

1. Flashcards



В рамках данного проекта необходимо разработать бэкэнд для приложения, позволяющего изучать иностранные слова по технике флэш-карточек.

Функционал приложения

В приложении будут реализованы четыре основные сущности: *пользователь, язык, карточка, задание*.

Пользователь:

- Пользователи могут иметь одну из двух ролей: user, admin.
- Регистрация пользователя происходит по электронной почте, имени пользователя и паролю. Email должен быть уникальным. Пароль должен состоять не менее чем из 8 латинских символов нижнего и верхнего регистра, как минимум одной цифры и как минимум одного спецсимвола (!@#\$%^&*()_+=). Имя пользователя должно иметь длину не менее 5 и не более 256 символов.
- Вход в аккаунт осуществляется по email и паролю.
- Пользователь с ролью "admin" создается с помощью миграции (установить стандартные параметры для входа: email: admin@admin.com, password: Admin123!, username: admin).

Язык:

- Пользователь с ролью “admin” может создавать, редактировать и удалять языки.
- Сущность **язык** включает в себя полное название (например, “English”) и уникальный код языка (например, “en”).
- Авторизованный пользователь может просматривать список всех доступных языков. Список должен включать пагинацию и возможность сортировки по дате добавления (по возрастанию и по убыванию) и по названию (по возрастанию и по убыванию), а также регистрационезависимый поиск по названию языка.
- Если админ добавляет новый язык, необходимо уведомить об этом всех пользователей (продемонстрировать работу уведомления через html страницу).

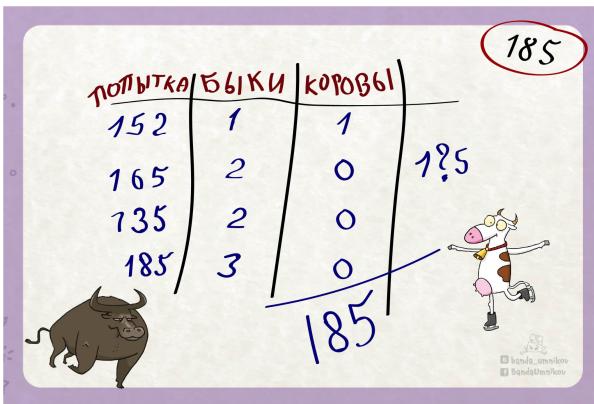
Карточка:

- Авторизованный пользователь может создавать, редактировать и удалять карточки. Необходимо запрещать создание одинаковых слов для одного и того же языка.
- Сущность **карточка** должна включать в себя слово на родном языке, перевод слова на иностранный язык (***реализовать возможность добавления нескольких вариантов перевода**).
- Авторизованный пользователь может просматривать список карточек выбранного языка. Список должен включать пагинацию и возможность сортировки по дате добавления (по возрастанию и по убыванию), по слову на родном/иностранным языке (по возрастанию и по убыванию), а также регистрационезависимый поиск по слову на родном/иностранным языке.

Задание:

- Авторизованный пользователь может получить задание. Для этого необходимо передать в запрос идентификатор языка и тип задания (to_foreign/to_native). Запрос возвращает идентификатор задания и слово на родном/иностранным языке в зависимости от типа задания.
- Авторизованный пользователь может ответить на задание. Запрос принимает идентификатор задания и перевод. Возвращает результат “верно/неверно”.
- Авторизованный пользователь может просмотреть список неотвеченных заданий для выбранного языка. Список должен включать пагинацию и возможность сортировки по дате добавления (по возрастанию и по убыванию), а также регистрационезависимый поиск по слову из формулировки задания.
- Авторизованный пользователь может просмотреть статистику по правильным/неправильным ответам за выбранный период и по выбранным языкам.

2. Игра “Быки и коровы”



В рамках данного проекта необходимо разработать бэкэнд для игры “Быки и коровы”. С правилами игры можно ознакомиться [тут](#).

Функционал приложения

В приложении будут реализованы три основные сущности: *пользователь, игра, ход*.

Пользователь:

- Пользователи могут иметь одну из двух ролей: user, admin.
- Регистрация пользователя происходит по электронной почте, имени пользователя и паролю. Email должен быть уникальным. Пароль должен состоять не менее чем из 8 латинских символов нижнего и верхнего регистра, как минимум одной цифры и как минимум одного спецсимвола (!@#\$%^&*()_+=). Имя пользователя должно иметь длину не менее 5 и не более 256 символов.
- Вход в аккаунт осуществляется по email и паролю.
- Пользователь с ролью “admin” создается с помощью миграции (установить стандартные параметры для входа: email: admin@admin.com, password: Admin123!, username: admin).

