SARANATHAN COLLEGE OF ENGINEERING

PANJAPPUR, TIRUCHIRAPPALLI – 620012.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MINI PROJECT NOTE BOOK

CS8611/ MINI PROJECT

JUNE-2023

ASSISTING GENERAL PUBLIC IN GOVERNMENT HOSPITALS USING BOT

A PROJECT REPORT

Submitted by

APARNA GAYATHRI N	**813820104011**
ANANDA DHARSHINEE M.S	**813820104009**
BHAVYA SRII A	**813820104018**
DHEEPIKA R	**813820104025**

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

SARANATHAN COLLEGE OF ENGINEERING, TRICHY

ANNA UNIVERSITY: CHENNAI 600 025

2023

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **“ASSISTING GENERAL PUBLIC IN GOVERNMENT HOSPITALS USING BOT”**

is the bonafide work of

APARNA GAYATHRI N	**813820104011**
ANANDA DHARSHINEE M.S	**813820104009**
BHAVYA SRII A	**813820104018**
DHEEPIKA R	**813820104025**

who carried out the Mini-project work under my supervision.

SIGNATURE
Dr. V PUNITHA,M.E.,Ph.D
HEAD OF THE DEPARTMENT
PROFESSOR
Computer Science and Engineering,
Saranathan College of Engineering,
Panjapur,
Tiruchirapalli-620 012

SIGNATURE
Dr.R SENTHAMILSELVI,M.E,Ph.D
SUPERVISOR
ASSOCIATE PROFESSOR
Computer Science and Engineering,
Saranathan College of Engineering,
Panjapur,
Tiruchirapalli-620 012

VIVA - VOCE EXAMINATION

ASSISTING GENERAL PUBLIC IN GOVERNMENT HOSPITALS USING BOT

Submitted by

APARNA GAYATHRI N	**813820104011**
ANANDA DHARSHINEE M.S	**813820104009**
BHAVYA SRII A	**813820104018**
DHEEPIKA R	**813820104025**

The Viva - Voce Examination of this Mini-Project (CS8611) work done as a part of B.E. Computer Science and Engineering was held on______________________.

INTERNAL EXAMINER

EXTERNAL EXAMINER

SARANATHAN COLLEGE OF ENGINEERING

Venkateswara Nagar ,Panjappur

Tiruchirapalli-620012

VISION OF THE INSTITUTION

Impart an inclusive engineering education that beyond being a facilitator for a career and rudimentary skills, equips the students to offer ethically & environmentally conscious solutions to societal issues.

MISSION OF THE INSTITUTION

Develop the Institution into a Model Self Financing College for Engineering and Technology. Deliver Professional Training to our students with state-of-the art laboratories and converting them into Technocrats of international repute.

1. Create a nurturing, holistic environment of engineering education to facilitate every student realize their full potential.
2. Strive to make the students strong in basic concepts armed with appropriate skills to enhance one’s ability to apply the knowledge to provide solutions to real time issues.
3. Maintain an ambience that facilitates the students to strengthen their ethical value systems.
4. Actively promote R&D and institute-industry interaction.

VISION OF THE DEPARTMENT

- To evolve as a centre of academic excellence and advanced research in Computer Science and Engineering discipline.

MISSION OF THE DEPARTMENT

- To inculcate in students a profound understanding of fundamentals related to discipline.
- To inculcate skills, attitudes and their applications in solving real world problems with an inclination towards societal issues and research.
- To promote research in the emerging areas of computer science and technology

PROGRAM EDUCATIONAL OBJECTIVES (PEOS)

PEO1: Acquire strong foundation in the mathematical, scientific and engineering fundamentals necessary to formulate, solve and analyze engineering problems.

PEO2: Develop the ability to analyze the requirements of the software, understand the technical specifications, design and provide novel engineering solutions and efficient software/hardware designs.

PEO3: Have exposure to emerging cutting edge technologies, adequate training & opportunities to work as teams on multidisciplinary projects with effective communication skills and leadership qualities.

PEO4: Have awareness on the life-long learning and prepare them for research development and consultancy.

PEO5: Have a successful career and work with values & social concern bridging the digital divide and meet the requirements of Indian and multinational companies.

PROGRAM SPECIFIC OUTCOME (PSO)

1. **PS01**: Foundation of mathematical concepts: To use mathematical methodologies to crack problem using suitable mathematical analysis, data structure and suitable algorithm.
2. **PSO2**: Foundation of Computer System: the ability to interpret the fundamental concepts and methodology of computer systems. Students can understand the functionality of hardware and software aspects of computer systems.
3. **PSO3**: Foundations of Software development: the ability to grasp the software development lifecycle and methodologies of software systems. Possess competent skills and knowledge of software design process. Familiarity and practical proficiency with a broad area of programming concepts and provide new ideas and innovations towards research

Program Outcomes: (POs)

Sno	Program Outcomes
1	**PO1 Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and engineering specialization to the solution of complex engineering problems.
2	**PO2 Problem Analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3	**PO3: Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4	**PO4 : Conduct Investigations of Complex Problems** Use research-based knowledge and research methods including design of exercises, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5	**PO5 : Modern Tool Usage** : Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6	**PO6 : The Engineer and Society** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7	**PO7: Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8	**PO8 : Ethics** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9	**PO9: Individual and Team Work** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10	**PO10 : Communication** Communicate effectively on complex engineering activities with the engineering community

	and with society at large, communication such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11	**PO11 : Project management and finance** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12	**PO12 :Life-long learning** Recognize the need for, and have the preparation n and ability to engage in independent and life-long learning in the broadest context of technological change

NBA Code for the Subject: C319 Subject Code: CS8611

Title: MINI PROJECT

COURSE OUTCOMES: At the end of this course, Student will be able to

Course Code	Course outcome Description
C319.1	**Gather and interpret technical literature to formulate a project proposal to solve challenging practical problems.**
C319.2	**Identify SDLC model and prepare software requirements specification.**
C319.3	**Design the software architecture.**
C319.4	**Apply modern tools for implementation using best coding practices and.testing at various levels of the project.**
C319.5	**Document the technical report on identified topic and present the ideas with effective communication skills**
C319.6	**Learn the concepts of project management and to work effectively as a member in team.**

