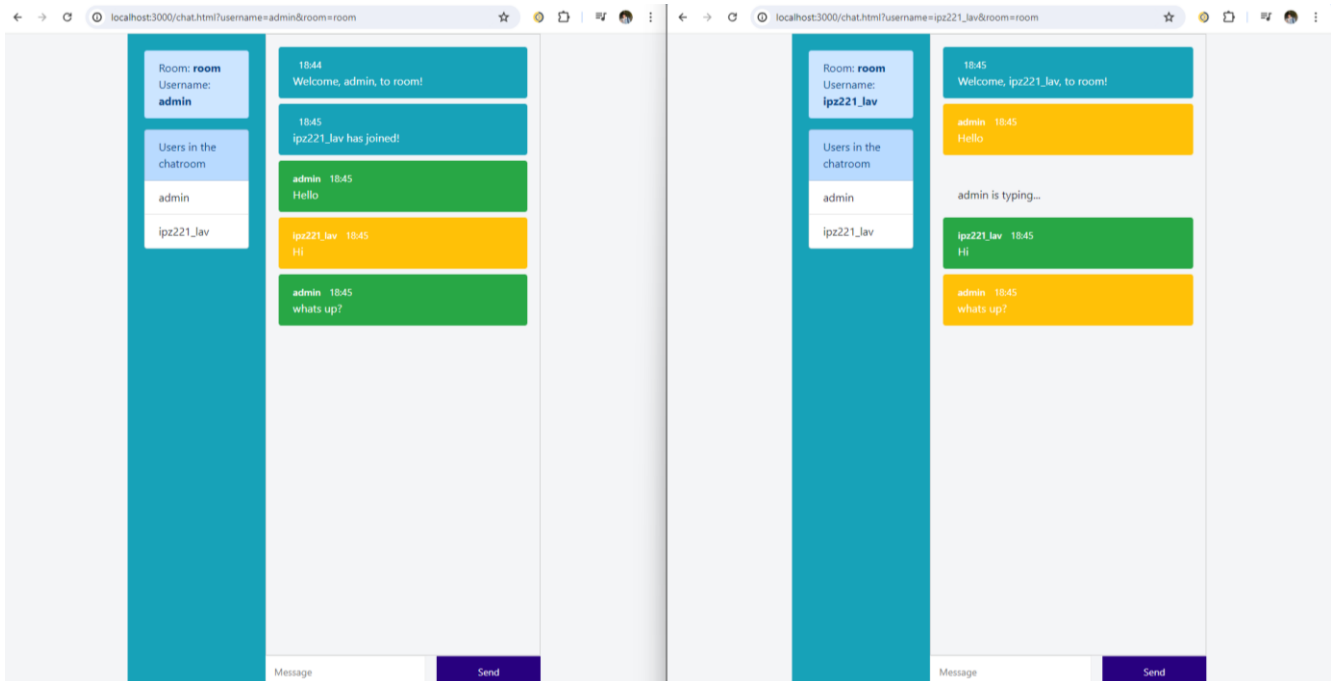


Лабораторна робота №6

Протокол WebSocket. Використання Socket.io для розробки чат-додатків



```
const express = require('express')
const app = express()
const server = require('http').createServer(app)
const io = require('socket.io')(server)
const { join } = require('path');
const { generateMessage, generateLocationMessage } = require("../utils/messages")
const { addUser, removeUser, getUser, getUsersInRoom, getUserByName } = require("../utils/users")

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

io.on('connection', (socket) => {
  console.log("user is connected")
  socket.on('join', (options, callback) => {
    console.log(options)
    console.log(callback)
    const { error, user } = addUser({ id: socket.id, ...options })

    if (error) {
      return callback(error)
    }

    socket.join(user.room)
  })
})
```

					ДУ «Житомирська політехніка».22.121.13.000 – Лр6			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Лаговський А.В.						
Перевір.		Сидорчук В.						
Керівник								
Н. контр.								
Зав. каф.								
					Лім.		Арк.	Аркушів
							1	15
					Звіт з лабораторної роботи			
					ФІКТ Гр. ІПЗ-22-1[1]			

```

        socket.emit('message', generateMessage(' ', `Welcome, ${user.username}, to
${user.room}!`))
        socket.broadcast.to(user.room).emit('message', generateMessage(' ', `${us-
er.username} has joined!`))
        io.to(user.room).emit('roomData', {
            users: getUsersInRoom(user.room)
        })

        callback()
    })

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

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

    socket.on('userIsTyping', () => {
        const user = getUser(socket.id)
        console.log(user.username)
        socket.broadcast.to(user.room).emit('messageTyping', user.username)
    })

    socket.on('privateMessage', (data, callback) => {
        console.log(data)
        const recipientSocket = io.sockets.sockets.get(getUserByName(data.to).id)
        console.log(recipientSocket)
        const user = getUser(socket.id)
        if (recipientSocket) {
            recipientSocket.emit('message', generateMessage(user.username, da-
ta.text))
        }
        callback()
    })

    socket.on('disconnect', () => {
        console.log("user is disconnected")
        const user = removeUser(socket.id)

        if (user) {
            io.to(user.room).emit('message', generateMessage(' ', `${user.username}
has left!`))
            io.to(user.room).emit('roomData', {
                users: getUsersInRoom(user.room)
            })
        }
    })
});

server.listen(3000, function () {
    console.log("listening on 3000 port")
});

