1. Symptom Checker

Interactive Questionnaire: Users answer a series of questions about their symptoms to receive tailored advice.

Severity Assessment: Provide a severity rating (e.g., mild, moderate, severe) and recommend appropriate actions.

Follow-Up Questions: Ask additional questions based on initial responses to refine the advice given.

Pseudocode :

```
def symptom_checker(symptoms):
    # Interactive Questionnaire
    questions = generate_questions(symptoms)
    responses = get_user_responses(questions)

    # Severity Assessment
    severity = assess_severity(responses)

    # Follow-Up Questions
    if severity == 'moderate' or severity == 'severe':
        follow_up_questions = generate_follow_up_questions(responses)
        follow_up_responses = get_user_responses(follow_up_questions)
        severity = reassess_severity(follow_up_responses)

    return provide_advice(severity)
```

2. Step-by-Step First Aid Instructions

Visual Aids: Include images or short videos demonstrating first aid techniques.

Voice Guidance: Offer audio instructions for hands-free assistance.

Quick Access Buttons: Provide buttons for common emergencies (e.g., "Burns," "Cuts," "Choking") for rapid access.

Pseudocode :

```
def first_aid_instructions(emergency_type):
    # Visual Aids
```

```
    visual_aids = get_visual_aids(emergency_type)

    # Voice Guidance
    audio_instructions = get_audio_instructions(emergency_type)

    # Quick Access Buttons
    quick_access = get_quick_access_buttons()

    return display_instructions(visual_aids, audio_instructions, quick_access)
```

3. Emergency Protocols

Countdown Timers: For time-sensitive actions (e.g., CPR), include countdown timers to guide users through the process.

Emergency Contacts: Allow users to store emergency contact information and quickly call or message them.

Automated Alerts: Send automated alerts to emergency contacts if the user indicates a severe emergency.

Pseudocode :

```
def emergency_protocols(emergency_type):
    # Countdown Timers
    timer = start_countdown(emergency_type)

    # Emergency Contacts
    contacts = get_emergency_contacts()

    # Automated Alerts
    if emergency_type == 'severe':
        send_automated_alerts(contacts)

    return guide_user_through_protocol(timer)
```

4. Medication Information

Search Functionality: Users can search for medications by name or symptom.

Dosage Reminders: Set reminders for taking medications at the correct times.

Interaction Warnings: Provide warnings about potential interactions between different medications.

Pseudocode :

```
def medication_information(medication_name):
    # Search Functionality
    medication_details = search_medication(medication_name)

    # Dosage Reminders
    set_dosage_reminders(medication_details)

    # Interaction Warnings
    interactions = check_interactions(medication_details)

    return display_medication_info(medication_details, interactions)
```

5. Health Tips and Preventive Care

Daily Tips: Send daily health tips or preventive care advice.

Seasonal Advice: Offer tips relevant to the current season (e.g., flu prevention in winter).

Healthy Habits Tracker: Help users track habits like hydration, exercise, and sleep.

Pseudocode :

```
def health_tips():
    # Daily Tips
    daily_tip = get_daily_tip()

    # Seasonal Advice
    seasonal_advice = get_seasonal_advice()

    # Healthy Habits Tracker
    habits_tracker = track_healthy_habits()

    return display_health_tips(daily_tip, seasonal_advice, habits_tracker)
```

6. Resource Links

Trusted Sources: Link to reputable health websites (e.g., NHS, Mayo Clinic) for more detailed information.

Local Services: Provide links to local health services, such as pharmacies or clinics.

Educational Articles: Share articles on various health topics for users to read and learn more.

Pseudocode :

```
def resource_links(topic):
    # Trusted Sources
    trusted_sources = get_trusted_sources(topic)

    # Local Services
    local_services = get_local_services()

    # Educational Articles
    articles = get_educational_articles(topic)

    return display_resource_links(trusted_sources, local_services, articles)
```

Improvements in the future?

7. Language Support

Multilingual Interface: Offer the chatbot in multiple languages to cater to a diverse audience.

Translation Feature: Allow users to input text in their language and receive translated responses.

Cultural Sensitivity: Ensure health advice is culturally appropriate and sensitive.

8. User-Friendly Interface

Intuitive Design: Simple, clean design with easy navigation.

Favorites: Allow users to save frequently accessed information or instructions.

Feedback Option: Provide a way for users to give feedback on the chatbot's performance and suggest improvements.

Additional Ideas

Interactive Scenarios: Create simulated emergency scenarios for users to practice their response.

Voice Assistance: Integrate with voice assistants like Alexa or Google Assistant for hands-free operation.

Location-Based Services: Use GPS to provide information on nearby medical facilities or pharmacies.

