Vivekanand Education Society's Institute Of Technology
Department Of Information Technology
DSA mini Project
A.Y. 2025-26

Title: Health Checker

Sustainability Goal : Advancing Health Awareness and Accessible Medical Guidance

Domain: Data Structures & Algorithms
Member: Yash Rai

Mentor Name: Kajal Jewani


1 NO POVERTY
2 ZERO HUNGER
3 GOOD HEALTH AND WELL-BEING
4 QUALITY EDUCATION
5 GENDER EQUALITY
6 CLEAN WATER AND SANITATION
7 AFFORDABLE AND CLEAN ENERGY
8 DECENT WORK AND ECONOMIC GROWTH
9 INDUSTRY, INNOVATION AND INFRASTRUCTURE
10 REDUCED INEQUALITIES
11 SUSTAINABLE CITIES AND COMMUNITIES
THE GLOBAL GOALS
For Sustainable Development
12 RESPONSIBLE CONSUMPTION AND PRODUCTION
13 CLIMATE ACTION
14 LIFE BELOW WATER
15 LIFE ON LAND
16 PEACE AND JUSTICE STRONG INSTITUTIONS
17 PARTNERSHIPS FOR THE GOALS

Content

1. Introduction to the Project
2. Problem Statement
3. Objectives of the Project
4. Scope of the Project
5. Requirements of the System (Hardware, Software)
6. ER Diagram of the Proposed System
7. Data Structure & Concepts Used
8. Algorithm Explanation
9. Time and Space Complexity
10. Front End
11. Implementation
12. Gantt Chart
13. Test Cases
14. Challenges and Solutions
15. Future Scope
16. Code
17. Output Screenshots
18. Conclusion
19. References (in IEEE Format)

Introduction to Project

Health Checker is a healthcare technology project developed using C programming with a binary decision tree architecture. This system serves as a medical assistant that conducts preliminary health assessments through intelligent symptom analysis.

By guiding users through an interactive question-and-answer process, Health Checker evaluates symptoms for over fifteen common medical conditions while providing reliable medication recommendations and implementing a crucial severity-based triage system to support informed healthcare decisions.

Problem Statement

In today's healthcare landscape, several critical issues exist like

1) Limited Access to Healthcare
2) Information Overload
3) Delayed Care

There is a need for a systematic, tree-based diagnostic tool that provides reliable preliminary assessment, clear recommendations, and guidance on when professional medical care is necessary.

Objectives of the project

- Implement binary tree for symptom assessment
- Provide preliminary health evaluation
- Offer treatment guidance (meds + home remedies)
- Determine urgency levels
- Health education & prevention tips
- Integrate maps using OpenStreetMaps API

Scope of the project

Target Users:

- General public seeking preliminary health guidance
- Students learning about common health conditions
- Individuals in remote areas with limited healthcare access
- People wanting to assess symptom severity before doctor visits

Requirements of the system (Hardware, software)

