### 1. Основная концепция

У нас есть три ключевые роли:

1. **Поставщик**: Присылает информацию о "ценных товарах" на почту. Он вне нашей системы.
2. **Брокер**: Администратор системы. Он вручную переносит данные от Поставщика в админку, управляет товарами, пользователями и доступами.
3. **Пользователь (Клиент)**: Конечный потребитель, который заходит на сайт и видит только те товары и ту информацию, которую ему разрешил Брокер.

### 2. Структура данных (Проектируем базу данных)

Это фундамент всего приложения. Нам понадобится несколько таблиц.

#### Таблица users (Пользователи)

Здесь храним и Брокеров, и Клиентов.

| **Имя поля** | **Тип данных** | **Описание** |
| --- | --- | --- |
| id |  | Уникальный номер пользователя. |
| login |  | Логин для входа. Генерируется Брокером. |
| password\_hash |  | **Никогда не храните пароль в открытом виде!** Храните его хэш. |
| role |  | Роль в системе. Чтобы отличать Брокера от Клиента. |

#### Таблица products (Товары)

Основная информация о товарах.

| **Имя поля** | **Тип данных** | **Описание** |
| --- | --- | --- |
| id |  | Уникальный номер товара. |
| name |  | Имя товара. |
| description |  | Длинное описание товара. |
| price |  | Цена товара. |
| category\_id |  | Ссылка на категорию товара (из таблицы categories). |

#### Таблица product\_features (Особенности товара)

Те самые "окошки с любым контентом".

| **Имя поля** | **Тип данных** | **Описание** |
| --- | --- | --- |
| id |  | Уникальный номер особенности. |
| product\_id |  | Ссылка на товар, к которому относится особенность. |
| content |  | **Самое важное поле.** Здесь можно хранить любой текст, HTML, JSON. |

* **Почему так?** Один товар (product) может иметь много (many) особенностей (product\_features). Это классическая связь "один ко многим".

#### Таблица categories (Категории товаров)

Для группировки товаров и выбора дизайна.

| **Имя поля** | **Тип данных** | **Описание** |
| --- | --- | --- |
| id |  | Уникальный номер категории. |
| name |  | Название категории |
| layout\_type |  | Тип отображения. Например: 'layout\_A', 'layout\_B', 'layout\_C'. |

#### **САМАЯ ГЛАВНАЯ ТАБЛИЦА:** user\_product\_visibility (Видимость товаров для пользователей)

Эта таблица — сердце всей логики доступов. Она связывает пользователей, товары и поля, которые они могут видеть.

| **Имя поля** | **Тип данных** | **Описание** |
| --- | --- | --- |
| id |  | Уникальный номер записи о доступе. |
| user\_id |  | Ссылка на пользователя. |
| product\_id |  | Ссылка на товар. |
| show\_description | BOOLEAN | Галочка "видит описание". true или false. |
| show\_price | BOOLEAN | Галочка "видит цену". true или false. |
| show\_features | BOOLEAN | Галочка "видит особенности". true или false. |

* **Как это работает?** Если для пары user\_id=5 и product\_id=10 есть запись в этой таблице, значит пользователь видит товар. А флаги show\_... определяют, какие именно поля он видит. Имя товара (name) обычно показывают всегда, если доступ есть.

### 3. Административная панель (Что делает Брокер)

#### 3.1. Раздел "Товары" (/admin/products)

* **Список товаров:**
  + Простая таблица: ID, Имя, Категория, Цена.
  + Кнопка "Добавить новый товар".
  + У каждой строки кнопки "Редактировать" и "Удалить".
* **Форма добавления/редактирования товара (/admin/products/new):**
  + **Имя:** Текстовое поле.
  + **Описание:** Большое текстовое поле (желательно с редактором WYSIWYG, чтобы можно было вставлять форматированный текст).
  + **Цена:** Поле для ввода числа.
  + **Категория:** Выпадающий список со всеми категориями из таблицы categories.
  + **Особенности:**
    - Это самый интересный элемент. Это должен быть "динамический список".
    - Изначально пусто. Есть кнопка [ + Добавить особенность ].
    - При нажатии на нее появляется новое текстовое поле (тоже желательно с WYSIWYG-редактором) и кнопка [ X ] для его удаления.
    - Можно добавлять сколько угодно таких полей. Каждое поле — это одна запись в таблице product\_features.

