

**“САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,
МЕХАНИКИ И ОПТИКИ”**

Факультет Инфокоммуникационных технологий

Образовательная программа: Интеллектуальные системы в гуманитарной сфере (Академический бакалавр, Очная ф/о)

Направление подготовки (специальность): 45.03.04 Интеллектуальные системы в гуманитарной сфере

О Т Ч Е Т

по курсовой работе

Тема задания: **«Танцевальные мастер-классы Санкт-Петербурга: система записи и администрирования»**

Обучающийся: Жабровец Е. А., группа К3343

Руководитель: Говоров А.И., ассистент кафедры ИТГС Университета ИТМО

Оценка ____

Подписи членов комиссии:

(подпись) Говоров А.И.

(подпись) Чунаев А.В.

(подпись) Антонов М.Б.

Дата ____

Санкт-Петербург
2020

Оглавление

ВВЕДЕНИЕ	3
1 НАЧАЛЬНЫЙ ЭТАП РАБОТЫ	4
1.1 Описание предметной области	4
1.2 Определение технических требований	4
2 РАЗРАБОТКА СЕРВЕРНОЙ ЧАСТИ	5
2.1. Характеристика средств разработки серверной части	5
2.2. Проектирование модели данных	6
2.3. Обзор полученных URL-шаблонов	7
3 РАЗРАБОТКА КЛИЕНТСКОЙ ЧАСТИ	10
3.1. Характеристика средств разработки клиентской части	10
3.2. Обзор полученных интерфейсов	10
А. Стартовая страница.....	10
В. Интерфейсы для обычного пользователя	11
С. Интерфейсы для администратора.....	14
ЗАКЛЮЧЕНИЕ.....	17
СПИСОК ИСТОЧНИКОВ	18

ВВЕДЕНИЕ

В настоящее время у людей наблюдается повышенный интерес к танцам – как к виду досуга, так и как к поприщу для профессионального роста и развития. Стремительно растет количество танцевальных школ, танцоры объединяются в сообщества внутри своих танцевальных стилей, организуют всевозможные чемпионаты и лагеря, ездят в туры по городам и странам, чтобы поделиться своим опытом с другими.

Несмотря на то, что в нашей стране, и в Санкт-Петербурге в частности, сфера танцев является достаточно развитой с точки зрения количества вовлеченных в нее профессионалов, существует проблема, связанная с информированием людей о проводимых танцевальных мероприятиях. Она возникла по причине того, что основным каналом для донесения информации являются социальные сети: о грядущих мастер-классах танцоры рассказывают на личных страницах, и таким образом единственно возможный инструмент для привлечения на занятия новых людей и расширения аудитории, - это рекомендация со стороны знакомых или, в лучшем случае, таргетированная реклама, которая в силу перегруженности платформ контентом часто остается незамеченной. Более того, даже люди, давно увлеченные танцами, испытывают проблемы с получением данных о потенциально интересных для них мероприятиях – существуют соответствующие рассылки, но об удобстве вычленения полезной информации из длинных сообщений говорить не приходится.

Иными словами, какого-либо единого и исчерпывающего ресурса по данной тематике не существует, чем и объясняется **актуальность** разработки подобного рода информационного портала.

Целью данной курсовой работы являлось создание системы, предоставляющей информацию об актуальных мастер-классах с возможностью записи. Для достижения поставленной цели были сформулированы следующие **задачи**:

1. определить функциональные требования для сервиса;
2. проектирование модели базы данных, соответствующей тематике заданной предметной области
3. реализация серверной части веб-сервиса;
4. реализация клиентской части веб-сервиса.

Каждая из представленных задач будет подробно описана далее в тексте отчета.

1 НАЧАЛЬНЫЙ ЭТАП РАБОТЫ

1.1 Описание предметной области

Танцевальные мастер-классы могут проводиться на различных площадках, представляющих собой танцевальные школы города. Допустимое количество посетителей определяется вместимостью конкретного зала, закрепленным за той или иной школой. Каждый зал имеет свою установленную цену аренды, обычно выраженной ценой пользования за один час.

Афиша мастер-класса обязательно содержит сведения о стоимости, дате и времени проведения. Помимо этого, каждый класс характеризуется каким-то одним танцевальным стилем. Количество хореографов, которые могут преподавать на одном мероприятии, не ограничено (как правило, хореографы, придумывающие танцевальные связки, будучи физически не в состоянии их исполнить в силу индивидуальных причин берут себе в помощь ассистентов).

Желающие посетить мастер-класс отправляют соответствующую заявку, утверждаемую после внесения 100% оплаты. Впоследствии участникам предоставляется возможность оставить отзыв о конкретном педагоге с привязкой к посещенному уроку.

1.2 Определение технических требований

На основании представленного описания предметной области с учетом конечной цели для разрабатываемого веб-ресурса были определены следующие технические требования:

- 1) предоставление пользователям возможности регистрации и авторизации в системе;
- 2) разделение ролей пользователей;
- 3) отображение информации о грядущих мастер-классах с возможностью записи;
- 4) отображение информации о посещенных мастер-классах с возможностью добавления отзыва к урокам;
- 5) наличие функционала для добавления мастер-класса и утверждения об участии со стороны администратора;
- 6) построение отчетов, содержащих информацию о рейтинге пользователей и выручке по школам;
- 7) предоставление сведений о педагогах, включающих в себя персональную информацию и отзывы.