Игра:

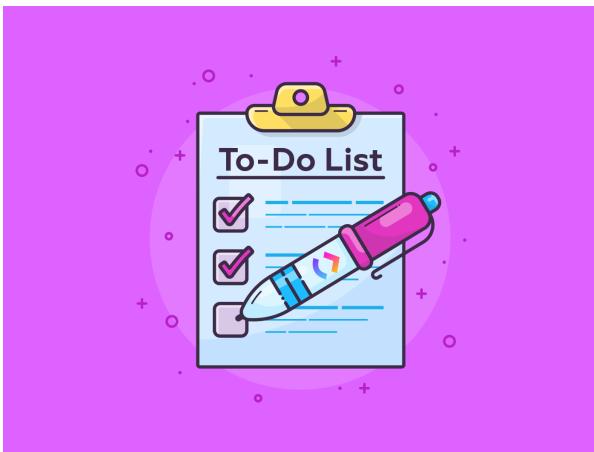
- Авторизованный пользователь может создавать новую игру. В запрос передается идентификатор пользователя, которого планируется пригласить в игру. Запрос возвращает идентификатор игры. Необходимо запрещать создавать новую игру, если у данного пользователя уже есть незавершенная игра с тем пользователем, которого приглашают. После создания игры каждый из игроков может загадать число с соответствующим количеством разрядов (цифры не должны повторяться). Запрещать изменение загаданного числа, если уже сделан хотя бы один ход в данной игре.
- Пользователь с ролью “admin” может задать параметр игры: количество разрядов загадываемых чисел (по умолчанию 4).

- Создатель игры может изменить противника, если он еще не загадал число. Создатель игры может также удалить игру, если в ней еще не было сделано ни одного хода.
- Авторизованный пользователь может получить информацию об игре, участником которой он является, список ходов статус игры).
- Авторизованный пользователь может получить список своих игр. Запрос предполагает пагинацию, фильтрацию по пользователям и статусам игр, сортировку по дате создания игры.
- Пользователь может просмотреть статистику по играм любого пользователя: количество побед, количество проигрыш, среднее количество ходов до победы.
- Пользователь может просмотреть лидерборд пользователей, сформированный по одному из критериев: количество выигранных игр, количество завершенных игр, количество ходов до победы. Лидерборд формируется по выбранному промежутку времени.

Ход:

- Авторизованный пользователь может сделать ход в игре. Запрещать делать ход, если хотя бы один из игроков еще не загадал число. Запрещать делать несколько ходов подряд. Запрос возвращает количество "быков" и "коров". Запрещать делать ходы в завершенной игре.
- Присыпать игроку уведомление о том, что его противник сделал ход (продемонстрировать работу уведомления через html страницу).

3. Список задач



В рамках данного проекта необходимо разработать бэкэнд для отслеживания списка дел пользователя.

Функционал приложения

В приложении будут реализованы три основные сущности: *пользователь*, *задача*, *группа пользователей*.

Пользователь:

- Пользователи могут иметь одну из двух ролей: user, admin.
- Регистрация пользователя происходит по электронной почте, имени пользователя и паролю. Email должен быть уникальным. Пароль должен состоять не менее чем из 8 латинских символов нижнего и верхнего регистра, как минимум одной цифры и как минимум одного спецсимвола (!@#\$%^&*()_+=). Имя пользователя должно иметь длину не менее 5 и не более 256 символов.
- Вход в аккаунт осуществляется по email и паролю.
- Пользователь с ролью “admin” создается с помощью миграции (установить стандартные параметры для входа: email: admin@admin.com, password: Admin123!, username: admin).

