



Міністерство освіти і науки України  
Національний технічний університет України  
“Київський політехнічний інститут імені Ігоря Сікорського” Факультет  
інформатики та обчислювальної техніки  
Кафедра інформатики та програмної інженерії

## Лабораторна робота №4

### **Основи WEB-технологій**

**«Протокол WebSocket. Використання Socket.io для розробки чат-  
додатків»**

Виконав студент групи IT-01:

Куксюк В.І.

Київ 2023

## Варіант 1

**Завдання 1.** Розробити додаток для обміну повідомленнями між учасниками в режимі реального часу за допомогою бібліотеки SocketIO

### Chat

User 1: qweaxszd
User 1: qweasd
User 1: qweasd
User 3: kasjdjh
User 3: qwefzxZZ

HTML:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Socket.IO Chat</title>
  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet">
  <script src="/socket.io/socket.io.js"></script>
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
  <style>
    .card-body {
      display: grid;
      grid-template-columns: 1fr 100px;
    }
  </style>
</head>
<body>
<div class="container chat-container">
  <h1 class="my-4" id="title">Chat</h1>
  <ul id="messages" class="list-group mb-2 card"></ul>
  <div class="card">
    <div class="card-body">
      <input id="messageInput" class="form-control" autocomplete="off"/>
```

```

        <button id="messageButton" class="btn btn-primary btn-
block">Send</button>
    </div>
</div>
</div>
<script> const socket = io();
    $('#messageButton').click(() => {
        const message = $('#messageInput').val();
        socket.emit('chat message', message);
        $('#messageInput').val('');
    });
    socket.on('chat message', (data) => {
        const {userId, message} = data;
        message !== "" && $('#messages').append($('- ').addClass('list-
group-item').text(`User ${userId}: ${message}`));
    });
</script>
</body>
</html>

```

JS:

```

const express = require('express');
const http = require('http');
const {Server} = require('socket.io');
const app = express();
const server = http.createServer(app);
const io = new Server(server);
app.use(express.static(__dirname + '/public'));
app.get('/', function (req, res) {
    res.sendFile(__dirname + '/index.html');
});
let userCounter = 0;
io.on('connection', (socket) => {
    userCounter++;
    const userId = userCounter;
    console.log(`User ${userId} connected`);
    socket.on('chat message', (message) => {
        io.emit('chat message', {userId, message});
    });
    socket.on('disconnect', () => {
        console.log(`User ${userId} disconnected`);
    });
});

```

```
server.listen(3000, () => {  
  console.log('Server is running on port 3000');  
});
```

[Код на GitHub](#)

### **Висновок:**

Під час виконання даної лабораторної роботи я навчився працювати з бібліотекою SocketIO.