2 РАЗРАБОТКА СЕРВЕРНОЙ ЧАСТИ

2.1. Характеристика средств разработки серверной части

Так как разрабатываемый ресурс предполагает взаимодействие двух «сторон» - сервера и клиента, необходимо определить набор функций, на основании которых осуществлялось бы данное сообщение. Для этих целей используется **API (Application Programming Interface)**, содержащее в себе описание способов (классов, процедур и проч.), которыми одна программа может взаимодействовать с другой.

Когда речь идет о предложении API пользователям, подразумевается создание ряда специальных URL, предназначением которых является передача каких-либо данных в качестве ответа. Подобные запросы отправляются через браузер – так как передача данных осуществляется по протоколу HTTP в текстовом виде, он всегда может отобразить ответ в формате JSON, из которого достаточно просто извлечь информацию и использовать её в коде по своему усмотрению.

В качестве инструмента для создания API в рамках работы использовался **Django REST Framework (3.11.0)**. Данная программная библиотека, основанная на идеологии python-фреймворка Django предоставляет функционал для создания Web API с опорой на архитектуру **REST (Representational State Transfer – «передача состояния-представления»)**. Можно отметить несколько основных принципов создания приложений в стиле REST:

- в процессе работы сервер не должен ничего знать о текущем состоянии клиента;
- запрос от клиента должен содержать всю необходимую информацию для обработки этого запроса сервером;
- каждый ресурс на сервере должен иметь определенный id, а также уникальный URL, по которому осуществляется доступ к этому ресурсу;
- действия для управления данными привязаны к определенным HTTP-методам.

Существует несколько стандартных действий для работы с данными:

- Create – «создание объекта» (используется http-метод POST).
- Read – «чтение» (http-метод GET).
- Update – «изменений» (http-метод PUT).
- Delete – «удаление» (http-метод DELETE).

```

HTTP 200 OK
Allow: GET, HEAD, OPTIONS
Content-Type: application/vnd.api+json
Vary: Accept

{
  "data": {
    "data": [
      {
        "id": 14,
        "name": "Shuffle Party",
        "place": {
          "id": 1,
          "name": "Синий",
          "cost": 3000,
          "capacity": 20,
          "school": {
            "id": 1,
            "name": "International Dance Center",
            "address": "Каменноостровский пр., 42"
          }
        }
      }
    ]
  }
},

```

Рисунок 1 – Пример предоставления информации от сервера клиенту в стиле REST

В качестве системы управления базами данных использовалась **PostgreSQL 10**. Будучи объектно-реляционной СУБД, она поддерживает пользовательские объекты и их поведение, включая типы данных, функции, операции, домены и индексы. Для взаимодействия с базой данных применялась самая популярная библиотека для работы с PostgreSQL на Python - **psycopg2**, написанная на языке C и обладающая такими характеристиками, как потоковая безопасность (исправное функционирование при использовании в нескольких потоках одновременно) и полное соответствие спецификации Python Database API Specification v2.0.

Языком для реализации серверной части являлся **Python (3.8)**, использовавшийся в связке с веб-инфраструктурой **Django (3.0.5)**.

