1. Project Setup & Configuration

Tools Required:

Node.js

MongoDB

Visual Studio Code

Postman (for API testing)

Steps:

Create the project folder:

bash

Copy

Edit

mkdir docspot

cd docspot

Initialize backend:

bash

Copy

Edit

mkdir backend

cd backend

npm init -y

npm install express mongoose dotenv cors

Initialize frontend:

bash

Copy

Edit

cd ..

npx create-react-app frontend

✅ 2. Backend Development

Use Node.js and Express.js.

Create files like:

server.js

routes/appointmentRoutes.js

models/Appointment.js

controllers/appointmentController.js

Sample route:

js

Copy

Edit

// routes/appointmentRoutes.js

const express = require('express');

const router = express.Router();

const { bookAppointment } = require('../controllers/appointmentController');

router.post('/book', bookAppointment);

module.exports = router;

✅ 3. Database (MongoDB)

Set up Mongoose model for appointments.

Example:

js

Copy

Edit

// models/Appointment.js

const mongoose = require('mongoose');

const appointmentSchema = new mongoose.Schema({

patientName: String,

doctorName: String,

date: String,

time: String

});

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

✅ 4. Frontend Development

Use React.js to build components.

Example files:

src/components/BookingForm.js

src/pages/Home.js

src/App.js

Sample form:

jsx

Copy

Edit

function BookingForm() {

const [form, setForm] = useState({ patientName: "", doctorName: "", date: "", time: "" });

const handleSubmit = async (e) => {

e.preventDefault();

await fetch("http://localhost:5000/api/book", {

method: "POST",

headers: { "Content-Type": "application/json" },

body: JSON.stringify(form),

});

};

return (

<form onSubmit={handleSubmit}>

<input type="text" placeholder="Patient Name" onChange={e => setForm({...form, patientName: e.target.value})} />

{/\* Add other inputs similarly \*/}

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

</form>

);

}

✅ 5. Project Implementation

Connect backend to frontend using REST APIs

Test APIs using Postman

Use .env for sensitive credentials

✅ 6. Extra

ER Diagram: Create using draw.io or DB Designer