

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>HealthAI - Intelligent Healthcare Assistant</title>

  <script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/3.9.1/chart.min.js"></script>

  <style>

    * {

      margin: 0;

      padding: 0;

      box-sizing: border-box;

    }

    body {

      font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;

      background: linear-gradient(135deg, #f5f7fa 0%, #c3cfe2 100%);

      min-height: 100vh;

    }

    .header {

      background: linear-gradient(90deg, #2E8B57, #4682B4);

      color: white;

      padding: 2rem;

      text-align: center;

      box-shadow: 0 4px 6px rgba(0,0,0,0.1);

    }
```

```
.header h1 {  
    font-size: 2.5rem;  
    margin-bottom: 0.5rem;  
}
```

```
.header p {  
    font-size: 1.2rem;  
    opacity: 0.9;  
}
```

```
.container {  
    max-width: 1400px;  
    margin: 0 auto;  
    padding: 2rem;  
}
```

```
.sidebar {  
    position: fixed;  
    left: -350px;  
    top: 0;  
    width: 350px;  
    height: 100vh;  
    background: white;  
    box-shadow: 2px 0 10px rgba(0,0,0,0.1);  
    transition: left 0.3s ease;  
    z-index: 1000;  
    overflow-y: auto;
```

```
}
```

```
.sidebar.active {
```

```
  left: 0;
```

```
}
```

```
.sidebar-toggle {
```

```
  position: fixed;
```

```
  top: 20px;
```

```
  left: 20px;
```

```
  background: #2E8B57;
```

```
  color: white;
```

```
  border: none;
```

```
  padding: 10px 15px;
```

```
  border-radius: 5px;
```

```
  cursor: pointer;
```

```
  z-index: 1001;
```

```
  font-size: 16px;
```

```
}
```

```
.sidebar-content {
```

```
  padding: 2rem;
```

```
}
```

```
.profile-section {
```

```
  background: #f8f9fa;
```

```
  padding: 1.5rem;
```

```
border-radius: 10px;  
margin-bottom: 1rem;  
}
```

```
.tabs {  
  display: flex;  
  background: white;  
  border-radius: 10px;  
  overflow: hidden;  
  box-shadow: 0 2px 10px rgba(0,0,0,0.1);  
  margin-bottom: 2rem;  
}
```

```
.tab-button {  
  flex: 1;  
  padding: 1rem 2rem;  
  background: white;  
  border: none;  
  cursor: pointer;  
  font-size: 1rem;  
  transition: all 0.3s ease;  
  border-bottom: 3px solid transparent;  
}
```

```
.tab-button:hover {  
  background: #f8f9fa;  
}
```

```
.tab-button.active {  
    background: #2E8B57;  
    color: white;  
    border-bottom-color: #1e5f3f;  
}
```

```
.tab-content {  
    display: none;  
    background: white;  
    border-radius: 10px;  
    padding: 2rem;  
    box-shadow: 0 2px 10px rgba(0,0,0,0.1);  
    min-height: 600px;  
}
```

```
.tab-content.active {  
    display: block;  
}
```

```
.form-group {  
    margin-bottom: 1.5rem;  
}
```

```
.form-group label {  
    display: block;  
    margin-bottom: 0.5rem;
```

```
font-weight: 600;

color: #333;

}
```

```
.form-group input,
.form-group select,
.form-group textarea {
  width: 100%;
  padding: 0.75rem;
  border: 2px solid #e0e0e0;
  border-radius: 5px;
  font-size: 1rem;
  transition: border-color 0.3s ease;
}
```

```
.form-group input:focus,
.form-group select:focus,
.form-group textarea:focus {
  outline: none;
  border-color: #2E8B57;
}
```

```
.btn {
  background: #2E8B57;
  color: white;
  border: none;
  padding: 0.75rem 1.5rem;
```

```
border-radius: 5px;

cursor: pointer;

font-size: 1rem;

transition: all 0.3s ease;

margin-right: 1rem;

margin-bottom: 1rem;
}
```

```
.btn:hover {

    background: #1e5f3f;

    transform: translateY(-2px);
}
```

```
.btn-secondary {

    background: #6c757d;
}
```

```
.btn-secondary:hover {

    background: #545b62;
}
```

```
.btn-danger {

    background: #dc3545;
}
```

```
.btn-danger:hover {

    background: #c82333;
}
```

```
}
```

```
.chat-container {  
  height: 400px;  
  border: 2px solid #e0e0e0;  
  border-radius: 10px;  
  overflow-y: auto;  
  padding: 1rem;  
  margin-bottom: 1rem;  
  background: #fafafa;  
}
```

```
.chat-message {  
  margin-bottom: 1rem;  
  padding: 1rem;  
  border-radius: 10px;  
  max-width: 80%;  
}
```

```
.user-message {  
  background: #e3f2fd;  
  margin-left: auto;  
  text-align: right;  
}
```

```
.ai-message {  
  background: #f1f8e9;
```



```
    margin-right: auto;
}
```

```
.metrics-grid {
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));
    gap: 1rem;
    margin-bottom: 2rem;
}
```

```
.metric-card {
    background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
    color: white;
    padding: 1.5rem;
    border-radius: 10px;
    text-align: center;
}
```

```
.metric-value {
    font-size: 2rem;
    font-weight: bold;
    margin-bottom: 0.5rem;
}
```

