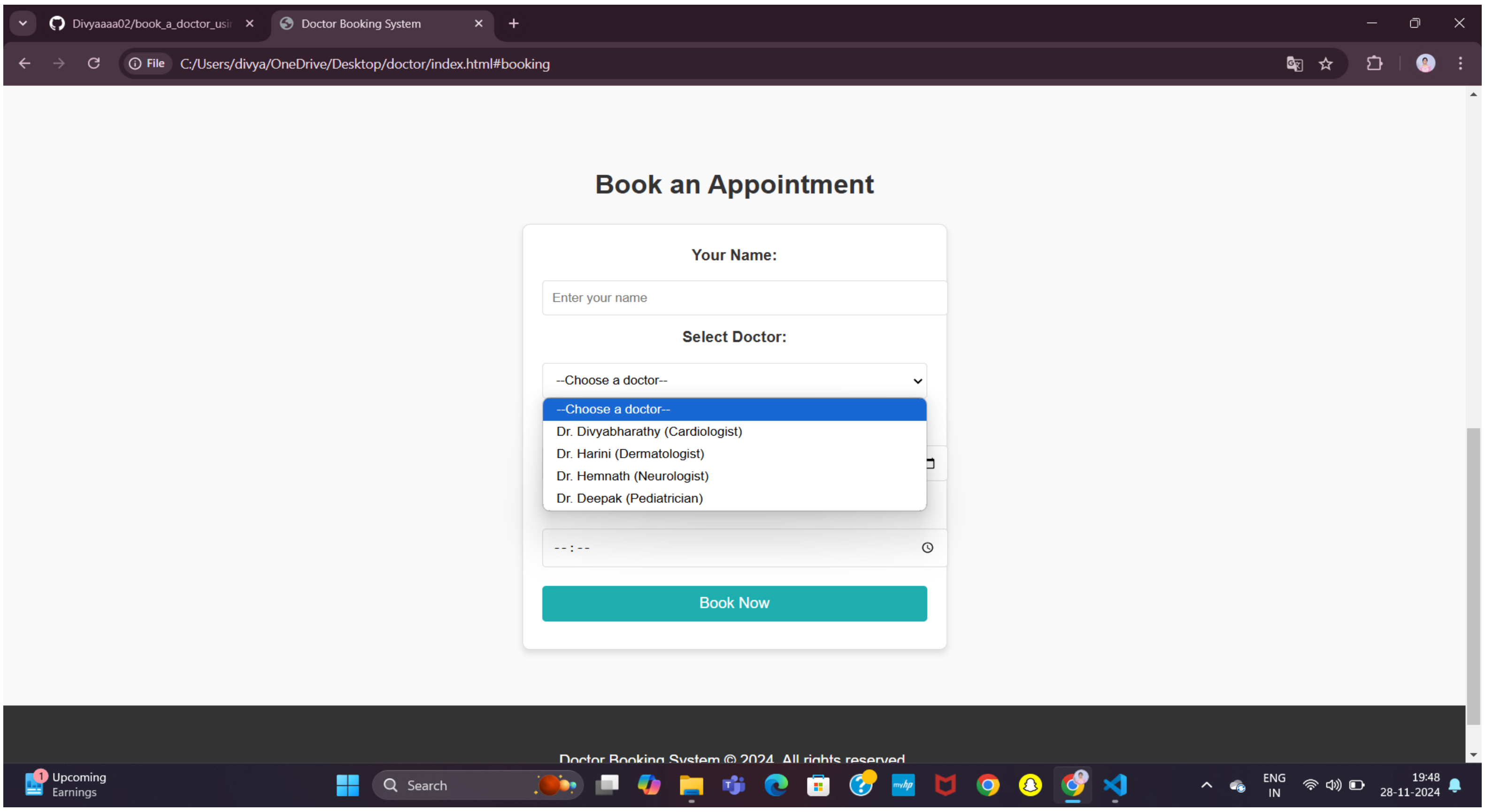
Date:

dd-mm-yyyy

Time:

--:--

Book Now


Divyaaaa02/book_a_doctor_usi
Doctor Booking System
C:/Users/divya/OneDrive/Desktop/doctor/index.html#booking

Book an Appointment

Your Name:

Enter your name

Select Doctor:

