

## ЛАБОРАТОРНА РОБОТА № 6

### Project03 “Chat Application”

Лабораторна робота 06. Протокол WebSocket. Використання Socket.io  
для розробки чат-додатків

Хід роботи:

### Завдання до проекту (Лабораторна №6)

- Розробити додаток для обміну повідомленнями між учасниками в режимі реального часу
- Спілкування проводиться в кімнатах користувачами, що здійснили вхід в кімнату
- Про вхід чи вихід користувачів з кімнати необхідно надсилати повідомлення та оновлені дані кімнати
- Використати зразки сторінок на наступних слайдах
- .

### Завдання до проекту (самостійно)

- Забезпечити внутрішній скролінг для контенту з повідомленнями та даними про кімнати (авто прокручування до останнього повідомлення)
- Додати функціональність “{user} is typing” (тривалістю 2 секунди): всі користувачі в межах кімнати отримують відповідну подію з повідомленням.
  - Крок1. Клієнт відправляє на сервер подію “userIsTyping”
  - Крок2. Сервер слухає подію “userIsTyping” та ретранслює її на всі інші клієнти в межах даної кімнати
  - Крок3. Клієнт слухає подію “userIsTyping” та відображає повідомлення в інформаційному блоці
- Додати функціональність для приватних повідомлень: в повідомленні якщо написати “@username: text”

Таблиця 1-2 - Завдання

					ДУ «Житомирська політехніка».22.121.05.000 – Лр6			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Пасічник Б.Р			Звіт з лабораторної роботи		Літ.	Арк.
Перевір.		Лисенко М.С.						1
Керівник								15
Н. контр.							ФІКТ Гр. ВТ-22-1[2]	
Зав. каф.								

## Лістинг програми:

### App.js

```
const express = require('express');
const http = require('http');
const socketio = require('socket.io');
const { join } = require('path');

const { generateMessage, generateLocationMessage, generateTypingMessage } =
  require('./utils/messages');
const { addUser, removeUser, getUser, getUsersInRoom, getUserByName } =
  require('./utils/users');

const app = express();
const server = http.createServer(app);
const io = socketio(server);

app.use(express.static(join(__dirname, '/../public')));

io.on('connection', (socket) => {
  console.log('User connected');

  socket.on('join', (options, callback) => {
    const { error, user } = addUser({ id: socket.id, ...options });

    if (error) {
      return callback(error);
    }

    socket.join(user.room);

    socket.emit('message', generateMessage('Admin', `Welcome,
    ${user.username}!`));
    socket.broadcast.to(user.room).emit('message', generateMessage('Admin',
    `${user.username} has joined!`));

    io.to(user.room).emit('roomData', {
      users: getUsersInRoom(user.room)
    });

    callback();
  });

  socket.on('sendMessage', (message, callback) => {
    const user = getUser(socket.id);

    io.to(user.room).emit('message', generateMessage(user.username, message));
    callback();
  });
});
```

		Пасічник Б.Р.			ДУ «Житомирська політехніка».22.121.05.000 – Лр6	Арк.
		Лисенко М.С.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

```

socket.on('sendPrivateMessage', (username, message, callback) => {
  const recipient = getUserByName(username);
  const user = getUser(socket.id);
  if (user && recipient) {
    socket.to(recipient.id).emit('message', generateMessage(user.username,
message));
    socket.emit('message', generateMessage(user.username, message));
    callback();
  } else {
    callback('User not found');
  }
});

socket.on('userIsTyping', (typing) => {
  const user = getUser(socket.id);
  if (typing === true)
    io.emit('display', generateTypingMessage(true, `${user.username} is
typing...`));
  else
    io.emit('display', generateTypingMessage(false, ""));
});

socket.on('disconnect', () => {
  const user = removeUser(socket.id);
  if (user) {
    io.to(user.room).emit('message', generateMessage('Admin',
`${user.username} has left!`));
    io.to(user.room).emit('roomData', {
      users: getUsersInRoom(user.room)
    });
  }
});
});

server.listen(3000, () => {
  console.log('Server is running on http://localhost:3000');
});

```

## Users.js

```

const users = []

const addUser = ({ id, username, room }) => {
  username = username.trim().toLowerCase()
  room = room.trim().toLowerCase()

  const existingUser = users.find((user) => {
    return user.room === room && user.username === username
  })

