Python 3.x installed
Flask: For creating the web server
NLTK / SpaCy: For natural language processing
Scikit-learn / TensorFlow: For implementing the ML model (if necessary)
SQLite / MongoDB: For storing user data securely
Twilio / SendGrid: For sending appointment reminders (optional)
API Integration: Access to verified medical APIs like Infermedica or Medscape (for accurate health information)

pip install flask nltk scikit-learn pandas numpy spacy

```
from flask import Flask, request, jsonify, render_template
import random
import spacy


# Initialize the Flask app
app = Flask(_name_)


# Load NLP model
nlp = spacy.load("en_core_web_sm")


# Sample medical responses (this would be replaced by an actual API in production)
medical_data = {
  "fever": "It sounds like you have a fever. Drink plenty of fluids and rest. If your temperature exceeds 102°F, please consult a doctor.",
  "headache": "Headaches can be caused by stress, dehydration, or other factors. Try drinking water and resting in a quiet place.",
  "cough": "A cough can be due to a cold, allergies, or something more serious. If it persists for more than a week, seek medical attention.",
}
```

```
# Route for the homepage
@app.route('/')
def home():
  return render_template('index.html')


# Route for chatbot responses
@app.route('/get-response', methods=['POST'])
def get_response():
  user_input = request.json.get('message')
  response = generate_response(user_input)
  return jsonify({"response": response})


# Function to generate a response based on user input
def generate_response(user_input):
  doc = nlp(user_input.lower())
  for token in doc:


       if token.text in medical_data:
         return medical_data[token.text]
     return random.choice(["I'm sorry, I didn't understand that. Can you please elaborate?",
                  "Could you describe your symptoms in more detail?",
                  "Let me know how you're feeling, and I can try to help!"])


# Run the Flask app
if _name_ == "_main_":
  app.run(debug=True)
```