Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение высшего образования  
«Московский политехнический университет»

Кафедра «Инфокогнитивные технологии»  
Образовательная программа «Веб-технологии»

Отчет по курсовому проекту  
по дисциплине «Инженерное проектирование»

Тема: «Каталог музыкальных композиций»

Выполнил:  
Студент группы 201-321

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Бухарцев М.А.  
подпись, дата

Принял:  
Старший преподаватель

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Даньшина М.В.  
подпись, дата

Москва 2021

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc75816989)

[ОСНОВНАЯ ЧАСТЬ 4](#_Toc75816990)

[ЗАКЛЮЧЕНИЕ 8](#_Toc75816991)

[СПИСОК ИСТОЧНИКОВ 9](#_Toc75816992)

[ПРИЛОЖЕНИЯ 10](#_Toc75816993)

# ВВЕДЕНИЕ

Задача курсового проекта – в течение 2 месяцев с нуля разработать и запустить сайт. Цель данного проекта: готовое SPA и PWA по теме «Каталог музыкальных композиций». Работа над проектом производится поэтапно, однако допускается смешанное и непоследовательное выполнение этапов.

Подготовительные этапы:

* Изучение задач и всех материалов, расположенных на странице с задание
* Разрешение возникших вопросов с преподавателем
* Анализ аналогов (поиск отечественных и зарубежных сайтов, анализ структуры сайта, юзабилити, функциональности)

Основные этапы:

* Проектирование (обзор кейсов использования сайтов, CJM, выбор функционала, проектирование бизнес-процессов)
* Разработка название, фирменного стиля и дизайна сайта
* Разработка мокапов и прототипов (мокап сайта, макет сайта, структура базы данных)
* Разработка и тестирование сайта
* Запуск сайта (deploy to production)
* Тестирование сайта в работе

Финальные этапы:

* Документирование
* Создание отчётных документов
* Оформление Git-репозитория

# ОСНОВНАЯ ЧАСТЬ

Во время подготовительного периода были всесторонне изучены материалы по проекту, а также сфера по теме проекта. Был сформирован и тщательно исследован следующий **список аналогов**:

* Яндекс.Музыка
* Deezer
* Spotify
* Soundcloud
* Google Play Music
* Apple Music
* Last.fm
* Музыка ВКонтакте

Решено использовать **английский** язык как язык интерфейса приложения.

На этапе проектирования сайта был сформирован **список функций** каталога музыкальных композиций:

* Просмотр в списке, просмотр подробной информации и поиск сущностей
* Административный доступ с подобным функционалом, а также CRUD
* SPA
* PWA
* Обработка ошибки 404 (не найдена страница)
* Смена темы
* Поиск
* Фильтрация (встроенная в соответствующие сущности)
* Просмотр главная страницы и страницы о проекте
* Дополнительные ссылки

Следующий этап – **этап брендинга** – был выполнен с особой филигранностью. С нуля создан оригинальный (без заимствований) набор материалов. **Название** Foamy случайно, но не случайно. Это сгенерированное (с 26 попытки) в рандомайзере слово, которое переводится как «пенный, пенистый». Это слово ассоциируется с вечеринкой и игристым вином. Ассоциации, в свою очередь, создают у пользователя праздничное настроение и могут быть связаны с музыкой.

К разработке **логотипа** применён не менее кропотливый подход. Целая нота символизирует цельность проекта и отражает музыкальную суть. Логотип заменяет букву «o» в названии и также может быть воспринят как пузырёк с той самой пенной вечеринки. Цвет (HEX: #00bfff) – «морозное небо» - хорошо отображается как в светлой теме, так и в тёмной. По моим устным опросам, оттенки зелёного и синего больше всего ассоциируются с музыкой. Именно этот цвет идеально ложится в разрабатываемый дизайн.

В перспективе есть возможность добавления цветного логотипа в качестве символа ноты в разные части проекта, а также заменять букву «o» в важных словах или заголовках. Ещё из логотипа можно сделать загрузочную заставку, пустив вращающийся градиент (с цветом #2bd9d0, близким к оттенку морской волны). Такой градиент будет отсылать к заказчику - Московскому политехническому университету, а также к преломлению света на поверхности пузыря.

Дальнейшие этапы – **разработка прототипов** – велись на разных стадиях работы над проектом. Основываясь на требованиях и полученном списке функций, решено создать многостраничный сайт, который будет одновременно являться SPA и PWA. К сожалению, функциональность PWA недоработана в связи с тем, что предоставленный домен не позволяет включить SSL и, соответственно, реализовать PWA.

Для фронтенд-стороны проекта **использовался фреймворк** Vue.js. Один из наиболее простых, он позволяет реализовать весь нужный функционал. Также был найден CSS-фреймворк Vuetify, использование которого ускоряет работу над проектом. Макет, таким образом, создаётся на основе готовых компонентов. **Адаптивный дизайн** же разрабатывается исходя из рамок и преимуществ фреймворков, при этом остаётся простор для фантазии и помещения брендинга.

