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: «Доработка информационной системы для салона красоты»

**Выполнил:**

Студент группы 191-361

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Козлов А.В.

подпись, дата

**Принял:**

Старший преподаватель

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Даньшина М.В.

подпись, дата

ВВЕДЕНИЕ

В рамках данного курсового проекта была произведена доработка информационной системы для салона красоты. В состав разработанной системы входят веб-приложение, серверное приложение и база данных. Система размещена на хостинге факультета информационных технологий. [8]

Помимо результатов работы за осенний семестр 2021/2022 учебного года, в отчёте преимущественно содержится доработанный текст из отчёта за прошлый семестр. Этот текст было решено оставить, чтобы вся система была описана в одном отчёте. Так же все базовые аспекты системы, например, архитектура серверного-приложения, были возложены в предыдущем семестре и не были подвержены изменениям.

Целью данного курсового проекта является расширение информационного поля небольшого салона красоты, а именно формирование информационной системы для более удобного способа информирования клиентов об оказываемых услугах и способа ведения учёта заказов.

Задачи данного курсового проекта:

* провести анализ отечественных и зарубежных аналогов;
* составить формальные модели для формирования представления о системе;
* составить модель данных для создания на её основе базы данных;
* создать веб-приложение и серверное приложение, ведя разработку в репозитории и использованием веток;
* для каждого приложения разъяснить структуру, описать особенности функционирования и провести тестирование;
* доработать проект в соответствии с требованиями осеннего семестра 2021/2022 учебного года.

Объектом исследования данного курсового проекта является небольшой салон красоты, а предметом исследования является формируемая информационная система для салона красоты.

В данной курсовой работе содержатся 7 таблиц и 9 рисунков, состоит она из 21-ой страницы.

# АНАЛИТИЧЕСКАЯ ЧАСТЬ

## Анализ аналогов

Согласно объекту исследования в качестве аналогов были выбраны салоны красоты, а не сети салонов красоты. Были выбраны два отечественных аналога и один зарубежный.

### Салон красоты «Силуэт»

«Силуэт» – это отечественный центр здоровья и красоты в Дедовске. [3]

Оценка критериев сайта салона красоты «Силуэт» приведена в таблице 1.1.

Таблица . – Оценка критериев сайта «Силуэт»

|  |  |
| --- | --- |
| Критерий | Оценка |
| СЕО-оптимизация | Отсутствуют ключевые слова. Присутствует содержательное описание. |
| Доступность | Нет специальных атрибутов для экранного диктора. Ссылки не обладают сопровождающей информацией. Плохая контрастность у описания услуг. |
| Структура | Сайт был создан при помощи «Joomla!».  В шапке сайта расположена навигационная панель по семи разделам. При прокрутке разделов она закрепляется в верхней части экрана. Каждый из семи разделов расположен в собственном html-файле, шапка дублируется во всех файлах. |
| Адаптивность | Сайт адаптивен под мобильные устройства и планшеты, но не под широкоформатные мониторы. При сокращении ширины шапки, навигационная панель сворачивается в выезжающее с лева окно. На мобильных телефонах на главной странице излишнее количество карточек с услугами. |
| Юзабилити | Сайт интуитивно понятен, но нет возможности вывести все услуги на одной странице и применить к ним фильтр по каким-либо параметрам или поиск. |
| Функциональность | На сайте можно узнать об оказываемых услугах, контактной информации, вакансиях, новостях и акциях. |
| Быстродействие | Не используются современные форматы изображений. Множество файлов со стилями блокируют первоначальную отрисовку страницы. Главная страница потребляет более четырёх МБ трафика. Нет оптимизации под мобильную версию. |

### Салон красоты «LUX»

«LUX» – это отечественный салон красоты в Красногорске. [4]

Оценка критериев сайта салона красоты «LUX» приведена в таблице 1.2.

Таблица . – Оценка критериев сайта «LUX»

|  |  |
| --- | --- |
| Критерий | Оценка |
| СЕО-оптимизация | Ключевые слова – два коротких предложения, содержащие вставку «Салон красоты LUX». Присутствует скудное описание. |
| Доступность | Нет специальных атрибутов для экранного диктора. Присутствуют пустые ссылки на главной странице. Хорошая контрастность. |
| Структура | В шапке сайта расположена навигационная панель по восьми разделам. При прокрутке разделов она закрепляется в верхней части экрана. Главная страница заполнена небольшими секциями из других разделов. |
| Адаптивность | Сайт адаптивен под мобильные устройства и планшеты, но не под широкоформатные мониторы. При сокращении ширины шапки, навигационная панель уходит за границы окна, но пункты доступны при её прокрутке. |
| Юзабилити | На главной странице представлено описание. Пользователь сразу видит навигационную панель и контактную информацию с расписанием при заходе на сайт. Прайс лист удобно выполнен, но в мобильной версии слишком много позиций из прайс-листа на главной странице. |
| Функциональность | На сайте можно узнать об оказываемых услугах, контактной информации и акциях. В прайс-листе присутствует навигация по категориям услуг и поиск услуг. Присутствует раздел со специалистами с фильтрацией по нескольким параметрам. Возможность запустить 3D-тур салона и встроенная карта с Яндекса с указанием географического местоположения салона. |
| Быстродействие | Замечаний по desktop-версии практически нет. Не используются современные форматы изображений. Главная страница потребляет более трёх с половиной МБ трафика. |

### Салон красоты «Beautify»

«Beautify» – это салон красоты, расположенный в столице Ирландии. [6]

Оценка критериев сайта салона красоты «Beautify» приведена в таблице 1.3.

Таблица . – Оценка критериев сайта «Beautify»

|  |  |
| --- | --- |
| Критерий | Оценка |
| СЕО-оптимизация | Присутствуют ключевые слова в виду двух словосочетаний. Присутствует содержательное описание. |
| Доступность | Нет специальных атрибутов для экранного диктора. Некоторые ссылки не обладают сопровождающей информацией. Интерфейс достаточно контрастен. |
| Структура | В шапке сайта расположена навигационная панель по шести разделам. Главная страница заполнена ссылками к описанию услуг, оформленными в виде блоков. В каждом разделе повторяется блок с контактной информацией. |
| Адаптивность | Сайт не обладает адаптивностью под мобильные устройства, планшеты или широкоформатные мониторы. |
| Юзабилити | Сайт интуитивно понятен, но нет возможности вывести все услуги на одной странице и применить к ним фильтр по каким-либо параметрам, однако очень удобно списком можно вывести все услуги в рамках одной категории и сравнить их. Блок с контактами в каждом разделе неуместен и занимает слишком много места. |
| Функциональность | На сайте можно узнать об оказываемых услугах, контактной информации, расписании, а также подать заявку на приём. Помимо этого, есть форма связи через электронную почту. |
| Быстродействие | Не используются современные форматы изображений. «Тяжёлая» кодировка изображений. У статических объектов короткое время жизни в кеше – в среднем пятнадцать минут. Объём сайта: менее одного МБ. |

## Формальные модели

В качестве формальной модели была построена функциональная модель для бизнес-процесса «Записаться на приём». Для создания модели использовалась методология IDEF0. В составе функциональной модели была построена графическая диаграмма (Рисунок 1.1) с декомпозицией до первого уровня (Рисунок 1.2). [2]

Для построения диаграмм использовалось программное обеспечение ERwin Process Modeler. [16]

Коротко говоря, данная модель отображает структуру и функции системы, так же потоки информации и материальные объекты, преобразуемые функциями.

