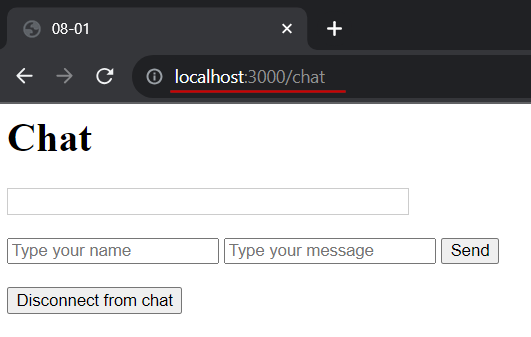
Лабораторная работа 07

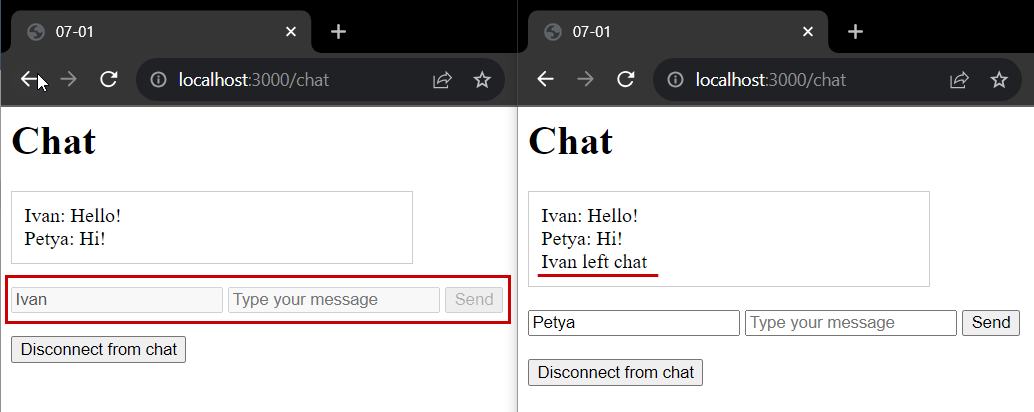
СТРWП

**Задание 01**

1. Разработайте приложение **07-01**, представляющее собой HTTP-сервер и WebSocket(WS)-сервер одновременно.
2. HTTP-сервер должен прослушивать порт **3000**.
3. WS-север должен прослушивать порт **4000**.
4. HTTP-сервер реагирует только на GET-запрос **/start**, на остальные запросы HTTP-сервер должен возвращать ответ со статусом **400**.
5. На GET-запрос **/chat** HTTP-сервер возвращает HTML-страницу следующего вида:



1. При загрузке страницы должно устанавливаться ws-соединение с сервером.
2. При нажатии кнопки **Send** должна запускаться JS-функция, отправляющая указанное имя пользователя и сообщение WS-серверу.
3. При нажатии кнопки **Disconnect from chat** должна запускаться JS-функция, в которой происходит закрытие соединения с сервером. При этом поля для ввода имени и сообщения, а также кнопка для отправки сообщения должны блокироваться. Остальным участникам чата должно приходить сообщение о том, что пользователь покинул чат.
4. JS-скрипт должен отображать сообщения, принятые от сервера, на странице в блоке выше, как показано на картинке ниже:



1. WS-сервер принимает сообщения от клиента и пересылает их всех подключенным клиентам.
2. WS-сервер отображает принятые сообщения от клиента.
3. Проверьте работоспособность приложения. Запустите несколько клиентовдля взаимодействия с одним общим сервером. Объясните наблюдаемый эффект.

**Задание 02.** Ответьте на следующие вопросы

1. Поясните разницу между полудуплексным и дуплексным каналами данными.
2. Какой тип канала применяется HTTP-протоколом.
3. Поясните понятие **WebSocket**.
4. Какой тип канала применяется WebSocket-протоколом.
5. Расскажите процедуру WebSocket-рукопожатия.
6. Полудуплексный канал данных позволяет передавать данные в обоих направлениях, но только в одном направлении одновременно. Это означает, что обмен данными происходит в форме дуплексного канала, но только в одном направлении за раз. После передачи данных в одном направлении канал должен переключиться на другое направление для передачи данных в обратном направлении. Например, радиосвязь на базе push-to-talk является примером полудуплексного канала. В отличие от этого, дуплексный канал данных позволяет одновременно передавать и принимать данные в обоих направлениях.
7. HTTP-протокол использует полудуплексный канал данных. Это означает, что клиент отправляет запрос на сервер, а сервер отправляет ответ обратно клиенту. Однако в любой момент времени только одно из этих действий может происходить.
8. WebSocket - это протокол связи поверх TCP-соединения, который обеспечивает возможность обмена данными между клиентом и сервером в режиме реального времени. Он предоставляет полнодуплексный канал связи, что означает, что клиент и сервер могут отправлять и получать данные одновременно.
9. WebSocket-протокол применяет дуплексный канал данных, что позволяет одновременно передавать и принимать данные в обоих направлениях.
10. Процедура WebSocket-рукопожатия включает в себя следующие шаги:
11. Клиент отправляет HTTP-запрос на сервер, в заголовке которого содержится информация о протоколах, среди которых указан WebSocket.
12. Сервер обрабатывает этот запрос и, если он поддерживает WebSocket, он отправляет ответ с кодом 101 (Switching Protocols), указывая, что соединение было успешно установлено и протокол переключен на WebSocket.
13. После этого соединение переходит в режим WebSocket, и клиент и сервер могут обмениваться данными в полнодуплексном режиме.