2.2. Проектирование модели данных

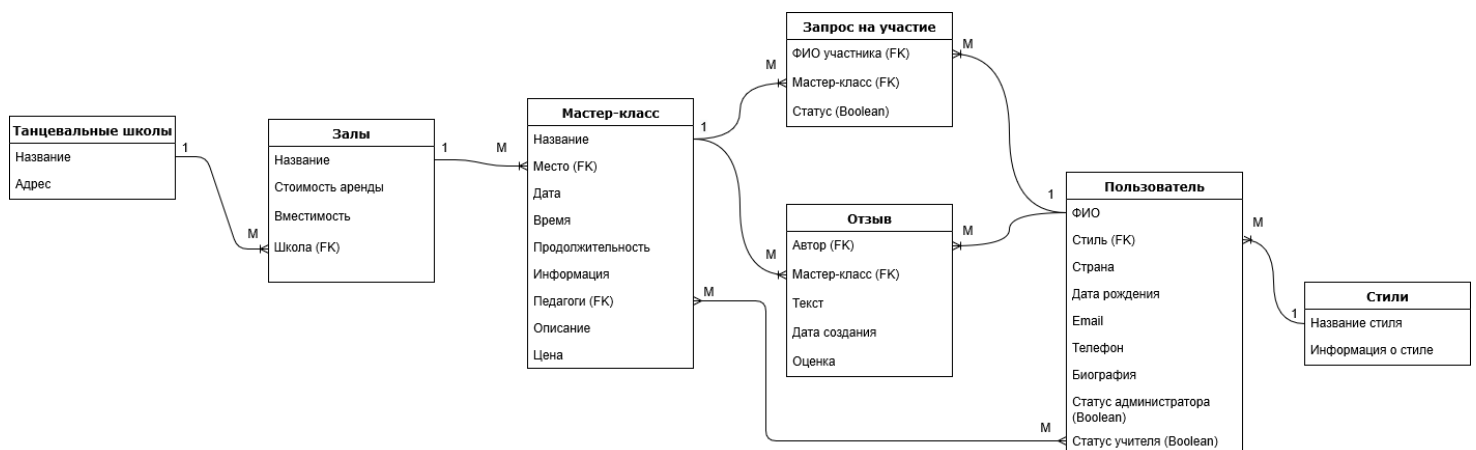


Рисунок 2 – Модель данных для предметной области «Танцевальные мастер-классы Санкт-Петербурга»

Структура базы данных веб-ресурса включает в себя 7 таблиц: «*Танцевальные школы*», «*Залы*», «*Мастер-класс*», «*Запрос на участие*», «*Отзыв*», «*Пользователь*» и «*Стили*»:

- Таблица «Танцевальные школы» содержит информацию о названии и адресе локации для мастер-класса;
- Таблица «Залы» содержит данные о вместимости, названии, цене аренды зала и школе, за которой этот зал закреплен. В одной танцевальной школе может быть несколько танцевальных залов;
- Таблица «Стили» содержит информацию о названии танцевальных стилей и их краткое описание;
- Таблица «Пользователь» хранит данные пользователей системы, включающие такие атрибуты, как ФИО, страна, адрес электронной почты, телефон, биография, а также два булевых поля – «Статус суперпользователя» и «Статус учителя». Для пользователей, обладающих статусом учителя, в обязательном порядке также указывается танцевальный стиль, который он преподает;
- Таблица «Запрос на участие» содержит информацию об имени заявителя, мастер-классе, на который была подана заявка и статусе заявки («Одобрена/Не одобрена»). Один человек может отправлять заявки на разные мастер-классы;
- Таблица «Отзыв» включает в себя такие атрибуты, как «Автор отзыва», «Текст отзыва», «Дата создания», «Оценка». Предполагается возможность оставлять несколько отзывов о посещенных мастер-классах;
- Таблица «Мастер-класс» имеет атрибуты «Название», «Дата», «Время», «Продолжительность», «Описание», «Цена посещения», «Место проведения» и «Хореографы». На одном мастер-классе может преподавать несколько хореографов при условии наличия одинакового танцевального стиля.

2.3. Обзор полученных URL-шаблонов

По завершении реализации серверной части были получены следующие URL-шаблоны:

- 1) '*wsh_all/*' – GET-запрос, возвращающий сериализованные данные о всех мастер-классах с не наступившей датой, включая данные из связанных таблиц
- 2) '*wsh_filter/*' – GET-запрос, аналогичным образом возвращающий актуальные мастер-классы с учетом указанного диапазона дат. Подразумевает указание параметров '*date_start*' («начальная дата») и '*date_end*' («конечная дата»)

- 3) **'user_info/'** – GET-запрос, возвращающий информацию о текущем авторизованном пользователе (на основании токена авторизации, содержащегося в HTTP-заголовке Authorization)
- 4) **'add_wsh/'** – POST-запрос, создающий новую запись в таблице «Мастер-класс». Принимает из тела запроса значения ключей на основании обозначенных в соответствующем сериализаторе полей. Доступен только для администратора.
- 5) **'halls_for_school/'** – GET-запрос, возвращающий список залов, закрепленных за конкретной танцевальной школой. Требуется указания значения параметра 'school' (id школы). Используется в форме для добавления мастер-класса. Доступен только для администратора.
- 6) **'all_schools/'** – GET-запрос, возвращающий список всех танцевальных школ. Используется в форме для добавления мастер-класса. Доступен только для администратора.
- 7) **'all_styles/'** – GET-запрос, возвращающий список всех танцевальных стилей. Используется в форме для добавления мастер-класса. Доступен только для администратора.
- 8) **'teachers_for_style/'** – GET-запрос, возвращающий список учителей, практикующих данный танцевальный стиль. Требуется указания значения параметра 'style' (id танцевального стиля). Используется в форме для добавления мастер-класса. Доступен только для администратора.
- 9) **'add_teachers_for_wsh/'** – POST-запрос, добавляющий педагога к последнему созданному мастер-классу. Принимает из тела запроса значение ключа 'teacher' (id педагога). Доступен только для администратора.
- 10) **'query_for_part/'** – POST-запрос, создающий новую запись в таблице «Запрос на участие». Принимает из тела запроса значение ключа «wsh» (id мастер-класса) id пользователя на основании токена в HTTP-заголовке.
- 11) **'query_part_info/'** – GET-запрос, возвращающий информацию о текущих заявках пользователя в одном словаре и список посещенных мастер-классов – в другом
- 12) **'profile_upd/<int:pk>/'** – POST-запрос, обновляющий данные авторизованного пользователя (представление наследуется от класса ModelViewSet GenericAPIView). В качестве значения переменной pk принимается id текущего пользователя.
- 13) **'write_feedback/'** – POST-запрос, добавляющий новую запись в таблицу «Отзыв». Принимает из тела запроса значение ключа 'wsh' (id мастер-класса) и значение id пользователя на основании токена авторизации.