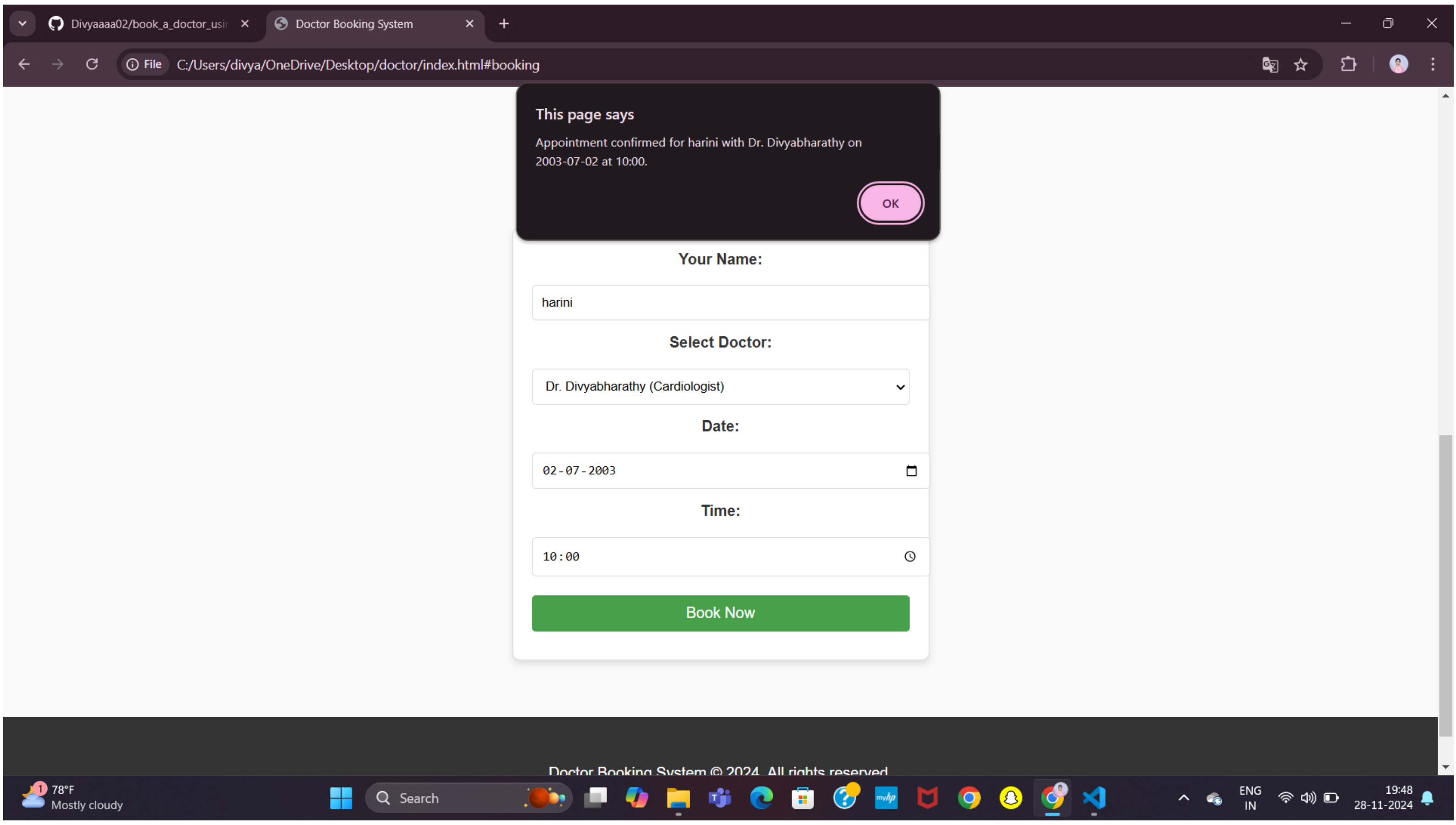
--Choose a doctor--

--Choose a doctor--
Dr. Divyabharathy (Cardiologist)
Dr. Harini (Dermatologist)
Dr. Hemnath (Neurologist)
Dr. Deepak (Pediatrician)

--:--

Book Now

Doctor Booking System © 2024. All rights reserved.



Divyaaaa02/book_a_doctor_usir
Doctor Booking System
C:/Users/divya/OneDrive/Desktop/doctor/index.html#booking

This page says

Appointment confirmed for harini with Dr. Divyabharathy on 2003-07-02 at 10:00.

OK

Your Name:
harini

Select Doctor:
Dr. Divyabharathy (Cardiologist)

Date:
02-07-2003

Time:
10:00

Book Now

Doctor Booking System © 2024. All rights reserved.

Conclusion

This Doctor Appointment Booking System provides an efficient way for patients to book appointments online. The system can be further extended with features like user authentication, notification systems, and more advanced time-slot management. The use of the MERN stack allows the application to be easily scalable and maintainable.

The application is reduced as much as possible to avoid errors while entering the data. It also provides error message while entering invalid data. No formal knowledge is needed for the user to use this system. Thus by this all it proves it is user-friendly. Doctor Appointment System, as described above, can lead to error free, secure, reliable and fast management system. It can assist the user to concentrate on their other activities rather to concentrate on the record keeping. Thus it will help organization in better utilization of resources.