```
.metric-label {
    font-size: 0.9rem;
    opacity: 0.9;
```

```
}
```

```
.alert {  
  padding: 1rem;  
  border-radius: 5px;  
  margin-bottom: 1rem;  
}
```

```
.alert-warning {  
  background: #fff3cd;  
  border: 1px solid #ffeaa7;  
  color: #856404;  
}
```

```
.alert-danger {  
  background: #f8d7da;  
  border: 1px solid #f5c6cb;  
  color: #721c24;  
}
```

```
.alert-info {  
  background: #d1ecf1;  
  border: 1px solid #bee5eb;  
  color: #0c5460;  
}
```

```
.alert-success {
```

```
background: #d4edda;
border: 1px solid #c3e6cb;
color: #155724;
}
```

```
.chart-container {
  position: relative;
  height: 400px;
  margin: 2rem 0;
}
```

```
.symptom-grid {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(150px, 1fr));
  gap: 1rem;
  margin-bottom: 1rem;
}
```

```
.symptom-checkbox {
  display: flex;
  align-items: center;
  padding: 0.75rem;
  background: #f8f9fa;
  border-radius: 5px;
  cursor: pointer;
  transition: background 0.3s ease;
}
```

```
.symptom-checkbox:hover {  
    background: #e9ecef;  
}
```

```
.symptom-checkbox input {  
    margin-right: 0.5rem;  
}
```

```
.progress-section {  
    background: #f8f9fa;  
    padding: 1.5rem;  
    border-radius: 10px;  
    margin-top: 2rem;  
}
```

```
.slider-group {  
    margin-bottom: 1rem;  
}
```

```
.slider-group label {  
    display: block;  
    margin-bottom: 0.5rem;  
    font-weight: 600;  
}
```

```
.slider {
```

```
width: 100%;  
height: 6px;  
border-radius: 3px;  
background: #ddd;  
outline: none;  
-webkit-appearance: none;  
}
```

```
.slider::-webkit-slider-thumb {  
  -webkit-appearance: none;  
  appearance: none;  
  width: 20px;  
  height: 20px;  
  border-radius: 50%;  
  background: #2E8B57;  
  cursor: pointer;  
}
```

```
.sample-questions {  
  display: grid;  
  gap: 0.5rem;  
  margin-top: 1rem;  
}
```

```
.sample-question {  
  background: #f8f9fa;  
  border: 1px solid #e0e0e0;
```

```
padding: 0.75rem;

border-radius: 5px;

cursor: pointer;

transition: all 0.3s ease;
}
```

```
.sample-question:hover {

    background: #e9ecef;

    transform: translateX(5px);
}
```

```
.hidden {

    display: none !important;
}
```

```
@media (max-width: 768px) {

    .tabs {

        flex-direction: column;
    }
}
```

```
.tab-button {

    padding: 0.75rem 1rem;
}
```

```
.container {

    padding: 1rem;
}
```

```
.metrics-grid {  
    grid-template-columns: 1fr;  
}  
  
}  
  
</style>  
  
</head>  
  
<body>  
  
    <div class="header">  
  
        <h1>🏥 HealthAI</h1>  
  
        <p>Intelligent Healthcare Assistant powered by IBM Watson</p>  
  
    </div>  
  
  
  
  
    <button class="sidebar-toggle" onclick="toggleSidebar()">🔍 Profile</button>  
  
  
  
  
  
  
  
  
  
    <div class="sidebar" id="sidebar">  
  
        <div class="sidebar-content">  
  
            <h3>Patient Profile</h3>  
  
            <div id="profileForm" class="profile-section">  
  
                <div class="form-group">  
  
                    <label for="age">Age</label>  
  
                    <input type="number" id="age" min="1" max="120" value="30">  
  
                </div>  
  
                <div class="form-group">  
  
                    <label for="gender">Gender</label>  
  
                    <select id="gender">  
  
                        <option value="Male">Male</option>
```

<option value="Female">Female</option>

<option value="Other">Other</option>

</select>

</div>

<div class="form-group">

<label for="weight">Weight (kg)</label>

<input type="number" id="weight" min="20" max="300" value="70" step="0.1">

</div>

<div class="form-group">

<label for="height">Height (cm)</label>

<input type="number" id="height" min="100" max="250" value="170">

</div>

<div class="form-group">

<label for="medicalHistory">Medical History</label>

<textarea id="medicalHistory" rows="3" placeholder="Enter conditions separated by commas"></textarea>

</div>

<div class="form-group">

<label for="medications">Current Medications</label>

<textarea id="medications" rows="3" placeholder="Enter medications separated by commas"></textarea>

</div>

<div class="form-group">

<label for="allergies">Allergies</label>

<textarea id="allergies" rows="3" placeholder="Enter allergies separated by commas"></textarea>

</div>

<button class="btn" onclick="saveProfile()">Save Profile</button>

</div>

<div id="profileDisplay" class="profile-section hidden">

<h4>Profile Active </h4>

<div id="profileInfo"></div>

<button class="btn btn-secondary" onclick="editProfile()">Edit Profile</button>

</div>

</div>

</div>

<div class="container">

<div class="tabs">

<button class="tab-button active" onclick="showTab('chat')"> Patient Chat</button>

<button class="tab-button" onclick="showTab('prediction')"> Disease Prediction</button>

<button class="tab-button" onclick="showTab('treatment')"> Treatment Plans</button>

<button class="tab-button" onclick="showTab('analytics')"> Health Analytics</button>

</div>

<div id="chat" class="tab-content active">

<h2> Healthcare Chat Assistant</h2>

<div class="chat-container" id="chatContainer"></div>

<div class="form-group">

<input type="text" id="chatInput" placeholder="Ask me anything about your health..."
onkeypress="handleChatKeyPress(event)">

</div>

<button class="btn" onclick="sendMessage()">Send</button>

<button class="btn btn-secondary" onclick="clearChat()">Clear Chat</button>

<h3> Sample Questions</h3>

<div class="sample-questions">

`<div class="sample-question" onclick="askSampleQuestion('What are the symptoms of high blood pressure?')">`

What are the symptoms of high blood pressure?

`</div>`

`<div class="sample-question" onclick="askSampleQuestion('How can I improve my sleep quality?')">`

How can I improve my sleep quality?

`</div>`

`<div class="sample-question" onclick="askSampleQuestion('What foods should I avoid with diabetes?')">`

What foods should I avoid with diabetes?

`</div>`

`<div class="sample-question" onclick="askSampleQuestion('When should I see a doctor for persistent headaches?')">`

When should I see a doctor for persistent headaches?

`</div>`

`</div>`

`</div>`

`<div id="prediction" class="tab-content">`

`<h2>❏ Disease Prediction System</h2>`

`<div class="alert alert-warning">`

`❏ Medical Disclaimer:` This tool provides general information only and should not replace professional medical advice.

Always consult healthcare professionals for proper diagnosis and treatment.

`</div>`

`<div style="display: grid; grid-template-columns: 2fr 1fr; gap: 2rem;">`

`<div>`

`<h3>Symptom Input</h3>`