- 14) ***'teachers/'***- GET-запрос, возвращающий список всех пользователей, у которых значение поля «Является учителем» равняется True.
- 15) ***'one_teacher/'*** – GET-запрос, возвращающий информацию об одном учителе. Требуется указания значения параметра 'teacher' (id педагога).
- 16) ***'feedbacks/'*** – GET-запрос, возвращающий данные об отзывах о мастер-классах данного хореографа. Требуется указания значения параметра 'teacher' (id педагога).
- 17) ***'participants/'*** – GET-запрос, возвращающий список участников для данного мастер-класса. Требуется указания значения параметра 'wsh' (id мастер-класса).
- 18) ***'queries_to_approve/'*** – GET-запрос, возвращающий информацию о неподтвержденных заявках при помощи сериализатора, унаследованного от класса BaseSerializer. Доступен только для администратора.
- 19) ***'approve_query/'*** – POST-запрос, меняющий статус заявки на «Одобренная» и добавляющий участников к мастер-классу. Доступен только для администратора.
- 20) ***'profile_new/'*** – POST-запрос, добавляющий имя к недавно зарегистрировавшемуся пользователю. Принимает из тела запроса значение ключа 'full_name' (ФИО)
- 21) ***'report/'*** – GET-запрос, возвращающий информацию о выручке и прогнозируемой прибыли танцевальных школ. Доступен только для администратора.
- 22) ***'rating/'***- GET-запрос, возвращающий список пользователей, ранжированный по количеству посещенных мастер-классов при помощи сериализатора, унаследованного от класса BaseSerializer.

Для реализации процесса регистрации и входа на сайт использовались URL-шаблоны, предлагаемые библиотекой *django* (***'/token/login/'*** и ***'/users/'*** соответственно).

3 РАЗРАБОТКА КЛИЕНТСКОЙ ЧАСТИ

3.1. Характеристика средств разработки клиентской части

Для разработки клиентской части веб-сервиса основным средством является фреймворк *Vue.js (2.6.11)*. В отличие от фреймворков-монолитов, таких как Angular.js, Vue создан пригодным для постепенного внедрения. Его ядро в первую очередь решает задачи уровня представления (view), что упрощает интеграцию с другими библиотеками и существующими проектами. С другой стороны, он полностью подходит и для создания сложных одностраничных приложений (SPA, Single-Page Applications), если использовать его совместно с современными инструментами и дополнительными библиотеками.

В качестве главного вспомогательного инструмента для оформления сайта использовалась библиотека *Muse-UI*. Она предоставляет набор компонентов для Vue.js, включая готовые решения для упрощения разметки, дизайна форм, окон и проч. Дополнительно к Muse-UI была подключена библиотека оконочных шрифтов *Material design icons* в целях создания возможности использования значков-изображений.

Также была произведена отдельная установка темы «*Carbon*» для Muse-UI и компонента *Video Background* для Vue.js для создания законченного облика стартовой страницы ресурса.

3.2. Обзор полученных интерфейсов

А. Стартовая страница

На стартовой странице пользователю предлагается выполнить вход в систему или пройти процедуру регистрации:

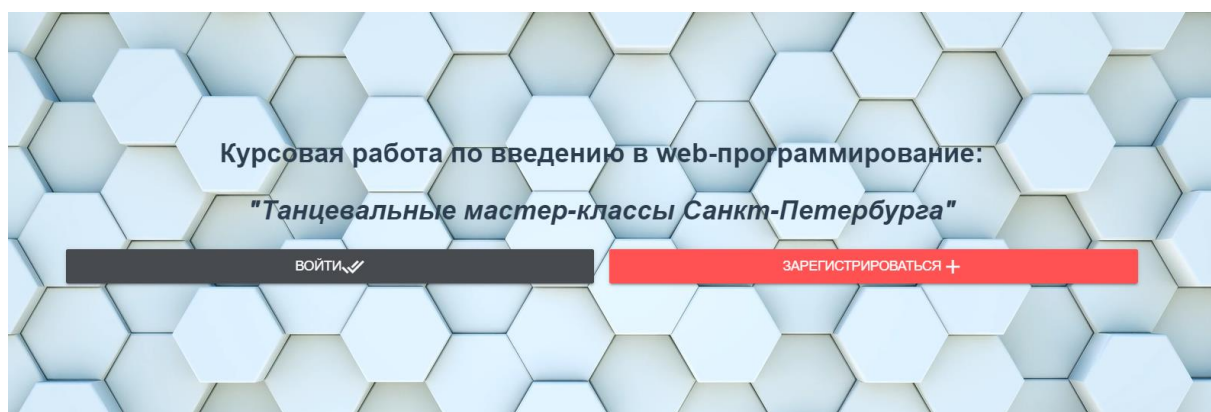


Рисунок 3 – Стартовая страница веб-сервиса

При нажатии на кнопку «Зарегистрироваться» открывается окно регистрации, включающей в себя три этапа: ввод логина, ввод и подтверждение пароля, ввод фамилии и имени пользователя. Доступ к каждому последующему этапу (появление кнопки «Далее»)

осуществляется после выполнения условий предыдущего (непустое поле в случае логина и имени и одинаковый пароль в полях второго этапа). Возможно осуществление возвращения на предыдущий этап при помощи кнопки «Назад».