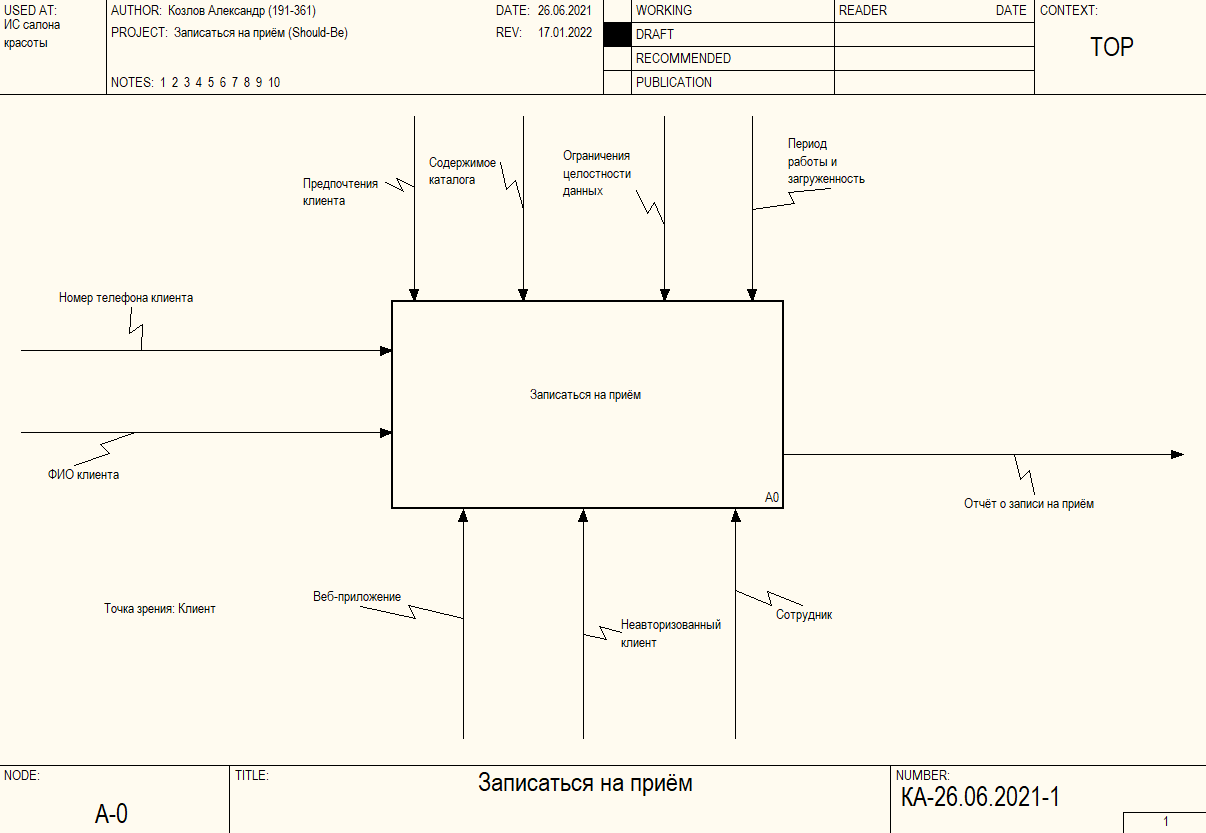


Рисунок . – Обязательная контекстная диаграмма верхнего уровня

Данная диаграмма определяет один из процессов, который должен быть реализован в системе в конечном итоге. Сам же процесс описан с точки зрения клиента, поэтому некоторые технические подробности, например, участие серверного-приложения, опущены.

В итоге клиент должен получить отчёт об успехе или провале подтверждения той заявки, которую он сформирует. Сам же отчёт выражен в устном подтверждении сотрудника салона красоты по телефону.