```
<div class="form-group">

  <label>Select your primary symptoms:</label>

  <div class="symptom-grid">

    <div class="symptom-checkbox">

      <input type="checkbox" id="headache" value="Headache">

      <label for="headache">Headache</label>

    </div>

    <div class="symptom-checkbox">

      <input type="checkbox" id="fever" value="Fever">

      <label for="fever">Fever</label>

    </div>

    <div class="symptom-checkbox">

      <input type="checkbox" id="fatigue" value="Fatigue">

      <label for="fatigue">Fatigue</label>

    </div>

    <div class="symptom-checkbox">

      <input type="checkbox" id="nausea" value="Nausea">

      <label for="nausea">Nausea</label>

    </div>

    <div class="symptom-checkbox">

      <input type="checkbox" id="dizziness" value="Dizziness">

      <label for="dizziness">Dizziness</label>

    </div>

    <div class="symptom-checkbox">

      <input type="checkbox" id="chestpain" value="Chest Pain">

      <label for="chestpain">Chest Pain</label>

    </div>

  </div>

</div>
```

```
</div>
```

```
</div>
```

```
<div class="form-group">
```

```
  <label for="symptomDescription">Describe your symptoms in detail:</label>
```

```
  <textarea id="symptomDescription" rows="4"></textarea>
```

```
</div>
```

```
<div class="form-group">
```

```
  <label for="duration">Duration of symptoms:</label>
```

```
  <select id="duration">
```

```
    <option value="Less than 24 hours">Less than 24 hours</option>
```

```
    <option value="1-3 days">1-3 days</option>
```

```
    <option value="1 week">1 week</option>
```

```
    <option value="1-4 weeks">1-4 weeks</option>
```

```
    <option value="More than a month">More than a month</option>
```

```
  </select>
```

```
</div>
```

```
<div class="form-group">
```

```
  <label for="severity">Severity (1-10):</label>
```

```
  <input type="range" id="severity" min="1" max="10" value="5" class="slider">
```

```
  <div style="display: flex; justify-content: space-between; font-size: 0.8rem;">
```

```
    <span>Mild</span>
```

```
    <span>Moderate</span>
```

```
    <span>Severe</span>
```

```
</div>
```

</div>

<button class="btn" onclick="analyzeSymptoms()">Analyze Symptoms</button>

<div id="analysisResults" class="hidden" style="margin-top: 2rem;"></div>

</div>

<div>

<h3>Risk Factors</h3>

<div id="riskFactors" class="alert alert-info">

Create a patient profile for personalized risk assessment.

</div>

</div>

</div>

</div>

<div id="treatment" class="tab-content">

<h2>📋 Personalized Treatment Plans</h2>

<div style="display: grid; grid-template-columns: 2fr 1fr; gap: 2rem;">

<div>

<div class="form-group">

<label for="condition">Select condition for treatment plan:</label>

<select id="condition">

<option value="">-- Select Condition --</option>

<option value="Hypertension">Hypertension</option>

<option value="Type 2 Diabetes">Type 2 Diabetes</option>

<option value="High Cholesterol">High Cholesterol</option>

```
<option value="Anxiety">Anxiety</option>

<option value="Depression">Depression</option>

<option value="Asthma">Asthma</option>

</select>
```

```
</div>
```

```
<div class="form-group">
```

```
  <label for="additionalInfo">Additional information about your condition:</label>
```

```
  <textarea id="additionalInfo" rows="4"></textarea>
```

```
</div>
```

```
<button class="btn" onclick="generateTreatmentPlan()">Generate Treatment Plan</button>
```

```
<div id="treatmentPlan" class="hidden" style="margin-top: 2rem;"></div>
```

```
<div id="progressSection" class="progress-section hidden">
```

```
  <h3>☒ Treatment Progress Tracker</h3>
```

```
  <div class="slider-group">
```

```
    <label>Medication Adherence: <span id="adherenceValue">80</span>%</label>
```

