

УДК 004.622

РАЗРАБОТКА МУЗЫКАЛЬНОГО СТРИМИНГОВОГО ВЕБ-СЕРВИСА

М.Р. ГРАДЮШКО*(Представлено: О.Н. ПЕТРОВИЧ)*

В данной статье объектом разработки представлен музыкальный стриминговый веб-сервис «Bro-Music». Рассмотрены основные принципы построения серверной части и графического интерфейса музыкального веб-сервиса.

Введение. Музыкальные сервисы пользуются большим спросом, нужно лишь подкупить пользователя, например, привлекательным внешним видом своего сервиса, уникальными возможностями или богатством музыкальной библиотеки. Тема, конечно, не новая, но совсем не утратила актуальности с течением времени.

Несмотря на довольно большое количество различных веб-сервисов, предоставляющих широкий спектр возможностей, связанных с прослушиванием музыки, все они обладают своими недостатками. К примеру, большинство таких сервисов платные, используют рекламу, не поддерживают оффлайн режим, не оповещают пользователя о выходе новых треков и альбомов любимых исполнителей или же не дают пользователю никакой интерактивности. Именно поэтому можно с уверенностью сказать, что даже сегодня многое можно улучшить в мире музыкальных сайтов.

Поэтому разработка музыкального стримингового веб-сервиса является актуальной.

Основной раздел. Разработанный музыкальный стриминговый веб-сервис предоставляет пользователям возможность прослушивать понравившуюся музыку. Программный продукт представляет собой веб-приложение, имеющее следующий функционал:

1. Прослушивание, комментирование, скачивание музыки.
2. Рассылка оповещений.
3. Предоставление информации о группах, треках, альбомах и т.д.
4. Возможность создания плейлистов.
5. Аутентификация и авторизация.

Для разработки такого веб-сервиса в качестве модульной платформы служит ASP .NET Core 2.2, поскольку она позволяет небольшим проектам и стартапам получить все преимущества платформы корпоративного уровня, при этом предоставляя удобные средства разработки, а также недорогую инфраструктуру. В будущем же неизбежен приход платформы и на крупный корпоративный рынок.

Преимущества платформы ASP.NET Core:

- единое решение для создания пользовательского веб-интерфейса и веб-API;
- страницы Razor делают создание кодов сценариев для страниц проще и эффективнее;
- Blazor позволяет использовать в браузере язык C# вместе с JavaScript. совместное использование серверной и клиентской логики приложений, написанных с помощью .NET;
- возможность разработки и запуска в ОС Windows, macOS и Linux;
- открытый исходный код и ориентация на сообщество;
- встроенное введение зависимостей;
- упрощенный высокопроизводительный модульный конвейер HTTP-запросов [1].

В качестве СУБД был выбран MySQL. Помимо универсальности и распространенности СУБД MySQL обладает целым комплексом важных преимуществ, которые делают её пригодной для решения поставленных задач. Данная система управления базами данных изначально создана таким образом, что множество встроенных функций безопасности в ней работают по умолчанию, что упрощает работу разработчику.

Взаимодействие с базой данных осуществляется с помощью современной технологии Entity Framework. Entity Framework предоставляет возможность работы с базами данных через объектно-ориентированный код C#. Этот подход предоставляет ряд существенных преимуществ: вам не нужно беспокоиться о коде доступа к данным, вам не нужно знать деталей работы СУБД SQL Server и синтаксиса языка запросов T-SQL, вместо этого вы работаете с таблицами базы данных как с классами C#, с полями этих таблиц как со свойствами классов, а синтаксис SQL запросов, который в ADO.NET раньше нужно было вставлять в код C# в виде команд, заменен на более удобный подход с LINQ. Entity Framework берет на себя обязанности по преобразованию кода C# в SQL-инструкции [2].

При работе с Entity Framework вам предоставляются огромные возможности по созданию модели базы данных с помощью интегрированной среды разработки (IDE) Visual Studio. Пример взаимодействия с базой данных представлен в листинге 1.

Листинг 1 – Получение самых популярных песен

```
1. model.AllSongsWithVoices = db.Voices
2. .Where(v => v.SongId != null)
3. .GroupBy(v => v.SongId)
4. .Select(g => new SongVoices { Song = allSongs[(int)g.Key], CountVoices =
g.Count() })
5. .OrderByDescending(v => v.CountVoices)
6. .Take(10);
```

Так как программа, работающая с базой данных, сделана в виде веб-приложения, все её формы выполнены в виде веб-страниц.

Административная часть сайта, где происходит полный контроль над содержимым веб-приложения, обладает теми же стилистическими особенностями, что и часть клиентская. Для удобства использования было принято отображать информацию в табличном виде. Администратор может добавлять, изменять, редактировать информацию и связывать данные из разных таблиц между собой.

Страницы сверстаны на языках HTML, CSS и JS, что позволило придать им понятный и привлекательный вид. Кроме того, стиль придаёт специальный фреймворк Bootstrap, позволяющий быстро стилизовать приложение, наполнив его красивыми элементами и применяя определенные заготовленные стили.

При проектировании приложения были учтены все возможные случаи некорректной работы программы, поэтому большинство нештатных ситуаций сопровождается оповещениями с описанием проблемы.

Ниже представлен скрипт обработки воспроизведения песни из листинга треков.

Листинг 2 – Воспроизведение песни из листинга

```
1. playlist.find('.song').click(function (e) {
2. e.preventDefault();
3. link = $(this);
4. if (link.attr("song-id") == song.id) {
5. playOrPauseSong();
6. } else {
7. $("#playlist.song").attr("src", "/img/listplay.png");
8. currentSong = link.parent().parent().index();
9. playSong();
11. $("#play img").attr("src", "/img/pause.png");
12. link.attr("src", "/img/list-pause.png");
13. }
14. });
```

Также клиенту доступен плеер для более удобного прослушивания музыкальных композиций. В нем доступен ряд возможностей, такие как переключение песен, воспроизведение и приостановка треков, регулирование уровня громкости, перемотка композиции. Скриншот плеера представлен на рисунке 1.

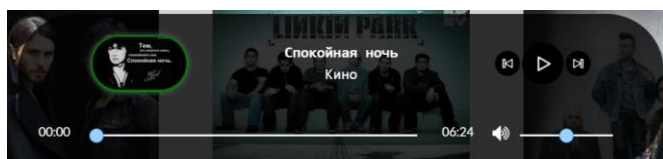


Рисунок 1. – Плеер

Заключение. В данной статье были рассмотрены средства для разработки веб-приложений, при помощи которых был реализован стриминговый веб-сервис, соответствующий современным требованиям.

ЛИТЕРАТУРА

1. Документация по ASP.NET Core [Электронный ресурс]. Режим доступа: <https://docs.microsoft.com/ru-ru/ASPNET/>. – Дата доступа: 22.09.2020.
2. ProfessorWeb [Электронный ресурс]. Режим доступа: <https://professorweb.ru/my/entityframework/6/level1/>. – Дата доступа: 23.09.2020.