Doctor Appointment Booking System Using AI & Digital Technology

Optimizing Doctor Availability and Appointment Allocation

INTRODUCTION:

- The medical sector is challenged by managing doctor appointments effectively.
- Patients have difficulty locating available physicians, resulting in extensive waiting times.
- This system offers real-time doctor availability check and automatic appointment scheduling.
- Leverages AI and digital technology to optimize hospital workflow and patient experience.

PROBLEM STATEMENT:

- Most hospitals are still utilizing manual appointment scheduling, which causes inefficiencies.
- Patients don't always know if a doctor is available before visiting the hospital.
- Overbooked physicians lead to longer waiting periods for patients.
- No smart system to recommend substitute doctors by availability.

SYSTEM WORKFLOW:

1. User Login & Search: The patient logs in and searches for a doctor based on specialization.
2. Availability Check: The system verifies if the selected doctor is available.
3. Appointment Booking:
    1. If available → The patient books an appointment.
    2. If unavailable → The system recommends other nearby doctors.
4. Confirmation & Notification:
    - The patient receives a confirmation message with appointment details.

5. Doctor Dashboard:

    - Doctors can view their schedule and upcoming appointments

KEY FEATURES:

- **Real-time Doctor Availability Check –**
  Prevents patients from booking unavailable slots
- **AI-based Doctor Recommendation**
  Recommends doctors nearby by location & specialty.
- **Automated Appointment Scheduling –**
  Enables simple and streamlined booking.
- **User-Friendly Interface -**
  Easy and interactive UI for patients & physicians
- **Doctor Dashboard -**
  Physicians are able to manage appointments and availability.

TECHNOLOGIES USED:

Frontend :

- React (Typescipt) –> Main Frontend framework
- **Vite** → Fast build tool for frontend development
- **Tailwind CSS** → Utility-first CSS framework for styling

Configuration & Build Tools:

- **ESLint** → Code linting
- **PostCSS** → CSS transformations
- **TypeScript** → Strongly typed JavaScript

SCREENSHOTS:

