№15
23.01.2024г.

Министерство науки и высшего образования Российской Федерации
ФБГОУ ВО «Кубанский государственный технологический университет»
(ФГБОУ ВО «КубГТУ»)

Институт компьютерных систем и информационной безопасности
Кафедра информатики и вычислительной техники
Направление подготовки 09.03.04 программная инженерия
Профиль проектирование и разработка программного обеспечения

КУРСОВОЙ ПРОЕКТ

по дисциплине «Технологии разработки программного обеспечения»
на тему: «Разработка программного обеспечения интернет-магазина»
Выполнил студент Заболотнов Дмитрий Алексеевич
курса 3 группы 21-ЗКБ-ПР1
Допущен к защите 25.12.23г.
Руководитель (нормоконтролёр) проекта _____ Ковалева К.А.
Защищен 23.01.24г. Оценка 5 (отл)

Члены комиссии _____ доц. Мурлин А.Г.
_____ доц. Тотухов К. Е
_____

Краснодар
2024 г.

ФБГОУ ВО «Кубанский государственный технологический университет»
(ФГБОУ ВО «КубГТУ»)

Институт компьютерных систем и информационной безопасности
Кафедра информатики и вычислительной техники
Направление подготовки 09.03.04 программная инженерия
Профиль проектирование и разработка программного обеспечения

УТВЕРЖДАЮ
Зав. кафедрой ИСП
доц. М.В. Янаева
28.09 2023 г.

ЗАДАНИЕ
на курсовой проект

Студенту Заболотнову Дмитрию Алексеевичу
курса 3 группы 21-ЗКБ-ПР1
Тема проекта: Разработка программного обеспечения интернет-магазина
(утверждена указанием директора института № 65 (от 28.09.23)
План проекта:
1. Описание предметной области
2. Проектирование
3. Реализация

Объем проекта:
А) пояснительная записка 61 с.
Б) Программа
В) Приложения
Рекомендуемая литература:
1. Гагарина Л.Г., Кокорева Е.В., Сидорова-Виснадул Б.Д. Технология разработки программного обеспе-чения: учеб. пособие [Электронный ресурс]. – М.: ИНФРА-М, 2019. – 400 с. Режим доступа: http://znanium.com/catalog/product/1011120.
2. Зубкова, Т. М. Технология разработки программного обеспечения : учебное пособие / Т. М. Зубкова. — Санкт-Петербург : Лань, 2022. — 324 с. — ISBN 978-5-8114-3842-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/206882 (дата обращения: 10.12.2023). — Режим доступа: для авториз. пользователей.

Срок выполнения работы: с "28" 09.23 по "05" 02 2024 г.
Срок защиты: "____" ____ 2024 г.
Дата выдачи задания: "____" ____ 2024 г.
Дата сдачи проекта на кафедру: "____" ____ 2024 г.

Руководитель проекта _____ Ковалева К.А.
Задание принял студент _____ Заболотнов Д. А.

Реферат

Курсовая работа содержит — 61 страница, 29 рисунков, 1 таблица, 25 схем, 13 источников, 10 приложения.

ИНТЕРНЕТ-МАГАЗИНЫ, САЙТЫ, WEB-ПРОГРАММИРОВАНИЕ, СЕРВЕР, ПРОЕКТИРОВАНИЕ, ДИЗАЙН, БАЗЫ ДАННЫХ, FIGMA, PHOTOSHOP, JAVASCRIPT, REACT, PHP, MYSQL, PHPMYADMIN, APACHE.

Цель курсового проекта состоит в изучении технологий разработки и реализации интернет-магазина в виде веб-приложения и получении практических навыков в этой области знаний на конкретном примере. Данный интернет-магазин позволяет покупателям удобно и быстро просматривать ассортимент магазина, а также оформлять заказы и их доставку. В рамках курсового проекта были рассмотрены различные подходы к разработке интернет-магазина, изучены имеющиеся сегодня инструменты построения диаграмм, так же корректно выполнен каждый этап жизненного цикла разработки интернет-магазина.

При выполнении курсового проекта были закреплены основы и углублённые знания в технологиях разработки интернет-магазина через практическую реализацию на индивидуальном примере. При работе над курсовым проектом были самостоятельно выполнены все этапы создания продукта, от постановки задачи до практической реализации, заканчивающейся верификацией и подготовкой отчетности. В процессе выполнения работы были получены навыки самостоятельного использования специальной литературы, справочников, стандартов.

Реализация была выполнена на языке JAVASCRIPT с использование библиотеки React, PHP с библиотекой Composer, MySql в качестве языка запросов и PHPmyAdmin для удобной работы с базой данных. Для проектирования программы использовались Process Modeller и Microsoft Visio.

Произведен анализ предметной области, разработана структура сайта и дизайн. Описана внутренняя структура движка сайта как для клиентской, так и для серверной части. Была написана тестовая база данных. Возможности интернет-магазина и как им пользоваться представлены в руководстве пользователя.

Содержание

Введение 4
1 Техническое задание 5
2 Планирование проекта 6
3 Функциональное проектирование 7
3.1 Функциональная схема 7
3.2 Моделирование IDEF0 8
3.1 Моделирование IDEF3 22
3.2 Моделирование IDEF1X 27
4 Объектно-ориентированное проектирование информационной системы ... 29
5 Проектирование и дизайн интернет-магазина 31
5.1 Построение структуры сайта 31
5.2 Дизайн страниц 32
6 Разработка интернет-магазина 34
6.1 Написание клиентской части 34
6.1.1 Клиентская маршрутизация 34
6.1.2 Статичная верстка страниц 35
6.1.2 Написание запросов к серверу и логики сайта 43
6.2 Разработка базы данных 45
6.2.1 Проектирование базы данных 45
6.2.2 Связи таблиц 46
6.3 Серверная часть 47
6.3.1 Создание базовой структуры сервера 47
6.3.2 Написание серверной маршрутизации 48
6.3.3 Написание логики обработки запросов с клиентской части 48
6.3.4 Написание запросов к базе данных и реализация сессий 49
6.3.5 Настройка серверной безопасности 55
6.4 Тестирование 56
Заключение 58
Список используемых источников 59

Введение

Интернет-магазин — интернет-программа, позволяющая пользователям онлайн в своём браузере, просматривать ассортимент магазина, сформировать заказ на покупку, выбрать способ оплаты и доставки заказа.

