

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Лабораторна робота №1

з дисципліни «Основи WEB - технологій»

Тема: «Введення в програму Cisco Packet Tracer, режим симуляції»

Перевірів:

Доц. Голубєв Л. П.

Виконав:

студент групи ІМ-22

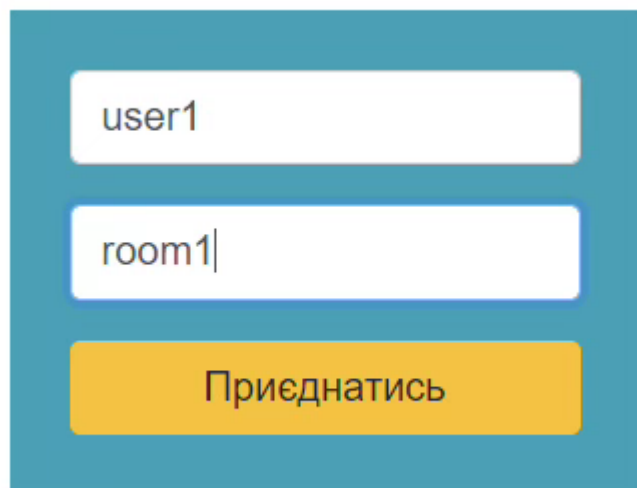
Балахон Михайло

Варіант 11

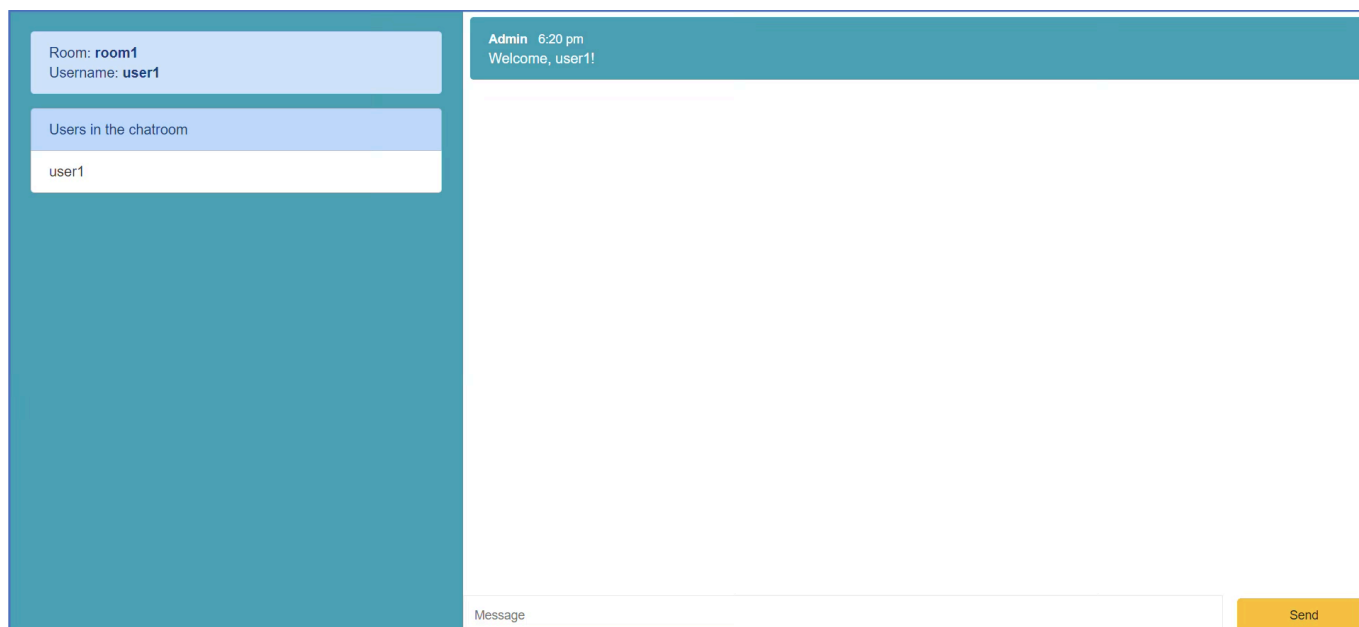
Київ - 2025

Завдання 1. Розробити додаток для обміну повідомленнями між учасниками в режимі реального часу (chat) за допомогою бібліотеки SocketIO (або іншої).