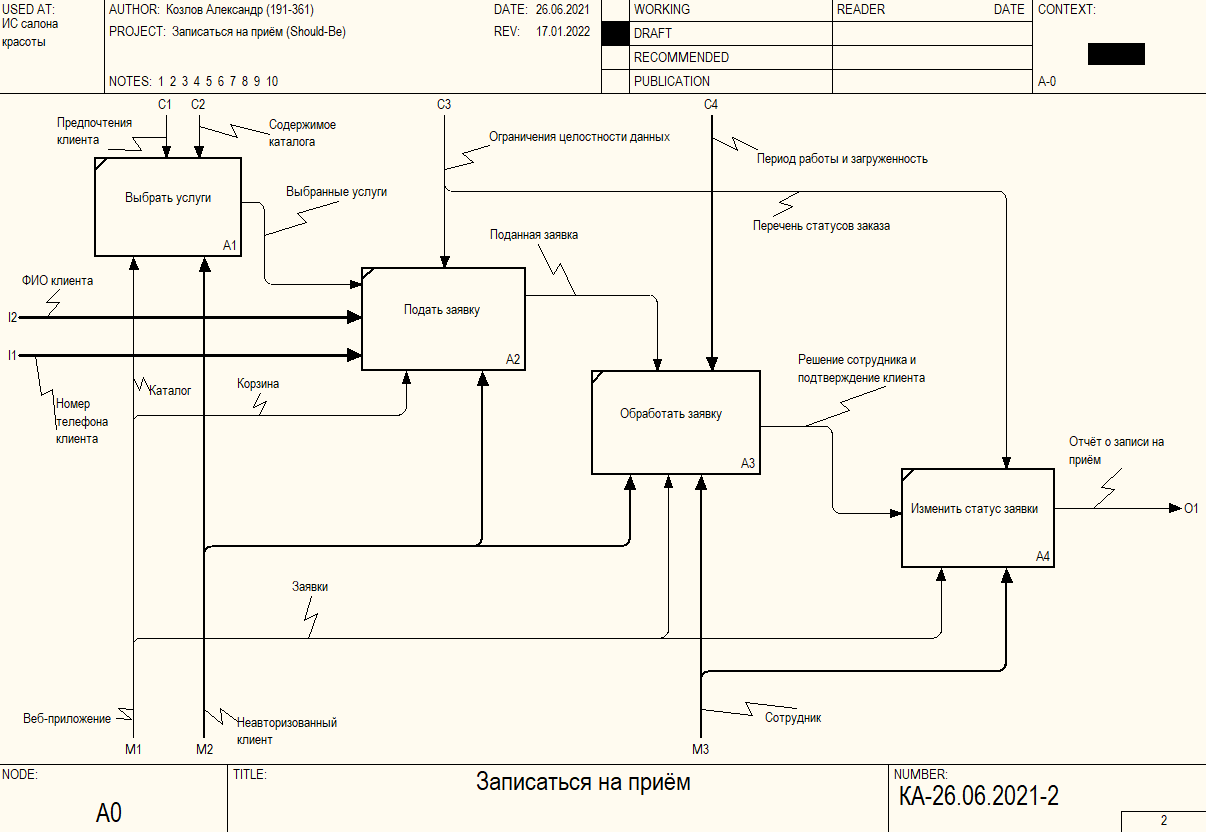


Рисунок . – Дочерняя диаграмма узла A-0

Как видно из диаграмм, все заявки должны обрабатываться сотрудниками салона красоты. Данный фактор в совокупности с накладываемыми ограничениями целостности данных позволит уменьшить число нежелательных заявок.

Прецеденты, возможные в рамках системы, определены на диаграмме прецедентов на рисунке 1.3.[15]

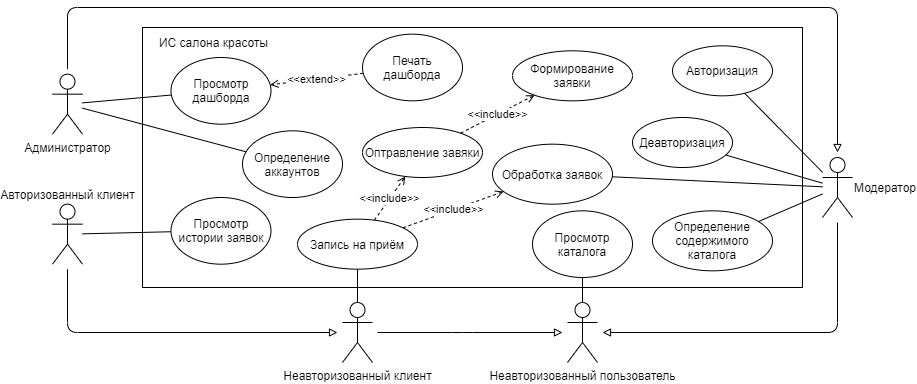


Рисунок . – Диаграмма прецедентов ИС салона красоты

«Неавторизованный клиент» — это слой абстракции, который характеризует разницу между посетителями сайта, которые просматривают каталог, и посетителями, которые отправляют заявку на приём.

На рисунке 1.4 представлена «карта пути клиента» обычного клиента салона красоты, который предпочитает производить поиск в Интернете.

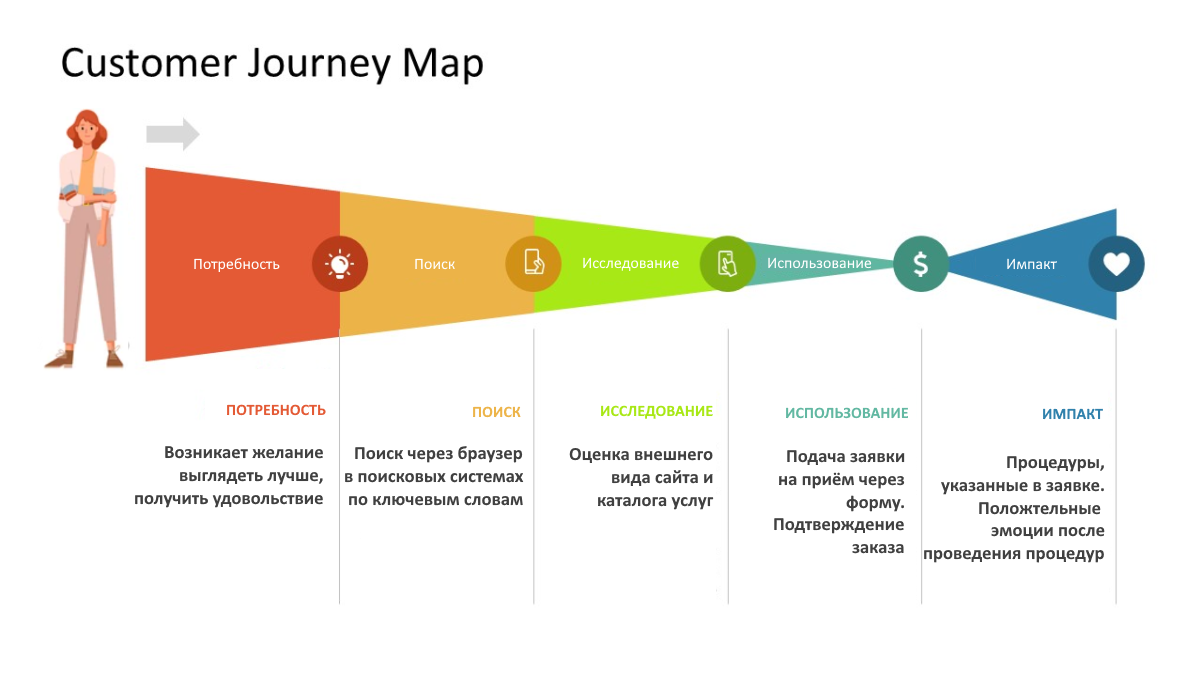


Рисунок . – Customer journey map

## Выводы по аналитической части

На основе данной теоретической части можно изложить следующие выводы:

* был проведён анализ аналогов разрабатываемого веб-приложения и во внимание будут взяты результаты этого анализа: наличие ключевых слов, адаптивной вёрстки, юзабилити и прочих приведённых в таблицах 1.1-1.3 критериев;
* была представлен диаграмма прецедентов, которой необходимо следовать при реализации системы;
* была представлена IDEF0-диаграмма с декомпозицией до первого уровня для подробного описания процесса «Записаться на приём» в рамках ИС;
* была представлена CJM потенциального клиента разрабатываемой ИС;

В рамках осеннего семестра были проанализированные новые аналоги, так как предыдущие представляли из себя сети салонов красоты. Так же была подправлена функциональная модель. На диаграмме прецедентов актёр «Клиент» был разбит на следующих актёров: «Неавторизированный пользователь», «Неавторизированный клиент», «Авторизированный клиент». Исправлены общие фактические ошибки и оформление.