Для разработки дизайна и макета сайта выбран онлайн-сервис для разработки интерфейсов Figma, Photoshop и ряд других технологий. Для написания клиентской части интернет-магазин используется браузерный язык JavaScipt в сочетании с библиотекой React. За работу сервера и обработку входящих запросов будет отвечать серверный язык программирования PHP с использованием веб-сервера Apache. Для разработки базы данных выбран язык запросов Sql, его диалект MySql с применением приложения для администрирования баз данных PHPMySql.

Задачи:

1. Смоделировать макет сайта, его структуру;
2. Дизайн сайта;
3. Клиентская часть;
4. Серверная часть и база данных;
5. Различные тесты готового продукта.

1 Техническое задание

Интернет-магазин — интернет программа с общим доступом, позволяющая пользователям онлайн просматривать товары и оформлять товары.

Основная цель этого интернет-ресурса — увеличения охвата продаж и комфортность продажи своего продукта.

Разрабатываемая программа должна обеспечить возможность выполнения перечисленных ниже функций:

1. Удобный просмотр ассортимента;
2. Просмотр ассортимента с выбранными фильтрами, сортировками и категориями;
3. Предоставлять общую информацию об интернет-магазине;
4. Предоставлять пользователю возможность регистрироваться на сайте;
5. Предоставлять пользователю иметь свой личный кабинет;
6. Предоставлять пользователю вести учет своим заказам;
7. Предоставлять пользователю возможность выбирать избранные товары;
8. Программа должна корректно отображать все данные, которые были заложены в базу данных.

2 Планирование проекта

Перед началом любой разработки нужно спланировать проект. С использованием программы Microsoft Project Document были отражены основные этапы разработки программного продукта.

Режи зада	Название задачи	Длительность	Начало	Окончание	Названия ресурсов	Трудозатраты
	Интернет-магазин	**27 дней**	**Вс 01.10.23**	**Пт 27.10.23**		**176 часов**
	Этап 1	**4 дней**	**Вс 01.10.23**	**Ср 04.10.23**		**32 часов**
	Разработка дизайна сайта	**4 дней**	**Вс 01.10.23**	**Ср 04.10.23**		**32 часов**
	Разработка логотипа и фонов с помощью нейросетей и фотошопа	1 день	Пн 02.10.23	Пн 02.10.23	Заболотнов Д. А.	8 часов
	Дизайн отдельных блоков	3 дней	Пн 02.10.23	Ср 04.10.23	Заболотнов Д. А.	24 часов
	Этап 2	**21 дней**	**Пт 06.10.23**	**Чт 26.10.23**		**136 часов**
	Написание клиентской части	**12 дней**	**Пт 06.10.23**	**Вт 17.10.23**		**72 часов**
	Разработка клиентской маршрутизации	1 день	Пт 06.10.23	Пт 06.10.23	Заболотнов Д. А.	8 часов
	Статичная верстка всех страниц	4 дней	Сб 07.10.23	Вт 10.10.23	Заболотнов Д. А.	24 часов
	Написание логики	**7 дней**	**Ср 11.10.23**	**Вт 17.10.23**		**40 часов**
	Написание логики авторизации пользователя	1 день	Ср 11.10.23	Ср 11.10.23	Заболотнов Д. А.	8 часов
	Написание запросов к серверу	3 дней	Пт 13.10.23	Вс 15.10.23	Заболотнов Д. А.	16 часов
	Оживление активных элементов - слайдеры, подгрузки, фильтры, спинеры. Добавление оставшейся анимации	2 дней	Пн 16.10.23	Вт 17.10.23	Заболотнов Д. А.	16 часов
	Разработка базы данных	**2 дней**	**Ср 18.10.23**	**Чт 19.10.23**		**16 часов**
	Создание представление базы, проектирование таблиц и их данных	1 день	Ср 18.10.23	Ср 18.10.23	Заболотнов Д. А.	8 часов
	Разработка базы данных - создание таблиц, связывание таблиц, прочие триггеры	1 день	Чт 19.10.23	Чт 19.10.23	Заболотнов Д. А.	8 часов
	Разработка серверной части	**6 дней**	**Сб 21.10.23**	**Чт 26.10.23**		**48 часов**
	Написание серверной маршрутизация	1 день	Сб 21.10.23	Сб 21.10.23	Заболотнов Д. А.	8 часов
	Написание логики обработки запросов с клиентской части	2 дней	Вс 22.10.23	Пн 23.10.23	Заболотнов Д. А.	16 часов
	Написание запросов к базе данных	2 дней	Вт 24.10.23	Ср 25.10.23	Заболотнов Д. А.	16 часов
	Серверная безопасность	1 день	Чт 26.10.23	Чт 26.10.23	Заболотнов Д. А.	8 часов
	Этап 3	**1 день**	**Пт 27.10.23**	**Пт 27.10.23**		**8 часов**
	Тестирование	**1 день**	**Пт 27.11.09**	**Пт 27.11.09**		**8 часов**
	Тестирование клиентской части	0,5 дней	Пт 27.11.09	Пт 27.11.09	Заболотнов Д. А.	4 часов
	Тестирование серверной части	0,5 дней	Пт 27.11.09	Пт 27.11.09	Заболотнов Д. А.	4 часов

Таблица 1 — Таблица общего плана проекта

3 Функциональное проектирование

3.1 Функциональная схема

Функциональная схема или схема данных (ГОСТ 19.701-90) – схема взаимодействия компонентов программного обеспечения с описанием информационных потоков, состава данных в потоках и указанием используемых файлов и устройств. Для изображения функциональных схем используют специальные обозначения, установленные стандартом.

