Информационный веб-сайт

**Описание**

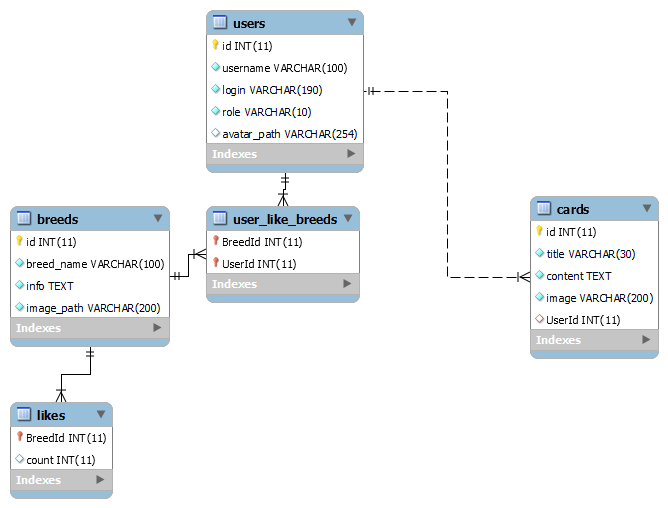
Web-сайт, в котором будет собрана и в удобном формате представлена информация о различных породах собак. Сайт будет состоять из 2 главных страниц:   
- Главная страница с различными породами. Породы будут представлены в виде карточек с описанием породы.  
- Вторая страница с пользовательскими карточками. Пользовательские карточки будут включать в себя информацию о конкретном питомце конкретного пользователя.  
  
**Функциональность**  
- возможность авторизоваться на сайте (для создания/просмотра пользовательских карточек)  
- возможность поиска по названию породы  
- возможность оценить понравившуюся породу  
- возможность отсортировать породы по количеству лайков  
- каждый пользователь имеет свой профиль  
- возможность зайти на профиль пользователя и посмотреть его карточки **Пользовательские роли  
Конечные пользователи** – те, кто будут непосредственно пользоваться сайтом, авторизовываться, добавлять свои карточки (сюда относятся как авторизованные пользователи, так и не авторизованные)  
**Неавторизованные пользователи** (VISITOR) – пользователи, которые имеют доступ к сайту, но могут просматривать ТОЛЬКО основную публичную страницу (public). Доступа к сообществу/к оцениванию/просмотру чужих профилей у неавторизованных пользователей нет.   
**Авторизованные пользователи** (USER) – пользователи, которые успешно авторизовались на сайте и имеют возможность просматривать/создавать свои карточки.  
**Модератор пользовательского контента (MODERATOR) –** тот, кто будет следить за пользовательским контентом. В его обязанности будет входить проверка пользовательских карточек на соблюдение определенного формата и определенных требований. Модератор по своему усмотрению может удалять пользовательские карточки (те, которые противоречат тематике сайта или являются источниками неположительной информации).  
**Контент-менеджер (MANAGER)** – тот, кто будет наполнять систему актуальной информацией. (То есть будет вносить определенные данные в БД).

**Архитектор системы/администратор данных/менеджер проекта** – проектировщик системы. Выдвигает основные требования к системе и следит за их соблюдением.   
**Разработчики системы** – прикладные программисты, использующие технологии для разработки описываемого продукта. (Сюда входят программисты всех уровней, то есть администратор баз данных тоже входит в эту группу)  
**Тестировщик** – специалист, который будет проверять корректность работы системы и соблюдение выдвигаемых требований посредством тестирования.

**Данные**  
Для хранения информации будет задействована база данных. Она будет состоять из 5 таблиц:  
- **Breeds**:   
**id** – INT (NOT NULL, AUTO INCREMENT, PRIMARY KEY)  
**breed\_name** – VARCHAR(100) (NOT NULL, UNIQUE)  
**info** – TEXT (NOT NULL)  
**image\_path** – VARCHAR(200) (NOT NULL)  
Таблица содержит информацию, которая показывается на карточках на главной странице (породы)

- **Likes**:   
**BreedId** – INT, (PRIMARY KEY + FOREIGN KEY, NOT NULL)  
**count** – INT (DEFAULT: 0)  
Таблица содержит информацию о количестве лайков у каждой породы (подробне см. транзакции)

- **Users**:   
**id** – INT (PRIMARY KEY, NOT NULL, AUTO INCREMENT)  
**username** – VARCHAR(100) (NOT NULL)  
**login** – VARCHAR(190) (NOT NULL, UNIQUE)  
**role** – VARCHAR(10) (NOT NULL, DEFAULT: ‘USER’)  
**avatar\_path** – VARCHAR(254) (ALLOW NULL)  
Таблица содержит информацию о пользователях, которые авторизовывались на сайте  
  
- **User\_like\_breeds**:   
**BreedId** – INT (FOREIGN KEY)  
**UserId** – INT (FOREIGN KEY)  
Таблица содержит информацию о том, каким породам поставил лайк каждый юзер (см. подробнее связи между таблицами ниже)  
  
- **Cards**:   
**id** – INT (NOT NULL, PRIMARY KEY, AUTO INCREMENT)  
**title** – VARCHAR(30) (NOT NULL)  
**content** – TEXT (NOT NULL)  
**image** – VARCHAR(200) (NOT NULL)  
**UserId** – INT (NOT NULL, FOREIGN KEY)  
Таблица содержит информацию, которая показывается на странице сообщества (пользовательские карточки)

EER диаграмма описанной базы данных:  