```
    <input type="range" id="adherence" min="0" max="100" value="80" class="slider"
oninput="updateSliderValue('adherence', 'adherenceValue')">
```

```
  </div>
```

```
  <div class="slider-group">
```

```
    <label>Lifestyle Changes: <span id="lifestyleValue">60</span>%</label>
```

```
    <input type="range" id="lifestyle" min="0" max="100" value="60" class="slider"
oninput="updateSliderValue('lifestyle', 'lifestyleValue')">
```

```
  </div>
```

```
  <div class="slider-group">
```

```

        <label>Symptom Improvement: <span id="improvementValue">70</span>%</
label>

        <input type="range" id="improvement" min="0" max="100" value="70" class="slider"
oninput="updateSliderValue('improvement', 'improvementValue')">

        </div>

        <canvas id="progressChart" width="400" height="400"></canvas>

    </div>

</div>

<div>

    <h3>📅 Next Steps</h3>

    <div class="alert alert-info">

        • Schedule follow-up appointment<br>

        • Monitor symptoms daily<br>

        • Take medications as prescribed<br>

        • Maintain healthy lifestyle

    </div>

    <h3>🔔 Reminders</h3>

    <div class="form-group">

        <label for="reminderTime">Set medication reminder:</label>

        <input type="time" id="reminderTime">

    </div>

    <button class="btn btn-secondary" onclick="setReminder()">Set Reminder</button>

</div>

</div>

<div id="analytics" class="tab-content">

```

```
<h2>🏥 Health Analytics Dashboard</h2>
```

```
<div class="metrics-grid">
```

```
  <div class="metric-card">
```

```
    <div class="metric-value" id="avgHeartRate">72</div>
```

```
    <div class="metric-label">Avg Heart Rate (bpm)</div>
```

```
  </div>
```

```
  <div class="metric-card">
```

```
    <div class="metric-value" id="avgBloodPressure">120/80</div>
```

```
    <div class="metric-label">Avg Blood Pressure</div>
```

```
  </div>
```

```
  <div class="metric-card">
```

```
    <div class="metric-value" id="avgGlucose">95</div>
```

```
    <div class="metric-label">Avg Blood Glucose (mg/dL)</div>
```

```
  </div>
```

```
  <div class="metric-card">
```

```
    <div class="metric-value" id="avgSleep">7.5</div>
```

```
    <div class="metric-label">Avg Sleep (hours)</div>
```

```
  </div>
```

```
</div>
```

```
<div class="form-group">
```

```
  <label for="chartType">Select metric to visualize:</label>
```

```
  <select id="chartType" onchange="updateChart()">
```

```
    <option value="heartRate">Heart Rate</option>
```

```
    <option value="bloodPressure">Blood Pressure</option>
```

```
    <option value="bloodGlucose">Blood Glucose</option>
```



```
<option value="weight">Weight</option>
```

```
<option value="sleep">Sleep Hours</option>
```

```
</select>
```

```
</div>
```

```
<div class="chart-container">
```

```
<canvas id="healthChart"></canvas>
```

```
</div>
```

```
<h3>🏥 AI Health Insights</h3>
```

```
<div id="healthInsights">
```

```
<div class="alert alert-success">
```

```
    🟢 Your health metrics are within normal ranges. Keep up the good work!
```

```
</div>
```

```
</div>
```

```
<button class="btn" onclick="exportHealthData()">📄 Export Health Data</button>
```

```
</div>
```

```
</div>
```

```
<script>
```

```
let patientProfile = null;
```

```
let chatHistory = [];
```

```
let healthData = generateHealthData();
```

```
let currentChart = null;
```

```
class IBMWatsonIntegration {  
  constructor(apiKey = "mock_api_key") {  
    this.apiKey = apiKey;  
    this.modelId = "granite-13b-instruct-v2";  
  }
```

```
  generateResponse(prompt, context = "") {  
  
    if (prompt.toLowerCase().includes('symptom') || context === 'symptom') {  
      return this.generateSymptomAnalysis(prompt);  
    } else if (prompt.toLowerCase().includes('treatment') || context === 'treatment') {  
      return this.generateTreatmentPlan(prompt);  
    } else if (context === 'chat') {  
      return this.generateChatResponse(prompt);  
    }  
  
    return "I'm here to help with your healthcare questions. Please provide more details.";  
  }
```

```
  generateSymptomAnalysis(symptoms) {  
    const responses = {  
      'headache': 'Persistent headaches can be caused by various factors including tension,  
dehydration, or underlying conditions.',  
      'fever': 'Fever is often a sign of infection or inflammation in the body.',  
      'fatigue': 'Fatigue can result from poor sleep, stress, nutritional deficiencies, or medical  
conditions.',  
      'chest pain': 'Chest pain requires immediate medical attention to rule out cardiac  
issues.',
```

'difficulty breathing': 'Breathing difficulties need urgent medical evaluation.'

};

let response = "Based on the symptoms you've described:\n\n";

for (const [symptom, explanation] of Object.entries(responses)) {

if (symptoms.toLowerCase().includes(symptom)) {

response += `• \${explanation}\n`;

}

}

response += "\n⚠ This is for informational purposes only. Please consult a healthcare professional for proper diagnosis.";

return response;

}

generateTreatmentPlan(condition) {

const plans = {

'hypertension': `

****Treatment Plan for Hypertension:****

****Medications:****

- ACE inhibitors or ARBs as first-line treatment
- Diuretics if needed
- Regular monitoring required

****Lifestyle Modifications:****

- Reduce sodium intake (<2300mg/day)
- Regular exercise (30min, 5 days/week)
- Maintain healthy weight

- Limit alcohol consumption

****Follow-up:****

- Blood pressure monitoring every 2-4 weeks initially
- Regular physician visits every 3-6 months`,
 'diabetes': `

****Treatment Plan for Type 2 Diabetes:****

****Medications:****

- Metformin as first-line treatment
- Additional medications based on HbA1c levels
- Regular glucose monitoring

****Lifestyle Modifications:****

- Carbohydrate counting and portion control
- Regular physical activity
- Weight management
- Foot care and regular check-ups

****Follow-up:****

- HbA1c testing every 3-6 months
- Annual eye and kidney function tests`

};