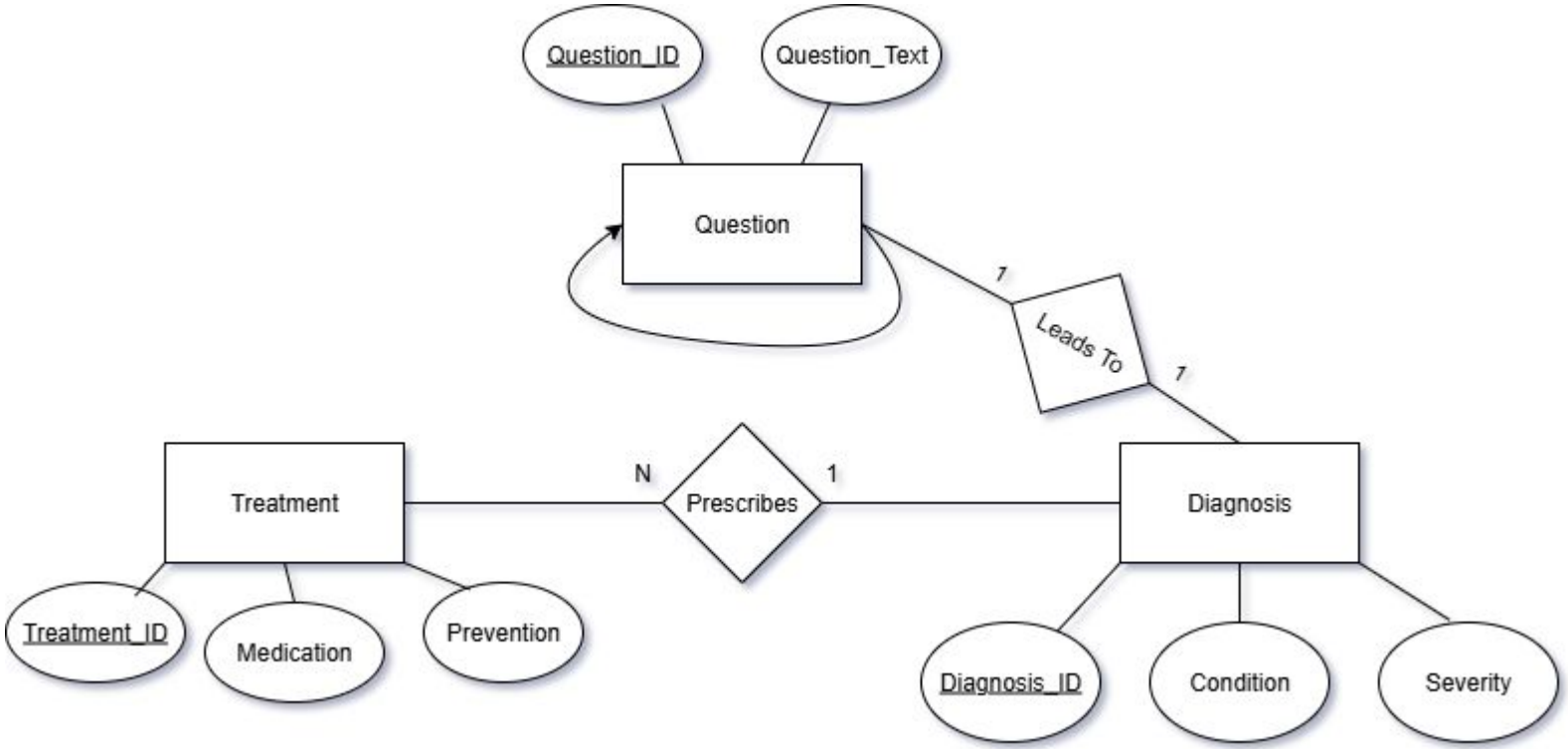
Hardware:

- Processor: Intel i3 or higher (Any modern processor)
- RAM: Minimum 2GB (4GB recommended)
- Storage: 50MB free disk space

Software:

- OS: Windows 10/11
- Compiler: GCC (GNU Compiler Collection) - MinGW/TDM-GCC for Windows
- Terminal: CMD/PowerShell

ER Diagram of the Proposed System



Data Structures and Concepts used

➢ **Binary Tree**

Each node represents either:

❑ QUESTION_NODE: A symptom-related question

❑ DIAGNOSIS_NODE: Final diagnosis (leaf node)

Time and Space Complexity

1) **Time Complexity:**

Each of the n nodes is created once during tree building.

Hence the time complexity is **O(n) .**

2) **Space complexity:**

We need to store all n nodes in memory simultaneously.

Hence the space Complexity is **O(n) .**

Front End

- **Interactive Q&A Flow** - Guided yes/no questions about symptoms
- **Severity Levels** - Emergency, Urgent, Moderate and Mild
- **Comprehensive Diagnosis Reports including:**
  - Condition description and severity assessment
  - Home remedies.
  - Specific medication guidance with dosages
- **Multiple Health Categories covering:** Respiratory issues, Digestive problems, Headaches and pain, General wellness, etc

Implementation

- Uses **binary decision tree** to navigate symptom assessment in C language.