Personalized Health Records: Allow users to store and access their medical history, allergies, and emergency contacts.

Common issues searched online:

1. **Falls and Slips**:
    a. How to treat a sprained ankle.
    b. What to do if someone falls and hits their head.
    c. First aid for broken bones.

```
def treat_sprained_ankle():
    # Steps to treat a sprained ankle
    return "Rest, ice, compression, and elevation (RICE). Avoid putting weight on
the ankle."

def head_injury_fall():
    # Steps to take if someone falls and hits their head
    return "Keep the person still, apply a cold compress, and seek medical
attention if they lose consciousness or show signs of a concussion."

def broken_bone_first_aid():
    # First aid for broken bones
    return "Immobilize the area, apply ice packs, and seek medical help
immediately."
```

2. **Burns**:
    a. How to treat a burn from hot water or steam.
    b. First aid for kitchen burns.
    c. What to do for a chemical burn.

```
def treat_burn_hot_water():
    # Steps to treat a burn from hot water or steam
    return "Cool the burn under running water for at least 10 minutes, cover with
a sterile dressing, and avoid using ice."

def kitchen_burn_first_aid():
    # First aid for kitchen burns
    return "Cool the burn, remove any tight items around the burn area, and cover
with a clean cloth."
```

```
def chemical_burn_first_aid():
    # What to do for a chemical burn
    return "Remove contaminated clothing, rinse the area with water for at least
20 minutes, and seek medical help."
```

3. **Cuts and Lacerations**:
    a. How to stop bleeding from a deep cut.
    b. First aid for minor cuts and scrapes.
    c. When to get stitches for a cut.

```
def stop_bleeding_cut():
    # How to stop bleeding from a deep cut
    return "Apply pressure with a clean cloth, elevate the wound, and seek
medical attention if bleeding doesn't stop."

def minor_cuts_first_aid():
    # First aid for minor cuts and scrapes
    return "Clean the wound with water, apply an antibiotic ointment, and cover
with a bandage."

def when_to_get_stitches():
    # When to get stitches for a cut
    return "Seek stitches if the cut is deep, gaping, or located on a joint or
face."
```

4. **Choking**:
    a. How to perform the Heimlich maneuver.
    b. What to do if a child is choking.
    c. First aid for choking in adults.

```
def heimlich_maneuver():
    # How to perform the Heimlich maneuver
    return "Stand behind the person, make a fist above their navel, and thrust
inward and upward."

def child_choking_first_aid():
    # What to do if a child is choking
    return "Perform back blows and chest thrusts, and call emergency services if
the object doesn't come out."

def adult_choking_first_aid():
```

```
    # First aid for choking in adults
    return "Perform the Heimlich maneuver and call emergency services if the
person cannot breathe."
```

5. **Poisoning**:
    a. What to do if someone ingests poison.
    b. Symptoms of carbon monoxide poisoning.
    c. First aid for accidental medication overdose.

```
def ingest_poison_first_aid():
    # What to do if someone ingests poison
    return "Call emergency services, do not induce vomiting, and provide
information about the poison."

def carbon_monoxide_poisoning():
    # Symptoms of carbon monoxide poisoning
    return "Symptoms include headache, dizziness, and confusion. Move to fresh
air immediately and seek medical help."

def medication_overdose_first_aid():
    # First aid for accidental medication overdose
    return "Call emergency services, provide information about the medication,
and do not induce vomiting."
```

6. **Electrical Injuries**:
    a. First aid for electric shock.
    b. What to do if someone is electrocuted.
    c. How to treat electrical burns.

```
Def electric_shock_first_aid():
    # First aid for electric shock
    return "Turn off the power source, call emergency services, and do not touch
the person if they are still in contact with the source."

def electrocution_first_aid():
    # What to do if someone is electrocuted
    return "Ensure the power is off, call emergency services, and perform CPR if
necessary."

def electrical_burn_first_aid():
    # How to treat electrical burns
```

```
    return "Cool the burn with water, cover with a sterile dressing, and seek
medical help."
```

7. **Allergic Reactions**:
    a. How to use an EpiPen.
    b. First aid for severe allergic reactions.
    c. Symptoms of anaphylaxis and what to do.

```
def use_epipen():
    # How to use an EpiPen
    return "Remove the safety cap, press the injector against the outer thigh,
and hold for 10 seconds."

def severe_allergic_reaction_first_aid():
    # First aid for severe allergic reactions
    return "Use an EpiPen if available, call emergency services, and keep the
person calm and lying down."

def anaphylaxis_symptoms():
    # Symptoms of anaphylaxis and what to do
    return "Symptoms include difficulty breathing, swelling, and hives. Use an
EpiPen and seek emergency medical help."
```