**Backend (Серверная часть):**

* **Python API:** Серверная часть приложения, написанная на языке программирования Python. Обрабатывает запросы от фронтенда и взаимодействует с базой данных PostgreSQL для получения и отправки данных.
* **API Контроллеры:** Часть Python API, ответственная за обработку запросов к базе данных для конкретных функциональных блоков (Catalog, News, Gallery, Map).
* **PostgreSQL:** Реляционная база данных, используемая для хранения данных о растениях, новостях, изображениях и картографических данных.

**Frontend (Клиентская часть):**

* **React Компоненты:** Часть клиентской части приложения, написанная с использованием библиотеки React.js. Каждый компонент (Catalog, News, Gallery, Map) отвечает за отображение данных для соответствующей функциональности.

**Функциональные блоки:**

* **Catalog (Каталог):** Функциональный блок, отвечающий за отображение данных о растениях.
* **News (Новости):** Функциональный блок, отвечающий за отображение новостей.
* **Gallery (Галерея):** Функциональный блок, отвечающий за отображение изображений, возможно, связанных с растениями.
* **Map (Карта):** Функциональный блок, отвечающий за отображение картографических данных, возможно, местоположения растений в ботаническом саду.

**Взаимодействие:**

* **Retrieve (Получение):** Процесс извлечения данных из базы данных. На диаграмме представлено взаимодействие между API Контроллерами и PostgreSQL для получения необходимой информации.

**Бэкенд (Backend):**

* Серверная сторона веб-приложения, ответственная за обработку запросов, взаимодействие с базами данных и выполнение бизнес-логики.

**API (Интерфейс программирования приложений):**

* Набор правил и инструментов, определяющих, как программы или компоненты могут взаимодействовать друг с другом.

**REST (Representational State Transfer):**

* Архитектурный стиль для разработки веб-сервисов, который использует стандартные HTTP-методы (GET, POST, PUT, DELETE) для выполнения операций.

**SQL (Structured Query Language):**

* Язык запросов, используемый для взаимодействия с реляционными базами данных. Примеры запросов включают SELECT, INSERT, UPDATE, DELETE.

**Фреймворк:**

* Структура и набор инструментов, предназначенных для упрощения разработки приложений. Например, Django, Flask, Ruby on Rails.

**Сервер:**

* Программное обеспечение, которое обрабатывает запросы от клиентов и предоставляет им необходимые ресурсы.

**База данных:**

* Система для хранения и организации данных. Реляционные базы данных (PostgreSQL, MySQL) и NoSQL базы данных (MongoDB) - примеры.

**Хостинг:**

* Предоставление инфраструктуры для размещения веб-приложений на серверах, обычно предоставляемое облачными провайдерами.

**Аутентификация:**

* Процесс проверки подлинности пользователя, чтобы удостовериться в его легитимности.

**Авторизация:**

* Управление доступом пользователей к ресурсам или функциональности приложения.

**HTTP (Hypertext Transfer Protocol):**

* Протокол передачи данных в сети, используемый для передачи информации между клиентами и серверами в вебе.

**Сессия:**

* Временный механизм хранения данных на сервере для отслеживания состояния пользователя между запросами.

**Журналирование (Logging) и Мониторинг:**

* Запись событий и данных для анализа, а также наблюдение за работой приложения для выявления проблем и улучшения производительности.

**Хэширование паролей:**

* Преобразование пароля в нечитаемую форму перед его сохранением, обеспечивая безопасность при хранении пользовательских учетных данных.