# ПРОЕКТНАЯ ЧАСТЬ

## Модель данных

### Концептуальная модель

При концептуальном проектировании базы данных была построена модель данных в виде ER-диаграммы, представленной на рисунке 2.1. [7]

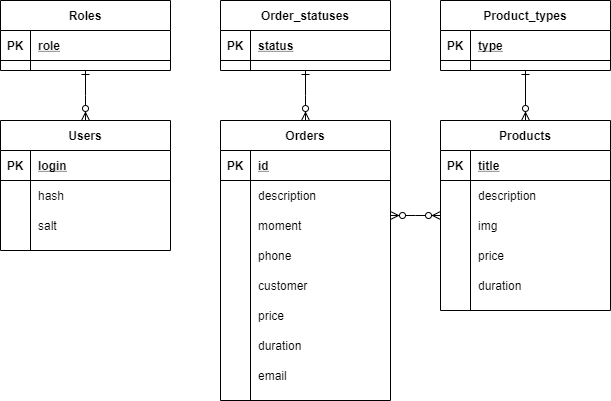


Рисунок . – Концептуальная модель данных

### Аспект манипуляции

Хранить пароли пользователей в чистом виде не безопасно, поэтому, как видно из рисунка, вместо пароля должны хранится хэш и «соль». «Соль» единожды генерируется на сервере при создании пользователя и нужна для того, чтобы хэш у пользователей с одинаковыми паролями отличался. «Соль» представляет из себя случайный набор символов.

Например, у двух пользователей пароль «123456». В таком случае в поле «hash» в таблице «Users» будет храниться не хэш от «123456», а хэш от «123456» с приписанной на конце «солью». [13]

### Аспект структуры

В базе данных должны быть три служебные таблицы с предопределёнными значениями. Одна для хранения видов ролей пользователей, другая для хранения видов статусов заказа и последняя для хранения типов услуг салона красоты.

Между сущностями «Orders» и «Products» объявлена связь «Многие-ко-многим», поэтому для реализации этой связи в базе данных необходимо создать промежуточную таблицу.

### Аспект целостности

Для поддержания целостности данных в базе данных необходимо использовать внешние ключи между зависящими сущностями. Для простоты добавления этих ограничений у каждой сущности должно быть поле «id», по которому будет построен внешний ключ.

Логины у пользователей должны быть уникальны, поэтому необходимо добавить ограничение уникальности значений в поле «login» в таблице «Users». Аналогичное можно сказать и про названия услуг.

### Физическая модель

На основе модели, приведённой в пункте 2.1.1, была построена физическая модель данных, описанная в виде ER-диаграммы на рисунке 2.2.

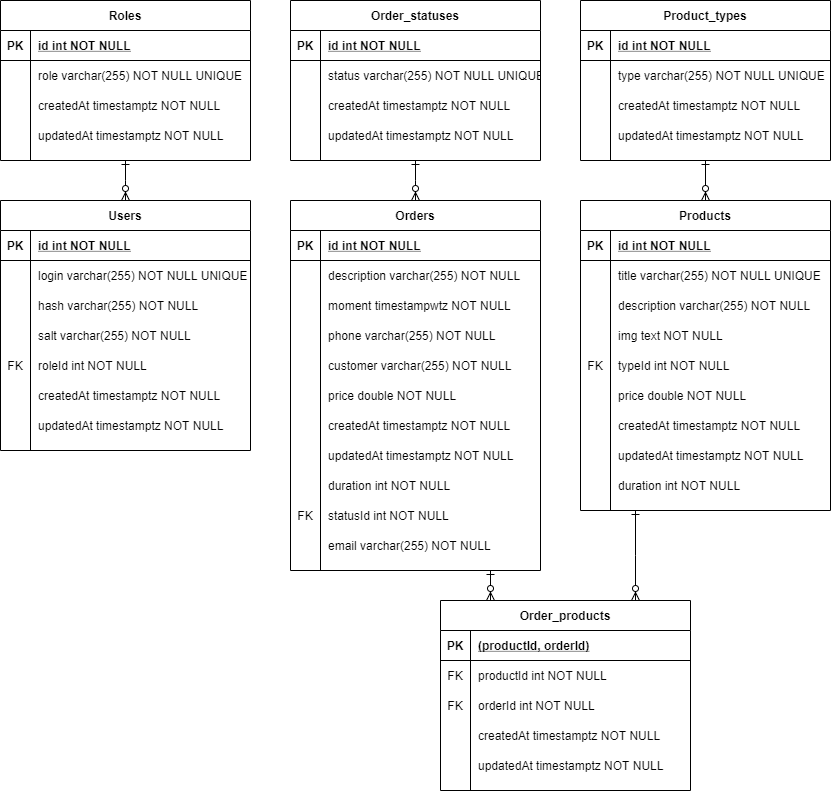


Рисунок . – Физическая модель данных

## Веб-приложение

Веб-приложение было создано на основе фреймворка «Vue» на языке программирования «JavaScript». [10]

Перечень используемых зависимостей в приложении:

* «axios»: используется для реализации API;
* «vue-cryptojs»: используется для хэширования паролей перед отправной к серверному приложению;
* «vue-router»: используется для поддержки общих у нескольких страниц html-элементов и удобной настройки маршрутизации;
* «vuex»: используется для реализации глобального хранилища в приложении;
* «node-sass», «sass-loader», «webpack»: используются для поддержки написания стилей на scss;
* «eslint»: используется для поддержания общего семантического стиля в рамках проекта.

### Структура проекта

Реализация API находится в папке «api» и основана на «axios»: объект «axios» и его конфигурация представлены в файле «instance.js». Импорт модулей API и объекта «axios» представлены в файле «index.js». Прочие файлы являются модулями, между которыми разделена реализация методов API. [5]

В папке «assets» хранятся статические файлы, такие как картинки, шрифты и стили.

Компоненты Vue-приложения распределены между папками «pages» и «components» и «layouts». В свою очередь, в папке «pages» хранятся компоненты, описывающие собой целую страницу. Аналогично, в папке «layouts» хранятся обёртки для страниц. В папке «components» описаны отдельные компоненты, которые могут встречаться на разных страницах.