```
for (const [key, plan] of Object.entries(plans)) {  
  if (condition.toLowerCase().includes(key)) {  
    return plan;  
  }  
}
```

```
    }  
  }  
  return "Please specify the condition for a personalized treatment plan.";  
}
```

```
generateChatResponse(question) {  
  if (question.toLowerCase().includes('medication')) {  
    return "I can provide general information about medications, but always consult your  
    pharmacist or doctor before starting, stopping, or changing any medication.";  
  } else if (question.toLowerCase().includes('emergency')) {  
    return "For medical emergencies, please call emergency services immediately (911 in  
    the US). I'm here for general health information only.";  
  } else if (question.toLowerCase().includes('blood pressure')) {  
    return "High blood pressure often has no symptoms, which is why it's called the 'silent  
    killer'. Common signs may include headaches, shortness of breath, or nosebleeds, but these  
    usually occur when blood pressure is severely high.";  
  } else if (question.toLowerCase().includes('sleep')) {  
    return "To improve sleep quality: maintain a consistent sleep schedule, create a  
    relaxing bedtime routine, avoid caffeine late in the day, keep your bedroom cool and dark, and  
    limit screen time before bed.";  
  } else if (question.toLowerCase().includes('diabetes')) {  
    return "With diabetes, it's important to avoid refined sugars, processed foods, white  
    bread, sugary drinks, and high-fat meats. Focus on vegetables, lean proteins, whole grains, and  
    monitor portion sizes.";  
  } else if (question.toLowerCase().includes('headache')) {  
    return "You should see a doctor for headaches if they're severe, sudden, accompanied  
    by fever/stiff neck, worsen over time, occur after a head injury, or significantly impact your daily  
    activities.";  
  }  
  return "I'm here to help with your health questions. Could you please provide more  
  specific details about what you'd like to know?";  
}
```

```
}
```

```
const watsonClient = new IBMWatsonIntegration();
```

```
function generateHealthData() {
```

```
    const data = [];
```

```
    const startDate = new Date('2024-01-01');
```

```
    for (let i = 0; i < 365; i++) {
```

```
        const date = new Date(startDate);
```

```
        date.setDate(date.getDate() + i);
```

```
        data.push({
```

```
            date: date.toISOString().split('T')[0],
```

```
            heartRate: Math.floor(Math.random() * 20 + 65),
```

```
            systolicBP: Math.floor(Math.random() * 30 + 110),
```

```
            diastolicBP: Math.floor(Math.random() * 20 + 70),
```

```
            bloodGlucose: Math.floor(Math.random() * 40 + 80),
```

```
            weight: Math.floor(Math.random() * 10 + 65),
```

```
            sleep: Math.round((Math.random() * 4 + 6) * 10) / 10
```

```
        });
```

```
    }
```

```
    return data;
```

```
}
```

```
function showTab(tabName) {

    const tabContents = document.querySelectorAll('.tab-content');
    tabContents.forEach(content => content.classList.remove('active'));

    const tabButtons = document.querySelectorAll('.tab-button');
    tabButtons.forEach(button => button.classList.remove('active'));

    document.getElementById(tabName).classList.add('active');

    event.target.classList.add('active');

    if (tabName === 'analytics') {
        setTimeout(() => updateChart(), 100);
    }
}
```

```
function toggleSidebar() {

    const sidebar = document.getElementById('sidebar');
    sidebar.classList.toggle('active');
}
```

```
document.addEventListener('click', function(event) {

    const sidebar = document.getElementById('sidebar');
    const toggle = document.querySelector('.sidebar-toggle');
```

```
if (!sidebar.contains(event.target) && !toggle.contains(event.target)) {  
    sidebar.classList.remove('active');  
}  
});  
  
function saveProfile() {  
    const age = document.getElementById('age').value;  
    const gender = document.getElementById('gender').value;  
    const weight = parseFloat(document.getElementById('weight').value);  
    const height = parseFloat(document.getElementById('height').value);  
    const medicalHistory = document.getElementById('medicalHistory').value;  
    const medications = document.getElementById('medications').value;  
    const allergies = document.getElementById('allergies').value;  
  
    patientProfile = {  
        age: parseInt(age),  
        gender: gender,  
        weight: weight,  
        height: height,  
        medicalHistory: medicalHistory.split(',').map(item => item.trim()).filter(item => item),  
        medications: medications.split(',').map(item => item.trim()).filter(item => item),  
        allergies: allergies.split(',').map(item => item.trim()).filter(item => item)  
    };  
  
    displayProfile();  
    updateRiskFactors();  
}
```



```

function displayProfile() {

    const profileForm = document.getElementById('profileForm');

    const profileDisplay = document.getElementById('profileDisplay');

    const profileInfo = document.getElementById('profileInfo');

    if (patientProfile) {

        const bmi = (patientProfile.weight / Math.pow(patientProfile.height / 100, 2)).toFixed(1);

        profileInfo.innerHTML = `

            <p><strong>Age:</strong> ${patientProfile.age} years</p>

            <p><strong>Gender:</strong> ${patientProfile.gender}</p>

            <p><strong>BMI:</strong> ${bmi}</p>

            ${patientProfile.medicalHistory.length > 0 ? `<p><strong>Medical History:</strong><br>${patientProfile.medicalHistory.map(item => `• ${item}`).join('<br>')}</p>` : ""}