```

```

const users = []

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

```

		Лаговський А.			ДУ «Житомирська політехніка».22.121.13.000 – Лр6	Арк.
		Сидорчук В.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    if (!username || !room) {
      return {
        error: 'Username and room are required!'
      }
    }
    const existingUser = users.find((user) => {
      return user.room === room && user.username === username
    })
    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()
    return users.filter((user) => user.room === room)
  }

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

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

  const generateMessage = (username, text) => {
    return {
      username,
      text,
      createdAt: new Date().getTime()
    }
  }

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

  module.exports = {
    generateMessage,

```

		Лаговський А.			ДУ «Житомирська політехніка».22.121.13.000 – Лр6	Арк.
		Сидорчук В.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        generateLocationMessage
    }
<!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-
Vkoo8x4CGsO3+HhXv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" cros-
sorigin="anonymous">
    <link rel="stylesheet" href="/styles/style.css">
    <title>Chat</title>
</head>
<body>
<div class="centered-form_box">
    <form action="/chat.html" class="centered-form">
        <div class="form-group">
            <input type="text" class="form-control" id="nickname" name="username"
placeholder="Login" required>
        </div>
        <div class="form-group">
            <input type="text" class="form-control" id="room" name="room" place-
holder="Room" required>
        </div>
        <button type="submit" class="btn btn-warning" style="width:
9em">Join</button>
    </form>
</div>
</body>
</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-
Vkoo8x4CGsO3+HhXv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" cros-
sorigin="anonymous">
    <link rel="stylesheet" href="/styles/style.css">
    <title>Chat</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"></div>
                    <div id="compose">
                        <form id="message-form" class="form-inline">
                            <div class="form-group flex-grow-1">
                                <input type="text" name="message" placehold-
er="Message" required autocomplete="off">
                            </div>

```

		Лаговський А.			ДУ «Житомирська політехніка».22.121.13.000 – Лр6	Арк.
		Сидорчук В.				
Змн.	Арк.	№ докум.	Підпис	Дата		4

```
<div class="form-group">  
    <!-- <button class="btn btn-warning btn-  
lg">Send</button> -->  
  
    <button>Send</button>  
    </div>  
</form>  
</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 cha-  
troom</li>  
        {{#users}}  
        <li class="list-group-item">{{username}}</li>  
        {{/users}}  
    </ul>  
</script>  
  
<script  
src="https://cdnjs.cloudflare.com/ajax/libs/mustache.js/3.1.0/mustache.min.js"></s  
cript>  
<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>  
    //elements  
    const $messageForm = document.querySelector("#message-form")  
    const $messageFormInput = $messageForm.querySelector("input");  
    const $messageFormButton = document.querySelector("button")  
    const $messages = document.querySelector("#messages")  
    //Templates  
    const messageTemplate = document.querySelector("#message-  
template").innerHTML  
    const sidebarTemplate = document.querySelector('#sidebar-  
template').innerHTML  
    // Options  
    const { username, room } = Qs.parse(location.search, { ignoreQueryPrefix:  
true })
```

		Лаговський А.			ДУ «Житомирська політехніка».22.121.13.000 – Лр6	Арк.
		Сидорчук В.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

```

let socket = io();

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 == '') {
    bgClass = 'bg-info text-white'
  }

  const html = Mustache.render(messageTemplate, {
    username: message.username,
    message: message.text,
    bgClass: bgClass,
    createdAt: moment(message.createdAt).format('HH:mm')
  })

  $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
})

$messageForm.addEventListener('submit', (e) => {
  e.preventDefault()

  $messageFormButton.setAttribute('disabled', 'disabled')

  const message = e.target.elements.message.value
  if (!message.startsWith('@')) {
    socket.emit('sendMessage', message, (error) => {
      $messageFormButton.removeAttribute('disabled')
      $messageFormInput.value = ''
      $messageFormInput.focus()

      if (error) {
        return console.log(error)
      }

      console.log('Message delivered!')
    })
  } else {
    const atIndex = message.indexOf(':');
    if (atIndex !== -1) {
      const username = message.slice(1, atIndex).trim();
      const newmessage = message.slice(atIndex + 1).trim();
      socket.emit('privateMessage', { to: username, text: newmessage
    }, (error) => {
      $messageFormButton.removeAttribute('disabled')
      $messageFormInput.value = ''
      $messageFormInput.focus()
    })
  }
})