В папке «router» расположен объект «VueRouter», реализующий навигацию в приложении. Навигация в нём реализована иерархически, что позволяет выводить на экране не только компонент, в котором находится пользователь, но и элементы «родителей» этого компонента.

В папке «plugins» расположен интерпретатор. Он нужен для того, чтобы преобразовывать или форматировать значения. Например, в объекте услуги из каталога в поле «duration» указано число «70». Интерпретатор преобразует это число в строку «1 час. 10 мин.».

В папке «store» реализовано глобальное хранилище приложения на основе «vuex». Во «vuex» существует четыре основных понятия: «state» – данные, «getter» – метод получения данных, «mutation» – метод размещения данных и «action» – асинхронный метод получения данных. В проекте активно используются возможности «vuex», а некоторые элементы хранилища вынесены в отдельные модули в отдельных файлах в подпапке «modules».

### Стили

Все стили описаны в папке «assets/css» и написаны на «scss» с использованием вложенностей, миксинов и переменных.

Используется методология БЭМ. [1]

В файле «variables.scss» описаны все переменные цветовых значений – ни одно цветовое значение не задано без использования этих переменных, что позволяет очень просто интегрировать в приложение возможность переключения на другую тему, не меняя при этом сами классы стилей.

Основные стили прописаны в файле «myStyle.scss». В начале файла импортируются переменные цветовых значений, затем сбрасываются некоторые стили по умолчанию, после этого задаются миксины: «form» применяется к каждой форме, а «layout» для «обёрток», у которых снизу должен быть футер.

Вёрстка адаптивна как для широких экранов, так и для мобильных устройств.

На рисунке 2.3 представлен пример выбранной цветовой гаммы.

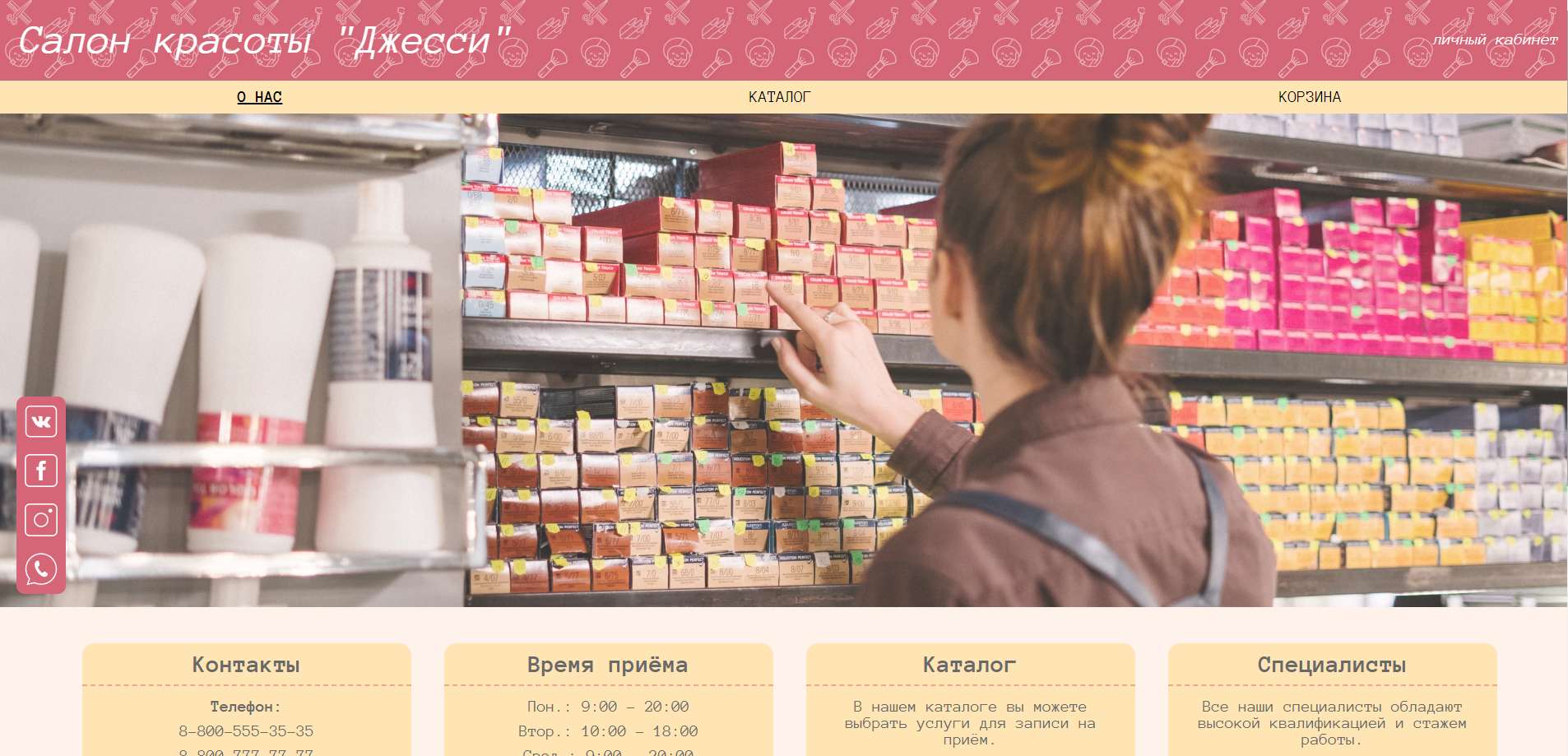


Рисунок . – Главная страница

К современным решениям дизайна можно отнести моменты:

* размеры шрифта и элементов преимущественно определены в метрике «em»;
* блок фильтрации в каталоге прилипает к верхней части экрана, когда пользователь прокручивает каталог;
* футер, прилипающий к нижней части экрана и не съезжающий вверх из-за пустого пространства после основного контента;
* использование небольшого пула сочетающихся цветов для оформления элементов;
* оформление элементов в виде карточек для упрощения зрительного восприятия;
* отсутствие лишних границ между элементами – мягкие переходы между ними;
* описание карточки заказа расположено в выпадающей части, а не в, например, открывающемся модальном окне, что позволяет держать раскрытыми сразу несколько карточек и сравнивать их между собой.

### Особенности функционирования

Фильтр на странице каталога реализован динамически – фильтр по категориям показывает лишь те категории, которые содержатся в пришедших с серверного приложения данных, значения диапазона минимальной и максимальной цены рассчитывается на основании содержимого каталога, то же самое можно сказать и про значения диапазона продолжительности услуг.

В случае, если пользователь перешёл в корзину, но она пуста, отрисовывается компонент перенаправления в каталог, в панели навигации рядом со вкладкой корзины выводится количество добавленных товаров.