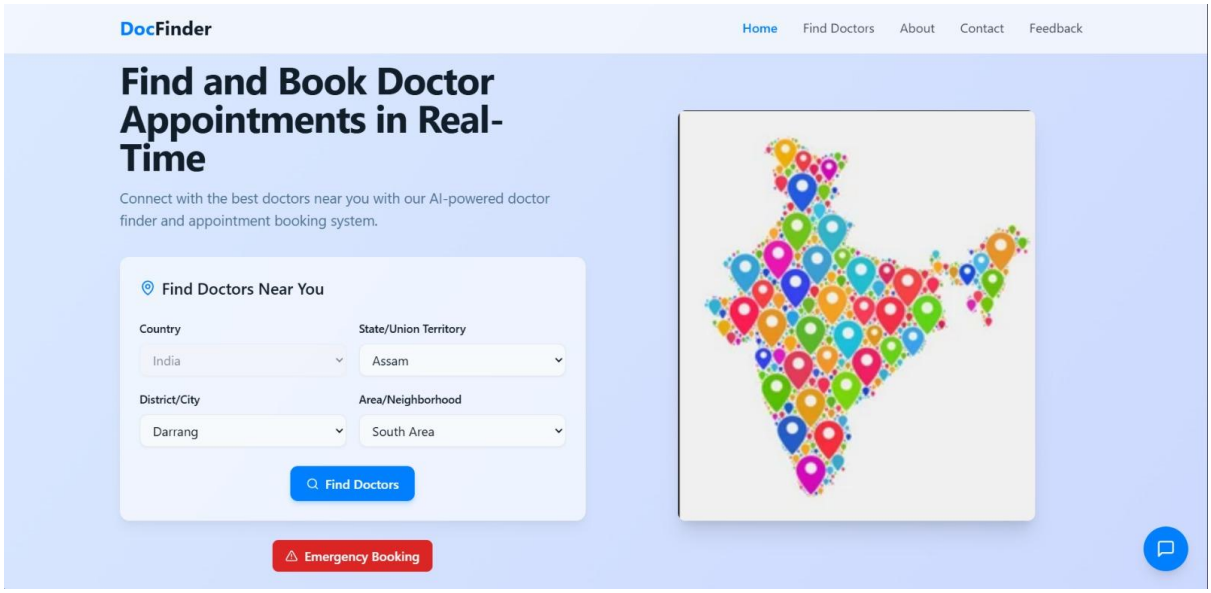

Figure a). Find Doctor Page

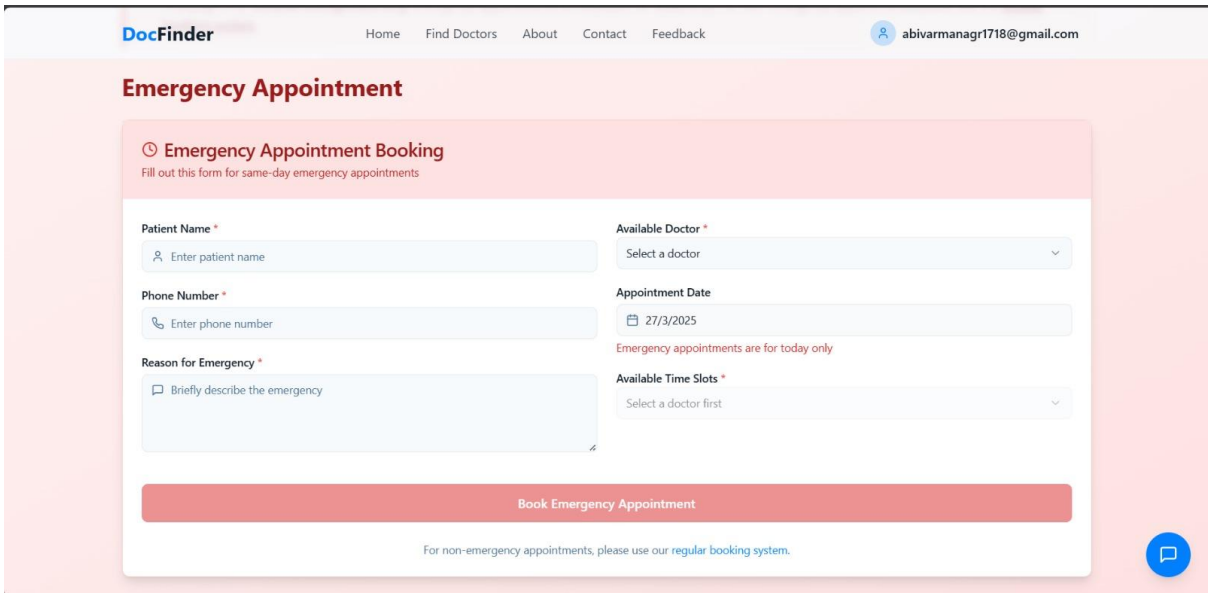

Figure b). Emergency page

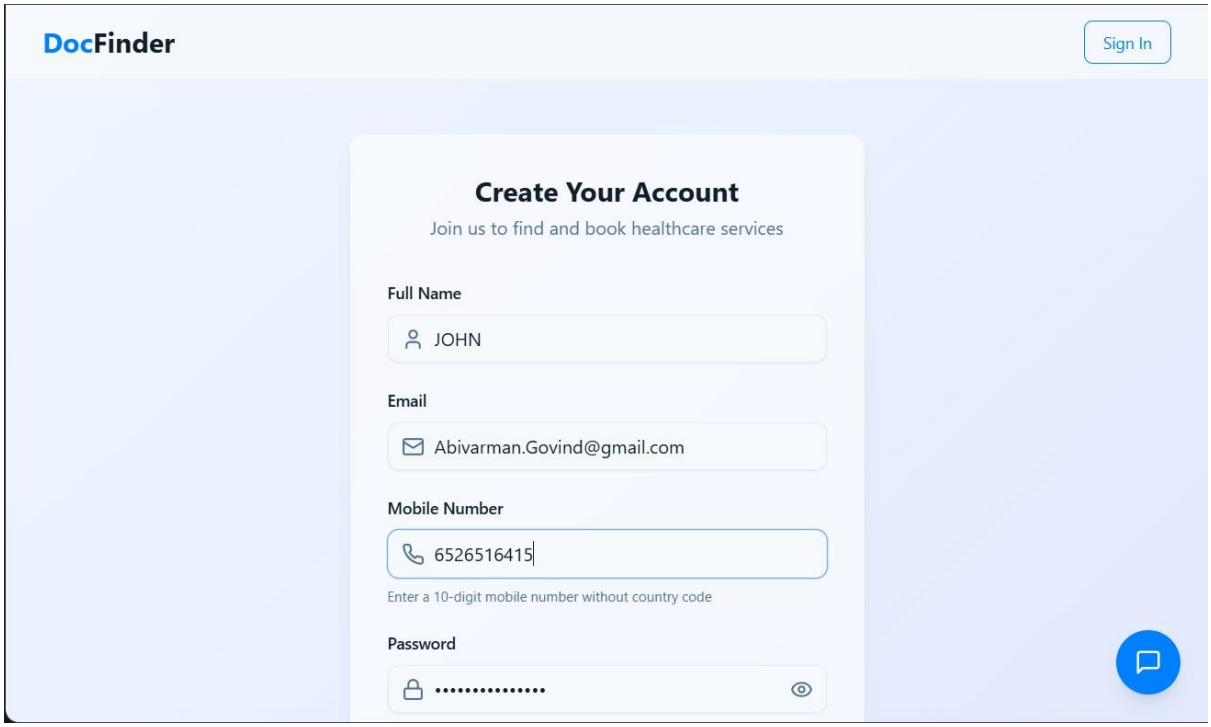

Figure c) Sign in page

CODE:

```
import { Doctor, LocationType } from '@/data/mockData';
import { getRecommendedDoctors } from './bookingUtils';

interface SymptomMatch {
  symptoms: string[];
  specialty: string;
  confidence: number;
}

// Symptom database with matching specialties
const symptomSpecialtyMap: SymptomMatch[] = [
  {
    symptoms: ['fever', 'cough', 'cold', 'sore throat', 'headache', 'flu', 'body pain'],
```

```
  specialty: 'General Physician',
  confidence: 0.8
 },
 {
  symptoms: ['chest pain', 'shortness of breath', 'palpitations', 'high blood pressure', 'heart', 'dizziness'],
  specialty: 'Cardiologist',
  confidence: 0.9
 },
 {
  symptoms: ['rash', 'acne', 'skin', 'itching', 'skin infection', 'mole', 'hair loss'],
  specialty: 'Dermatologist',
  confidence: 0.85
 },
 {
  symptoms: ['headache', 'migraine', 'seizure', 'memory loss', 'tremor', 'balance', 'numbness'],
  specialty: 'Neurologist',
  confidence: 0.85
 },
 {
  symptoms: ['joint pain', 'fracture', 'bone', 'back pain', 'knee pain', 'muscle', 'sprain'],
  specialty: 'Orthopedic',
  confidence: 0.9
 },
 {
  symptoms: ['ear', 'nose', 'throat', 'sinus', 'hearing loss', 'tonsil', 'voice hoarse'],
  specialty: 'ENT Specialist',
  confidence: 0.85
```

```
},
{
  symptoms: ['eye', 'vision', 'glasses', 'red eye', 'blurry vision', 'eye pain', 'cataract'],
  specialty: 'Ophthalmologist',
  confidence: 0.9
},
{
  symptoms: ['pregnancy', 'menstrual', 'vaginal', 'ovary', 'uterus', 'breast pain'],
  specialty: 'Gynecologist',
  confidence: 0.95
},
{
  symptoms: ['depression', 'anxiety', 'stress', 'insomnia', 'mood', 'panic', 'mental health'],
  specialty: 'Psychiatrist',
  confidence: 0.8
},
{
  symptoms: ['breathing', 'cough', 'asthma', 'tuberculosis', 'pneumonia', 'lung'],
  specialty: 'Pulmonologist',
  confidence: 0.85
},
{
  symptoms: ['diabetes', 'thyroid', 'hormone', 'weight gain', 'growth', 'metabolism'],
  specialty: 'Endocrinologist',
  confidence: 0.85
},
{
  symptoms: ['kidney', 'urinary', 'bladder', 'prostate', 'urine', 'testicular'],
