

NAME : PRACHI PATEL
SAP ID: 60004200049
BRANCH : COMPUTER ENGINEERING(A1)
SUBJECT : Ubiquitous Computing
EXPERIMENT - 5
Ubiquitous Communication

PRAKHI PATEL

60004200049

Computer Engineering

Experiment - 5 :-

Aim - To implement Ubiquitous Communication

Theory -

Ubiquitous applications use communication network to access relevant remote external information and tasks, anywhere and anytime. Although, communication access can be modelled as part of the internal system, the role of the communication network infrastructure is considered to be external to the Ubiquitous system and part of the system's virtual computing environment. Different applications require different combinations of network functions and services eg data streaming, minimal jitter, type of media access control etc. Different networks support different sets of communication functions in different ways. Key design issue concern, first, whether or not these communication functions are largely transparent to services (network oriented) or whether or not these communication functions are largely exposed via some interfaces and able to be configured and controlled by services (service oriented) or whether or not these communication functions are exposed via some interfaces and able to be configured and controlled by services (service oriented). Second, there is the issue of whether or not to make all networked services ubiquitous, all networked services ubiquitous, attached anywhere and accessible.

Flask Server:

```
from flask import Flask, render_template, request
from flask_socketio import SocketIO, emit
app = Flask(__name__, static_folder='templates', static_url_path='')
socketio = SocketIO(app)

users = {}

@app.route('/')
def index():
    return render_template('index.html')

@socketio.on('connect')
def handle_connect():
    user_id = request.sid
    users[user_id] = {'name': 'IP '+request.remote_addr}
    print(f'User {user_id} connected')

@socketio.on('disconnect')
def handle_disconnect():
    user_id = request.sid
    del users[user_id]
    print(f'User {user_id} disconnected')

@socketio.on('message')
def handle_message(data):
    user_id = request.sid
    emit('message', {'name': users[user_id]['name'], 'text': data['text']},
        broadcast=True)
if __name__ == '__main__':
    socketio.run(app, host='0.0.0.0', port=5000)
```

