

Конструктор тестов

Оглавление

1 Описание	3
2 Схема базы данных	3
3 Описание сущностей.....	4
3.1 Пользователь [user]	4
3.2 Тест [test]	4
3.3 Вопрос [question].....	4
3.4 Ответ [answer]	5
3.5 Пройденные тесты [completed_test]	5
3.6 Ответы пользователей [User_answers].....	6
4 Эндпоинты	6
4.1 Пользователи	6
4.2 Тесты.....	6
4.3 Ответы.....	7
5 Дизайн	7
5.1 Регистрация	7
5.2 Список созданных тестов	8
5.3 Меню	8
5.4 Список пройденных тестов	9
5.5 Прохождение теста	9
5.6 Создание нового теста	10
Список полезных ссылок.....	11

1 Описание

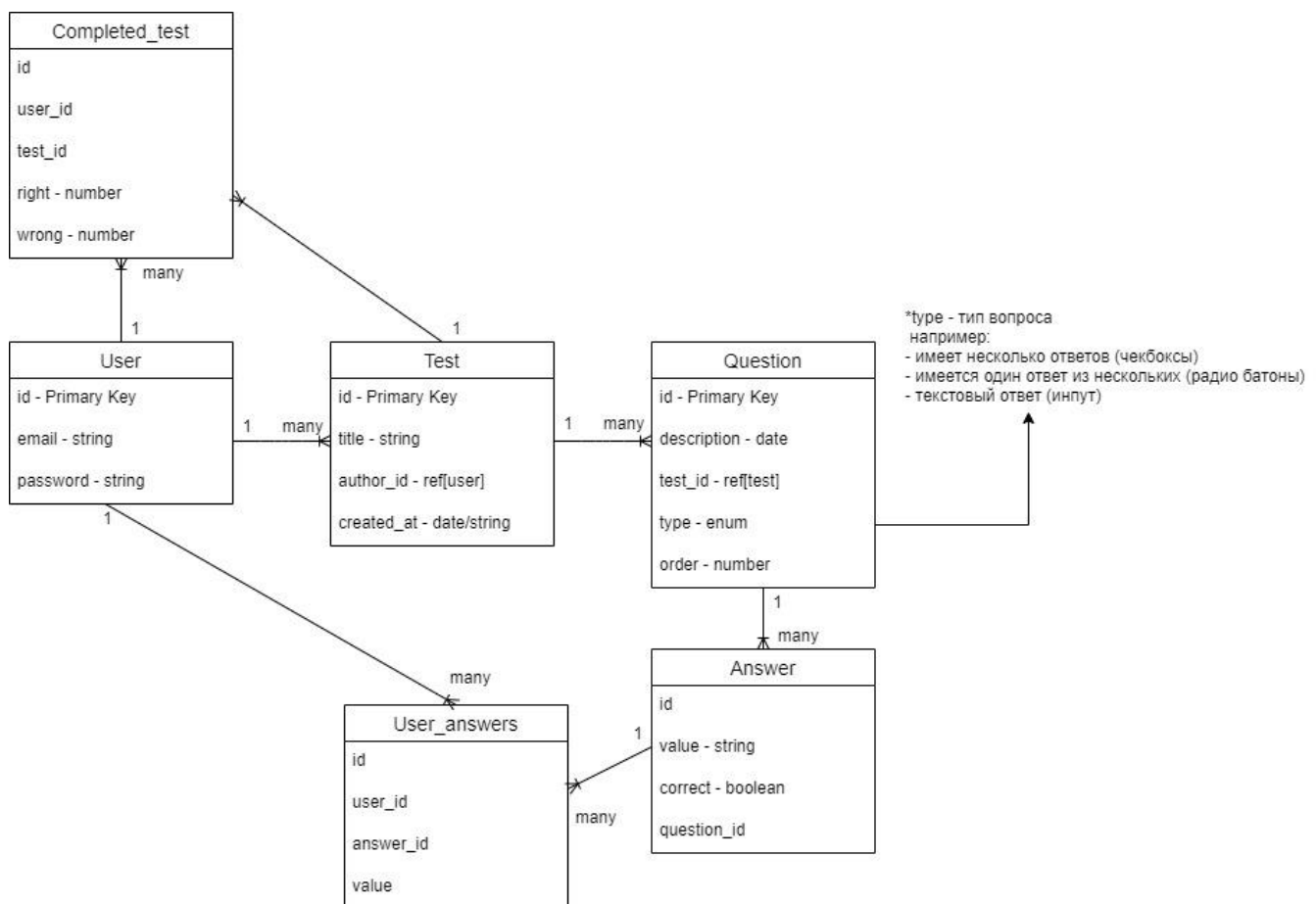
Приложение представляет из себя конструктор тестов, в котором каждый пользователь может создать тест, заполнить его вопросами разных типов и указать правильные варианты ответа.

2 Схема базы данных

Можно использовать как реляционную, так и NoSQL базу данных. В зависимости от выбора необходимо правильно продумать организацию данных.

Ролик про реляционные БД – <https://youtu.be/IK6e1SFCdow>

Ролик про NoSQL на примере MongoDB - <https://youtu.be/LNvmI8a9jwY>



3 Описание сущностей

3.1 Пользователь [user]

Поле	Тип	Описание
id	Число строка (Первичный ключ)	Уникальный идентификатор
email	Строка	Почтовый адрес
password	Строка	Пароль пользователя (храним в захешированном виде)

По желанию можно добавить аватар и любые другие поля для пользователя.