```

		Пасічник Б.Р.			ДУ «Житомирська політехніка».22.121.05.000 – Лр6	Арк.
		Лисенко М.С.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    if (!username || !room) {
      return {
        error: 'Username and room are required!'
      }
    }

    if (existingUser) {
      return {
        error: 'Username is in use!'
      }
    }

    const user = { id, username, room }
    users.push(user)
    return { user }
  }

const removeUser = (id) => {
  const index = users.findIndex((user) => user.id === id)

  if (index !== -1) {
    return users.splice(index, 1)[0]
  }
}

const getUser = (id) => {
  return users.find((user) => user.id === id)
}

const getUsersInRoom = (room) => {
  room = room.trim().toLowerCase()
  return users.filter((user) => user.room === room)
}

const getUserByName = (name) => {
  return users.find((user) => user.username === name)
}

module.exports = {
  addUser,
  removeUser,
  getUser,
  getUsersInRoom,
  getUserByName
}

```

## Messages.js

```
const generateMessage = (username, text) => {
```

		Пасічник Б.Р.			ДУ «Житомирська політехніка».22.121.05.000 – Лр6	Арк.
		Лисенко М.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		4

```

    return {
      username,
      text,
      createdAt: new Date().getTime()
    }
  }
}
const generateTypingMessage = (typing, text) => {
  return {
    text,
    typing
  };
};
const generateLocationMessage = (username, url) => {
  return {
    username,
    url,
    createdAt: new Date().getTime()
  }
}

module.exports = {
  generateMessage,
  generateLocationMessage,
  generateTypingMessage
}

```

## Index.html

```

<!doctype html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"
integrity="sha384-Vkoo8x4CGs03+Hhvx8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh"
crossorigin="anonymous">
  <link rel="stylesheet" href="/css/index_style.css">
  <title>Приєднатись до чату</title>
</head>
<body>
<div class="vw-100 vh-100 d-flex justify-content-center align-items-center">
  <div class="form-container bg-info p-4">
    <form action="/chat.html">
      <div class="form-group">
        <input type="text" class="form-control" id="nickname" name="username"
placeholder="Логін">
      </div>
      <div class="form-group">

```

		Пасічник Б.Р.			ДУ «Житомирська політехніка».22.121.05.000 – Лр6	Арк.
		Лисенко М.С.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        <input type="text" class="form-control" id="room" name="room"
placeholder="Кімната">
    </div>
    <button type="submit" class="btn btn-warning">Приєднатись</button>
</form>
</div>
</div>

</body>
</html>

```

## Chat.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"
integrity="sha384-Vkoo8x4CGs03+Hhvx8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh"
crossorigin="anonymous">
    <link rel="stylesheet" href="/css/chat_style.css">
    <title>Document</title>
</head>
<body>
<section>
    <div class="container-fluid">
        <div class="row">
            <div id="sidebar" class="col-4 vh-100 bg-info p-4">
                </div>

            <div id="chat" class="col-8 d-flex flex-column flex-grow-1 vh-100 p-0">

                <div id="messages" class="flex-grow-1" style="overflow-y: auto; max-
height: calc(120vh - 120px);">

                    </div>

                <div id="compose">
                    <form id="message-form" class="form-inline">
                        <div class="form-group flex-grow-1">
                            <input type="text" name="message" placeholder="Message"
required autocomplete="off">
                        </div>
                        <div class="form-group ml-2">
                            <button class="btn btn-warning btn-lg">Send</button>
                        </div>
                    </form>
                </div>
            </div>
        </div>
    </div>

```

		Пасічник Б.Р.			ДУ «Житомирська політехніка».22.121.05.000 – Лр6	Арк.
		Лисенко М.С.				6
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        </form>
      </div>
    </div>
  </div>
</div>

</section>

<script id="message-template" type="text/html">
  <div class="alert message {{bgClass}}">
    <div>
      <span class="message-name">{{username}}</span>
      <span class="message-meta">{{createdAt}}</span>
    </div>
    <div>{{message}}</div>
  </div>
</script>

<script id="sidebar-template" type="text/html">
  <div class="alert alert-primary">
    Room: <strong>{{room}}</strong><br>
    Username: <strong>{{currentUser}}</strong>
  </div>

  <ul class="list-group">
    <li class="list-group-item list-group-item-primary">Users in the
chatroom</li>
    <li class="list-group-item">{{username}}</li>
    </ul>
</script>

<script
src="https://cdnjs.cloudflare.com/ajax/libs/mustache.js/3.1.0/mustache.min.js"></scri
pt>
<script src="https://cdnjs.cloudflare.com/ajax/libs/moment.js/2.24.0/moment-with-
locales.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/qs/6.9.1/qs.min.js"></script>

<script src="/socket.io/socket.io.js"></script>
<script>
  const $messageForm = document.querySelector("#message-form");
  const $messageFormInput = $messageForm.querySelector("input");
  const $messageFormButton = document.querySelector("button");
  const $messages = document.querySelector("#messages");

  const messageTemplate = document.querySelector("#message-template").innerHTML;