Функциональные схемы, более информативны, чем структурные. Все компоненты структурных и функциональных схем должны быть описаны. При структурном подходе особенно тщательно необходимо прорабатывать спецификации межпрограммных интерфейсов, так как от качества их описания зависит количество самых дорогостоящих ошибок. К самым дорогим относятся ошибки, обнаруживаемые при комплексном тестировании, так как для их устранения могут потребоваться серьезные изменения уже отлаженных текстов.

Функциональная схема главного действия интернет-магазина — оформление заказа, выполнена с помощью программы Microsoft Visio и представлена ниже.

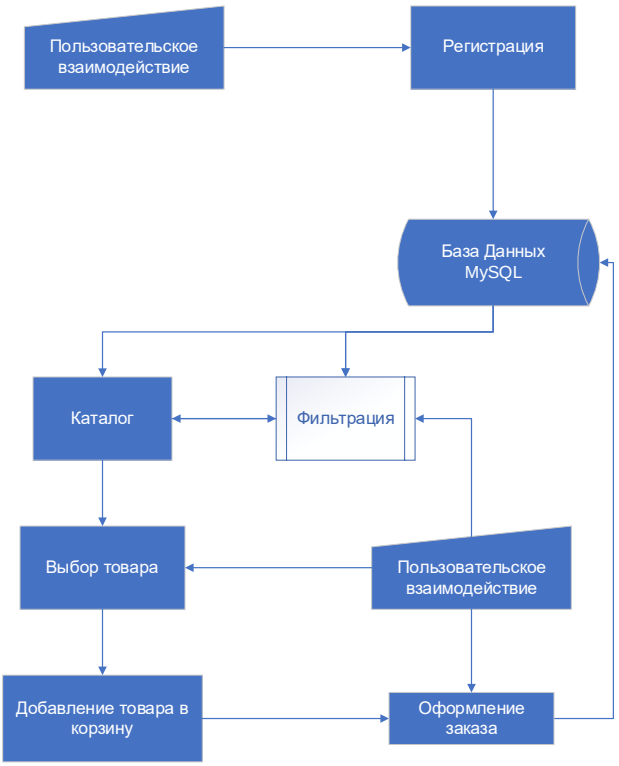


Схема 1 — Функциональная схема оформления заказа

3.2 Моделирование IDEF0

IDEF0 – это графическая нотация, предназначенная для описания бизнес-процессов. Система, описываемая в данной нотации, проходит через декомпозицию или, иными словами, разбиение на взаимосвязанные функции. Для каждой функции существует правило сторон:

– стрелкой слева обозначаются входные данные;

– стрелкой сверху – управление;

– стрелкой справа – выходные данные;

– стрелкой снизу – внешние воздействия.

Когда бизнес-аналитик выявил входные и выходные данные, установил механизм и способы управления, можно свести всю эту информацию в первую диаграмму, называемую контекстной диаграммой. Схема — 2 Контекстная диаграмма главного предназначения интернет-магазина — полный путь заказа товаров.

Данные управления схемы:

– Браузер клиента — это инструмент клиента для связи с интернет-магазином;

– Протокол HTTP/S — коридор, с помощью которого клиент будет обмениваться данными с сервером;

– Веб сервер Apache — находится на стороне сервера и служит для обработки всех приходящих данных;

– MySQL — база данных, где будет хранится вся и обо всем информация.

Данные воздействий:

– Пользователь — непосредственно тот, ради которого это все задумано;

– Администратор — человек отвечающий за подтверждение заказов, заполнением сайта контентом и поддержки клиентов;

– PHP — язык программирования на стороне сервера, служит для связи клиента с сервером и базой данных;

– JavaScript — язык программирования на стороне клиента, служит для показа всего контента пользователю и отправки запросов на сервер.

Контекстная диаграмма не даёт полного видения процесса, а лишь общий взгляд. Для того, чтобы просмотреть последовательность выполнения процесса, нужно декомпозировать диаграмму, т.е. дать чуть более детальное описание процесса. Контекстная диаграмма будет декомпозирована на 4 процесса:

– Регистрация;

– Выбор товаров;

– Оформление заказа;

– Администрирование.

Далее все процессы будут тоже декомпозированы:

Регистрация: 1. Войти; 2. Зарегистрироваться; 3. Проверка существования пользователя.

Выбор товаров: 1. Каталог; 2. Фильтрация; 3. Выборка товаров; 4. Корзина.

Оформление заказа: 1. Корзина — кнопка оформить заказ; 2. Форма оформления заказа; 3. Создание заказа в базе данных.

Администрирование: 1. Проверка заказа; 2. Распределение; 3. Смена статуса.

Для построения функциональных диаграмм IDEF0 был использован программный продукт Process Modeller позволяющий проводить описание, анализ и моделирование бизнес-процессов.