- **Tree nodes contain:**
  - Question nodes for symptom interrogation
  - Diagnosis nodes for final condition assessment

- **Tree Architecture:**
  - **Root Node**: Starts with emergency symptom detection (chest pain, breathing difficulty)
  - **Internal Nodes**: Question nodes that branch based on Yes/No responses
  - **Leaf Nodes:** Diagnosis nodes that provide final assessment

Gantt Chart

Task	Week 1	Week 2	Week 3	Week 4
Project Planning	✓			
Data Structure Design	✓			
Core Logic (Tree)		✓		
Traversal Algorithm		✓		
UI/CLI Implementation			✓	
Testing & Debugging			✓	✓
Documentation & Report				✓

Test Cases

Test Case ID	Input Path	Expected Output	Status
TC-01	Severe chest pain = YES	EMERGENCY: Potential Medical Emergency	✅ Pass
TC-02	Fever >3 days + body aches + cough	Influenza (MODERATE)	✅ Pass
TC-03	Fever <3 days + mild symptoms	Common Viral Infection (MILD)	✅ Pass

Future Scope

- Develop Desktop application
- Develop mobile application (Android/IOS)
- Expand diagnosis database to 50+ conditions .
- Voice interface for hands-free operation
- Multi-language support
- Connect with wearable devices for vital signs

Code

Diagnosis Structure

```
typedef struct {
    char condition[MAX_TEXT];
    Severity severity;
    char description[MAX_TEXT];
    char remedies[MAX_TEXT];
    char medications[MAX_TEXT];
    char when_to_see_doctor[MAX_TEXT];
    char prevention[MAX_TEXT];
} Diagnosis;
```

TreeNode Structure

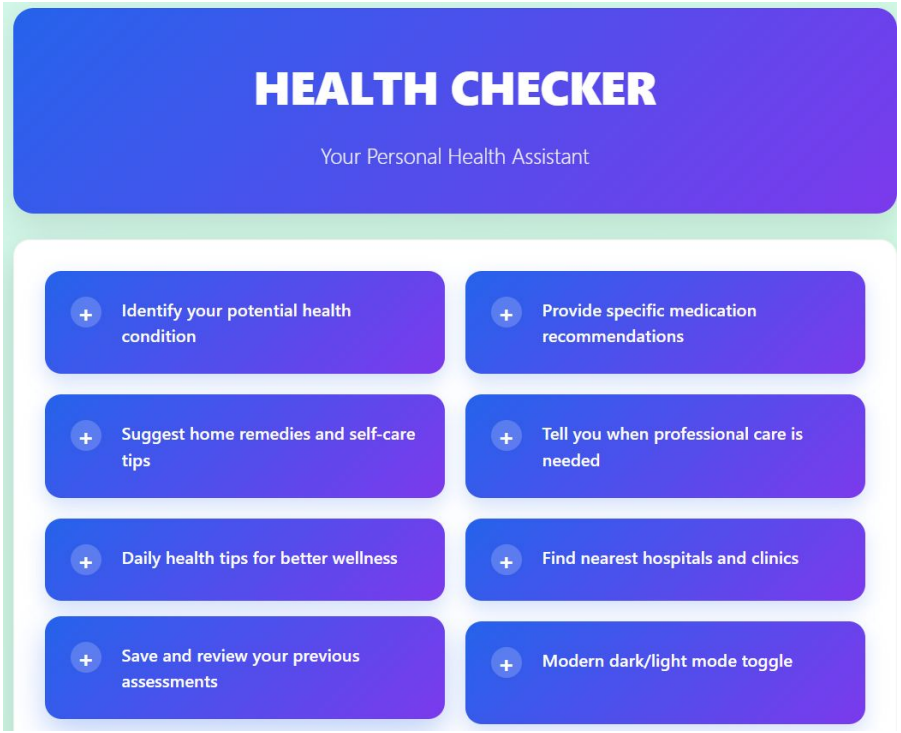
```
typedef struct TreeNode {
    NodeType type;
    char text[MAX_TEXT];
    struct TreeNode *yes_branch;
    struct TreeNode *no_branch;
    Diagnosis *diagnosis;
} TreeNode;
```