```

		Пасічник Б.Р.			ДУ «Житомирська політехніка».22.121.05.000 – Пр6	Арк.
		Лисенко М.С.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

```

const sidebarTemplate = document.querySelector('#sidebar-template').innerHTML;

const { username, room } = Qs.parse(location.search, { ignoreQueryPrefix: true
});

var socket = io();

socket.on('display', (message) => {
  if (message.typing) {
    const html = Mustache.render(messageTemplate, {
      message: message.text
    });
    $messages.insertAdjacentHTML('beforeend', html);
  } else {
    const typingMessages = $messages.querySelectorAll(".alert.message");
    typingMessages.forEach((message) => {
      if (message.textContent.trim().match(/^\\w+ is typing\\.\\.\\.\\$/)) {
        message.remove();
      }
    });
  }
});

socket.on('message', (message) => {
  console.log(message);
  let bgClass = 'bg-warning text-white';
  if (message.username === username.toLowerCase()) {
    bgClass = 'bg-success text-white pull-right';
  } else if (message.username === 'Admin') {
    bgClass = 'bg-info text-white';
  }
  const html = Mustache.render(messageTemplate, {
    username: message.username,
    message: message.text,
    bgClass: bgClass,
    createdAt: moment(message.createdAt).format('h:mm a')
  });
  $messages.insertAdjacentHTML('beforeend', html);
  $messages.scrollTop = $messages.scrollHeight;
});

socket.on('roomData', ({ users }) => {
  const html = Mustache.render(sidebarTemplate, {
    room,
    currentUser: username,
    users
  });
  document.querySelector('#sidebar').innerHTML = html;
});

```



```

let typing = false;
let timeout = undefined;

$messageFormInput.addEventListener('input', e => {
  clearTimeout(timeout);
  if (!typing) {
    socket.emit('userIsTyping', true);
  }
  typing = true;
  timeout = setTimeout(() => {
    typing = false;
    socket.emit('userIsTyping', false);
  }, 2000);
});

$messageForm.addEventListener('submit', (e) => {
  e.preventDefault();
  typing = false;
  $messageFormButton.setAttribute('disabled', 'disabled');

  const messageInput = e.target.elements.message.value;
  let message = messageInput.trim();
  let mentionedUser = null;

  const mentionRegex = /^@(\w+):\s*(.+)$/;
  const match = message.match(mentionRegex);

  if (match) {
    mentionedUser = match[1];
    message = match[2];
    socket.emit('sendPrivateMessage', mentionedUser, message, (error) => {
      $messageFormButton.removeAttribute('disabled');
      $messageFormInput.value = '';
      $messageFormInput.focus();
      socket.emit('userIsTyping', false);
      if (error) {
        return console.log(error);
      }
      console.log('Message delivered!');
    });
  } else {
    socket.emit('sendMessage', message, (error) => {
      $messageFormButton.removeAttribute('disabled');
      $messageFormInput.value = '';
      $messageFormInput.focus();
      if (error) {
        return console.log(error);
      }
      console.log('Message delivered!');
      socket.emit('userIsTyping', false);
    });
  }
});

```

		Пасічник Б.Р.			ДУ «Житомирська політехніка».22.121.05.000 – Лр6	Арк.
		Лисенко М.С.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        });
    }
});

socket.emit('join', { username, room }, (error) => {
    if (error) {
        alert(error);
        location.href = '/';
    }
});
});
</script>

</body>
</html>

```

## Index\_style.css

```

.form-container button {
    width: 100%;
}

* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

html {
    font-size: 16px;
}

input {
    font-size: 14px;
}

body {
    line-height: 1.4;
    color: #333;
    font-family: Arial, sans-serif;
    background-color: #f0f2f5;
}

h1 {
    margin-bottom: 16px;
}

label {
    display: block;
    font-size: 14px;
}

```

		Пасічник Б.Р.			ДУ «Житомирська політехніка».22.121.05.000 – Лр6	Арк.
		Лисенко М.С.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

```

margin-bottom: 8px;
color: #777;
}

input {
border: 1px solid #ddd;
padding: 12px;
outline: none;
border-radius: 4px;
}

button {
cursor: pointer;
padding: 12px;
background: #007bff;
border: none;
color: white;
font-size: 16px;
transition: background 0.3s ease;
border-radius: 4px;
}

button:hover {
background: #0056b3;
}

button:disabled {
cursor: default;
background: #007bff94;
}

.centered-form {
background: #333;
width: 100vw;
height: 100vh;
display: flex;
justify-content: center;
align-items: center;
}