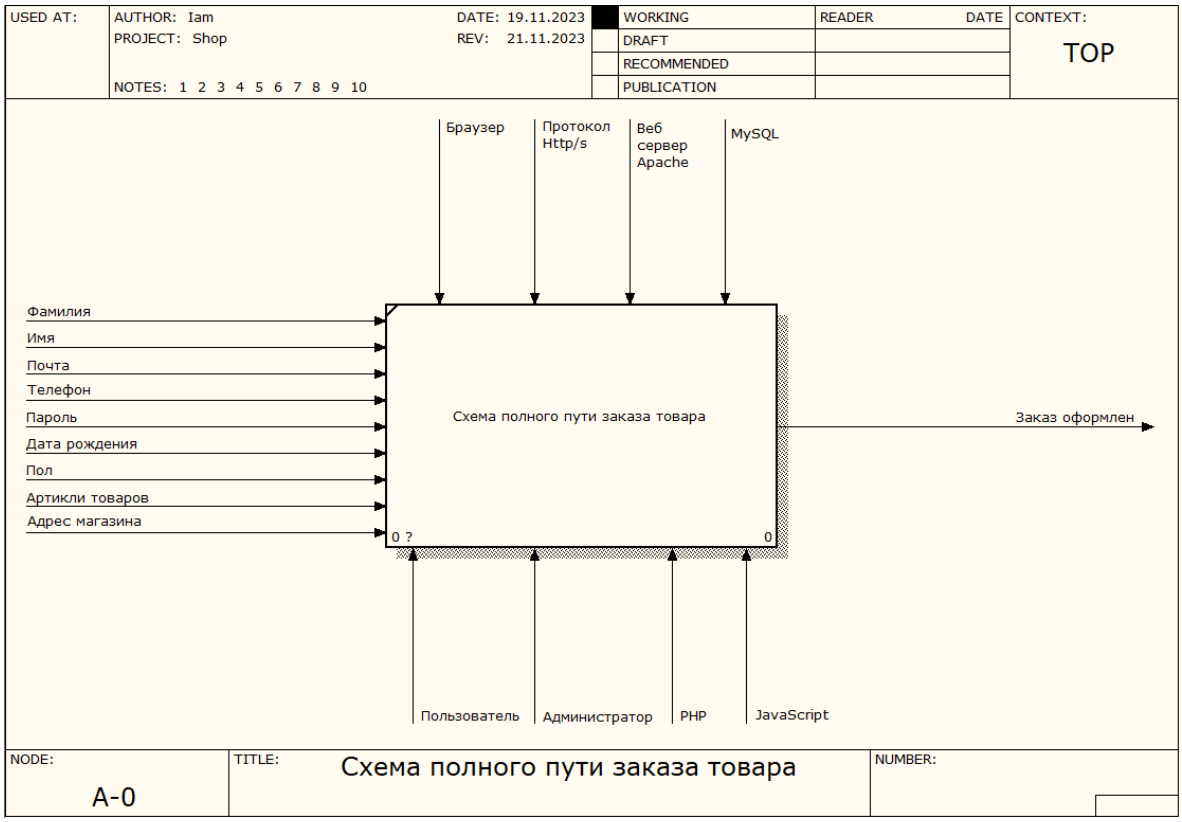


Схема 2 — Контекстная диаграмма IDEF0 полный путь заказа товара

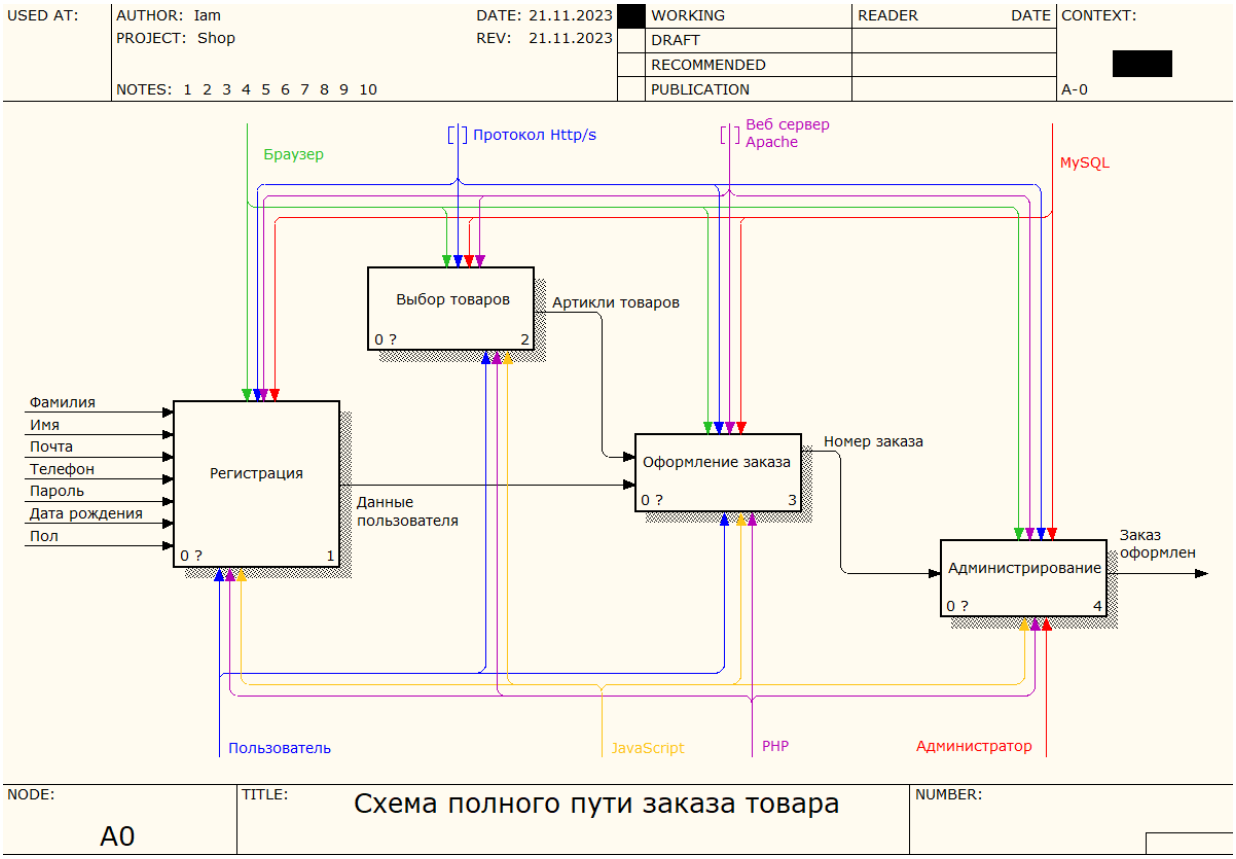


Схема 3 — Декомпозиция диаграммы полного пути заказа товара