Приклад роботи chat-додатку



A registration form with a teal background. It contains three input fields: the first contains 'user1', the second contains 'room1|'. Below the inputs is a yellow button with the text 'Приєднатись' (Join).



A screenshot of a chat application interface. On the left is a sidebar with a teal background. It has a light blue box at the top showing 'Room: room1' and 'Username: user1'. Below it is another light blue box labeled 'Users in the chatroom' containing a white box with 'user1'. The main area on the right has a teal header bar with the message 'Admin 6:20 pm Welcome, user1!'. Below the header is a large white chat area. At the bottom, there is a white input field labeled 'Message' and a yellow 'Send' button.

Room: **room1**
Username: **user1**

Users in the chatroom

user1

Admin 6:20 pm
Welcome, user1!

user1 6:21 pm
Hello!

Message

Send

user2

room1|

Приєднатись

Room: **room1**
Username: **user2**

Users in the chatroom

user1

user2

Admin 6:23 pm
Welcome, user2!

Message

Send

Room: **room1**
Username: **user2**

Users in the chatroom

user1

user2

Admin 6:23 pm
Welcome, user2!

user2 6:23 pm
Hi!

Message

Send

Room: **room1**
Username: **user1**

Users in the chatroom

user1

user2

Admin 6:20 pm
Welcome, user1!

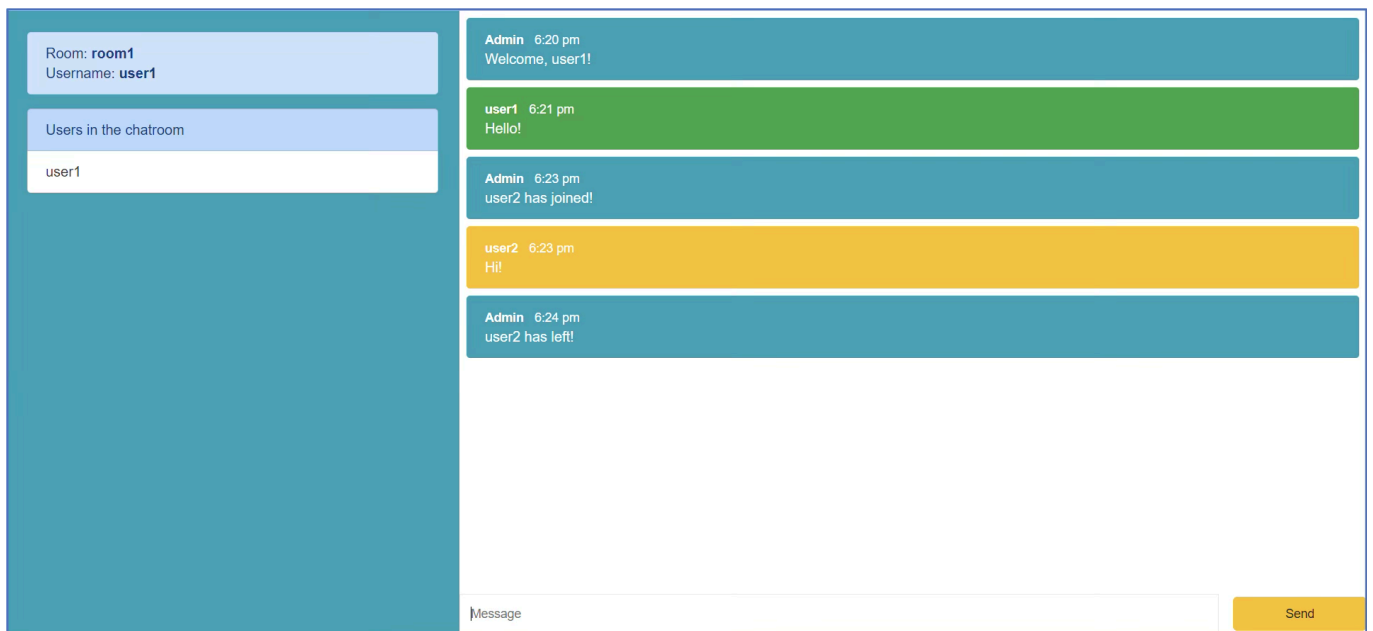
user1 6:21 pm
Hello!