У администратора на дашборде выводится информация, агрегированная на сервере. Дашборд адаптирован для печати. [14]

Из-за использования модуля для удобной навигации «vue-route» путь, отображаемый в url-строке браузера, заменён не настоящим, поэтому после перезагрузки страницы веб-сервер может выдать ошибку «503», так как браузер попытается загрузить контент по не настоящему пути. Для предотвращения подобной ситуации необходимо подредактировать конфигурационной файл веб-сервера. Но редактирование данного файла на хостинге факультета информационных технологий требует прав администратора. На localhost или на собственном сервере проблемы не будет.

Модальное окно реализовано при помощи обёртки с фиксированным позиционированием. Его можно переиспользовать, так как оно реализовано при помощи слота, куда вставляется содержимое.

Авторизация у модераторов и администраторов происходит через единую форму посредством ввода логина и пароля, а у клиента происходит посредством ввода кода из письма, которое отправляется на указанную им электронную почту в том случае, если данная электронная почта значится хотя бы у одной заявки на приём в базе данных.

В качестве ещё одной особенности можно привести переиспользуемые компоненты с разными режимами работы. Например, карточка заказа в личном кабинете клиента и карточка заказа в личном кабинете модератора – это один и тот же компонент, которому на вход поступает булево значение. И в зависимости от этого карточка меняет свои свойства. Например, модератор может использовать её чтобы изменить статус заказа, но клиент может использовать ту же карточку чтобы отслеживать статус заказа.

Так же навигационные панели в личных кабинетах у администратора, модератора и клиента организованны схожим образом. Компонент принимает на вход массив названий для маршрутизации, которые задаются во «vue-router», и преобразовывает их в ссылки на страницы, которые определены под этими названиями для маршрутизации.

### Тестирование

Для тестирования веб-приложения был привлечён метод ручного тестирования на основе прохождения тест-кейсов. Сценарии тест-кейсов начинаются с главной страницы веб-приложения. Примеры тест-кейсов представлены в таблицах 2.1-2.4.

Таблица . – тест-кейс «подача заявки на приём»

|  |  |
| --- | --- |
| Описание действий | Ожидаемый результат |
| 1. Нажать ЛКМ по надписи «Каталог» в навигационной панели в верхней части сайта; 2. путём нажатия ЛКМ по кнопкам «В корзину» на карточках в каталоге товаров выбрать как минимум одну услугу; 3. нажать ЛКМ по надписи «Корзина» в навигационной панели в верхней части сайта; 4. заполнить форму отправки заявки на приём, введя лишь ФИО, номер телефона и предполагаемое время приёма; 5. нажать ЛКМ по кнопке «Отправить заказ». | Клиент должен быть перенаправлен на страницу, где указан его номер заказа и сопутствующая информация. |

Таблица . – тест-кейс «подача заявки на приём: проверка валидности значений»

|  |  |
| --- | --- |
| Описание действий | Ожидаемый результат |
| 1. Повторить первые три действия из тест-кейса в таблице 2.1; 2. заполнить форму отправки заявки на приём, введя лишь ФИО; 3. нажать ЛКМ по кнопке «Отправить заказ»; 4. заполнить форму отправки заявки на приём, введя лишь номер телефона; 5. нажать ЛКМ по кнопке «Отправить заказ»; 6. заполнить форму отправки заявки на приём, введя лишь предполагаемое время приёма; 7. нажать ЛКМ по кнопке «Отправить заказ»; 8. заполнить форму отправки заявки на приём, введя лишь ФИО, некорректный номер телефона и предполагаемое время приёма; 9. нажать ЛКМ по кнопке «Отправить заказ». | После каждого нажатия по кнопке «Отправить заказ» на форме должно появляться соответствующее предупреждение об ошибке. |

Таблица . – тест-кейс «Авторизация администратора»

|  |  |
| --- | --- |
| Описание действий | Ожидаемый результат |
| 1. Нажать ЛКМ по надписи «личный кабинет» в верхней правой части сайта; 2. нажать ЛКМ по надписи «Для сотрудников»; 3. в открывшейся форме авторизации ввести случайные значения логина и пароля; 4. нажать ЛКМ по кнопке «Авторизоваться»; 5. ввести случайные пять символов в поле для ввода логина; 6. нажать ЛКМ по кнопке «Авторизоваться»; 7. в форме авторизации ввести корректные значения логина и пароля; 8. нажать ЛКМ по кнопке «Авторизоваться». | Нажатия ЛКМ по кнопке «Авторизоваться» после ввода случайных значений должны сопровождаться соответствующими сообщениями об ошибке.  Нажатие ЛКМ по кнопке «Авторизоваться» после ввода корректных значений должно перенаправить пользователя на страницу для администрации. |

Таблица . – тест-кейс «Добавление нового пользователя»

|  |  |
| --- | --- |
| Описание действий | Ожидаемый результат |
| 1. Повторить действия 1, 2 и 7 из тест-кейса в таблице 2.3; 2. нажать ЛКМ по надписи «Настройка пользователей» на навигационной панели наверху. 3. в форме добавления пользователей ввести логин уже существующего пользователя (существующие пользователи отображены в виде списка выше формы добавления пользователей), произвольный пароль, отвечающий критериям, и выбрать любую предложенную роль из выпадающего списка; 4. нажать ЛКМ по кнопке «Добавить учётную запись»; 5. изменить логин на произвольный, отвечающий критериям; 6. нажать ЛКМ по кнопке «Добавить учётную запись». | Попытка добавления пользователя с уже существующим логином должна сопровождаться соответствующей ошибкой.  Добавление пользователя с корректными логином и паролем и выбранной ролью должно сопровождаться очищением формы и возникновением новой карточки в списке существующих учётных записей выше формы для добавления пользователей. |

По результатам прохождения тест-кейсов не было выявлено ошибок.

## Серверное приложение

Серверное приложение было создано на основе фреймворка «Express» на языке программирования «JavaScript». [11]

Перечень используемых зависимостей в приложении:

* «dotenv»: используется для вынесения параметров конфигурации для подключения к базе данных;
* «jsonwebtoken»: используется для реализации получения токенов доступа и проверки токенов доступа.
* «sequelize»: используется в качестве интерфейса для обращения к базе данных, что позволило мигрировать с postgreSQL на mySQL в середине реализации проекта с минимальными затратами по времени.

### Структура проекта

В папке «controllers» реализованы методы добавления, удаления, редактирования и получения данных из базы данных. Методы распределены между файлами с говорящими названиями. Например, вся логика относительно заказов определена в файле «Orders.js».

