Задание “Чат”

Исходные данные

Участникам олимпиады предлагается организовать чат.
Чат состоит из страницы инициализации чата, страницы присоединения к чату и страницы самого чата.

Страница инициализации чата

Состоит ровно поля “Имя пользователя” и кнопки “Начать чат”. После ее нажатия создается новый чат, с уникальным ключом и пользователь перебрасывается на страницу чата.

Страница присоединения к чату

Если пользователь открыл страницу чата первый раз - у него спрашивается его имя, и если имя уникально - он попадает на страницу чата. Соответственно - страница состоит из поля “Имя пользователя ” и кнопки “Присоединиться”.

Страница чата

Состоит из надписи с именем пользователя, блока с историей чата и поля с новым сообщением.
Когда вводится новое сообщение - он тут же появляется в истории у пользователя. У остальных пользователей оно появляется с задержкой не более 5 секунд и на своем месте
История показывается в виде списка сообщения, где каждое сообщение содержит имя пользователя, время сообщения и текст сообщения. URL страницы можно выделить и отправить другим пользователям. Они будут попадать в чат через страницы присоединения к чату.

Базовая задача (обязательная)

1. Реализовать все 3 страницы
2. Обновление страницы чата сделать через AJAX
Максимальное количество баллов: 232

Дополнительные задачи

1. Для отрисовки страниц использовать bootstrap 4.x
Максимальное количество баллов: 60

2. Для обновления страницы чата использовать не просто AJAX, а передачу данных через JSON.
Причем - как для обновления истории, так и для отправки нового сообщения.
Максимальное количество баллов: 100

3. Показывать число пользователей онлайн на странице чата с точностью в 30 секунд. То есть - если пользователь закрыл вкладку - то не позже, чем через 30 секунд он должен исчезнуть из списка активных пользователей у всех участников чата.
Максимальное количество баллов: 120

Максимальное количество баллов за задание: 512

Задание "Игра в Шашки"

Исходные данные

Участникам олимпиады предлагается реализовать RESTful API для игры в Шашки.
API включает в себя следующие методы:
1. Регистрация игрока. Поля для регистрации нового пользователя - имя и email. Результат метода - ID пользователя
2. Получение списка игроков с возможностью поиска по имени и email. Результат - список игроков.
3. Создание игры для двух существующих игроков по их идентификаторам. Результат - ID игры.
4. Выполнение хода. Игрок делает свой ход путем передачи в метод координат начальной и конечной клетки доски, показывая какая шашка и куда перемещается. Например, {"from":"a3", "to":"b4"}. При этом между начальной и конечной клеткой может находиться шашка противника, в этом случае она считается битой. В реализации метода должна быть проверка на возможность выполнения предлагаемого хода. Результат метода - успех или ошибка.
5. Получить текущее состояние игры. Результат - массив с состоянием всех клеток поля, состояние игры (игра продолжается, ничья, определен победитель), чей следующий ход (если игра продолжается), победитель (если он определен).

Базовая задача (обязательная)

Реализовать перечисленные выше методы
Максимальное количество баллов: 252

Дополнительные задачи

1. Добавить пагинацию и сортировку для метода получения списка игроков. Метод должен принимать дополнительные параметры - номер страницы, поле сортировки (имя, email), порядок сортировки (по возрастанию или убыванию).
Максимальное количество баллов: 80

2. Добавить рейтинги в метод получения списка игроков. За каждую победу значение рейтинга увеличивается на 1, за проигрыш - уменьшается на 1.
Максимальное количество баллов: 60

3. Для метода получения состояния игры добавить суммарное время ходов по игрокам (время начинает считаться с момента первого хода в игре)
Максимальное количество баллов: 60

4. Реализовать аутентификацию игроков. Для этого необходимо в регистрационные данные добавить поле для пароля и добавить новый метод для аутентификации игрока по имени и паролю. Результат метода - строка, содержащая access token. Метод выполнения хода теперь должен принимать access token вместо ID пользователя.
Максимальное количество баллов: 120

Максимальное количество баллов: 572

Система документаоборота компании.

Пример - всякого рода должностные инструкции, примеры договоров.

Исходные данные

Каждый документ должен иметь следующие поля:

- название
- краткое описание
- Содержание документа
- Родительский документ (нужно для различного рода приложений для договоров)
- Позиция в списке
- Возможность прикрепить файлы к документу

Базовая задача (обязательная)

1. Реализовать страницу с древовидным списком всех документов

2. Реализовать страницу добавления/редактирования документов
3. Реализовать страницу просмотра документа

Максимальное количество баллов: 232

Дополнительные задачи

1. Показывать иконки тех пользователей, кто сейчас просматривает документ.

Максимальное количество баллов: 60

2. Показывать правки онлайн.

Максимальное количество баллов: 60

3. Используем bootstrap 4.x

Максимальное количество баллов: 60

4. Реализовать простейшую систему регистрации, авторизации и механизм групп пользователей

- Предполагается наличие логина и пароля;
- Предполагается, что все пользователи будут одного типа - только администраторы;
- Предполагается создание групп пользователей - отдельная сущности. В настройках пользователя-администратора можно будет выбрать только одну группу пользователей;
- Предполагается создание двух новых полей-настроек для документов:
    - Пользовательские группы, которые могут редактировать данный документ;
    - Пользовательские группы, которые могут просматривать данный документ;

Максимальное количество баллов: 150

Максимальное количество баллов за задание: 562