Немаловажен в проекте **интерфейс администрирования**. В нём реализована авторизация (ААА, по всем правилам информационной безопасности) и CRUD. Разработка основана на Django Admin, с присоединением модуля Jazzmin. Эта тема практически идентична разработанному дизайну, что полезно для юзабилити. Главная страница админ-панели – дашборд (dashboard). С неё можно перейти на редактирование каталога или пользователей. Также сохраняется история действий пользователей.

На любой странице проекта отображается **footer** (кроме панели администратора, но там тоже есть все нужные ссылки). Из него можно открыть гитхаб репозиторий, админ-панель, API-интерфейс, Support. GitHub-репозиторий проекта ведётся с использованием веток и публичных данных. Например, все документы, включая этот, можно найти в директории «\_docs», а большую часть информации, в укороченном виде, можно получить, прочитав README.md. **API**-взаимодействие построено на Django Rest framework, при этом без авторизации API доступен только для GET запросов и возвращает данные в формате JSON. Ссылка support (поддержка) ведёт в почтовый клиент пользователя и автоматически заполняет некоторые поля письма, предлагая написать мне, как автору этого проекта.

Кроме всего перечисленного, были учтены ещё многие **требования**. Со всеми можно ознакомиться в оценочном листе в приложении. К примеру, агрегирующая информация тоже присутствует в проекте. Она выводится на главной странице и странице «About». Здесь собраны основные данные по проекту. Распечатать можно эту информацию, а также любую другую страницу (например, страницу с информацией о треке). Ещё проект обладает высокой доступностью (правильно работает tabindex, масштабирование и изменение шрифтов). При разработке стилей применены БЭМ и препроцессоры SASS, SCSS.

Немного о **backend**. Он написан на языке программирования Python с помощью Django. Все материалы оригинальны или являются open source проектами. Отдельного внимания заслуживает база данных, модель которой можно найти в приложении. Долгие поиски готовой базы данных, содержащей в себе каталог информации о музыке, не дали удовлетворительного результата. Даже самые близкие к проекту решения, вроде [MusicBrainz](https://musicbrainz.org/), не подходили по различным причинам. Помощь пришла с неожиданной стороны. Open source приложение [Kodi](https://kodi.tv/) для смарт ТВ, на которое я случайно наткнулся, позволило создать базу данных из треков, скачанных мной за годы жизни. Проведя множество манипуляций, я переместил все данные в свою SQLite базу данных. Теперь там треки 26 жанров, исполненные различными артистами.

**Проект Foamy запущен на сервере** c помощью WSGI.

При разработке проекта было применено ручное тестирование, в частности приложение Postman для тестирования API. Проект идеально проходит валидацию и имеет **user friendly** дизайн.

Список использованных технологий:

* Django
  + Django Rest
  + Jazzmin
* SQLite
* Vue
  + VueX
  + Vue-Router
  + Vuetify
  + Vuetify-loader
* Axios
* MDI Icons
* FontAwesome
* Fonts
  + 'Please write me a sing' ByTheButterfly
  + 'Tchaikovsky' JoannaVu

Возможные улучшения:

* Взаимодействие по API с существующей базой музыкальных композиций
* Создание аккаунтов (не только администраторов) с возможностью добавления собственных плейлистов
* Воспроизведение музыки на сайте
* Реализация PWA – осталось реализовать https-взаимодействие и добавить «service worker»
* Loader с градиентом, описанный в разделе про брендинг

# ЗАКЛЮЧЕНИЕ

В результате работы над проектом разработан каталог музыкальных композиций. Согласно отметкам о выполнении в оценочном листе набрано **99** баллов (с учётом получения максимального балла за каждый пункт). Это оценка «**Отлично**».

**Все ссылки на итоговые материалы:**

Сайт Foamy: <http://foamy.std-1374.ist.mospolytech.ru/>

[GitHub репозиторий проекта](https://github.com/bouhartsev/Foamy)

**Контакты для связи:**

Бухарцев Матвей

[programmer@bouhartsev.top](mailto:programmer@bouhartsev.top)

[VKонтакте](https://vk.com/bouhartsev) (@bouhartsev)

# СПИСОК ИСТОЧНИКОВ

[Vue.js](https://vuejs.org/)

[Vuetify](https://vuetifyjs.com/en/)

[Django](https://www.djangoproject.com/)

[djbook](https://www.djbook.ru/)

[Django REST framework](https://www.django-rest-framework.org/)

[Tproger](https://tproger.ru/)

[Stack Overflow](https://stackoverflow.com/)

[MDN](https://developer.mozilla.org/)

[LMS](https://lms.mospolytech.ru/)

[FIT Mospolytech](https://fit.mospolytech.ru/)

[MusicBrainz](https://musicbrainz.org/)

[Kodi](https://kodi.wiki/view/Databases)

[Инструкция по размещению на сервере](https://drive.google.com/file/d/1BF84WhOL-zs3pe7I5YmcSlerHMod3Wm0/view)

# ПРИЛОЖЕНИЯ

Оценочный лист

Мокап сайта

Макет сайта

ER-модель базы данных

Данные для авторизации