```

		Лаговський А.			ДУ «Житомирська політехніка».22.121.13.000 – Лр6	Арк.
		Сидорчук В.				6
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        if (error) {
            return console.log(error)
        }

        console.log('Message delivered!')
    });
}
}
})

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

let typingTimer;
const doneTypingInterval = 2000;

function sendTypingMessage() {
    socket.emit('userIsTyping');
}

document.querySelector("input[name='message']").addEventListener('input',
function () {
    clearTimeout(typingTimer);
    typingTimer = setTimeout(sendTypingMessage, doneTypingInterval);
});

socket.on('messageTyping', (message) => {
    console.log(message)
    if (document.querySelectorAll('.typing').length === 0) {
        let bgClass = 'text-black typing'
        const html = Mustache.render(messageTemplate, {
            message: `${message} is typing...`,
            bgClass: bgClass,
        })
        $messages.insertAdjacentHTML('beforeend', html)
        $messages.scrollTop = $messages.scrollHeight;
        setupClearTypingTimer()
    }
})

function clearTypingMessage() {
    const typingElements = document.querySelectorAll('.typing');
    typingElements.forEach(function (typingElement) {
        typingElement.remove();
    });
}

function setupClearTypingTimer() {
    clearTimeout(typingTimer);
    typingTimer = setTimeout(clearTypingMessage, doneTypingInterval);
}

</script>
</body>

</html>
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;

```

		Лаговський А.			ДУ «Житомирська політехніка».22.121.13.000 – Лр6	Арк.
		Сидорчук В.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

```

}

html {
  font-size: 16px;
}

input {
  font-size: 14px;
}

body {
  display: flex;
  align-items: center;
  justify-content: center;
  height: 100vh;
  background-color: #f4f5f7;
}

h1 {
  margin-bottom: 16px;
}

label {
  display: block;
  font-size: 14px;
  margin-bottom: 8px;
  color: rgba(4,18,100,0.55);
}

input {
  border: 1px solid #eeeeee;
  padding: 12px;
  outline: none;
}

button {
  cursor: pointer;
  padding: 12px;
  background: #28007f;
  border: none;
  color: white;
  font-size: 16px;
  transition: background .3s ease;
}

button:hover {
  background: #6b47b8;
}

button:disabled {
  cursor: default;
  background: #7c5cbf94;
}

.centered-form {
  display: flex;
  flex-direction: column;
  align-items: center;
  width: 100%;
  background: rgba(0,19,170,0.38);
  padding: 20px;
  border-radius: 10px;
}

```

		Лаговський А.			ДУ «Житомирська політехніка».22.121.13.000 – Лр6	Арк.
		Сидорчук В.				8
Змн.	Арк.	№ докум.	Підпис	Дата		


```

.centered-form__box {
  background-color: #ffffff;
  padding: 2rem;
  border-radius: 15px;
  box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
  width: 100%;
  max-width: 400px;
}

.btn-custom {
  background-color: #ffc107;
  border: none;
  border-radius: 10px;
  width: 100%;
}

.btn-custom:hover {
  background-color: #e0a800;
}

.centered-form input {
  border-radius: 10px;
}

.centered-form__box p {
  text-align: center;
  margin-bottom: 1.5rem;
  font-size: 0.9rem;
  color: #888;
}

/* Chat Page Layout */

.chat {
  display: flex;
}

.chat__sidebar {
  height: 100vh;
  color: white;
  background: #333744;
  width: 100%;
  overflow-y: scroll
}

/* Chat styles */

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

.chat__messages {
  flex-grow: 1;
  padding: 24px 24px 0 24px;
  overflow-y: scroll;
}

/* Message Styles */

.message {
  margin: 8px;
}

```

		Лаговський А.			ДУ «Житомирська політехніка».22.121.13.000 – Лр6	Арк.
		Сидорчук В.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

```

}

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

.message-meta {
  font-size: 14px;
}

.message a {
  color: #0070CC;
}

/* Message Composition Styles */

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

#compose button {
  font-size: 14px;
  width: 150px;
}

/* Chat Sidebar Styles */

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

.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 {
  max-height: 100%;
  overflow-y: auto;
  border: 1px solid #ccc;
  padding: 10px;
}

```

		Лаговський А.			ДУ «Житомирська політехніка».22.121.13.000 – Лр6	Арк.
		Сидорчук В.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

Висновок: під час виконання даної лабораторної роботи, я навчився користуватися протоколами WebSocket. Використовувати Socket.io для розробки чат-додатків.

GitHub: <https://github.com/AdamLahovskyi/NodeJS>

		Лаговський А.			ДУ «Житомирська політехніка».22.121.13.000 – Лр6	Арк.
		Сидорчук В.				11
Змн.	Арк.	№ докум.	Підпис	Дата		