SUBJECT CODE : CS8611 NBA CODE : C319 TITLE : MINI PROJECT														
Course Code	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
C319.1	2	2	2	2	2	2	-	1	2	2	2	2	2	2
C319.2	2	2	2	2	2	2	-	1	2	2	2	2	2	2
C319.3	2	2	2	2	2	2	-	1	2	2	2	2	2	2
C319.4	2	2	2	2	2	2	-	1	2	2	2	2	2	2
C319.5	2	2	2	2	2	2	-	1	2	2	2	2	2	2
C319.6	2	2	2	2	2	2	-	1	2	2	2	2	2	2

TABLE OF CONTENTS

ABSTRACT

The use of chat bots in the healthcare industry has revolutionized the way patients and visitors access information and services in hospitals. Government hospitals, in particular, face the challenge of assisting a large number of patients and visitors, especially during peak periods of demand. To address this challenge, we have come up with a solution by developing a bot which helps the general public by providing guidance specifically in finding locations inside the hospital. By using the bot, government hospitals can provide an efficient and seamless experience for patients and visitors. The bot can help individuals navigate complex hospital layouts and find specific locations quickly and easily, such as the emergency room, pharmacy, or specific departments. To implement a chat bot to assist the general public in government hospitals, a map of the hospital's layout is required, including the location of all departments, rooms, and key landmarks. This map can be used to create a graph, with the hallways and rooms representing the nodes and the distances between them representing the edges. Using the Dijkstra algorithm, the bot can find the shortest path between the user's current location and the desired location. In addition to providing assistance with finding locations using chat bots, voice bots are also included that can offer a personalized experience to visually impaired people. The bot guides both the general public and the visually impaired people as it acts as both voice and text assistant. By using a Braille keyboard, visually impaired people can access the bot in the same way as the normal people.

CHAPTER 1
INTRODUCTION

1.1 About the Project

Generally, In Government Hospitals there exists experienced and skilled doctors and Technicians, But still some public feels hesitated to get service in these hospitals. So, to avoid this from happening we have founded a way to put up with these, that is our Bot.

By using this bot, government hospitals can provide an efficient and seamless experience for patients and visitors. The user interacts with the bot by means of Keyboard. Especially the most beneficiary is the Visually impaired people because the bot takes input from the user in means of keyboard so, it is very for the visually impaired people to interact with it. The bot consists the layout of the hospital and which in turns it assists the user by directing them with the distance from the user's place to the requested destination by the user.

The objective of our project is to assist the general people as well as the visually impaired people by means of direction within the government Hospitals. This application is very much user friendly and users can also place their queries for the future enhancement of the application

Our application includes the following:

- Animated Guide(Bot)
- Hospital layout
- Numbered Destinations

Here, many bots are placed in a distinctive manner in the hospital, so the distance will be based on the distance between the bots.

1.2 Purpose

The purpose of this application is to assist the general public and the visually impaired people in terms of guiding them from one place to another within the hospital Building. This makes people to be more comfortable in using the Government Hospital and it makes the visually impaired people to be more confident and independent while coming to the hospital.

1.3 Scope

Our platform focuses on reaching out all the people and visually impaired people who are hesitated to get serviced by Government Hospital. It includes all kinds of people , without any specification of gender, race, culture etc,

The public can take help from the bot to reach their destination by using the bot. So, they did not want to take help from anybody in the hospital.

CHAPTER 2
PROPOSED SYSTEM

2.1 Working of the System

The Bot will always remain in the active state for the benefit of the user especially the visually impaired people. The user side of the application is that the user interaction will be made by pressing enter on the keyboard and that’s how the bot is will start to interact with the user. The user can exit the application by clicking the exit option regardless of what stage the user id in the application.

2.2 Destination Finding by the System

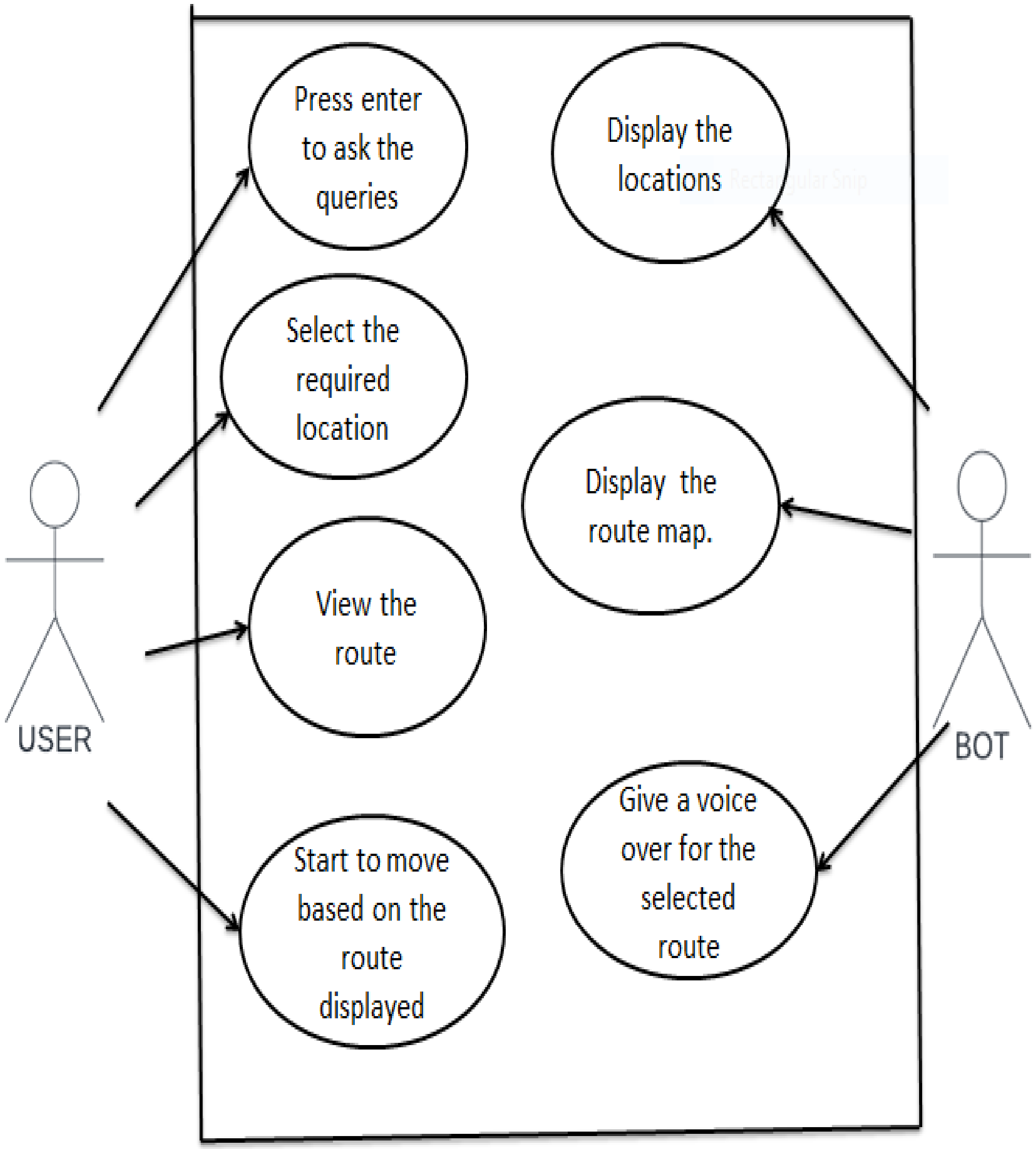
The user will then enter the required destination place by pressing the mapped integers which is displayed on to them with the help of display and that’s how the bot find’s the user’s destination.