#### 3.2. Раздел "Пользователи" (/admin/users)

* **Список пользователей:**
  + Таблица: ID, Логин, Роль.
  + Кнопка "Добавить пользователя".
  + У каждой строки кнопки "Редактировать" и "Удалить".
* **Форма добавления/редактирования пользователя (/admin/users/new):**
  + **Логин:** Текстовое поле. Можно с кнопкой "Сгенерировать".
  + **Пароль:** Поле для ввода пароля. Можно с кнопкой "Сгенерировать".
  + **Смена пароля (для редактирования):**
    - Поля "Новый пароль" и "Подтверждение пароля" скрыты.
    - Есть кнопка [ Сменить пароль ].
    - При нажатии появляется окно, где Брокер должен ввести свой **секретный код**.
    - Если код верный, поля для смены пароля становятся активными.
    - *Техническая реализация:* Секретный код — это просто строка, которая хранится в конфигурации приложения на сервере и известна только Брокеру.

#### 3.3. Управление доступами (Два пути к одной цели)

Это самая важная часть. Логика здесь одна и та же, просто интерфейс представлен в двух разных местах. Оба интерфейса управляют записями в таблице user\_product\_visibility.

* **A. Со страницы Пользователя (/admin/users/123):**
  1. Открываем страницу конкретного пользователя. Видим его данные (логин и т.д.).
  2. Ниже блок: **"Доступы к товарам"**.
  3. **Строка поиска:** [ Поиск по товарам... ] — позволяет быстро найти нужный товар в длинном списке.
  4. Далее идет **список всех товаров** из таблицы products.
  5. Каждая строка в списке выглядит так:  
     [✓] Товар "Ferrari F40"  
      [✓] Видит описание  
      [ ] Видит цену  
      [✓] Видит особенности
  6. Главная галочка [✓] Товар "..." создает или удаляет запись в user\_product\_visibility. Вложенные галочки обновляют поля show\_description, show\_price и т.д. в этой записи.
* **B. Со страницы Товара (/admin/products/456):**
  1. Открываем страницу редактирования товара.
  2. Под основной информацией о товаре (цена, описание) есть блок **"Доступы пользователей"**.
  3. Далее идет **список всех пользователей** из таблицы users (с ролью client).
  4. Каждая строка в списке выглядит абсолютно так же:  
     [✓] Пользователь "user\_best"  
      [✓] Видит описание  
      [ ] Видит цену  
      [✓] Видит особенности
  5. Логика работы галочек точно такая же. Изменение здесь должно мгновенно отражаться на странице пользователя, и наоборот, потому что они меняют одну и ту же таблицу.

#### 3.4. Раздел "Категории" (/admin/categories)

* Простой раздел для управления категориями.
* **Список категорий:** ID, Название, Тип шаблона.
* **Форма добавления/редактирования:**
  + **Название:** Текстовое поле.
  + **Тип шаблона:** Выпадающий список с заранее заготовленными вариантами (layout\_A, layout\_B, layout\_C, layout\_D, layout\_E).

### 4. Клиентская часть (Что видит Пользователь)

1. Пользователь логинится со своим логином и паролем.
2. Система проверяет его user\_id.
3. Система делает запрос в таблицу user\_product\_visibility и получает **все** product\_id, к которым у этого user\_id есть доступ.
4. Для каждого product\_id система также получает информацию о том, какие поля можно показывать (show\_price и т.д.).
5. Пользователю отображается каталог только из этих разрешенных товаров.
6. Когда пользователь кликает на товар, открывается страница этого товара.
7. Система смотрит на category\_id этого товара, определяет layout\_type и рендерит (отрисовывает) страницу, используя нужный шаблон (лейаут).
8. На странице товара отображаются только разрешенные поля (например, есть описание и особенности, но нет цены).

Важно что бы код был удобно читаемым