В папке «db» определены модели данных и объект подключения к базе данных. Для каждой сущности определение реализовано в отдельном файле. Синтаксис реализации прост и интуитивно понятен благодаря зависимости «sequelize».

В папке «middleware» определён middleware, проверяющий токен в заголовке запросов на наличие уровня доступа модератора, администратора или клиента. [12]

В папке «routes» определены маршрутизаторы, для перенаправления запросов на нужный метод из контроллеров. Как и в случае с контроллерами, каждый маршрутизатор покрывает запросы к определённой сущности из базы данных. Например, маршрутизатор «auth.js» перенаправляет на методы из контролера, связанного с авторизацией и проверкой токена. Условия перенаправления на эти маршрутизаторы определены в файле «app.js» в корне проекта.

### Особенности функционирования

В каждом токене зашифрованы данные по идентификатору пользователя и его роли. Идентификатор у администратора и модератора выражается числовым значением и соответствует id из базы данных, а у клиента соответствует адресу электронной почты.

Из базы данных не может быть удалён последний оставшийся аккаунт администратора.

Связи между сущностями базы данных определены в файле «db/index.js». Конфигурация для подключения к базе данных содержится в файле «.env».

На рисунке 2.4 представлен типовой запрос к базе данных. Благодаря «sequelize» синтаксис запросов не зависит от базы данных, что позволило в середине разработки курсового проекта перейти на базу данных mySQL без модифицирования запросов.

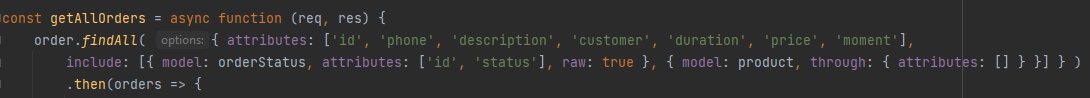


Рисунок . – Типовой запрос к базе данных

Поскольку внешние ключи в базе данных настроены, в запросах можно даже не указывать через какое поле совершается JOIN в запросе. В данном примере запрос возвращает список заказов с перечисленными атрибутами, статусом, взятом из другой таблицы, и списком услуг в заказе благодаря ключевому слову «through» без указания атрибутов.

### Тестирование

Для тестирования серверного приложения был привлечён метод автоматизированного тестирования на основе приложения «Postman». В данном приложении была составлена коллекция запросов и для каждого запроса своя функция тестирования и свой предварительный скрипт. Предварительный скрипт, например, может позволить вставить какой-либо заголовок в запрос, основываясь на данных, пришедших из запроса ранее. Таком образом перед отправкой запросов, требующих токена, например, запрос на удаление позиции из каталога, отправляется промежуточный запрос на получение этого самого токена через логин и хэш пароля. Также к каждому из запросов применяются общие тесты на проверку кода ответа и заголовки ответа. [9]

На рисунке 2.4 представлен отчёт об автоматическом тестировании серверного приложения. Файл для импорта тестов в приложение «Postman» предоставлен в архиве курсового проекта.

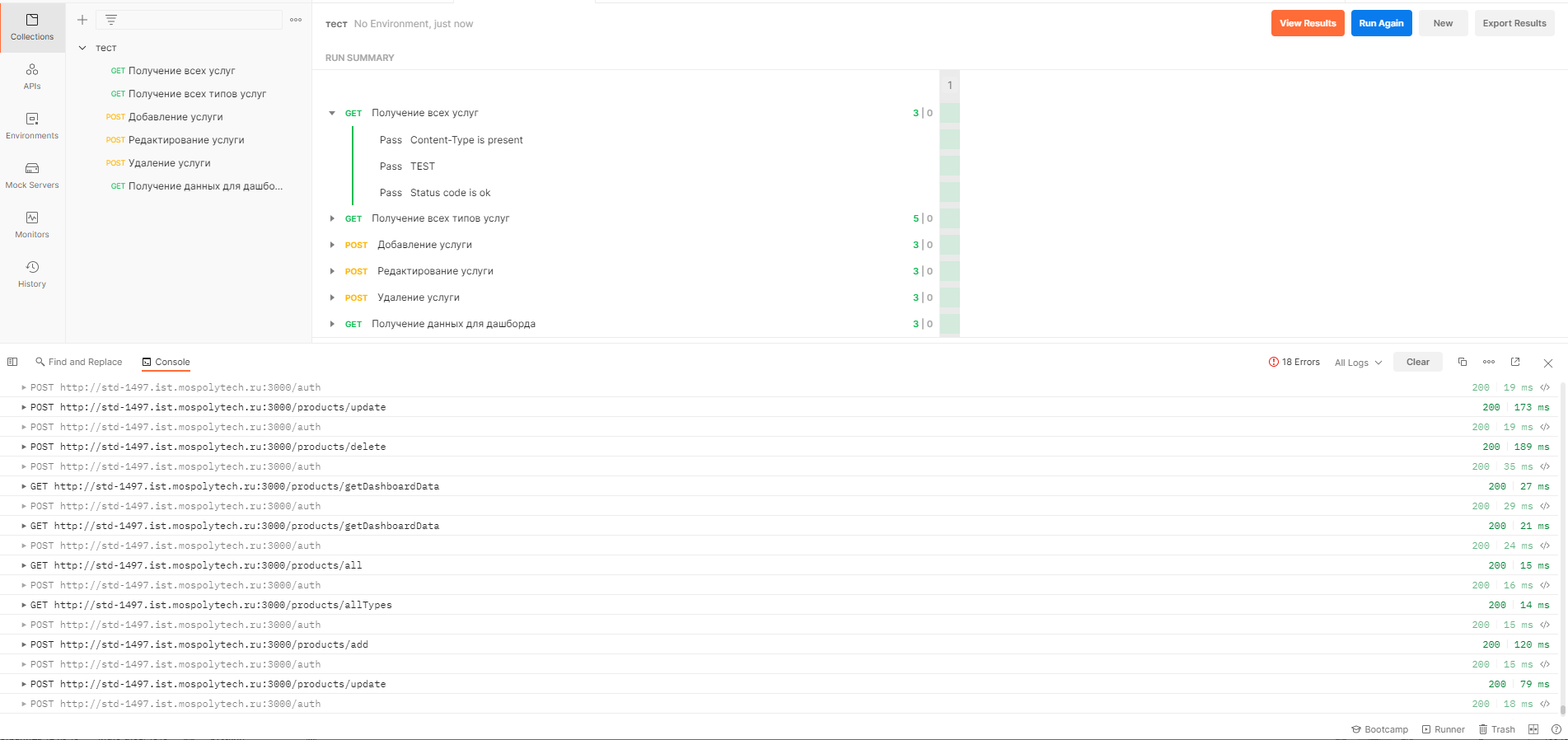


Рисунок . – Отчёт об автоматизированном тестировании серверного приложения

## Выводы по проектной части

На основе данной проектной части можно изложить следующие выводы:

* была представлена модель данных системы салона красоты — были расписаны три её аспекта: аспект манипуляции, структурный аспект и аспект целостности;
* было представлено подробное описание веб-приложения: краткое описание, структура проекта, реализация стилей, особенности функционирования и ручное тестирование в виде тест-кейсов;
* было представлено подробное описание серверного приложения: краткое описание, структура проекта, особенности функционирования и автоматическое тестирование при помощи приложения «Postman».