2.3 Directing the Public by the System

Upon knowing the source and the required destination of the user the bot will display the shortest route between the places and provide the distance in foots because visually impaired people tends to count the foots while walking.

The bot will do both the job of displaying the directions and the route with the help of Chatbot and as well as the audio of the directions is said by the voicebot.

CHAPTER 3
USECASE DIAGRAM


Press enter to ask the queries
Display the locations
Rectangular Snip
Select the required location
Display the route map.
View the route
USER
BOT
Give a voice over for the selected route
Start to move based on the route displayed

CHAPTER 4
REQUIREMENTS

4.1 Functional Requirements

- The application will allow any type of users including general public as well as visually imapaired to interact with the user.
- The application will allow the user to view the hospital layout and to enter the required destination
- The application will allow the user to exit regardless of what satge the user is in the application

4.2 Software Requirements:

1.Language: Python, JavaScript, HTML, CSS

2.Tool: Visual Studio Code

3.Framework: Bootstrap

4.Operating System: Windows

4.3 Non-Functional Requirements

4.3.1 Security

This application uses the hospital layout and thus it maintains the security of the hospital by guiding the people to the required location.

4.3.2 Availability

The application is available anywhere and at any time. It is not time bounded. The bot will be present in the different blocks, floors etc.., Even the bot can be present in the same floor within certain distance where the people can use it more efficiently.

4.3.3 Maintainability

There should be a facility to store and maintain the hospital layout inn the database if required. It will also provide a backup to maintain it. The bot is easily accessible, so it does not need any extra care.

4.3.4 Reliability

As our bot does not need any login credentials to access it. So, it so reliable as every general public can use it without any use pre-requisite.

4.3.5 Reusability

The same application can be used as the base for creation of any further improvement in the application as it is reusable .

Usability: User can achieve their target of reaching their required destination by using our bot.

CHAPTER 5
MODULE EXPLANATION

5.1 Modules Present in the System

1. Welcome Page

2. User Query Page

3. Route highlighted map

4 . Bot Guidance Page

5.1.1 Welcome Page

This module enables the user to find out the existence of the bot by using the voice bot. Here, the voice bot will repeatedly say that "I AM HERE TO HELP YOU" so, the general public and the visually impaired can find the existence of the bot.

5.1.2 User Query Page

This page will display the query of the users where they want to reach. For the particular query there will be the availability of number, by clicking the particular number on the braille keyboard , the general public as well as visually impaired can reach the destination without any problem.

5.1.3 Route highlighted map

After clicking the particular query, this module will display the map where the user can find their route from source to destination. The route will be highlighted in different colors, so that the route will be clearly visible and the user can reach the destination.

5.1.4 Bot Guidance Page

Our bot will act as both chat and voice assistance. The bot will guide the public by giving the assistance in form of voice and chat.

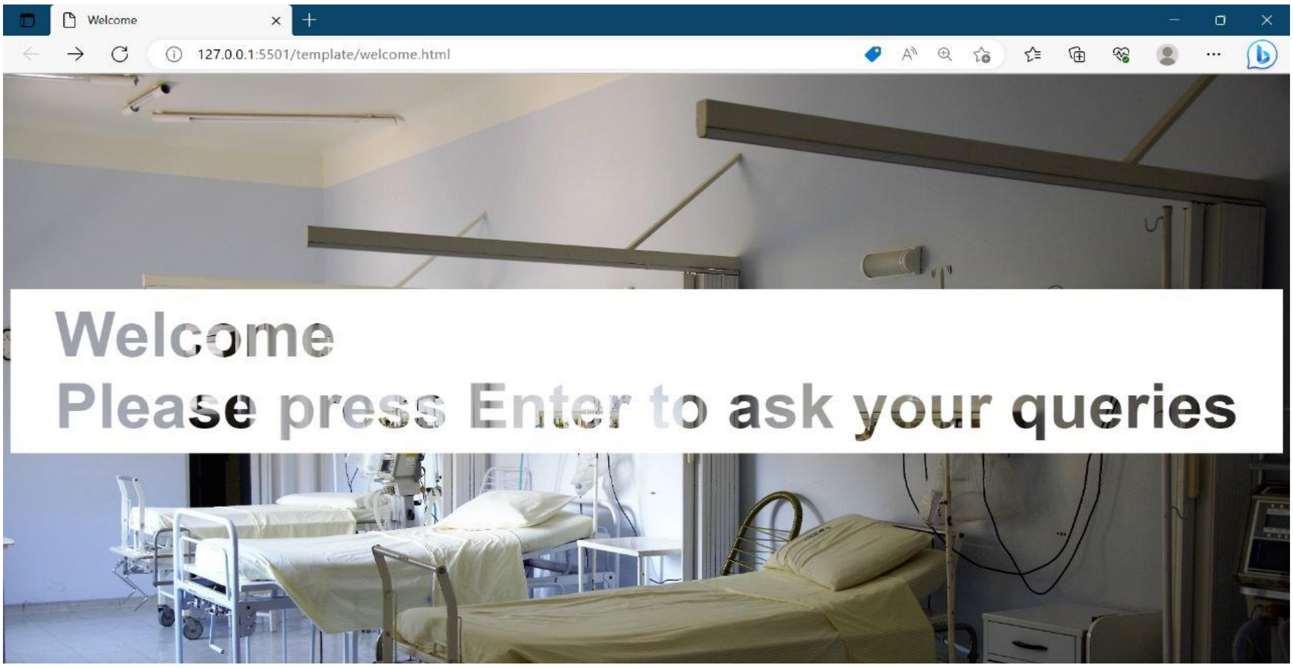
CHAPTER 6

CONCLUSION

Thus , the project serves as a good and friendly platform where users can interact with the bot via keyboard to reach the required destination(place/block) within the hospital premises. By using bot the users will find it very much comfortable to reach out to the government hospital if needed and it will in turn increase the rate of public coming to the hospital and it will develop the government hospital's facilities and funding by the government