```

```
    specialty: 'Urologist',
    confidence: 0.9
  },
  {
    symptoms: ['stomach', 'digestion', 'diarrhea', 'constipation', 'abdominal pain', 'vomiting',
'nausea'],
    specialty: 'Gastroenterologist',
    confidence: 0.85
  },
  {
    symptoms: ['child', 'infant', 'baby', 'vaccination', 'growth', 'development'],
    specialty: 'Pediatrician',
    confidence: 0.9
  }
];

// NLP-like function to analyze symptoms and recommend specialists
export const analyzeSymptoms = (symptoms: string): { specialty: string; confidence: number
} => {
  // Convert to lowercase and split into words
  const symptomWords = symptoms.toLowerCase().split(/[\s,;.!?]+/);

  // Count matches for each specialty
  const matches = symptomSpecialtyMap.map(item => {
    let matchCount = 0;
    let matchedSymptoms = new Set<string>();

    // Check each symptom word against our database
    symptomWords.forEach(word => {
```

```
    if (word.length < 3) return; // Skip short words

    item.symptoms.forEach(symptom => {
      if (symptom.includes(word) || word.includes(symptom)) {
        matchedSymptoms.add(symptom);
        matchCount++;
      }
    });
  });

  // Calculate a confidence score based on number of matches and strength of match
  const uniqueMatches = matchedSymptoms.size;
  const confidence = uniqueMatches > 0
    ? Math.min(item.confidence * (uniqueMatches / 3), 0.95) // Cap confidence at 95%
    : 0;

  return {
    specialty: item.specialty,
    confidence,
    matchCount,
    uniqueMatches
  };
});

// Sort by confidence score
const sortedMatches = matches.sort((a, b) => b.confidence - a.confidence);

// If no strong matches, default to General Physician
```

```
  if (sortedMatches[0].confidence < 0.3) {
    return { specialty: 'General Physician', confidence: 0.5 };
  }

  return {
    specialty: sortedMatches[0].specialty,
    confidence: sortedMatches[0].confidence
  };
};

// Get AI-recommended doctors based on symptoms and location
export const getAIRecommendedDoctors = (
  doctors: Doctor[],
  symptoms: string,
  location: LocationType
): { doctors: Doctor[]; recommendedSpecialty: string; confidence: number } => {
  // First analyze symptoms to get specialty