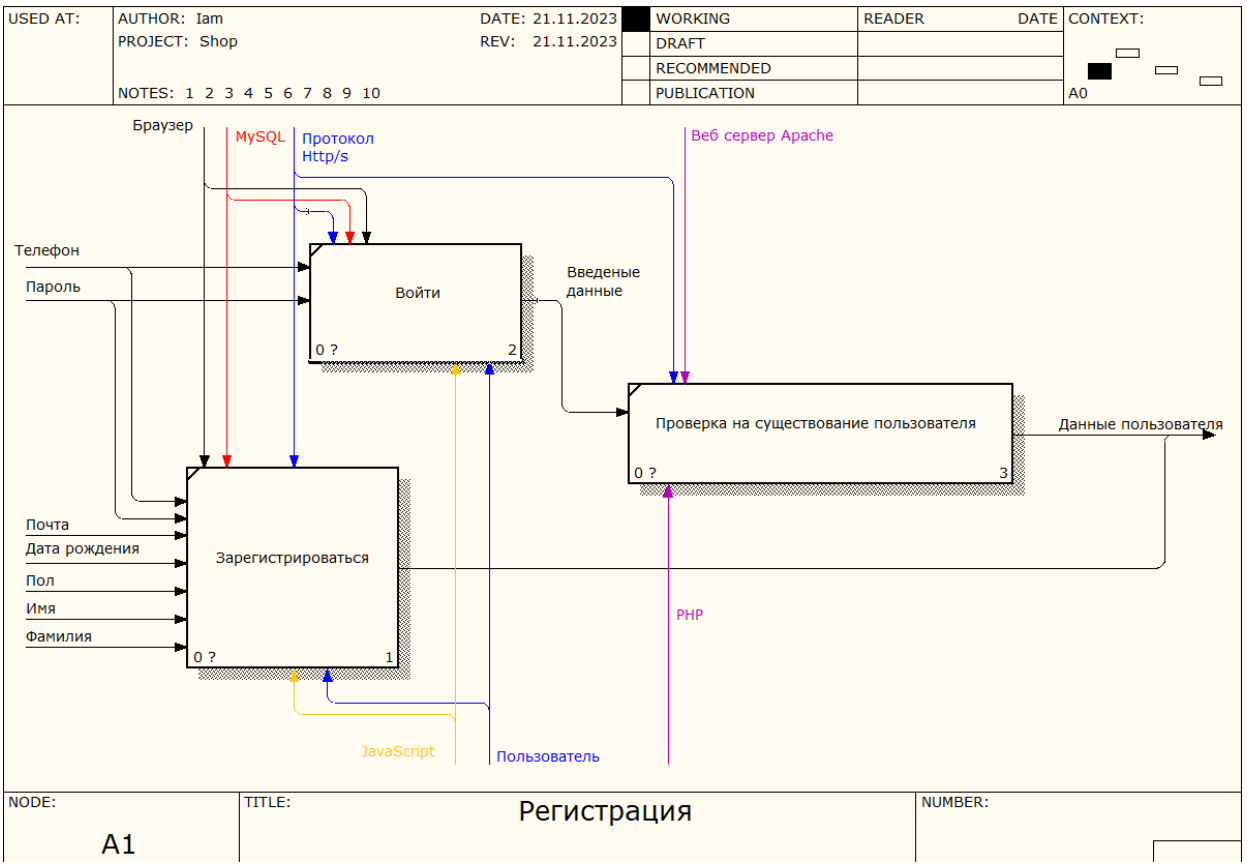


Схема 4 — Декомпозиция блока Регистрация

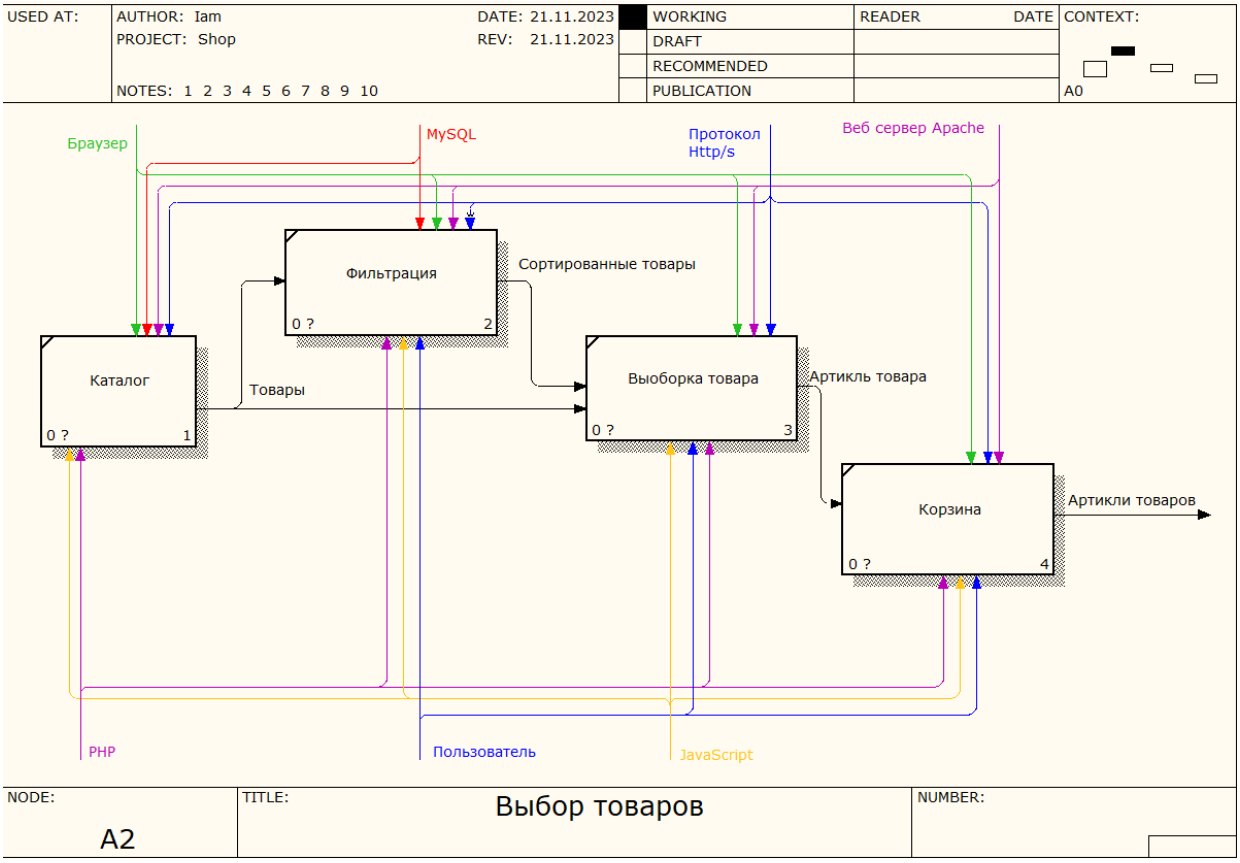


Схема 5 — Декомпозиция блока Выбор товаров

