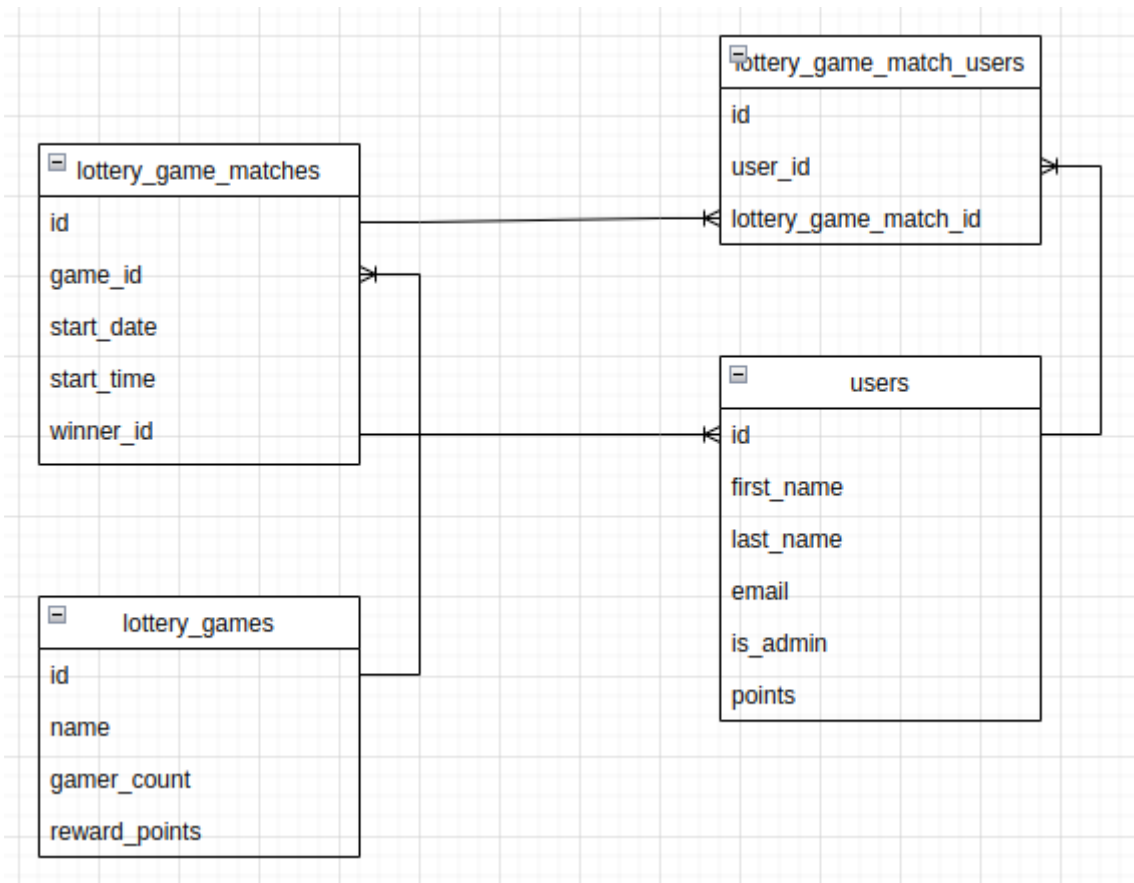


Задача

- 1. Развернуть docker контейнер с проектом на Lumen и контейнер для БД с СУБД PostgreSQL.
- 2. Составить README.md, включающий в себя:
 - необходимые шаги для разворота проекта
 - необходимые конфигурации проекта
- 3. Реализовать следующую схему БД с помощью Laravel миграций:



- 4. Создать Eloquent модели для представленных в БД сущностей.
- 5. Заполнить таблицы `users` и `lottery_games`, используя `seeder` классы и `factory` классы.
- 6. Реализовать следующий список API (ответ должен быть в формате json):

Адрес	Описание	Ролевой доступ
POST /api/users/register	Создание пользователя	Гость
POST /api/users/login	Авторизация. Получение jwt-токена авторизации	Гость
PUT /api/users/1	Редактирование пользователя	Авторизованный пользователь, владелец учетной записи
DELETE /api/users/1	Удаление пользователя	Авторизованный пользователь, владелец учетной записи

GET /api/lottery_games	Получение списка всех лотерейных игр	Гость, авторизованный пользователь, администратор
POST /api/lottery_game_matches	Создание матча лотерейной игры	Авторизованный администратор
PUT /api/lottery_game_matches (Обновляется только поле is_finished)	Завершение матча лотерейной игры	Авторизованный администратор
POST /api/lottery_game_match_users	Запись игрока на лотерейную игру	Авторизованный пользователь, владелец учетной записи
GET /api/lottery_game_matches? lottery_game_id=1	Получение списка всех матчей по id лотерейной игры	Гость, авторизованный пользователь, администратор
GET /api/users	Получение списка всех пользователей	Авторизованный администратор

7. Реализовать JWT авторизацию используя библиотеку <https://github.com/firebase/php-jwt>

JWT токен передается в Headers - Authorization: JWT-auth каждого запроса

8. Реализовать валидацию клиентских запросов:

Валидация на типы данных и обязательность полей.

Дата начала старта матча не может быть раньше сегодня.

9. С помощью Laravel Event/Listeners реализовать логику:

Один пользователь не может дважды записаться на один и тот же матч.

На матч может записаться не больше требуемого в игре количества участников.

При завершении матча победитель должен определяться случайным образом.

Победителю матча должно зачисляться указанное в игре число очков за победу.

11. С помощью отношений расширить следующие запросы:

GET /api/lottery_games - добавить к каждой игре вывод всех ее матчей,

отсортированных по дате и времени начала

GET /api/users - добавить к каждому пользователю вывод всех его выигранных матчей

12. Подготовить Postman коллекцию с примерами запросов реализованных в п.6 API.

Экспортировать коллекцию в json файл

Проект должен содержать:

Папку deploy с docker-compose file

Папку project с Lumen проектом

Папку collection с экспортированной Postman коллекцией

Проект разместить в github/gitlab репозитории