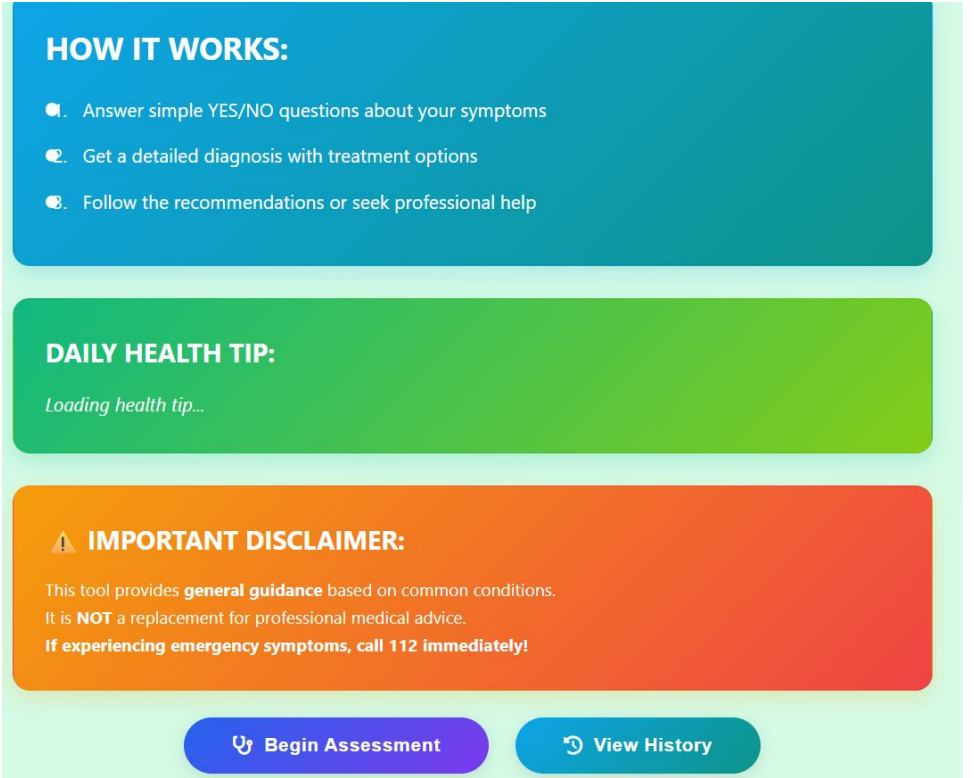
Code

Overpass API :

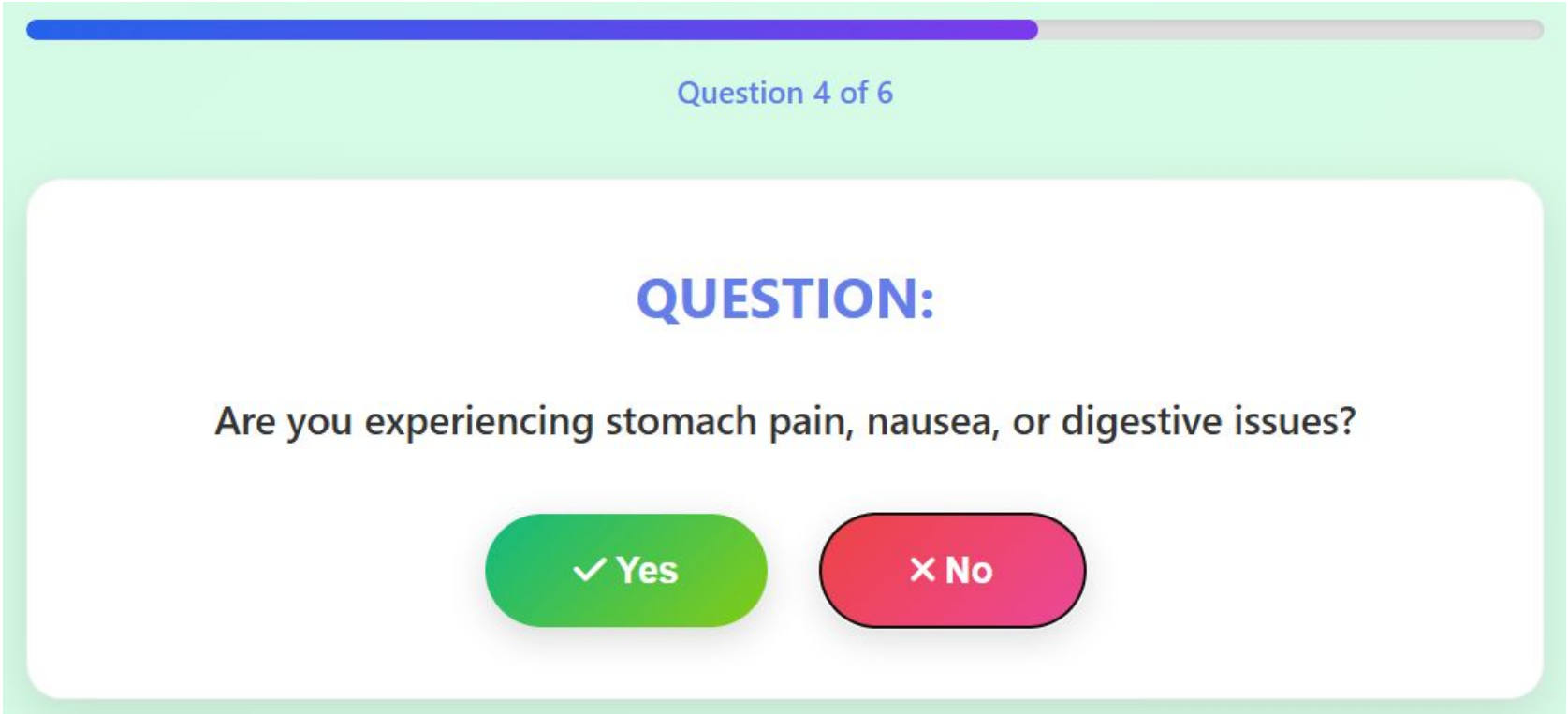
```
async function searchNearbyMedicalFacilities(lat, lng) {
  try {
    const radius = 10000;
    const query = `
      [out:json][timeout:25];
      (
        node["amenity"="hospital"](around:${radius},${lat},${lng});
        way["amenity"="hospital"](around:${radius},${lat},${lng});
        node["amenity"="clinic"](around:${radius},${lat},${lng});
        way["amenity"="clinic"](around:${radius},${lat},${lng});
        node["amenity"="doctors"](around:${radius},${lat},${lng});
        way["amenity"="doctors"](around:${radius},${lat},${lng});
      );
      out body;
      >;
      out skel qt;
    `;
    const response = await fetch("https://overpass-api.de/api/interpreter", {
      method: "POST",
      body: query,
    });
```