CHAPTER 7
SCREENSHOTS

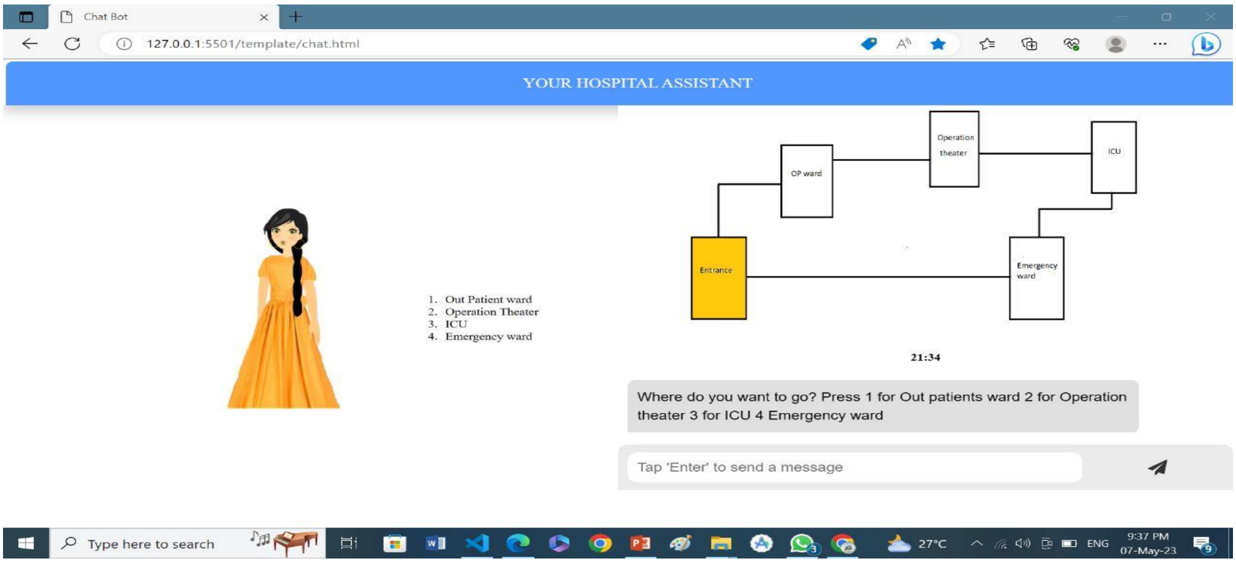
SEQUENCE OF STEPS:



Welcome
Please press Enter to ask your queries

Fig 1: Welcome Page

STEP 2:



YOUR HOSPITAL ASSISTANT

1. Out Patient ward
2. Operation Theater
3. ICU
4. Emergency ward

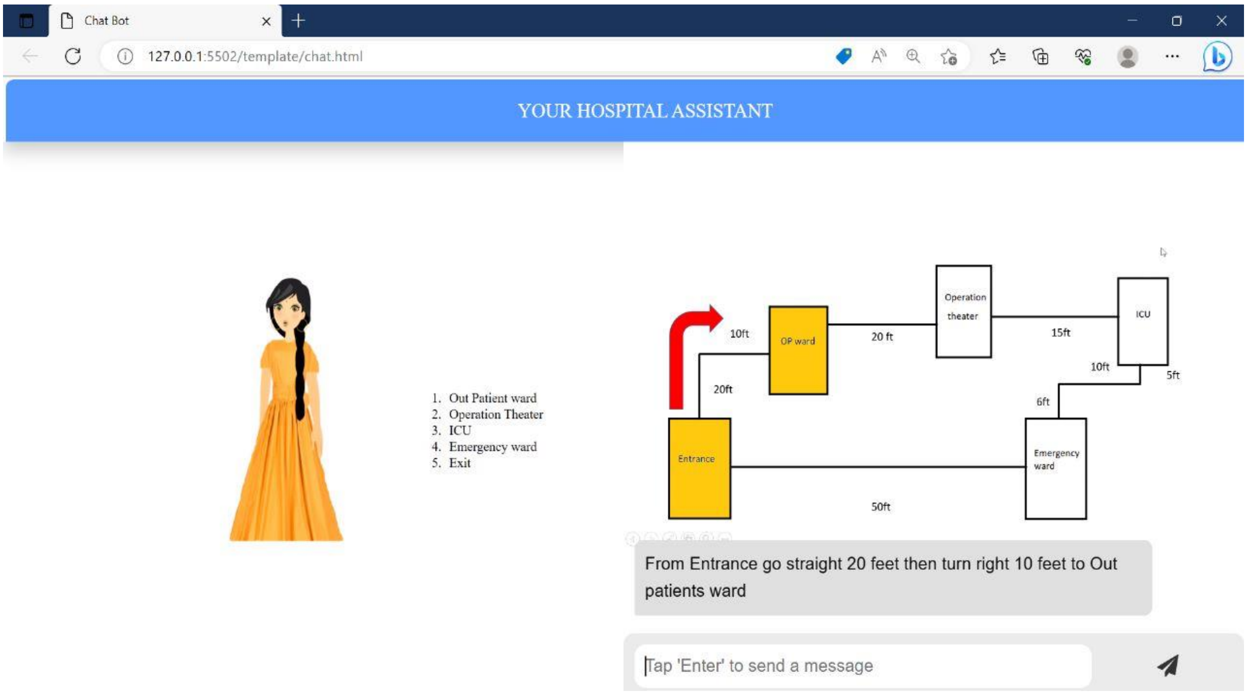
21:34

Where do you want to go? Press 1 for Out patients ward 2 for Operation theater 3 for ICU 4 Emergency ward