.centered-form__box {
box-shadow: 0px 0px 17px 1px #1d1f26;
background: #f7f7fa;
padding: 24px;
width: 300px;
border-radius: 8px;
}

.centered-form button {
width: 100%;

```

		Пасічник Б.Р.			ДУ «Житомирська політехніка».22.121.05.000 – Лр6	Арк.
		Лисенко М.С.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

```

}

.centered-form input {
    margin-bottom: 16px;
    width: 100%;
}

```

## Chat\_style.css

```

* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

html {
    font-size: 16px;
}

input {
    font-size: 14px;
}

body {
    line-height: 1.4;
    color: #333;
    font-family: Arial, sans-serif;
    background-color: #f0f2f5;
}

.chat {
    display: flex;
}

.chat__sidebar {
    height: 100vh;
    color: white;
    background: #007bff;
    width: 225px;
    overflow-y: scroll;
    padding: 16px;
    border-right: 1px solid #ddd;
}

.chat__main {
    flex-grow: 1;
    display: flex;
    flex-direction: column;
    max-height: 100vh;
}

```

		Пасічник Б.Р.			ДУ «Житомирська політехніка».22.121.05.000 – Лр6	Арк.
		Лисенко М.С.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

```

}

.chat__messages {
  flex-grow: 1;
  padding: 24px;
  overflow-y: scroll;
  background-color: #fff;
  border-bottom: 1px solid #ddd;
}

.message {
  margin: 8px;
  padding: 8px;
  border-radius: 4px;
  background: #f1f1f1;
}

.message-name {
  font-weight: 600;
  font-size: 14px;
  margin-right: 8px;
}

.message-meta {
  font-size: 12px;
  color: #999;
}

.message a {
  color: #007bff;
}

#compose input {
  border: 1px solid #ddd;
  width: 100%;
  padding: 12px;
  margin: 0 16px 0 0;
  flex-grow: 1;
  border-radius: 4px;
}

#compose button {
  font-size: 14px;
  width: 150px;
  border-radius: 4px;
  background: #007bff;
  border: none;
  color: white;
  padding: 12px;
  transition: background 0.3s ease;
}

```

		Пасічник Б.Р.			ДУ «Житомирська політехніка».22.121.05.000 – Лр6	Арк.
		Лисенко М.С.				13
Змн.	Арк.	№ докум.	Підпис	Дата		

```

}

#compose button:hover {
    background: #0056b3;
}

.room-title {
    font-weight: 400;
    font-size: 22px;
    background: #2c2f3a;
    padding: 24px;
    color: white;
}

.list-title {
    font-weight: 500;
    font-size: 18px;
    margin-bottom: 4px;
    padding: 12px 24px 0 24px;
}

.users {
    list-style-type: none;
    font-weight: 300;
    padding: 12px 24px 0 24px;
}

#messages {
    overflow-y: auto;
}

#compose {
    margin-top: auto;
    margin-bottom: 8px;
    padding: 8px;
    border-top: 1px solid #ddd;
    background: #f7f7fa;
}

#message-form {
    display: flex;
    align-items: flex-end;
    width: 100%;
}

#message-form input[type="text"] {
    flex-grow: 1;
    margin-right: 8px;
}

```

		Пасічник Б.Р.			ДУ «Житомирська політехніка».22.121.05.000 – Лр6	Арк.
		Лисенко М.С.				14
Змн.	Арк.	№ докум.	Підпис	Дата		

Результат виконання програми:

The image displays two screenshots of a web application. The top screenshot shows a login form with two input fields: the first contains '6555' and the second contains '22'. Below these fields is a yellow button labeled 'Приєднатись' (Join). The bottom screenshot shows the chatroom interface. On the left, a sidebar contains a header with 'Room: 22' and 'Username: 6555', and a section titled 'Users in the chatroom' listing '2' users, with '6555' listed below. The main chat area has a teal header with 'Admin 10:31 pm' and 'Welcome, 6555!'. At the bottom, there is a 'Message' input field and a blue 'Send' button.

Рис. 1-2. Результат виконання програми

		Пасічник Б.Р.			ДУ «Житомирська політехніка».22.121.05.000 – Лр6	Арк.
		Лисенко М.С.				15
Змн.	Арк.	№ докум.	Підпис	Дата		