**Project Structure**

book-a-doctor/

├── client/

│ ├── node\_modules/

│ ├── public/

│ ├── src/

│ │ ├── components/

│ │ ├── pages/

│ │ ├── App.js

│ │ ├── index.js

│ │ └── index.css

│ ├── package.json

│ └── package-lock.json

├── server/

│ ├── node\_modules/

│ ├── controllers/

│ ├── models/

│ ├── routes/

│ ├── app.js

│ └── package.json

└── package.json

**Server-side (Node.js, Express.js, MongoDB)**

1. Install the required dependencies:

bash

cd server

npm init -y

npm install express mongoose

1. Create an Express.js server (app.js):

javascript

const express = require('express');

const mongoose = require('mongoose');

const app = express();

*// Connect to MongoDB*

mongoose.connect('mongodb://localhost/book-a-doctor', {

useNewUrlParser: true,

useUnifiedTopology: true,

});

*// Middleware*

app.use(express.json());

*// Routes*

const doctorsRouter = require('./routes/doctors');

app.use('/api/doctors', doctorsRouter);

*// Start the server*

const PORT = process.env.PORT || 5000;

app.listen(PORT, () => console.log(`Server started on port ${PORT}`));

1. Create a model for doctors (models/Doctor.js):

javascript

const mongoose = require('mongoose');

const doctorSchema = new mongoose.Schema({

name: { type: String, required: true },

specialty: { type: String, required: true },

availability: [{ type: Date }],

});

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

1. Create a controller for handling doctor-related operations (controllers/doctorsController.js):

javascript

= require('../models/Doctor');

exports.getAllDoctors = async (req, res) => {

try {

const doctors = await Doctor.find();

res.json(doctors);

} catch (err) {

res.status(500).json({ message: err.message });

}

};

exports.bookAppointment = async (req, res) => {

try {

const { doctorId, appointmentDate } = req.body;

const doctor = await Doctor.findById(doctorId);

doctor.availability.push(appointmentDate);

await doctor.save();

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

} catch (err) {

res.status(500).json({ message: err.message });

}

};

1. Create routes for handling doctor-related endpoints (routes/doctors.js):

javascript

const express = require('express');

const router = express.Router();

const doctorsController = require('../controllers/doctorsController');

router.get('/', doctorsController.getAllDoctors);

router.post('/book-appointment', doctorsController.bookAppointment);

module.exports = router;

**Client-side (React.js)**

1. Create a new React.js project:

bash

cd client

npx create-react-app .

1. Install additional dependencies:

bash

npm install axios react-router-dom

1. Create a component for displaying the list of doctors (components/DoctorList.js):

jsx

import React, { useState, useEffect } from 'react';

import axios from 'axios';

const DoctorList = () => {

const [doctors, setDoctors] = useState([]);

useEffect(() => {

const fetchDoctors = async () => {

const response = await axios.get('/api/doctors');

setDoctors(response.data);

};

fetchDoctors();

}, []);

return (

<div>

<h2>Available Doctors</h2>

<ul>

{doctors.map((doctor) => (

<li key={doctor.\_id}>

{doctor.name} - {doctor.specialty}

</li>

))}

</ul>

</div>

);

};

export default DoctorList;

1. Create a component for booking an appointment (components/BookAppointment.js):

jsx

import React, { useState } from 'react';

import axios from 'axios';

const BookAppointment = () => {

const [doctorId, setDoctorId] = useState('');

const [appointmentDate, setAppointmentDate] = useState('');

const handleSubmit = async (e) => {

e.preventDefault();

try {

await axios.post('/api/doctors/book-appointment', {

doctorId,

appointmentDate,

});

alert('Appointment booked successfully');

} catch (err) {

console.error(err);

alert('Error booking appointment');

}

};

return (

<div>

<h2>Book an Appointment</h2>

<form onSubmit={handleSubmit}>

<label>

Doctor ID:

<input

type="text"

value={doctorId}

onChange={(e) => setDoctorId(e.target.value)}

/>

</label>

<br />

<label>

Appointment Date:

<input

type="datetime-local"

value={appointmentDate}

onChange={(e) => setAppointmentDate(e.target.value)}

/>

</label>

<br />

<button type="submit">Book Appointment</button>

</form>

</div>

);

};

export default BookAppointment;