            ${patientProfile.allergies.length > 0 ? `<p><strong>Allergies:</strong><br>${patientProfile.allergies.map(item => `• ${item}`).join('<br>')}</p>` : ""}

        `;

        profileForm.classList.add('hidden');

        profileDisplay.classList.remove('hidden');

    }

}

function editProfile() {

    document.getElementById('profileForm').classList.remove('hidden');

    document.getElementById('profileDisplay').classList.add('hidden');

}

```

```

function updateRiskFactors() {

    const riskFactors = document.getElementById('riskFactors');

    if (patientProfile) {

        const bmi = (patientProfile.weight / Math.pow(patientProfile.height / 100, 2)).toFixed(1);

        let riskLevel = 'Low';

        let riskColor = 'alert-success';

        if (patientProfile.age > 65 || bmi > 30) {

            riskLevel = 'Moderate';

            riskColor = 'alert-warning';

        }

        if (patientProfile.age > 75 || bmi > 35) {

            riskLevel = 'High';

            riskColor = 'alert-danger';

        }

        riskFactors.className = `alert ${riskColor}`;

        riskFactors.innerHTML = `

            <h4>Risk Assessment</h4>

            <p><strong>Overall Risk Level:</strong> ${riskLevel}</p>

            <p><strong>Age:</strong> ${patientProfile.age} years</p>

            <p><strong>BMI:</strong> ${bmi}</p>

            ${patientProfile.medicalHistory.length > 0 ? `<p><strong>Medical History:</strong>
            ${patientProfile.medicalHistory.join(', ')}</p>` : ""}

        `;

    }
}

```

```
}
```

```
function handleChatKeyPress(event) {
```

```
  if (event.key === 'Enter') {
```

```
    sendMessage();
```

```
  }
```

```
}
```

```
function sendMessage() {
```

```
  const input = document.getElementById('chatInput');
```

```
  const message = input.value.trim();
```

```
  if (message) {
```

```
    addMessageToChat(message, 'user');
```

```
    input.value = "";
```

```
    setTimeout(() => {
```

```
      const response = watsonClient.generateResponse(message, 'chat');
```

```
      addMessageToChat(response, 'ai');
```

```
    }, 1000);
```

```
  }
```

```
}
```

```
function addMessageToChat(message, type) {
```

```
  const chatContainer = document.getElementById('chatContainer');
```

```
  const messageDiv = document.createElement('div');
```

```
  messageDiv.className = `chat-message ${type}-message`;
```

```
const sender = type === 'user' ? 'You' : 'HealthAI';
```

```
messageDiv.innerHTML = `${sender}</strong> ${message.replace(/\n/g, '<br>')}`;
```

```
chatContainer.appendChild(messageDiv);
```

```
chatContainer.scrollTop = chatContainer.scrollHeight;
```

```
chatHistory.push({ type, message });
```

```
}
```

```
function clearChat() {
```

```
    document.getElementById('chatContainer').innerHTML = '';
```

```
    chatHistory = [];
```

```
}
```

```
function askSampleQuestion(question) {
```

```
    document.getElementById('chatInput').value = question;
```

```
    sendMessage();
```

```
}
```

```
function analyzeSymptoms() {
```

```
    const selectedSymptoms = [];
```

```
    const checkboxes = document.querySelectorAll('input[type="checkbox"]:checked');
```

```
    checkboxes.forEach(checkbox => {
```

```
        selectedSymptoms.push(checkbox.value);
```

```
    });
```

```
const description = document.getElementById('symptomDescription').value;
const duration = document.getElementById('duration').value;
const severity = document.getElementById('severity').value;
```

```
if (selectedSymptoms.length === 0 && !description.trim()) {
    alert('Please select symptoms or provide a description.');
```



```
    return;
}
```

```
const analysisPrompt = `Symptoms: ${selectedSymptoms.join(', ')}. Description:
${description}. Duration: ${duration}. Severity: ${severity}/10`;
```

```
const analysis = watsonClient.generateResponse(analysisPrompt, 'symptom');
```

```
const resultsDiv = document.getElementById('analysisResults');
resultsDiv.classList.remove('hidden');
```

```
let alertClass = 'alert-info';
```

```
let recommendation = 'Monitor your symptoms. If they worsen or persist, consult a
healthcare professional.';
```

```
if (parseInt(severity) >= 8 || selectedSymptoms.includes('Chest Pain')) {
```

```
    alertClass = 'alert-danger';
```

```
    recommendation = '⚠ URGENT: These symptoms may require immediate medical
attention. Please contact emergency services or visit the nearest emergency room.';
```

```
} else if (parseInt(severity) >= 6) {
```

```
    alertClass = 'alert-warning';
```

```
    recommendation = '⚠ Consider scheduling an appointment with your healthcare provider
within the next few days.';
```