Рисунок 4 – Процесс регистрации

При нажатии на кнопку «Вход» появляется форма ввода логина и пароля:

Рисунок 5 – Форма входа на сайт

В. Интерфейсы для обычного пользователя

После завершения процесса регистрации/входа пользователь перенаправляется на страницу с афишей актуальных мастер-классов:

Рисунок 6 – Содержимое вкладки «Афиша»

Мастер-классы можно фильтровать по дате при помощи формы справа (рис. 6).

Чтобы записаться на какой-либо мастер-класс, пользователю необходимо нажать на кнопку «Записаться» в карточке интересующего мастер-класса.

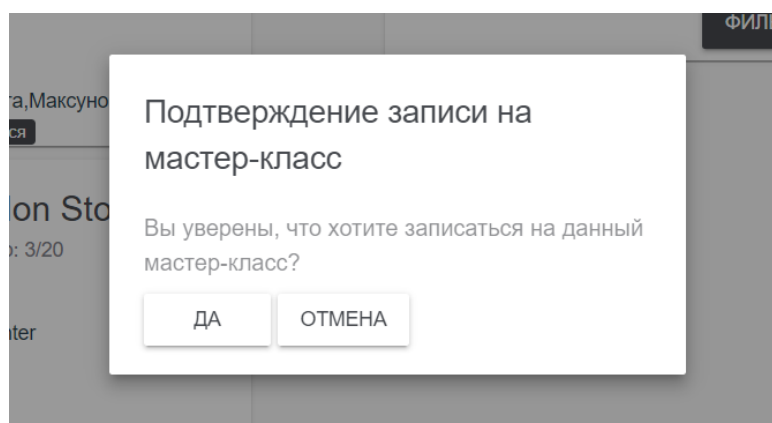


Рисунок 7 – Окно подтверждения записи на мастер-класс

После подтверждения записи при переходе на вкладку «Мои мастер-классы», пользователь может увидеть свою заявку с информацией о мастер-классе и статусом «Заявка находится на стадии рассмотрения» (рис. 8):

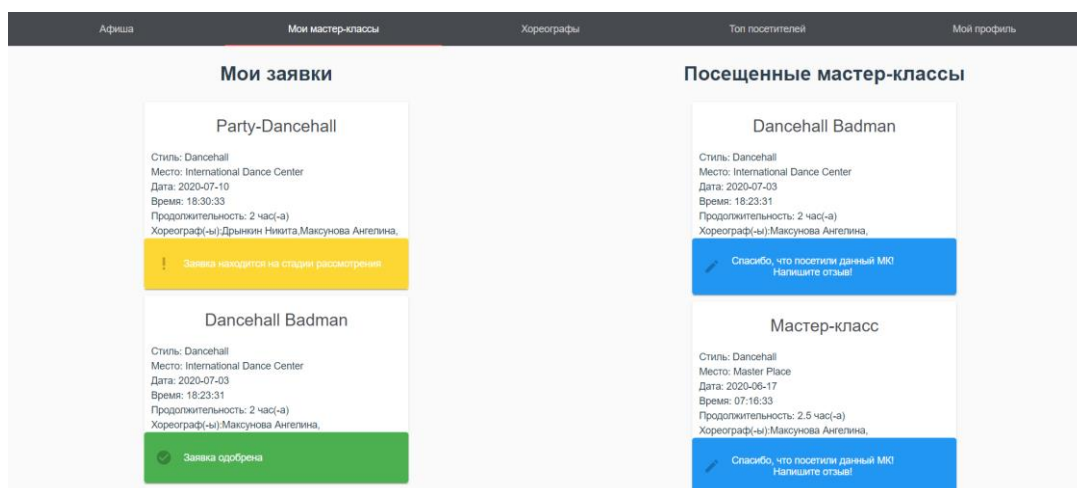


Рисунок 8 – Содержимое вкладки «Мои мастер-классы»

После того, как пользователь-администратор подтверждает внесение участником платы за мастер-класс, статус заявки меняется на «Заявка одобрена».

Для посещенных мастер-классов предусмотрена процедура написания отзыва. Форма отзыва включает в себя поле выбора оценки мастер-класса и поле ввода текста. Отправленный отзыв становится доступным к прочтению в окне персональной информации о педагоге, проводившем мастер-класс. Чтобы его открыть, необходимо перейти во вкладку «Хореографы», в которой представлен список всех пользователей со статусом «Является учителем» и нажать на значок «Узнать больше» справа напротив имени хореографа.