Tap 'Enter' to send a message

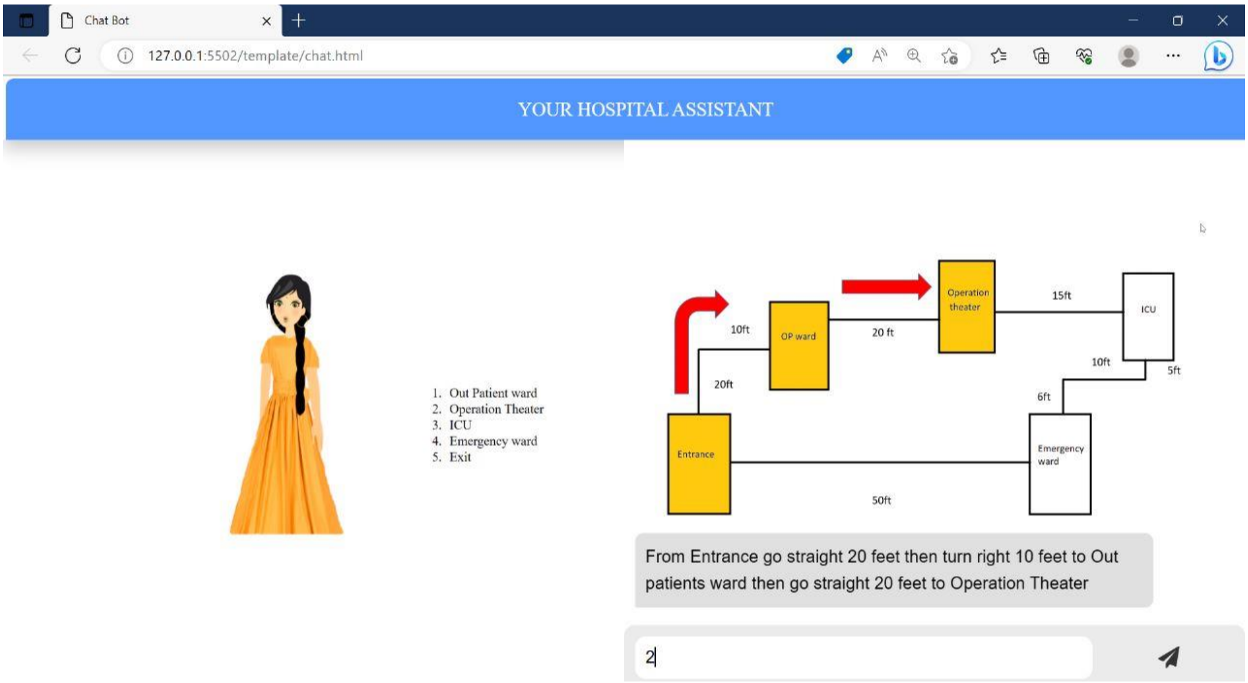
Fig 2: Acknowledgment of user’s current location.

STEP 3:



From Entrance go straight 20 feet then turn right 10 feet to Out patients ward

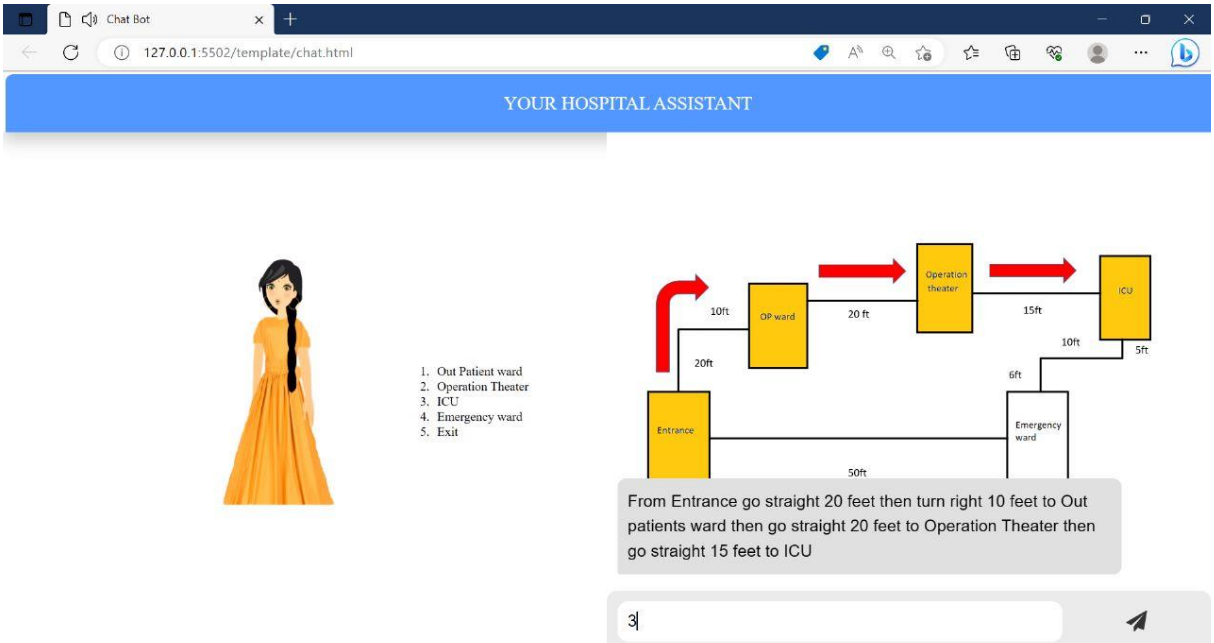
Fig 3: Direction from Entrance to Outpatient Ward



From Entrance go straight 20 feet then turn right 10 feet to Out patients ward then go straight 20 feet to Operation Theater