Группа пользователей:

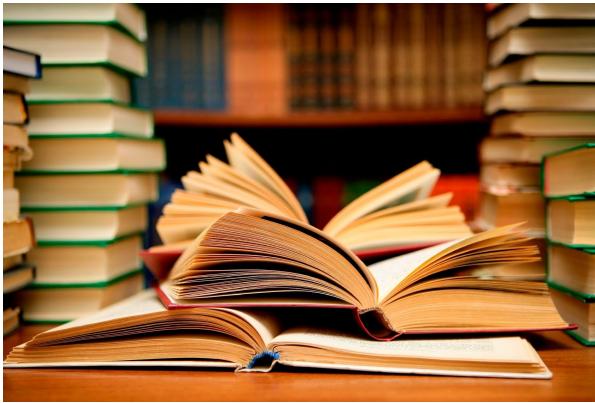
- Пользователь с ролью “admin” может создать группу пользователей. Группа включает название и описание. Админ может редактировать группы и удалять их, если туда еще не добавлено ни одного пользователя. Админ может добавить в группу пользователя и удалить пользователя из группы.
- Админ может просматривать список групп. Список должен включать пагинацию, сортировку по названию (по возрастанию и по убыванию), по дате создания (по возрастанию и по убыванию). Для каждой группы в списке

возвращать идентификатор, название, дату создания и количество участников. Также можно просмотреть информацию о конкретной группе по идентификатору. Данный запрос должен возвращать идентификатор, название, описание, дату создания и список участников.

Задача:

- Авторизованный пользователь может создать задачу. Задача включает название, описание, крайний срок исполнения, приоритет (low, normal, high), статус (to do, in progress, done). Проверять, что крайний срок исполнения не меньше, чем текущая дата.
- Авторизованный пользователь может редактировать задачу (только созданную им). Статус задачи нельзя отредактировать после дедлайна.
- Авторизованный пользователь может удалить задачу (только созданную им).
- Авторизованный пользователь может просмотреть список задач (своих и пользователей из групп, в которых он состоит). Запрос предполагает пагинацию, фильтрацию по пользователям, статусам и приоритетам, сортировку по дате дедлайна и приоритету.
- Авторизованный пользователь может просмотреть статистику по задачам любого пользователя из групп, в которых он состоит: количество задач в каждом из статусов; для задач, у которых прошел дедайн, статистику, сколько задач было выполнено и сколько нет.
- Уведомлять всех пользователей группы, о том, что была добавлена новая задача.

4. Библиотека



В рамках данного проекта необходимо разработать бэкэнд, моделирующий работу библиотеки.

Функционал приложения

В приложении будут реализованы три основные сущности: *пользователь, книга, бронирование книги*.

Пользователь:

- Пользователи могут иметь одну из двух ролей: user, admin.
- Регистрация пользователя происходит по электронной почте, имени пользователя и паролю. Email должен быть уникальным. Пароль должен состоять не менее чем из 8 латинских символов нижнего и верхнего регистра, как минимум одной цифры и как минимум одного спецсимвола (!@#\$%^&*()_+=). Имя пользователя должно иметь длину не менее 5 и не более 256 символов.
- Вход в аккаунт осуществляется по email и паролю.
- Пользователь с ролью “admin” создается с помощью миграции (установить стандартные параметры для входа: email: admin@admin.com, password: Admin123!, username: admin).

Книга:

- Пользователь с ролью “admin” может добавлять, редактировать и удалять книги. Книга содержит название, жанр, автора, описание.
- Любой пользователь может просматривать список книг. Список предполагает пагинацию, сортировку по названию, автору, фильтрацию по жанру и по автору, поиск по названию и по имени автора. В списке видны названия, жанры и авторы, состояние книги (забронирована/не забронирована).
- Можно просмотреть конкретную книгу по идентификатору, в таком случае запрос возвращает название, жанр, автора и описание книги, состояние книги (забронирована/не забронирована).

Бронирование:

- Авторизованный пользователь может забронировать книгу на определенный промежуток времени, начиная с момента бронирования, но не более, чем на 1 месяц. Нельзя забронировать книгу, которая в данный момент забронирована кем-то другим.
- Авторизованный пользователь может вернуть книгу.
- Авторизованный пользователь может просмотреть список забронированных им книг на данный момент, а также список книг, которые он бронировал ранее. Списки должны включать пагинацию, поиск по названию книги и автору, сортировку по дате бронирования.
- У пользователей есть рейтинг от 0 до 5, за каждую вовремя возвращенную книгу рейтинг увеличивается на 0.1, если он еще не достиг 5. За каждую просроченную книгу рейтинг уменьшается на 0.1 в день. Как только рейтинг становится 0, пользователь блокируется и не может бронировать книги, пока админ его не разблокирует.
- Любой пользователь может просматривать лидерборд юзеров по рейтингу.
- Когда админ добавляет новую книгу, уведомлять всех пользователей.