Мой отзыв

Оценка
Хорошо

Текст отзыва
Довольно неплохо
Но можно лучше

ОТПРАВИТЬ ОТЗЫВ ОТМЕНА

Рисунок 9 – Окно добавления отзыва

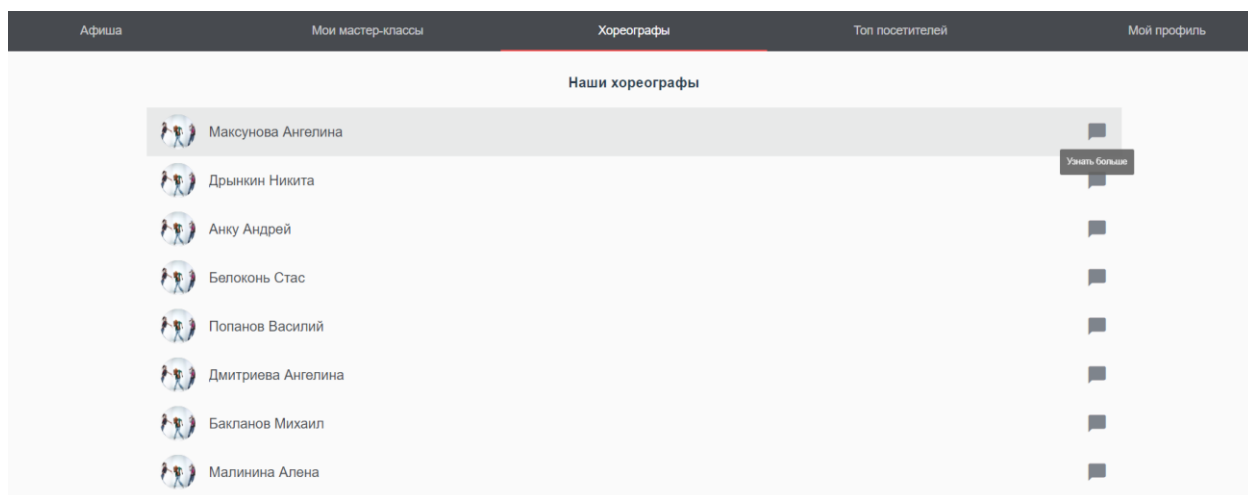


Рисунок 10 – Вкладка «Хореографы»

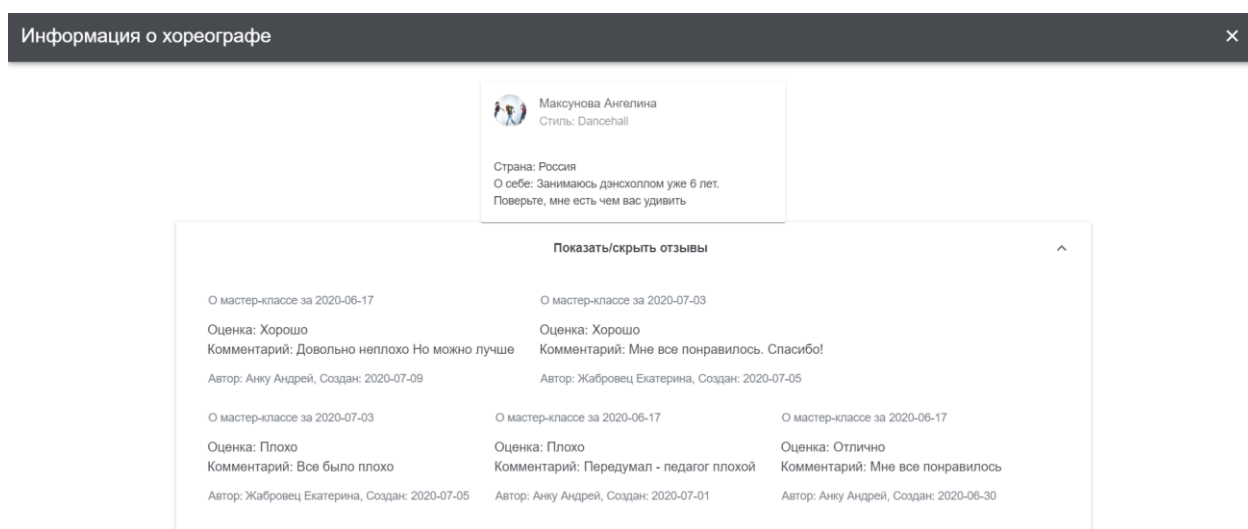


Рисунок 11 – Страница информации о хореографе с отзывами на его/её мастер-классы

Во вкладке «Топ посетителей» представлен рейтинг пользователей. Ранжирование производится на основании количества посещенных человеком мастер-классов

Афиша	Мои мастер-классы	Хореографы	Топ посетителей	Мой профиль
Рейтинг пользователей по количеству посещенных мастер-классов				
★	Жабровец Екатерина			3
★	Анку Андрей			2
★	Максунова Ангелина			1
	Леонидова Марина			0
	Братийчук Марина			0
	Дрынкин Алексей			0
	Дрынкин Никита			0
	Иванов Василий			0

Рисунок 12 – Вкладка «Топ посетителей»

Нажав на вкладку «Мой профиль», пользователь может выбрать одно из двух возможных действий: редактировать данные профиля или выйти из системы.