2

Fig 4: Direction from Entrance to Operation Theater



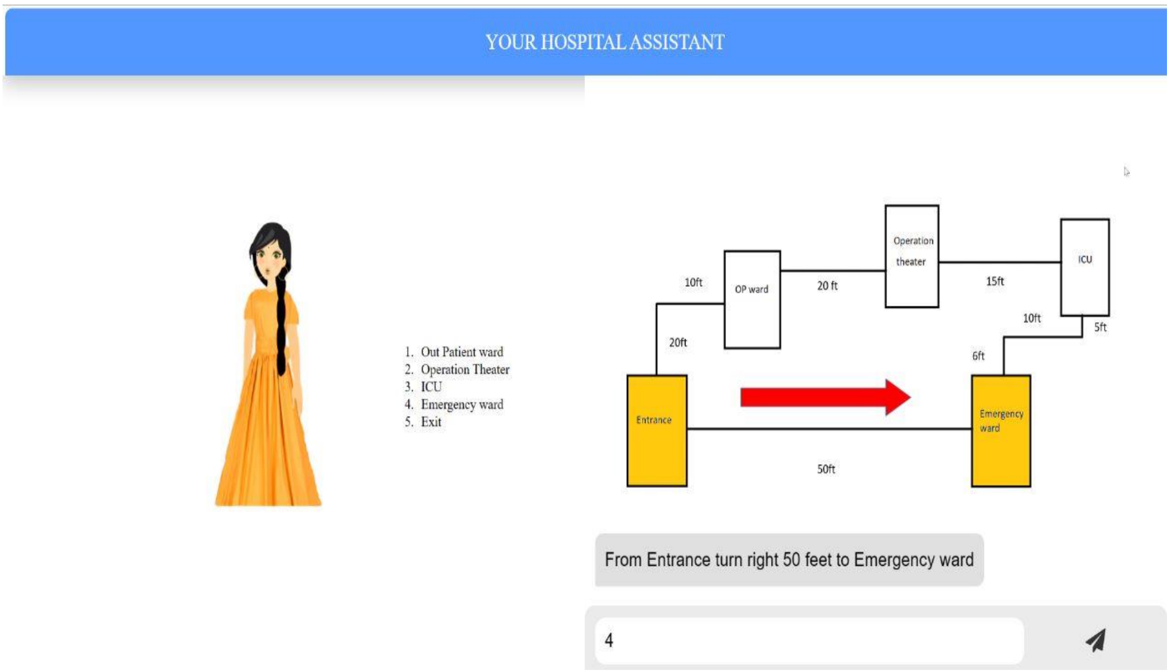
YOUR HOSPITAL ASSISTANT

1. Out Patient ward
2. Operation Theater
3. ICU
4. Emergency ward
5. Exit

From Entrance go straight 20 feet then turn right 10 feet to Out patients ward then go straight 20 feet to Operation Theater then go straight 15 feet to ICU

3

Fig 5: Direction from Entrance to ICU



YOUR HOSPITAL ASSISTANT

1. Out Patient ward
2. Operation Theater
3. ICU
4. Emergency ward
5. Exit

From Entrance turn right 50 feet to Emergency ward

4

Fig 6: Direction from Entrance to Emergency Ward

CHAPTER 8
CODE

Index.html

```
 <!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Welcome</title>
    <style>
        *{
            margin: 0;
            padding: 0;
            box-sizing: border-box;
            font-family:'Poppins',sans-serif ;
        }
        section{
            position: absolute;
            top:0;
            left: 0;
            width: 100%;
            height: 100vh;
            background: url(images/hospital1.jpg);
            background-repeat: no-repeat;
            background-size: cover;
            transition: opacity 3s linear;
            display: flex;
            justify-content: center;
            align-items: center;
        }
        section h2{
            font-size: 5vw;
            color: #000;
            background:#fff ;
            padding:  10px 40px;
            mix-blend-mode: screen;
        }
    </style>
</head>
<body style="background-color:#7FE9DE;" id="enterDemo"
onkeypress="enterKeyPressed(event)">
    <section>
    <h2>Welcome<br>Please press Enter to ask your queries</h2>
    </section>
    <script>
    function changeBg(){
        const images = [
'url("../images/hospital1.jpg")','url("../images/hospital2.jpg")','url("../images/hospital3.jpg")','url("../images/hospital4.jpg")'
        ]
        const section = document.querySelector('section');
        const bg = images[Math.floor(Math.random()* images.length)];
        section.style.backgroundImage = bg;
    }
    setInterval(changeBg,1000);
    function enterKeyPressed(event) {
        while(event.keyCode!==13){
            readOutLoud("I am here to help you");
            setTimeout(() => {
            getHardResponse(sampleText);
        }, 1000)
        }
        if (event.keyCode == 13) {
              window.location='chat.html';
            return true;
        } else {
            return false;
        }
    }
    </script>
</body>
<script src="chatBot.js"></script>
</html>
```

chat.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Chat Bot</title>
    <link rel="stylesheet" type="text/css" href="../chat.css"/>
    <link rel="stylesheet" type="text/css"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css"/>
    <style>
        video{
          float:left;
          margin-top:7%;
          height:100%;
        }
    </style>
  </head>
  <body>
     <div class="chat-bar-collapsible" >
      <div id="chat-button" class="collapsible">
        YOUR HOSPITAL ASSISTANT
      </div>
      <div class="guide" width="50%">
      <video id="hospital-guide" width=50% autoplay muted>
        <source src="../Bot.mp4" type="video/mp4"/>
          Your browser does not support HTML5 video.
      </video>
    </div>
    <div class="text-center p-2">
        <div class="full-chat-block">
           <div class="outer-container">
            <div id="mapRedirect"><img src="../map.png" alt="" width="100%"
height="300px"/></div>
            <div class="chat-container">
              <div id="chatbox">
                <p id="botStarterMessage" class="botText">
                  <span>Loading...</span>
                </p>
              </div>
              <!-- User input box -->
              <div class="chat-bar-input-block">
                <div id="userInput">
                  <input id="textInput" class="input-box" type="text" name="msg"
placeholder="Tap 'Enter' to send a message"/>
                  <p></p>
                </div>
                <div class="chat-bar-icons">
                  <i
                    id="chat-icon"
                    style="color: #333;"
                    class="fa fa-fw fa-send"
                    onclick="sendButton()"
                  ></i>
                </div>
              </div>
              <div id="chat-bar-bottom">
                <p></p>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </body>
  <script>
    document.getElementById('textInput').focus();
    document.getElementById('textInput').select();
  </script>
  <script
src="https://code.responsivevoice.org/responsivevoice.js?key=WUeEy8ji"></script>
  <script
```

```
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
  <script src="../chatBot.js"></script>
</html>
```

chat.css