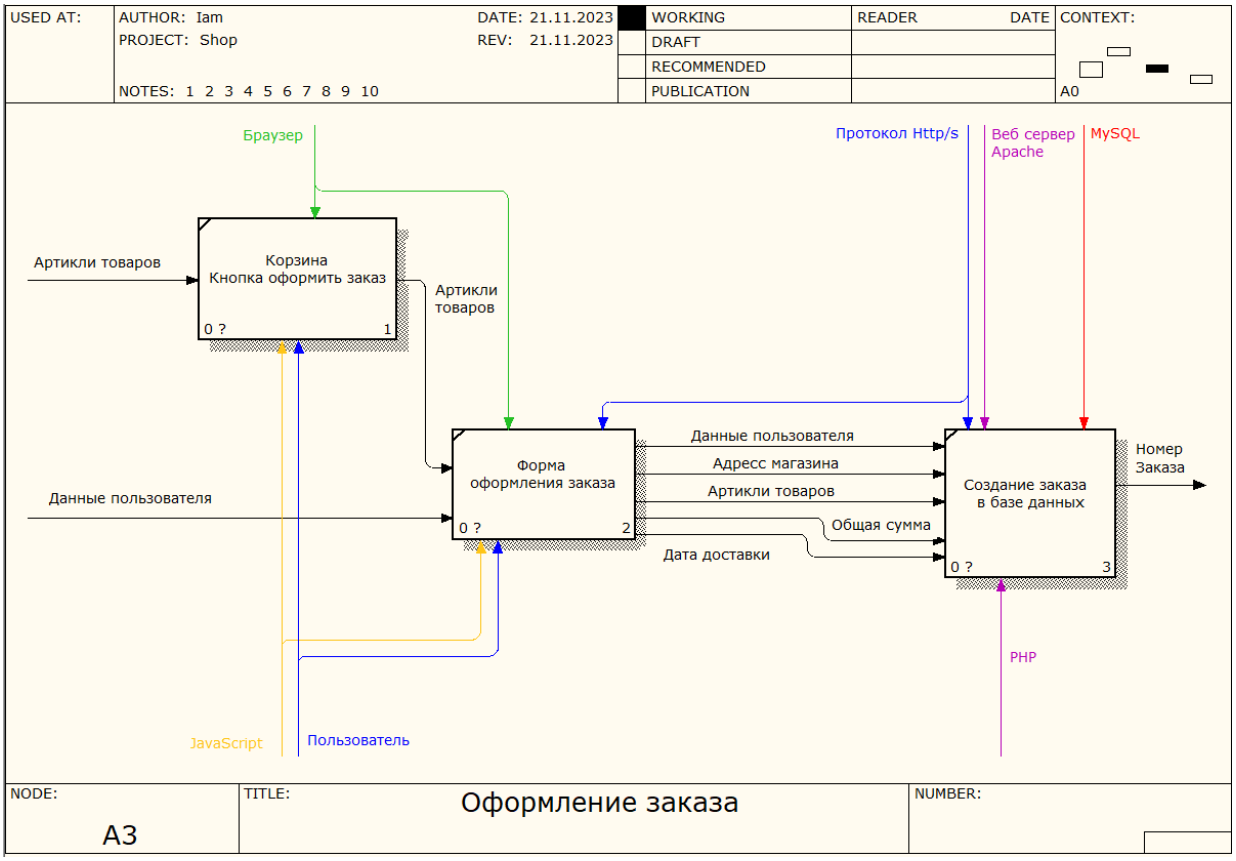


Схема 6 — Декомпозиция блока Оформление заказа

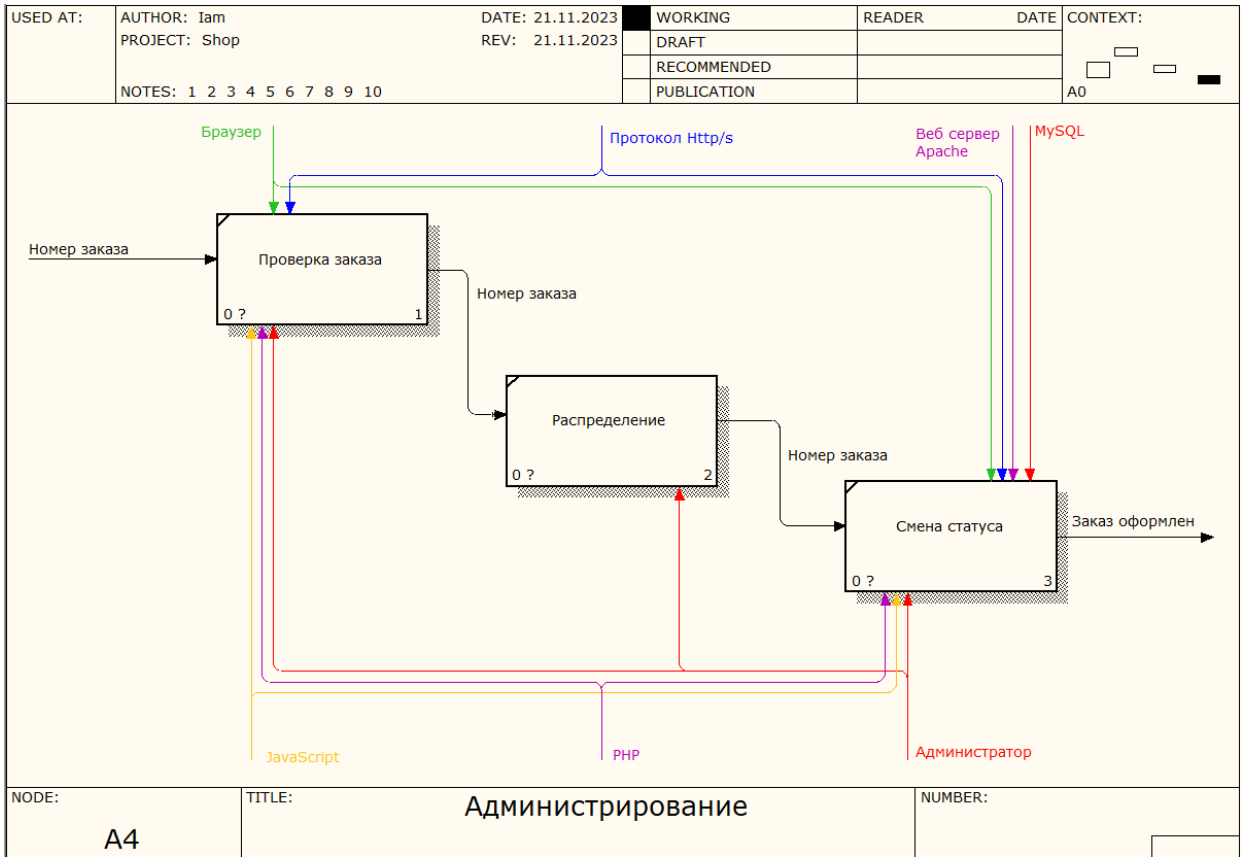


Схема 7 — Декомпозиция блока Администрирование

Вторая IDEF0 диаграмма будет об разработке интернет-магазина:

Управление производится стандартами разработки и требований к программному продукту;

Внешние воздействия оказывают программист и дизайнер.