Афиша	Мои мастер-классы	Хореографы	Топ посетителей	Мой профиль
<p>Изменить данные профиля</p> <p>Фамилия-Имя Анку Андрей</p> <p>Электронная почта 19@gmail.co</p> <p>Телефон 21314134</p> <p>Страна Беларусь</p> <p>Биография Я загадка</p> <p>РЕДАКТИРОВАТЬ</p>				<p>Мой профиль</p> <p>Данные профиля</p> <p>Выход</p>

Рисунок 13 – Редактирование данных профиля

С. Интерфейсы для администратора

Администратору доступны все действия, которые могут быть реализованы обычным пользователем, с некоторыми дополнениями:

Афиша	Мои мастер-классы	Хореографы	Топ посетителей	Отчет по школам	Мой профиль
<p>Party-Dancehall Мест занято: 3/20</p> <p>Стиль: Dancehall Место: International Dance Center Дата: 2020-07-10 Время: 18:30:33 Продолжительность: 2 час(-а) Цена посещения: 1400 руб. Хореограф(-ы): Дрынкин Никита, Максунова Ангелина, Заняться Открыть список участников</p> <p>Хип-хоп - Non Stop Мест занято: 3/20</p> <p>Стиль: Hip-hop Место: International Dance Center Дата: 2020-07-20 Время: 19:00:06 Продолжительность: 1 час(-а) Цена посещения: 600 руб. Хореограф(-ы): Анку Андрей, Заняться Открыть список участников</p>	<p>Фильтровать мастер-классы по дате</p> <p>С По</p> <p>ФИЛЬТРОВАТЬ</p> <p>Доступные действия</p> <p>ДОБАВИТЬ МАСТЕР-КЛАСС 1</p>				

Рисунок 14 – Главная страница пользователя «Администратор»

На странице с афишей он может просматривать список участников, утвержденных для посещения занятия:

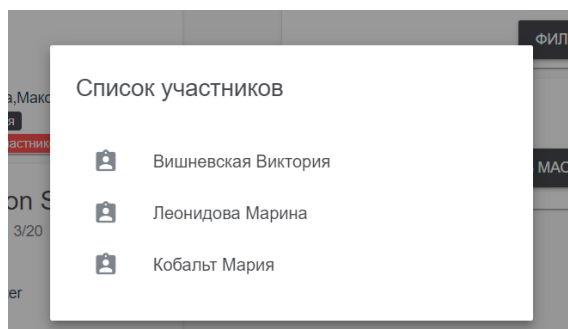


Рисунок 15 – Просмотр списка участников

Помимо этого, он может одобрять заявки на участие в мастер-классах, подтверждая внесение оплаты потенциальным посетителем. Окно с таблицей неутвержденных заявок появляется при нажатии на значок колокольчика. Количество неодобренных заявок отображается в красном кружке справа от него (см. рис. 14). Одобрять можно как одну, так и несколько заявок одновременно.

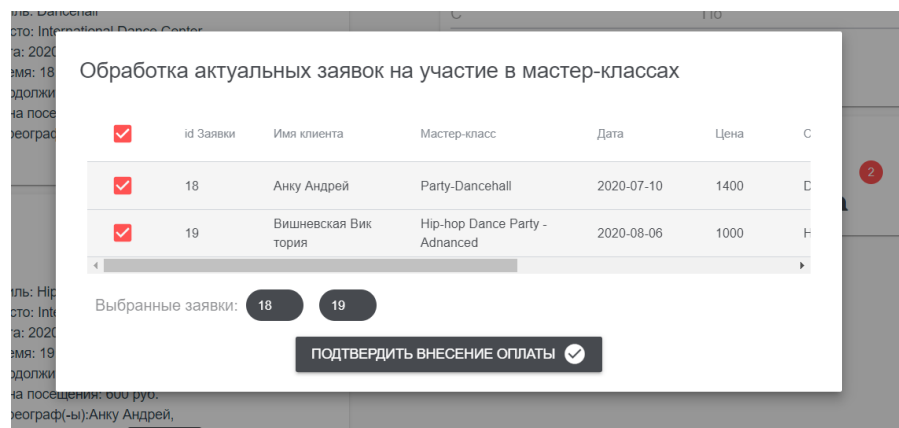


Рисунок 16 – Обработка актуальных заявок на участие в мастер-классах

Администратору также доступно действие добавления нового мастер-класса. Соответствующее окно появляется при нажатии на кнопку «Добавить мастер-класс»