```
html,
body {
  width: 100%;
  height: 100%;
  position: fixed;
}
body {
  margin: 0 auto;
}
.locations{
  border: solid black 2px;
  width: 45%;
  height: 100%;
}
.chat-btn {
  position: absolute;
  right: 20px;
  bottom: 30px;
  cursor: pointer;
}
.text-center{
    width: 50%;
    float: right;
}
.chat-btn i {
  transition: all 0.9s ease;
}
#check:checked ~ .chat-btn i {
  display: block;
  pointer-events: auto;
  transform: rotate(180deg);
}
.wrapper {
  position: absolute;
  right: 20px;
  bottom: 100px;
  width: 300px;
  border-radius: 5px;
  opacity: 0;
  transition: all 0.4s;
}
#check:checked ~ .wrapper {
  opacity: 1;
}
#check {
  display: none;
}
.guide {
  justify-content: end;
  margin-bottom: 20px;
  position: relative;
display: inline;
width: 50%;
}
.chat-bar-collapsible {
  /* position: fixed; */
  bottom: 0;
  width: 100%;
  justify-content: center;
  box-shadow: 0 8px 16px 0 rgba(0, 0, 0, 0.2);
}
.collapsible {
  background-color: rgb(82, 151, 255);
  color: white;
  cursor: pointer;
  padding: 18px;
  width: 100%  ;
  text-align: center;
  outline: none;
  font-size: 18px;
  border-radius: 10px 10px 0px 0px;
  border: 3px solid white;
  border-bottom: none;
}
.content {
  max-height: 0;
  /* overflow: hidden; */
  transition: max-height 0.2s ease-out;
  background-color: #f1f1f1;
}
.outer-container {
  min-height: 500px;
  bottom: 0%;
  position: relative;
}
```

```
.chat-container {
  /* max-height: 500px; */
  width: 100%;
  position: absolute;
  bottom: 0;
  left: 0;
  scroll-behavior: smooth;
  hyphens: auto;
}
.chat-bar-input-block {
  display: flex;
  float: left;
  box-sizing: border-box;
  justify-content: space-between;
  width: 100%;
  align-items: center;
  background-color: rgb(235, 235, 235);
  border-radius: 10px 10px 0px 0px;
  padding: 10px 0px 10px 10px;
  position: static;
}
.chat-bar-icons {
  display: flex;
  justify-content: space-evenly;
  box-sizing: border-box;
  width: 25%;
  float: right;
  font-size: 20px;
}
#userInput {
  width: 75%;
}
.input-box {
  float: left;
  border: none;
  box-sizing: border-box;
  width: 100%;
  border-radius: 10px;
  padding: 10px;
  font-size: 16px;
  color: #000;
  background-color: white;
  outline: none;
}
.userText {
  color: white;
  font-family: Helvetica;
  font-size: 16px;
  font-weight: normal;
  text-align: right;
  clear: both;
}
.userText span {
  line-height: 1.5em;
  display: inline-block;
  background: #5ca6fa;
  padding: 10px;
  border-radius: 8px;
  border-bottom-right-radius: 2px;
  max-width: 80%;
  margin-right: 10px;
  animation: floatup 0.5s forwards;
}
#mapRedirect{
  position:fixed;
  top: 10%;
}
.botText {
  color: #000;
  font-family: Helvetica;
  font-weight: normal;
  font-size: 16px;
  text-align: left;
}
.botText span {
  line-height: 1.5em;
  display: inline-block;
  background: #e0e0e0;
  padding: 10px;
  border-radius: 8px;
  border-bottom-left-radius: 2px;
  max-width: 80%;
  margin-left: 10px;
  animation: floatup 0.5s forwards;
}
```

chatbot.js

```
var coll = document.getElementsByClassName("chat-btn");
for (let i = 0; i < coll.length; i++) {
    coll[i].addEventListener("click", function () {
        this.classList.toggle("active");
        var content = this.nextElementSibling;
        if (content.style.maxHeight) {
            content.style.maxHeight = null;
        } else {
            content.style.maxHeight = content.scrollHeight + "px";
        }
    });
}
//voice bot
function readOutLoud(message) {
    const speech = message
    speech.text = message
    speech.volume = 1
    speech.rate = 1.1
    speech.pitch = 1
    responsiveVoice.speak(speech, 'UK English Female');
}
function firstBotMessage() {
    let welcomeMsg = "Hello, You are at the Entrance";
    let firstMessage = "Where do you want to go? ";
    let secondMessage =`Press 1 for Out patients ward 2 for Operation theater 3 for
ICU 4 Emergency ward`;
    readOutLoud(welcomeMsg);
    document.getElementById("botStarterMessage").innerHTML = '<p
class="botText"><span>' + firstMessage +'\n'+secondMessage+'</span></p>';
    readOutLoud(firstMessage);
    readOutLoud(secondMessage);
    let time = getTime();
    $("#chat-timestamp").append(time);
    document.getElementById("userInput").scrollIntoView(false);}
```

```
firstBotMessage();
 function fun(){
 var graph = new WeightedGraph();
 graph.addVertex("Entrance");
 graph.addVertex("OP ward");
 graph.addVertex("Operation Theater");
 graph.addVertex("Emergency ward");
 graph.addVertex("ICU");
 graph.addVertex("Out patients ward");
 graph.addVertex("j1");
 graph.addVertex("j2");
 graph.addVertex("j3");
 graph.addEdge("Entrance", "j1", 20);
 graph.addEdge("j1", "Out patients ward", 10);
 graph.addEdge("Out patients ward", "Operation Theater", 20);
 graph.addEdge("Operation Theater", "ICU", 15);
 graph.addEdge("ICU","j2", 5);
 graph.addEdge("j2", "j3", 10);
 graph.addEdge("j3", "Emergency ward", 6);
 graph.addEdge("Entrance", "Emergency ward", 50);
 return graph;
 }
function getHardResponse(userText) {
   graph = fun();
   let starting = "Entrance";
   let [path,g] = graph.Dijkstra(starting,userText);
   let botResponse=" ";
   if(userText>=1 && userText <=4){
      var key=["go straight","turn left","turn right"];
      if(path[1]=="Emergency ward")
```