```
}
```

```
resultsDiv.innerHTML = `
```

```
  <h3>📊 Analysis Results</h3>
```

```
  <div class="alert alert-info">
```

```
    ${analysis.replace(/\n/g, '<br>')}
```

```
  </div>
```

```
  <h3>📌 Recommendations</h3>
```

```
  <div class="alert ${alertClass}">
```

```
    ${recommendation}
```

```
  </div>
```

```
`;
```

```
}
```

```
function generateTreatmentPlan() {
```

```
  const condition = document.getElementById('condition').value;
```

```
  const additionalInfo = document.getElementById('additionalInfo').value;
```

```
  if (!condition) {
```

```
    alert('Please select a condition.');
```

```
    return;
```

```
  }
```

```
  const treatmentPrompt = `Condition: ${condition}. Additional info: ${additionalInfo}`;
```

```
  const plan = watsonClient.generateResponse(treatmentPrompt, 'treatment');
```

```
  const planDiv = document.getElementById('treatmentPlan');
```

```
  const progressSection = document.getElementById('progressSection');
```

```
planDiv.classList.remove('hidden');  
progressSection.classList.remove('hidden');
```

```
planDiv.innerHTML = `  
  <h3>☒ Treatment Plan for ${condition}</h3>  
  <div class="alert alert-info">  
    ${plan.replace(/\n/g, '<br>')}  
  </div>  
`;  
;
```

```
setTimeout(() => updateProgressChart(), 100);  
}
```

```
function updateSliderValue(sliderId, valueId) {  
  const slider = document.getElementById(sliderId);  
  const valueSpan = document.getElementById(valueId);  
  valueSpan.textContent = slider.value;  
  
  if (window.progressChart) {  
    updateProgressChart();  
  }  
}
```

```
function updateProgressChart() {  
  const canvas = document.getElementById('progressChart');  
  const ctx = canvas.getContext('2d');
```

```
if (window.progressChart) {  
  window.progressChart.destroy();  
}
```

```
const adherence = document.getElementById('adherence').value;  
const lifestyle = document.getElementById('lifestyle').value;  
const improvement = document.getElementById('improvement').value;
```

```
window.progressChart = new Chart(ctx, {  
  type: 'radar',  
  data: {  
    labels: ['Medication Adherence', 'Lifestyle Changes', 'Symptom Improvement'],  
    datasets: [{  
      label: 'Treatment Progress',  
      data: [adherence, lifestyle, improvement],  
      backgroundColor: 'rgba(46, 139, 87, 0.2)',  
      borderColor: 'rgba(46, 139, 87, 1)',  
      borderWidth: 2,  
      pointBackgroundColor: 'rgba(46, 139, 87, 1)'  
    }]  
  },  
  options: {  
    responsive: true,  
    maintainAspectRatio: false,  
    scales: {  
      r: {
```



```
        beginAtZero: true,  
        max: 100,  
        ticks: {  
            stepSize: 20  
        }  
    }  
}  
});  
}
```

```
function setReminder() {  
    const time = document.getElementById('reminderTime').value;  
    if (time) {  
        alert(`Reminder set for ${time}. You'll receive a notification to take your medication.`);  
    } else {  
        alert('Please select a time for the reminder.');    }  
}
```

```
function updateChart() {  
    const chartType = document.getElementById('chartType').value;  
    const canvas = document.getElementById('healthChart');  
    const ctx = canvas.getContext('2d');  
  
    if (currentChart) {  
        currentChart.destroy();  
    }  
}
```

```
}
```

```
const labels = healthData.slice(-30).map(d => d.date); // Last 30 days
```

```
let datasets = [];
```

```
let title = "";
```

```
switch (chartType) {
```

```
  case 'heartRate':
```

```
    title = 'Heart Rate Over Time';
```

```
    datasets = [{
```

```
      label: 'Heart Rate (bpm)',
```

```
      data: healthData.slice(-30).map(d => d.heartRate),
```

```
      borderColor: 'rgba(255, 99, 132, 1)',
```

```
      backgroundColor: 'rgba(255, 99, 132, 0.2)',
```

```
      tension: 0.4
```

```
    }];
```

```
    break;
```

```
  case 'bloodPressure':
```

```
    title = 'Blood Pressure Over Time';
```

```
    datasets = [
```

```
      {
```

```
        label: 'Systolic BP (mmHg)',
```

```
        data: healthData.slice(-30).map(d => d.systolicBP),
```

```
        borderColor: 'rgba(54, 162, 235, 1)',
```

```
        backgroundColor: 'rgba(54, 162, 235, 0.2)',
```

```
        tension: 0.4
```

```
},  
  
{  
  label: 'Diastolic BP (mmHg)',  
  data: healthData.slice(-30).map(d => d.diastolicBP),  
  borderColor: 'rgba(75, 192, 192, 1)',  
  backgroundColor: 'rgba(75, 192, 192, 0.2)',  
  tension: 0.4  
}  
];  
  
break;
```

case 'bloodGlucose':

```
  title = 'Blood Glucose Over Time';  
  datasets = [{  
    label: 'Blood Glucose (mg/dL)',  
    data: healthData.slice(-30).map(d => d.bloodGlucose),  
    borderColor: 'rgba(153, 102, 255, 1)',  
    backgroundColor: 'rgba(153, 102, 255, 0.2)',  
    tension: 0.4  
  }];  
  
  break;
```

case 'weight':