1. Update the App.js file to render the components:

jsx

import React from 'react';

import { BrowserRouter as Router, Switch, Route, Link } from 'react-router-dom';

import DoctorList from './components/DoctorList';

import BookAppointment from './components/BookAppointment';

function App() {

return (

<Router>

<div>

<nav>

<ul>

<li>

<Link to="/">Doctors</Link>

</li>

<li>

<Link to="/book-appointment">Book Appointment</Link>

</li>

</ul>

</nav>

<Switch>

<Route path="/book-appointment">

<BookAppointment />

</Route>

<Route path="/">

<DoctorList />

</Route>

</Switch>

</div>

</Router>

);

}

export default App;

const express = require('express');

const mongoose = require('mongoose');

const app = express();

*// Connect to MongoDB*

mongoose.connect('mongodb://localhost/book-a-doctor', {

useNewUrlParser: true,

useUnifiedTopology: true,

});

*// Middleware*

app.use(express.json());

*// Routes*

const doctorsRouter = require('./routes/doctors');

app.use('/api/doctors', doctorsRouter);

*// Start the server*

const PORT = process.env.PORT || 5000;

app.listen(PORT, () => console.log(`Server started on port ${PORT}`));

server/models/Doctor.js

javascript

const mongoose = require('mongoose');

const doctorSchema = new mongoose.Schema({

name: { type: String, required: true },

specialty: { type: String, required: true },

availability: [{ type: Date }],

});

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

server/controllers/doctorsController.js

javascript

const Doctor = require('../models/Doctor');

exports.getAllDoctors = async (req, res) => {

try {

const doctors = await Doctor.find();

res.json(doctors);

} catch (err) {

res.status(500).json({ message: err.message });

}

};

exports.bookAppointment = async (req, res) => {

try {

const { doctorId, appointmentDate } = req.body;

const doctor = await Doctor.findById(doctorId);

doctor.availability.push(appointmentDate);

await doctor.save();

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

} catch (err) {

res.status(500).json({ message: err.message });

}

};

server/routes/doctors.js

javascript

const express = require('express');

const router = express.Router();

const doctorsController = require('../controllers/doctorsController');

router.get('/', doctorsController.getAllDoctors);

router.post('/book-appointment', doctorsController.bookAppointment);

module.exports = router;

**Client-side (React.js)**

client/src/components/DoctorList.js

jsx

import React, { useState, useEffect } from 'react';

import axios from 'axios';

const DoctorList = () => {

const [doctors, setDoctors] = useState([]);

useEffect(() => {

const fetchDoctors = async () => {

const response = await axios.get('/api/doctors');

setDoctors(response.data);

};

fetchDoctors();

}, []);

return (

<div>

<h2>Available Doctors</h2>

<ul>

{doctors.map((doctor) => (

<li key={doctor.\_id}>

{doctor.name} - {doctor.specialty}

</li>

))}

</ul>

</div>

);

};

export default DoctorList;

client/src/components/BookAppointment.js

jsx

import React, { useState } from 'react';

import axios from 'axios';

const BookAppointment = () => {

const [doctorId, setDoctorId] = useState('');

const [appointmentDate, setAppointmentDate] = useState('');

const handleSubmit = async (e) => {

e.preventDefault();

try {

await axios.post('/api/doctors/book-appointment', {

doctorId,

appointmentDate,

});

alert('Appointment booked successfully');

} catch (err) {

console.error(err);

alert('Error booking appointment');

}

};

return (

<div>

<h2>Book an Appointment</h2>

<form onSubmit={handleSubmit}>

<label>

Doctor ID:

<input

type="text"

value={doctorId}

onChange={(e) => setDoctorId(e.target.value)}

/>

</label>

<br />

<label>

Appointment Date:

<input

type="datetime-local"

value={appointmentDate}

onChange={(e) => setAppointmentDate(e.target.value)}

/>

</label>

<br />

<button type="submit">Book Appointment</button>

</form>

</div>

);

};

export default BookAppointment;

client/src/App.js

jsx

import React from 'react';

import { BrowserRouter as Router, Switch, Route, Link } from 'react-router-dom';

import DoctorList from './components/DoctorList';

import BookAppointment from './components/BookAppointment';

function App() {

return (

<Router>

<div>

<nav>

<ul>

<li>

<Link to="/">Doctors</Link>

</li>

<li>

<Link to="/book-appointment">Book Appointment</Link>

</li>

</ul>

</nav>

<Switch>

<Route path="/book-appointment">

<BookAppointment />

</Route>

<Route path="/">

<DoctorList />

</Route>

</Switch>

</div>

</Router>

);

}

export default App;

To run the application, you'll need to start both the server and the client:

1. Start the server:

bash

cd server

npm install

node app.js

1. Start the client:

bash

cd client

npm install

npm start

This will start the development server for the React application, and you can access it at http://localhost:3000.