```
          botResponse = "From "+starting+" "+key[2]+" "+g[path[0]]+" feet to "+path[1];
        else
          botResponse = "From "+ starting+" "+ key[0]+" "+ g[path[0]] +" feet ";
        for(let i=1;i<path.length-1;i++){
          if(path[i+1]=="Emergency ward" || path[i+1]== "Out patients ward" )
            botResponse = botResponse + " "+" then " +key[2]+" "+ g[path[i]] +" feet to "
+path[i+1];
          if(path[i+1]=="ICU" || path[i+1]=="Operation Theater")
          botResponse = botResponse + " "+" then " +key[0]+" "+ g[path[i]] +" feet to "
+path[i+1];
        }
        let botVideo = `<video id="botText" width="100%" height=300px autoplay
muted> <source src="../maps/v`+userText+`.mp4" type="video/mp4" /></video>`
          document.getElementById("mapRedirect").innerHTML=botVideo;
    }
    else{
       botResponse = "Enter valid choice";
    }
    let botHtml = '<p class="botText"><span>' + botResponse + '</span></p>';
    document.getElementById("chatbox").innerHTML= botHtml;
    console.log("Response "+botResponse);
    // $("#chatbox").append(botHtml);
    readOutLoud(botResponse.toString());
    document.getElementById("chat-bar-bottom").scrollIntoView(true);
}
function getResponse() {
    let userText = $("#textInput").val();
    let userHtml = '<p class="userText"><span>' + userText + '</span></p>';
    $("#textInput").val("");
    $("#chatbox").append(userHtml);
```

```
   document.getElementById("chat-bar-bottom").scrollIntoView(true);
    setTimeout(() => {
        getHardResponse(userText);
    }, 1000)
}
function buttonSendText(sampleText) {
    let userHtml = '<p class="userText"><span>' + sampleText + '</span></p>';
    $("#textInput").val("");
    $("#chatbox").append(userHtml);
    document.getElementById("chat-bar-bottom").scrollIntoView(true);
    setTimeout(() => {
        getHardResponse(sampleText);
    }, 1000)
}
function sendButton() {
    getResponse();
}
$("#textInput").keypress(function (e) {
    if (e.which == 13) {
        getResponse();
    }
});
class Node {
    constructor(val, priority) {
      this.val = val;
      this.priority = priority;
    }
  }
  class PriorityQueue {
    constructor() {
```

```
    this.values = [];
  }
  enqueue(val, priority) {
   let newNode = new Node(val, priority);
   this.values.push(newNode);
   this.bubbleUp();
  }
  bubbleUp() {
   let idx = this.values.length - 1;
   const element = this.values[idx];
   while (idx > 0) {
    let parentIdx = Math.floor((idx - 1) / 2);
    let parent = this.values[parentIdx];
    if (element.priority >= parent.priority) break;
    this.values[parentIdx] = element;
    this.values[idx] = parent;
    idx = parentIdx;
   }
  }
  dequeue() {
   const min = this.values[0];
   const end = this.values.pop();
   if (this.values.length > 0) {
    this.values[0] = end;
    this.sinkDown();
   }
   return min;
  }
  class WeightedGraph {
   constructor() {
      this.adjacencyList = {};
     }
    addVertex(vertex) {
      if (!this.adjacencyList[vertex]) this.adjacencyList[vertex] = [];
     }
    addEdge(vertex1, vertex2, weight) {
      this.adjacencyList[vertex1].push({ node: vertex2, weight });
     }
    Dijkstra(start, finish) {
    var end = {1:"Entrance",2:"Out patients ward",3:"Operation
Theater",4:"ICU",5:"Emergency ward"};
      finish = end[(parseInt(finish)+1).toString()];
      const nodes = new PriorityQueue();
      const distances = {};
      const previous = {};
      let path = [];
      let g=[];
      let smallest;
      for (let vertex in this.adjacencyList) {
        if (vertex === start) {
          distances[vertex] = 0;
          nodes.enqueue(vertex, 0);
        } else {
          distances[vertex] = Infinity;
          nodes.enqueue(vertex, Infinity);
        }
        previous[vertex] = null;
      }
      // as long as there is something to visit
      while (nodes.values.length) {
```

```
     smallest = nodes.dequeue().val;
     if (smallest === finish)
      while (previous[smallest]) {
       path.push(smallest);
       var neigh = smallest;
       smallest = previous[smallest];
    for (var n in this.adjacencyList[smallest]){
         var nextNode = this.adjacencyList[smallest][n];
         if(nextNode.node == neigh){
           g[smallest]=nextNode.weight;
           break;}
      }
      }
      break;
     }
     if (smallest || distances[smallest] !== Infinity) {
      for (let neighbor in this.adjacencyList[smallest]) {
        let nextNode = this.adjacencyList[smallest][neighbor];
        let candidate = distances[smallest] + nextNode.weight;
        let nextNeighbor = nextNode.node;
        if (candidate < distances[nextNeighbor]) {
         distances[nextNeighbor] = candidate;
         previous[nextNeighbor] = smallest;
         nodes.enqueue(nextNeighbor, candidate);
       }}}  }
     console.log(g);
     console.log(path.concat(smallest).reverse());
     return [path.concat(smallest).reverse(),g];
 }
}
```

CHAPTER 9

REFERENCE

1. https://www.researchgate.net/profile/Nivedita-Bhirud **(AUG-2019)**

2. https://www.academia.edu/39649809/A_Survey_Paper_on_Chatbots

3. https://www.researchgate.net/profile/Sisi-Zlatanova **(MAP - OCT 2019)**

4. https://pdf.sciencedirectassets.com/280203/1-s2.0-S1877050916X00208/1-s2.0-S1877050916317859/main.pdf?X-Amz-Security-Token=IQoJb3JpZ2luX2VjEHgaCXVzLWVhc3QtMSJHMEUCIQDhCeD2Puso74Xg3Rbqa2w9hVFQlwbIgQR%2FHtRgz4VxHgIgBdvB%2FnKzL4Q%2BUsv4TMzAvAhfgZM2rGAp5snfc66WapgquwUIkf%2F%2F%2F%2F%2F%2F%2F%2F%2F%2FARAFGgwwNTkwMDM1NDY4NjUiDPA7YPQ5U0H2hTdoeSqPBWWIYh0FE8YvwLnAaNfS3gASXoEmpnSxpQCu2qd6vQ8tXq3Ha4RfLKenxTvvZs827s3ed2TdU8tYhfVdhYV%2FvzTSld%2Fu3xw8me6EgiOn5Ssg2pmWemEmgjod8TqYHyQxGsk4IO0VcCZ5H6z2UF9kNUGyyGX **(BRAILLE – 2016)**

5. https://ieeexplore.ieee.org/document/8947825 **(API - DEC 2018)**

6. https://www.atlantis-press.com/proceedings/iccese-20/125937024 **(MAR-2020)**

7. https://www.chatbot.com/

8. https://ieeexplore.ieee.org/document/9498508 **(2021)**