```
  title = 'Weight Over Time';  
  datasets = [{  
    label: 'Weight (kg)',  
    data: healthData.slice(-30).map(d => d.weight),
```

```
borderColor: 'rgba(255, 159, 64, 1)',  
backgroundColor: 'rgba(255, 159, 64, 0.2)',  
tension: 0.4  
});
```

```
break;
```

```
case 'sleep':
```

```
title = 'Sleep Hours Over Time';  
datasets = [{  
  label: 'Sleep Hours',  
  data: healthData.slice(-30).map(d => d.sleep),  
  borderColor: 'rgba(201, 203, 207, 1)',  
  backgroundColor: 'rgba(201, 203, 207, 0.2)',  
  tension: 0.4  
}];
```

```
break;
```

```
}
```

```
currentChart = new Chart(ctx, {  
  type: 'line',  
  data: {  
    labels: labels,  
    datasets: datasets  
  },
```

```
options: {  
  responsive: true,  
  maintainAspectRatio: false,
```

```
plugins: {  
  title: {  
    display: true,  
    text: title,  
    font: {  
      size: 16  
    }  
  },  
  legend: {  
    position: 'top'  
  }  
},  
scales: {  
  x: {  
    display: true,  
    title: {  
      display: true,  
      text: 'Date'  
    }  
  },  
  y: {  
    display: true,  
    title: {  
      display: true,  
      text: 'Value'  
    }  
  }  
}
```

```
    }  
  }  
});
```

```
    updateHealthInsights();  
  }
```

```
function updateHealthInsights() {  
  const insights = [];  
  
  const avgHeartRate = healthData.reduce((sum, d) => sum + d.heartRate, 0) /  
healthData.length;  
  
  const avgSystolic = healthData.reduce((sum, d) => sum + d.systolicBP, 0) /  
healthData.length;  
  
  const avgSleep = healthData.reduce((sum, d) => sum + d.sleep, 0) / healthData.length;  
  
  const avgGlucose = healthData.reduce((sum, d) => sum + d.bloodGlucose, 0) /  
healthData.length;  
  
  if (avgHeartRate > 100) {  
    insights.push('<div class="alert alert-warning">⚠ Your average heart rate is elevated.  
Consider stress management techniques.</div>');  
  }  
  
  if (avgSystolic > 130) {  
    insights.push('<div class="alert alert-warning">⚠ Your blood pressure readings suggest  
monitoring by a healthcare provider.</div>');  
  }  
  
  if (avgSleep < 7) {  
    insights.push('<div class="alert alert-warning">⚠ You\'re getting less than recommended  
sleep. Aim for 7-9 hours nightly.</div>');
```

```
}
```

```
if (avgGlucose > 100) {
```

```
    insights.push('<div class="alert alert-warning">⚠ Your glucose levels are slightly elevated.  
Consider dietary adjustments.</div>');
```

```
}
```

```
if (insights.length === 0) {
```

```
    insights.push('<div class="alert alert-success">✅ Your health metrics are within normal  
ranges. Keep up the good work!</div>');
```

```
}
```

```
document.getElementById('healthInsights').innerHTML = insights.join("");
```

```
}
```

```
function exportHealthData() {
```

```
    const csvContent = "data:text/csv;charset=utf-8," +
```

```
        "Date,Heart Rate,Systolic BP,Diastolic BP,Blood Glucose,Weight,Sleep Hours\n" +
```

```
        healthData.map(d => `${d.date},${d.heartRate},${d.systolicBP},${d.diastolicBP},  
${d.bloodGlucose},${d.weight},${d.sleep}`).join('\n');
```

```
    const encodedUri = encodeURIComponent(csvContent);
```

```
    const link = document.createElement('a');
```

```
    link.setAttribute('href', encodedUri);
```

```
    link.setAttribute('download', 'health_data.csv');
```

```
    document.body.appendChild(link);
```

```
    link.click();
```

```
    document.body.removeChild(link);
```

```
}
```

```
document.addEventListener('DOMContentLoaded', function() {

  const tabButtons = document.querySelectorAll('.tab-button');
  tabButtons.forEach((button, index) => {

    button.addEventListener('click', function() {

      const tabs = ['chat', 'prediction', 'treatment', 'analytics'];

      showTab(tabs[index]);

    });

  });

  const avgHeartRate = Math.round(healthData.reduce((sum, d) => sum + d.heartRate, 0) /
healthData.length);

  const avgSystolic = Math.round(healthData.reduce((sum, d) => sum + d.systolicBP, 0) /
healthData.length);

  const avgDiastolic = Math.round(healthData.reduce((sum, d) => sum + d.diastolicBP, 0) /
healthData.length);

  const avgGlucose = Math.round(healthData.reduce((sum, d) => sum + d.bloodGlucose, 0) /
healthData.length);

  const avgSleep = (healthData.reduce((sum, d) => sum + d.sleep, 0) /
healthData.length).toFixed(1);

  document.getElementById('avgHeartRate').textContent = avgHeartRate;

  document.getElementById('avgBloodPressure').textContent = `${avgSystolic}/
${avgDiastolic}`;

  document.getElementById('avgGlucose').textContent = avgGlucose;

  document.getElementById('avgSleep').textContent = avgSleep;

  addMessageToChat('Hello! I\'m your HealthAI assistant. How can I help you with your
health questions today?', 'ai');

});
```


</script>

</body>

</html>