  const { specialty, confidence } = analyzeSymptoms(symptoms);

  // Filter doctors by the recommended specialty
  const specialtyDoctors = doctors.filter(
    doctor => doctor.specialty === specialty || doctor.specialty === 'General Physician'
  );

  // If we have doctors in this specialty, use our recommendation engine to rank them
  if (specialtyDoctors.length > 0) {
    const recommendedDoctors = getRecommendedDoctors(specialtyDoctors, {
      location,
```

```
    medicalConditions: [symptoms]
   });

   return {
    doctors: recommendedDoctors,
    recommendedSpecialty: specialty,
    confidence
   };
  }

  // Fallback: return all available doctors ranked by recommendation algorithm
  const recommendedDoctors = getRecommendedDoctors(doctors, {
   location,
   medicalConditions: [symptoms]
  });

  return {
   doctors: recommendedDoctors,
   recommendedSpecialty: specialty,
   confidence
  };
};

// Function to provide AI explanation of recommendation
export const generateAIRecommendationExplanation = (
  specialty: string,
  confidence: number,
  symptoms: string
```

```
): string => {
  if (confidence >= 0.8) {
    return `Based on your symptoms "${symptoms}", I'm confident (${Math.round(confidence * 100)}%) that you should see a ${specialty}. They specialize in treating these conditions.`;
  } else if (confidence >= 0.5) {
    return `Your symptoms "${symptoms}" suggest you may need a ${specialty} (${Math.round(confidence * 100)}% confidence), but you might also benefit from seeing a General Physician first for an evaluation.`;
  } else {
    return `I'm not entirely sure which specialist best matches your symptoms "${symptoms}". I recommend starting with a General Physician who can provide a proper referral after examination.`;
  }
};
```

CONCLUSION:

- The Doctor Appointment System decreases patient waiting time and improves hospital efficiency.
- Artificial intelligence-powered doctor allocation optimizes the workload and enhances the accessibility of healthcare.
- Digital integration provides hassle-free scheduling and an improved patient experience.
- Future upgrades will encompass telemedicine, predictive analytics, and chatbot support for greater automation.