В рамках осеннего семестра были внесены следующие существенные изменения:

1. Полностью была переработана реализация работы модальных окон – ранее каждое модальное окно в веб-приложении представляло из себя отдельной компонент, лежащий в корне DOM-дерева. Теперь есть только один компонент модального окна и работает он при помощи слотов, а булево значение для его показа теперь храниться не в глобальном хранилище, а в компоненте, в котором импортирован и расположен компонент модального окна.
2. Похожее изменение коснулось навигационной панели в личных кабинетах пользователей. Ранее было несколько одинаковых компонентов, но теперь это один компонент, который принимает на вход массив названий страниц для маршрутизации.
3. Переработана форма для авторизации.
4. Переработано API для проверки прав доступа при заходе в личный кабинет.
5. Добавлен личный кабинет для клиента, в котором он может посмотреть историю своих заявок. Авторизация происходит через почту, которую клиент ранее указывал в заявке. Для этого внесены изменения в таблицу «Orders» в модели данных.
6. Вынесение однотипных функций в плагин.
7. Рефракторинг scss – замена метрики с «px» на «em» везде, кроме ширины границ блоков, и избавление от ненужных атрибутов.
8. Переписаны стили для «checkbox» и «range» в каталоге, чтобы был общий кросс-браузерный стиль.
9. Исправление ошибки, связанной с «cors» и настройка «eslint» в веб-приложении.
10. Отказ от модального окна для показа подробной информации из заявки на приём – вместо этого информация выводится в выпадающем блоке.

ЗАКЛЮЧЕНИЕ

В рамках осеннего семестра 2021/2022 учебного года была проведена доработка курсового проекта прошлого семестра. Данная пояснительная записка является доработанной версией пояснительной записки с прошлого семестра. Существенные изменения в проекте, осуществлённые в осеннем семестре, отображены в выводах по аналитической и проектной частях.

В аналитической части курсовой работы был проведён анализ аналогов разрабатываемой ИС, а именно frontend-части.

Также были представлены формальные модели, описывающие те или иные аспекты разрабатываемой системы: функциональная диаграмма в нотации IDEF0 с декомпозицией до первого уровня, диаграмма прецедентов и CJM.

В проектной части данной курсовой работы была представлена модель данных разрабатываемой ИС: были представлены концептуальная и физическая ER-диаграммы, а также описаны аспекты манипуляции, структуры и целостности модели данных.

Помимо этого, в проектной части было описано разработанное веб-приложение: было дано кратное описание с перечнем используемых зависимостей, описана структура проекта, описаны методика применения стилей и элементы дизайна, приведены некоторые особенности функционирования и произведено ручное тестирование по составленным тест-кейсам.

Также в проектной части было описано разработанное серверное приложение: было дано кратное описание с перечнем используемых зависимостей, описана структура проекта, приведены некоторые особенности функционирования и произведено автоматическое тестирование при помощи приложений «Postman».

СПИСОК ИСТОЧНИКОВ

1. БЭМ [Электронный ресурс]. – URL: https://ru.bem.info/ (дата обращения: 27.06.2021).

2. Р 50.1.028-2001 Информационные технологии поддержки жизненного цикла продукции. [Электронный ресурс]. – URL: https://gosthelp.ru/text/R5010282001Informacionnye.html (дата обращения: 27.06.2021).

3. Салон красоты «Силуэт» [Электронный ресурс]. – URL: http://siluet-salon.ru/ (дата обращения: 18.01.2022).

4. Салон красоты «Lux» [Электронный ресурс]. – URL: https://centr-lux.ru/ (дата обращения: 18.01.2022).

5. Что такое API? Простое объяснение для начинающих [Электронный ресурс]. – URL: https://dev.by/news/chto-takoe-api-prostym-yazykom (дата обращения: 01.12.2020).

6. Beauty Salon Lakelands, Stillorgan | Beautician Lakelands, Stillorgan [Электронный ресурс]. – URL: https://www.beautifysalon.ie/ (дата обращения: 27.06.2021).

7. ER-диаграмма (ERD): определение и обзор | Lucidchart [Электронный ресурс]. – URL: https://www.lucidchart.com/pages/ru/erd-диаграмма (дата обращения: 27.06.2021).

8. Front курсового проекта [Электронный ресурс]. – URL: http://kozlov.std-1497.ist.mospolytech.ru/ (дата обращения: 28.06.2021).

9. Postman | The Collaboration Platform for API Development [Электронный ресурс]. – URL: https://www.postman.com/ (дата обращения: 27.06.2021).

10. Vue.js [Электронный ресурс]. – URL: https://vuejs.org/ (дата обращения: 27.06.2021).

11. Express – фреймворк веб-приложений Node.js [Электронный ресурс]. – URL: https://expressjs.com/ru/ (дата обращения: 31.03.2021).

12. Использование промежуточных обработчиков Express [Электронный ресурс]. – URL: https://expressjs.com/ru/guide/using-middleware.html (дата обращения: 27.06.2021).

13. Солим пароли [Электронный ресурс]. – URL: https://habr.com/ru/post/145648/ (дата обращения: 27.06.2021).

14. Гридасов А. Что такое дашборд: примеры и способы применения [Электронный ресурс]. – URL: https://blog.calltouch.ru/dashbord-chto-eto-takoe-dlya-chego-nuzhen-dashboard-i-kak-ego-ispolzovat/ (дата обращения: 27.06.2021).

15. Диаграмма прецедентов (вариантов использования) UML | Планёрка [Электронный ресурс]. – URL: https://planerka.info/item/diagramma-precedentov-variantov-ispolzovaniya-uml/ (дата обращения: 27.06.2021).

16. AllFusion Process Modeler (Bpwin) — ERwin Process Modeler [Электронный ресурс]. – URL: https://businessarchitecture.ru/bwin/ (дата обращения: 27.06.2021).