Туркин Игнат 131б-ПИо

номер зачётки 20221065 –> вариант 1

1) Описание структуры БД

Проект содержит 4 таблицы: “Users”, “Plants”, “exchange\_offers”. “exchange\_history”

Users

|  |  |  |
| --- | --- | --- |
| Поле | тип | описание |
| Id | Serial | Первичный ключ |
| Name | Varchar(100) | Имя пользователя |

Plants

|  |  |  |
| --- | --- | --- |
| Поле | тип | описание |
| Id | Serial | Первичный ключ |
| Name | Varchar(100) | Название растения |

exchange\_offers

|  |  |  |
| --- | --- | --- |
| Поле | тип | описание |
| Id | Serial | Первичный ключ |
| user\_id | Int | нешний ключ на users(id) — кто создал |
| offered\_plant\_id | Int | Внешний ключ на plants(id) — что предлагает |
| desired\_plant\_id | int | Внешний ключ на plants(id) — что хочет |

exchange\_history

|  |  |  |
| --- | --- | --- |
| Поле | тип | описание |
| Id | Serial | Первичный ключ |
| user\_id | Int | нешний ключ на users(id) — кто создал |
| offered\_plant\_id | Int | Внешний ключ на plants(id) — что предлагал |
| desired\_plant\_id | int | Внешний ключ на plants(id) — что хотел |
| accepted\_by |  | Внешний ключ на users(id) — Кто принял обмен |
| timestamp | TIMESTAMP | |  | | --- | |  |  |  | | --- | | Дата и время завершения | |

-- Таблица пользователей

CREATE TABLE users (

id SERIAL PRIMARY KEY,

name VARCHAR(100) NOT NULL

);

-- Таблица растений

CREATE TABLE plants (

id SERIAL PRIMARY KEY,

name VARCHAR(100) NOT NULL UNIQUE

);

-- Таблица активных предложений

CREATE TABLE exchange\_offers (

id SERIAL PRIMARY KEY,

user\_id INTEGER REFERENCES users(id) ON DELETE CASCADE,

offered\_plant\_id INTEGER REFERENCES plants(id) ON DELETE CASCADE,

desired\_plant\_id INTEGER REFERENCES plants(id) ON DELETE CASCADE

);

-- Таблица завершённых обменов

CREATE TABLE exchange\_history (

id SERIAL PRIMARY KEY,

exchange\_id INTEGER,

user\_id INTEGER REFERENCES users(id) ON DELETE SET NULL,

offered\_plant\_id INTEGER REFERENCES plants(id) ON DELETE SET NULL,

desired\_plant\_id INTEGER REFERENCES plants(id) ON DELETE SET NULL,

accepted\_by INTEGER REFERENCES users(id) ON DELETE SET NULL,

timestamp TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

**2) Документирование API. Список всех маршрутов API с методами, описанием, параметрами и кодами ответа.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Маршрут | Метод | Описание | Параметры в body | Заголовки | Коды ответа |
| /api/plants | GET | Получение всех растений | - | Content-Type: application/json | 200, 500 |
| /api/plants | POST | Добавить растение | { "name": "Фикус" } | Content-Type: application/json | 201, 409, 500 |
| /api/plants/:id | PUT | Обновить растение | { "name": "Алоэ" } | Content-Type: application/json | 200, 400, 404 |
| /api/plants/:id | DELETE | Удалить растение | - | - | 204, 404 |
| /api/users | GET | Получить всех пользователей | - | Content-Type: application/json | 200, 500 |
| /api/users | POST | Добавить пользователя | { "name": "Анна" } | Content-Type: application/json | 201, 400, 500 |
| /api/users/:id | PUT | Изменить имя пользователя | { "name": "Мария" } | Content-Type: application/json | 200, 400, 404 |
| /api/users/:id | DELETE | Удалить пользователя | - | - | 204, 404 |
| /api/exchange | GET | Получение всех обменов | - | - | 200, 500 |
| /api/exchange | POST | Создать обмен | { "user\_id", "offered\_plant\_id", "desired\_plant\_id" } | Content-Type: application/json | 201, 400, 500 |
| /api/exchange/:id | DELETE | Удалить обмен | - | - | 204, 404 |
| /api/exchange/:id/accept | PUT | Принять обмен | { "accepted\_by": user\_id } | Content-Type: application/json | 200, 400, 500 |
| /api/exchange/history | GET | История завершённых обменов | - | - | 200, 500 |
| /api/reports | GET | JSON-отчёт по активности | - | - | 200, 500 |
| /api/reports/download | GET | Скачать отчёт (.docx) | - | Accept: application/vnd.openxmlformats-officedocument.wordprocessingml.document | 200, 500 |

200 – Успех

201 – Создано

204 – Удалено

400 – Ошибка

404 – Не найдено

409 – уже существует

500 – Ошибка сервера