3.2 Тест [test]

Поле	Тип	Описание
id	Число строка (Первичный ключ)	Уникальный идентификатор
title	Строка	Почтовый адрес
author_id	Число строка	Внешний ключ на таблицу User
Created_at	Дата строка	Дата создания теста

3.3 Вопрос [question]

Поле	Тип	Описание
id	Число строка (Первичный ключ)	Уникальный идентификатор
description	Строка	Сам вопрос, например: в каком году был создан язык JavaScript?
Type	Перечисление	Тип вопроса, например: - Несколько вариантов ответа (чекбоксы) - Один вариант ответа из нескольких (радио батон) - Письменный ответ (инпут)
order	Число	Номер вопроса
Test_id	Число строка	Внешний ключ на таблицу Test

3.4 Ответ [answer]

Поле	Тип	Описание
id	Число строка (Первичный ключ)	Уникальный идентификатор
value	Строка	Ответ на вопрос, например: - в 1907 году
correct	Логический	Является ли ответ на вопрос правильным true false
question_id	Число строка	Внешний ключ на таблицу Question

3.5 Пройденные тесты [completed_test]

Таблица, в которой хранятся сведения о пройденных тестах с количеством правильных\неправильных ответов.

Поле	Тип	Описание
id	Число строка (Первичный ключ)	Уникальный идентификатор
User_id	Число строка	Внешний ключ на таблицу User
Test_id	Число строка	Внешний ключ на таблицу Test
right	Число	Количество правильных ответов
wrong	Число	Количество неправильных ответов

3.6 Ответы пользователей [User_answers]

Таблица, в которой хранятся ответы пользователей. Сохраняем сразу же, после того как пользователь выбрал тот или иной вариант. Это делается на случай, если сессия прервется и ответы не были утеряны.

Поле	Тип	Описание
id	Число строка (Первичный ключ)	Уникальный идентификатор
User_id	Число строка	Внешний ключ на таблицу User
Anser_id	Число строка	Внешний ключ на таблицу Answer
value	Строка	Ответ на вопрос, важен только в случае типа вопроса с инпутом

4 Эндпоинты

4.1 Пользователи

Все про JWT авторизацию - <https://youtu.be/fN25fMQZ2v0>

- POST /api/users/registration - регистрация
- POST /api/users/login - логин
- GET /api/users/refresh – обновление токена
- GET /api/users/activate/:link – подтверждение почты
- POST /api/users/password – сброс пароля

4.2 Тесты

- POST /api/tests – создать тест
- GET /api/tests – получить все тесты, которые созданы текущим пользователем
- GET /api/tests/:id – получить тест по id
- PUT /api/tests – обновить тест
- DELETE /api/tests/:id– удалить тест по id

- GET /api/tests/completed – получить список пройденных тестов текущим пользователем
- GET /api/tests/completed/:id – получить детальную информацию по пройденному тесту (с правильными\неправильными ответами)

4.3 Ответы

- POST /api/ answers/answer_id – ответить
 - GET /api/answers/:test_id – получить ответы пользователя на тест по ID теста
- Возможно в ходе разработки появится необходимость добавить другие эндпоинты.

5 Дизайн

Прототип дизайна (масштабы не соблюдены, делалось на глаз) –

<https://www.figma.com/file/lfB2DBjGayUOf7iiGW31KP/Constructor?node-id=0%3A1>

Имплементацию делаете по своему вкусу.

5.1 Регистрация

Регистрация

Войти

Регистрация

Забыли пароль?

Войти

5.2 Список созданных тестов

Мои тесты

Создать

Название	Дата создания	Кол-во участников	Действия
Тест 1	03.06.2021	5	 
Тест 2	03.06.2021	5	 
Тест 3	03.06.2021	5	 
Тест 4	03.06.2021	5	 

5.3 Меню

Мои тесты

Мои тесты

Завершённые

Профиль

Выйти

Название	Дата создания	Кол-во участников	Дейст
Тест 1	03.06.2021	5	
Тест 2	03.06.2021	5	 
Тест 3	03.06.2021	5	 
Тест 4	03.06.2021	5	 

5.4 Список пройденных тестов



Пройденные тесты

Название	Дата прохождения	Результат	Действия
Тест 1	03.06.2021	32/40	<button>открыть</button>
Тест 1	03.06.2021	32/40	<button>открыть</button>
Тест 1	03.06.2021	32/40	<button>открыть</button>

5.5 Прохождение теста



Программистский тест

1. Когда появился JavaScript?

- ☐ 1987
- ☐ 1995
- ☐ 2003

2. Какие языки используются для разработки backend

- ☐ Java
- ☐ Swift

[Выйти](#)

[Закончить](#)

5.6 Создание нового теста



Создание нового теста

1. Когда появился JavaScript?

Один вариант ответа

- ☐ 1987
- ☐ 1995
- ☐ 2003

+ добавить



2. Какие языки используются для разработки backend

Несколько вариантов

- ☐ Java
- ☐ Swift

+ добавить



Отменить Сохранить

Список полезных ссылок

Создание fullstack интернет магазина. Много пересечений с этим проектом, можно взять за основу - <https://youtu.be/H2GCkRF9eko>

Все про JWT авторизацию - <https://youtu.be/fN25fMQZ2v0>

Ролик про реляционные БД – <https://youtu.be/IK6e1SFCdow>

Ролик про NoSQL на примере MongoDB - <https://youtu.be/LNvmI8a9jwY>

Курс по Redux - https://www.youtube.com/playlist?list=PL6DxKON1uLOHsBCJ_vVuvRsW84VnqmPp6

Макет - <https://www.figma.com/file/lfB2DBjGayUOf7iiGW31KP/Constructor?node-id=0%3A1>