Output Screenshots





Output Screenshots



Output Screenshots

INFLUENZA (FLU)

[!] MODERATE

SEVERITY LEVEL:

MODERATE

WHAT IS THIS?

You likely have the flu, a viral infection affecting the respiratory system. Most people recover within 1-2 weeks.

HOME REMEDIES & SELF-CARE:

Rest, drink plenty of fluids (water, warm soups), use a humidifier, gargle with salt water

RECOMMENDED MEDICATIONS:

Acetaminophen (Tylenol) 500-1000mg every 6 hours OR Ibuprofen (Advil) 400mg every 6 hours for fever/pain. Antiviral medications (Tamiflu) if prescribed within 48 hours of symptom onset

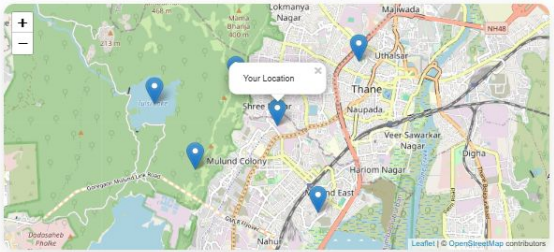
WHEN TO SEE A DOCTOR:

If fever persists beyond 5 days, difficulty breathing develops, or symptoms worsen

PREVENTION TIPS:

Annual flu vaccination, frequent handwashing, avoid close contact with sick individuals

NEAREST HOSPITALS & CLINICS:



Nearby Medical Facilities:

General Hospital
123 Main St, City Center
1.2 km

City Medical Center
456 Health Ave, Downtown
2.5 km

Community Clinic
789 Wellness Blvd, Suburb
3.1 km

Conclusion

- **Impacts:**
  - Provides preliminary health assessment without immediate doctor visit
  - Helps users make informed decisions about seeking medical care
  - Reduces unnecessary emergency room visits for minor conditions
  - Educates users about common health conditions

References

- The C Programming Language - Brian W. Kernighan & Dennis M. Ritchie
- C Programming: A Modern Approach - K. N. King
- Data Structures Using C - Aaron M. Tenenbaum
- Programming in ANSI C - E. Balagurusamy
- C Programming Tutorial - GeeksForGeeks