2. Index.html

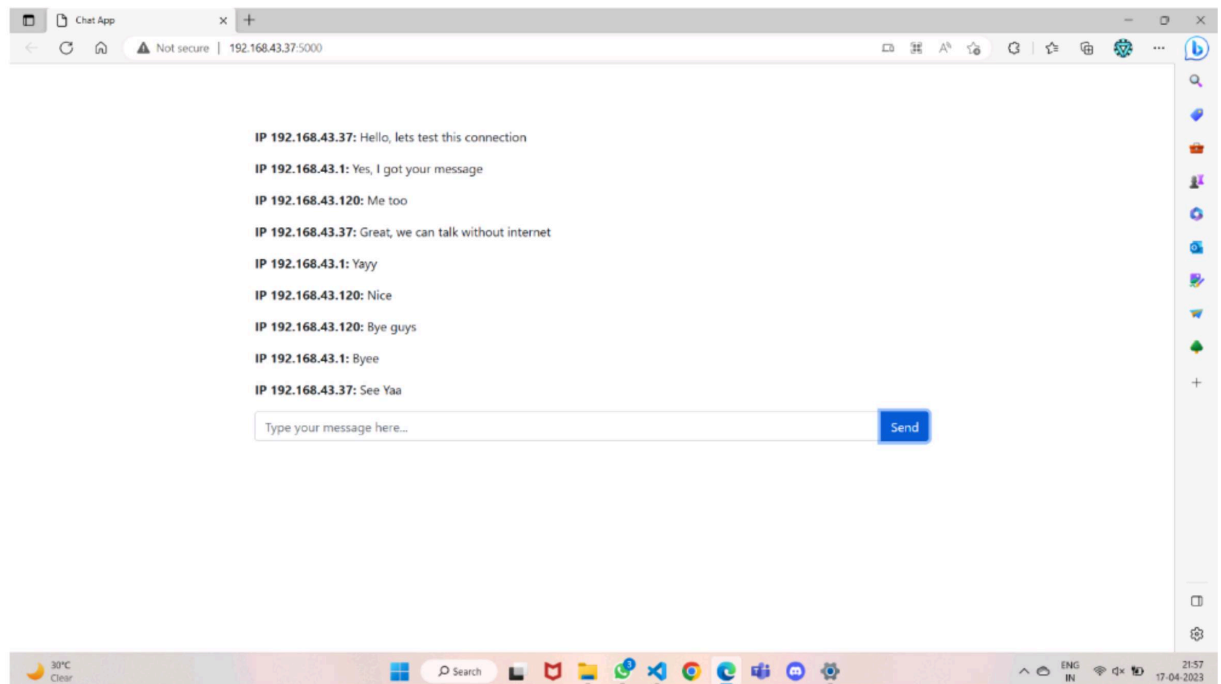
```
<!DOCTYPE html>
<html>
<head>
  <title>Chat App</title>
  <!--<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0/dist/css/bootstrap.mi
n.css" /> -->
<link rel="stylesheet" href="bootstrap.min.css" />
<style>
body {
  padding-top: 5rem;
}
.chat-message {
  margin-bottom: 1rem;
}
.chat-message .name {
  font-weight: bold;
}
</style>
</head>
<body>
  <div class="container">
    <div class="row">
      <div class="col-md-8 offset-md-2">
        <div id="messages"></div>
        <form id="message-form">
          <div class="input-group mt-3">
            <input type="text" class="form-control"
id="message-input" placeholder="Type
your
message here..." autocomplete="off" />
            <button class="btn btn-primary"
type="submit">Send</button>
          </div>
        </form>
      </div>
    </div>
  </div>
</body>
</html>
```

```

        </form>
    </div>
</div>
</div>
<!--<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script src="https://cdn.socket.io/4.5.0/socket.io.min.js"
crossorigin="anonymous"></script>-->
<script src="jquery.min.js"></script>
<script src="socket.io.min.js"></script>
<script>
$(document).ready(function() {
//var socket = io.connect('http://' + document.domain + ':' +
location.port);
var socket = io();
$('#message-form').submit(function(e) {
e.preventDefault();
var message = $('#message-input').val();
socket.emit('message', { text: message });
$('#message-input').val("");
});
socket.on('message', function(data) {
var messageHtml = '<div class="chat-message"><span class="name">'
+ data.name + ':'
</span><span class="text">' + data.text + '</span></div>';
$('#messages').append(messageHtml);
});
});
</script>
</body>
</html>

```

Output:



9:57 PM | 0.2KB/s

82%

No internet connection



192.168.43.37:5000



IP 192.168.43.1: Yes, I got your message

IP 192.168.43.120: Me too

IP 192.168.43.37: Great, we can talk without internet

IP 192.168.43.1: Yayy

IP 192.168.43.120: Nice

IP 192.168.43.120: Bye guys


IP 192.168.43.1: Byee

IP 192.168.43.37: See Yaa

Type your message here...

Send

Personal Hotspot : 2 connections, Used 0 B

 No internet connection



92.168.43.37:5000



1



IP 192.168.43.37: Hello, lets test this connection

IP 192.168.43.1: Yes, I got your message

IP 192.168.43.120: Me too

IP 192.168.43.37: Great, we can talk without internet

IP 192.168.43.1: Yayy

IP 192.168.43.120: Nice

IP 192.168.43.120: Bye guys

IP 192.168.43.1: Byee

IP 192.168.43.37: See Yaa

Type your message here...

Send

from anywhere just in case these services may be needed, versus selectively accessing networked services eg. some services may be limited to a locality.

Many general and introductory texts and descriptions about networking are oriented towards specific types of network. For example, data communication, traditionally focuses most on the communication of alpha numeric data. Telecoms focuses on voice communication and its use as an underlay network for data and audio video over telecoms. Broadcast audio video networks use separate radio and TV networks or wireless network. Because one interpretation of Ubiquitous, is any content on any network, anytime, anywhere, the complete range of different media networks is treated holistically.

Conclusion:-

The concept of Ubiquitous communication was studied by implementing a simple message transfer system between two devices and establishing a communication channel.