Как можно заметить, между таблицами действуют различные связи:  
- Между таблицами **Breeds** и **Likes** построена связь «Один к одному» (см. транзакции), причем связь identifying (запись в Likes не может существовать, если в графе BreedId указан id несуществующей породы)  
- Между таблицами **Users** и **Cards** построена связь«Один ко многим», причем non-identifying (одному пользователю могут соответствовать несколько карточек, при этом каждой карточке соответствует один пользователь)  
- Между таблица **Users** и **Breeds** построена связь «Многие ко многим», поскольку одному и тому же пользователю могут понравиться разные породы и в то же время одна порода может понравиться разным пользователям (связь non-identifying, т.к. потенциально порода может никому не понравиться)  
  
**Транзакции**В ходе разработки было принято решение использовать транзакции там, где это было уместно. Основная причина такого подхода заключается в том, что транзакции дают возможность выполнить последовательный набор инструкций как атомарное действие. При этом, если возникнет ошибка в ходе выполнения одной из внутренних инструкции транзакции, то в этом случае транзакция откатит систему до состояния, в котором она была до начала выполнения транзакции. Этот подход был удобен при проектировании исходной базы данных:  
1) При создании новой породы выполняются две инструкции:  
CREATE для таблицы **Breeds** (создание самой породы непосредственно)  
CREATE для таблицы **Likes** (по умолчанию каждая новая порода имеет 0 лайков)  
Таким образом, данная транзакция позволяет автоматически создавать запись в подчиненной таблице.  
  
2) Когда пользователь хочет поставить/убрать лайк:  
SELECT в таблице **User\_likes\_breeds** (если пользователь уже ставил лайк данной породе, то запись в таблице будет найдена)  
В зависимости от результата SELECT:  
 - если запись не была найдена, то CREATE в таблице **User\_like\_breeds** и UPDATE поля count в таблице **Likes** (увеличения счетчика лайков для той породы, лайк которой был поставлен)  
 - если запись была найдена, то DELETE в таблице **User\_like\_breeds** и UPDATE в таблице **Likes** (уменьшение счетчика лайков для той породы, лайк который был убран)

**Инструкция по использованию UI веб-сайта.**

В зависимости от роли, пользователь обладает следующим UI функционалом:

* **VISITER**.

Незарегистрированный пользователь имеет возможность зарегистрироваться/войти на сайте перейдя на страницу регистрации. Для этого ему требуется нажать на кнопку “WOOF” на главной навигационной панеле. На странице регистрации нажать на кнопку “присоединиться” и войти в аккаунт. Незарегистрированный пользователь имеет доступ к главной странице с карточками пород. На главной странице пользователь имеет возможность:

1. Перейти на другие страницы с карточками, нажав на номер страницы/стрелочку на пагинации.
2. Осуществить поиск по породам, набрав текст в поле поиска на верхней панели и, осуществить поиск, нажав на соответствующую кнопку.
3. Осуществить вывод карточек пород по популярности/количестве лайков. Для этого необходимо нажать на кнопку “отсортировать по лайкам” на верхней панели.
4. Перейти на страницу отдельной породы, кликнув на карточку.

* **USER**.

Зарегистрированный пользователь обладаем всем функционалом VISITER. Помимо этого, он может:

1) На странице конкретной породы поставить ей “лайк”, нажав на сердечко. Эта отметка сохраняется для пользователя и сайт понимает, что “лайк” был поставлен. По необходимости, “лайк” также можно убрать на той же странице породы.

2) На навигационной панели появляется кнопка “мой профиль”. Она позволяет перейти пользователю на страницу своего профиля. На этой страницу пользователь видит краткую информацию о себе и кнопку “показать карточки пользователя”. Кнопка позволяет пользователю увидеть все карточки, которые он создал на странице сообщества (об этом см. следующий пункт).

3) Пользователь имеет доступ к странице “сообщество”. Переход на страницу осуществляется при нажатии на соответствующую кнопку на навигационной панели. На странице в верхней панели имеется переключающийся фильтр, который отображает карточки всех пользователей или карточки текущего пользователя. Создаются пользовательские карточки при нажатии на кнопку “+добавить карточку” в той же верхней панели. Заполнив необходимую форму и нажав на кнопку “создать”, карточка сохраняется в базе данных. Если пользователь нажмет на имя “хозяина” на карточке, то он перейдет на страницу владельца карточки.

* **MODERATOR**.

Обладает всем функционалом, доступного для USER. Кроме этого, модератор может удалять пользовательские карточки на странице “сообщество”. Для этого ему требуется нажать на “крестик” (который ему становится виден) на карточке, и подтвердить свое действие в появившейся форме, нажав на “удалить”.

* **MANAGER**.

Обладает всем функционалом, доступного для USER. Кроме этого, менеджер может добавлять карточки на “главной” странице. Для этого требуется нажать на кнопку “+ добавить карточку” и заполнить форму для создания. Также, менеджер способен удалять карточку породы на ее странице. Для этого ему нужно нажать на кнопку “удалить” в левом нижнем углу.

* **ADMIN**.

Обладает всем вышеперечисленным функционалом.

**Технологии разработки**В качестве СУБД будет использоваться MySQL   
Для проектирования баз данных будет использоваться MySQL Workbench  
Для создания оболочки сайта (frontend часть) будет использоваться React  
Для создания серверной части сайта (backend часть) будет использоваться NodeJS с фреймворком Express  
Язык программирования - JavaScript  
Для поддержки совместной разработки будет использоваться GitHub.  
  
**Команда**Адамович Анатолий, Кудряшов Егор (группа 331)