Admin 6:23 pm
user2 has joined!

user2 6:23 pm
Hi!

Message

Send



Хід роботи:

1. Підготовка проекту

1.1. Ініціалізація Node.js проекту

```
npm init -y
```

Створено файл package.json з базовою конфігурацією проекту.

1.2. Встановлення необхідних залежностей

```
npm install express socket.io
```

Встановлено:

- Express - веб-фреймворк для Node.js
- Socket.io - бібліотека для real-time двостороннього зв'язку

2. Створення серверної частини

2.1. Файл server.js

Створено основний серверний файл з наступною функціональністю:

```
const express = require('express');  
const http = require('http');  
const socketio = require('socket.io');
```

Основні компоненти сервера:

- Налаштування Express сервера
- Інтеграція Socket.io
- Зберігання активних користувачів та кімнат
- Обробка подій підключення/відключення
- Обробка повідомлень чату

Реалізовані Socket.io події:

- connection - нове підключення користувача
- join - приєднання до кімнати
- chatMessage - відправка повідомлення
- disconnect - відключення користувача

3. Створення клієнтської частини

3.1. Сторінка входу (index.html)

Створено форму для введення:

- Імені користувача
- Назви кімнати

Дані зберігаються в sessionStorage для передачі на сторінку чату.

3.2. Сторінка чату (chat.html)

Реалізовано інтерфейс чату з:

- Відображенням поточної кімнати та користувача
- Списком активних користувачів
- Областю для повідомлень
- Формою відправки повідомлень

JavaScript функціональність:

- Підключення до Socket.io сервера
- Приєднання до кімнати
- Відправка та отримання повідомлень
- Оновлення списку користувачів

4. Стилiзація додатку (styles.css)

Створено стилі відповідно до макету з лабораторної роботи:

- Блакитна колірна схема (#17a2b8)
- Жовті акценти для кнопок (#ffc107)
- Адаптивний дизайн для мобільних пристроїв
- Різні стилі для власних повідомлень, повідомлень інших користувачів та системних повідомлень

5. Структура проекту

lab1/

```
|— server.js      # Серверна логіка
|— package.json  # Конфігурація та залежності
|— public/
|  |— index.html  # Сторінка входу
|  |— chat.html   # Інтерфейс чату
|  |— styles.css  # Стилi додатку
|— README.md     # Документація
```

6. Запуск та тестування

6.1. Запуск сервера

`npm start`

Сервер запускається на порту 3001.

6.2. Тестування функціональності

1. Відкрито декілька вікон браузера
2. В кожному вікні виконано вхід з різними іменами користувачів
3. Перевірено:
 - Відправка та отримання повідомлень в реальному часі
 - Відображення списку активних користувачів
 - Повідомлення про приєднання/вихід користувачів
 - Ізоляція повідомлень між різними кімнатами

Результати роботи

Розроблено повнофункціональний чат-додаток з наступними можливостями:

1. Реальний час - повідомлення доставляються миттєво завдяки WebSocket протоколу
2. Множинні кімнати - користувачі можуть створювати та приєднуватися до різних кімнат
3. Відслідковування користувачів - відображається список активних користувачів у кімнаті
4. Системні повідомлення - сповіщення про приєднання та вихід користувачів
5. Адаптивний дизайн - додаток коректно працює на різних пристроях

Висновки

В ході виконання лабораторної роботи:

- Вивчено принципи роботи WebSocket протоколу
- Освоєно бібліотеку Socket.io для real-time комунікації
- Реалізовано повноцінний чат-додаток з підтримкою кімнат
- Отримано практичні навички роботи з Node.js та Express

Розроблений додаток демонструє ефективність WebSocket для створення інтерактивних real-time додатків, де важлива миттєва доставка даних між клієнтами.