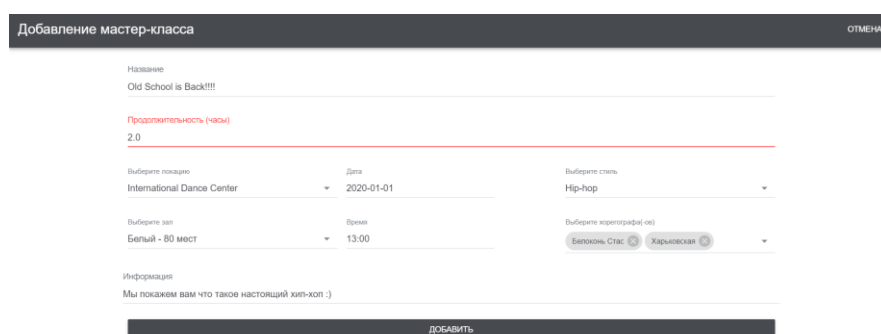


Рисунок 17 – Окно добавления мастер-класса

Для ввода даты и времени проведения мероприятия используются при нажатии на соответствующие поля отображаются специальные формы:

Дата

2020

Ср, Янв 1

<

Январь 2020

>

Вс	Пон	Вт	Ср	Чт	Пт	Суб
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

13:00

10	00
11	01
12	02
13	03
14	04
15	05
16	06

Рисунок 18 – Формы ввода даты и времени

При выборе танцевальной школы в выпадающем списке отображаются соответствующие этой школе залы с информацией об их вместимости. Аналогичным образом происходит фильтрация для назначаемых на класс хореографов, в соответствии с указанным администратором стилем.

Выберите стиль

Dancehall

Выберите хореографа(-ов)

Katrin Wow

Максунова Ангелина

☒ Katrin Wow

☒ Максунова Ангелина

Выберите стиль

Hip-hop

Выберите хореографа(-ов)

Белоконь Стас

Харьковская

Рисунок 19 – Пример фильтрации хореографов для разных стилей

Во вкладке «Отчет по школам» администратор может просмотреть информацию о сумме вырученной каждой школой денег и величиной ожидаемой прибыли.

Афиша	Мои мастер-классы	Хореографы	Топ посетителей	Отчет по школам	Мой профиль
Отчет о выручке по школам					
Название школы	Количество прошедших мастер-классов	Выручка(руб.)	Количество грядущих мастер-классов	Ожид. прибыль(руб.)	
International Dance Center	4	21000	3	13500	
Master Place	1	5000	2	6000	

Рисунок 20 – Отчет о выручке по школам

ЗАКЛЮЧЕНИЕ

В результате выполнения курсовой работы был разработан веб-сервис, предоставляющий информацию о мастер-классах с возможностью записи и администрирования, удовлетворяющий всем поставленным техническим требованиям.

Весь функционал сайта, связанный с взаимодействием пользователя и системы, был спроектирован при помощи средств, предлагаемых фреймворком Django REST Framework, использующим Python, что обеспечило вариативность относительно реализации поставленных задач, связанных с созданием API. В целях реализации фронтенд-части сайта было произведено знакомство с принципами работы и архитектурой Vue.js, а также были получены базовые навыки работы с библиотекой jQuery и языком программирования JavaScript.

Особенности спроектированной базы данных не исключают внесение дополнений для функционала веб-сервиса. В частности, не были разработаны идеи для личных кабинетов пользователей-педагогов и процедуры утверждения данного статуса у новых пользователей системы. Помимо этого, не исключается добавление на сайт мультимедийного контента, который мог бы включать видеозаписи мастер-классов и фотоотчеты.

Тем не менее, так как все сформулированные задачи были выполнены, работу можно считать законченной. Отдельно хотелось бы отметить тот факт, что аналогичных платформ в рунете на данный момент не существует - представленные в работе идеи могут быть заимствованы и доработаны в дальнейшем до приемлемого для конечного пользователя вида.

СПИСОК ИСТОЧНИКОВ

- 1) Документация Django REST framework [Электронный ресурс].
URL.: <https://www.django-rest-framework.org/> (дата обращения: 01.06.2020)
- 2) Kurylev A. Что такое Djsngo REST Framework? [Электронный ресурс].
URL: <https://mkdev.me/posts/chto-takoe-django-rest-framework> (дата обращения: 02.07.2020)
- 3) Brian Mulloy. Web API Design: The Missing Link. Best Practices for Crafting Interfaces that Developers Love [Электронный ресурс].
URL: <https://docs-apis.apigee.io/files/Web-design-the-missing-link-ebook-2016-11.pdf>
(дата обращения: 02.07.2020)
- 4) Vue.js official guide [Электронный ресурс]. URL.: <https://vuejs.org/v2/guide/> (дата обращения: 03.06.2020)
- 5) Muse-UI v.3.0.2 Documentation [Электронный ресурс]
URL: <https://muse-ui.org/#/en-US> (дата обращения: 20.06.2020)
- 6) Material Icons [Электронный ресурс]
URL.: <https://material.io/resources/icons/?style=baseline> (дата обращения: 20.06.2020)
- 7) Muse-UI carbon theme [Электронный ресурс]
URL.: <https://github.com/museui/muse-ui-carbon-theme> (дата обращения: 20.06.2020)
- 8) Video Background [Электронный ресурс]
URL.: <https://github.com/pespantelis/vue-videobg> (дата обращения: 03.07.2020)