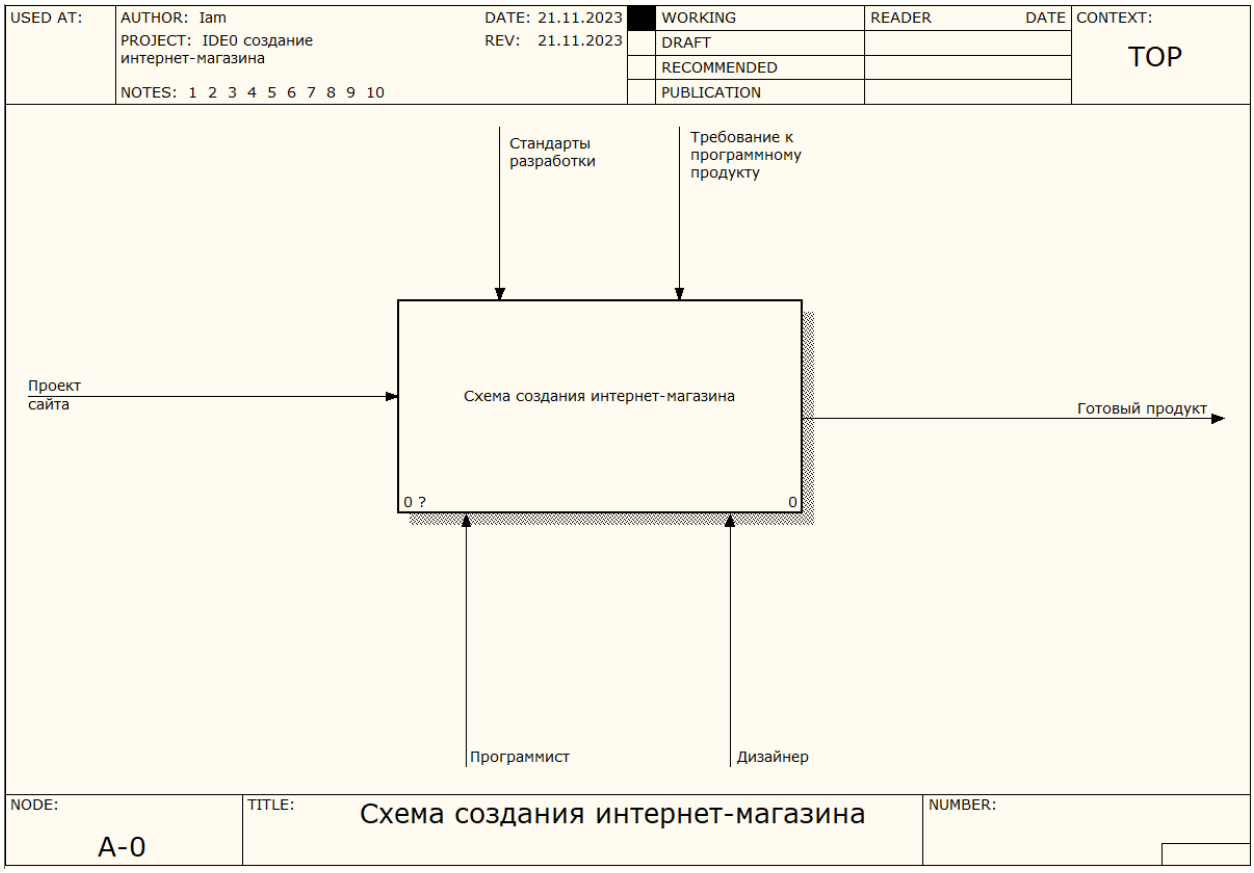


Схема 8 — Диаграмма IDEF0 создание интернет-магазина

Декомпозируется диаграмма на 4 уровня со своими подуровнями.

Разработка дизайна сайта — занимается этим дизайнер, входные данные — проект сайта. Управление производится стандартами разработками и требованиями к программному продукту.

Разработка сайта — программист действующее лицо, принимает готовый макет сайта. Управление так же производится стандартами разработками и требованиями к программному продукту.

Тестирование — тестирование проводит программист без всяких управлений.

Размещение сайта на хостинге — проводит программист с требованием к программному продукту.

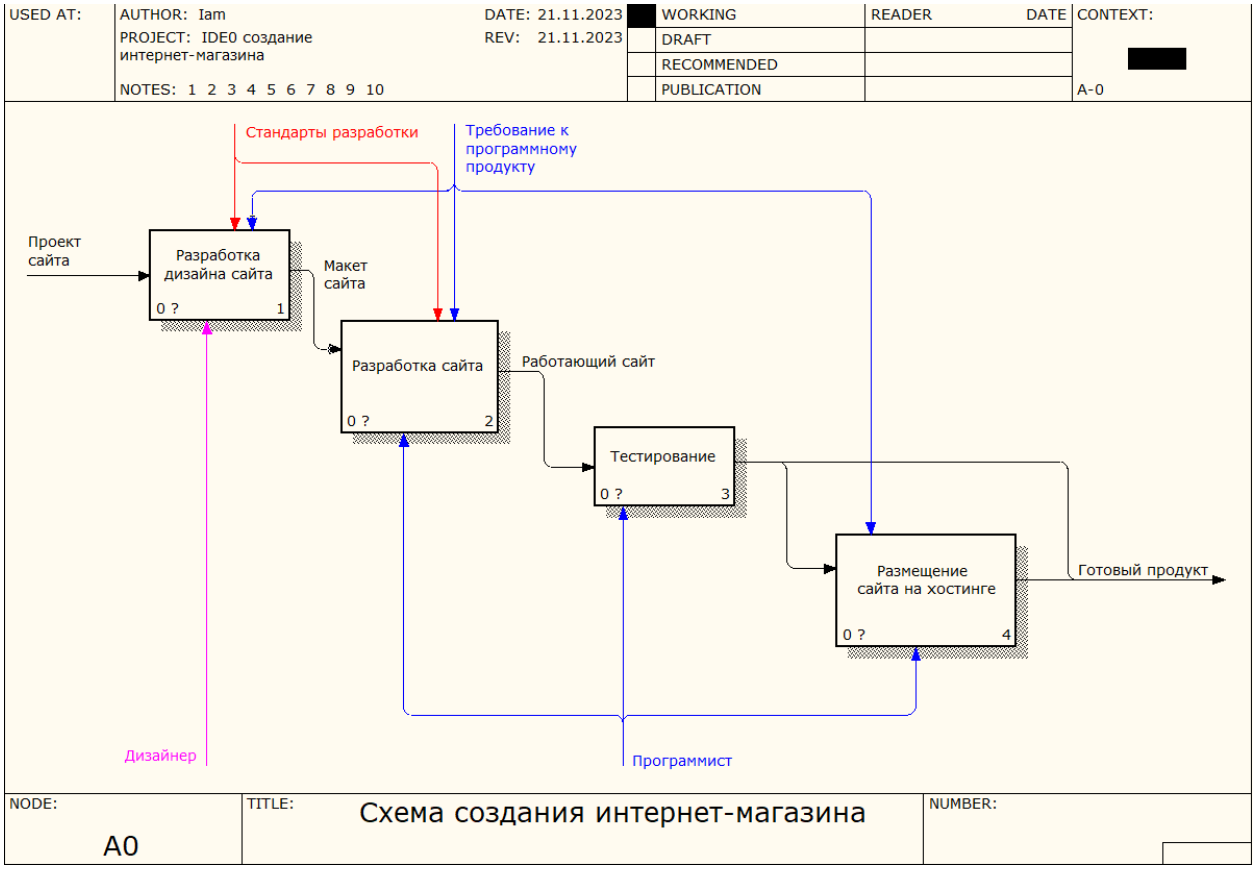


Схема 9 — Декомпозиция диаграммы создания интернет-магазина

Разработка дизайна сайта декомпозируется на следующие уровни (все уровни управляются требованием к программному продукту):

Предварительный дизайн — принимает в себя проект сайта, внешние воздействия — это дизайнер и Photoshop;

Разработка логотипов и фонов — принимает полученный набросок сайта, и разрабатывается дизайнером с помощью нейросетей и Photoshop;

Дизайн отдельных блоков — принимает готовые медиа файлы, разработку введет дизайнер с помощью интернет-сервиса Figma, в дополнении со стандартами разработки.

Дизайн всех страниц — принимает готовые некоторые отдельные блоки, разрабатывает дизайнер при помощи Figma и стандартами разработки.

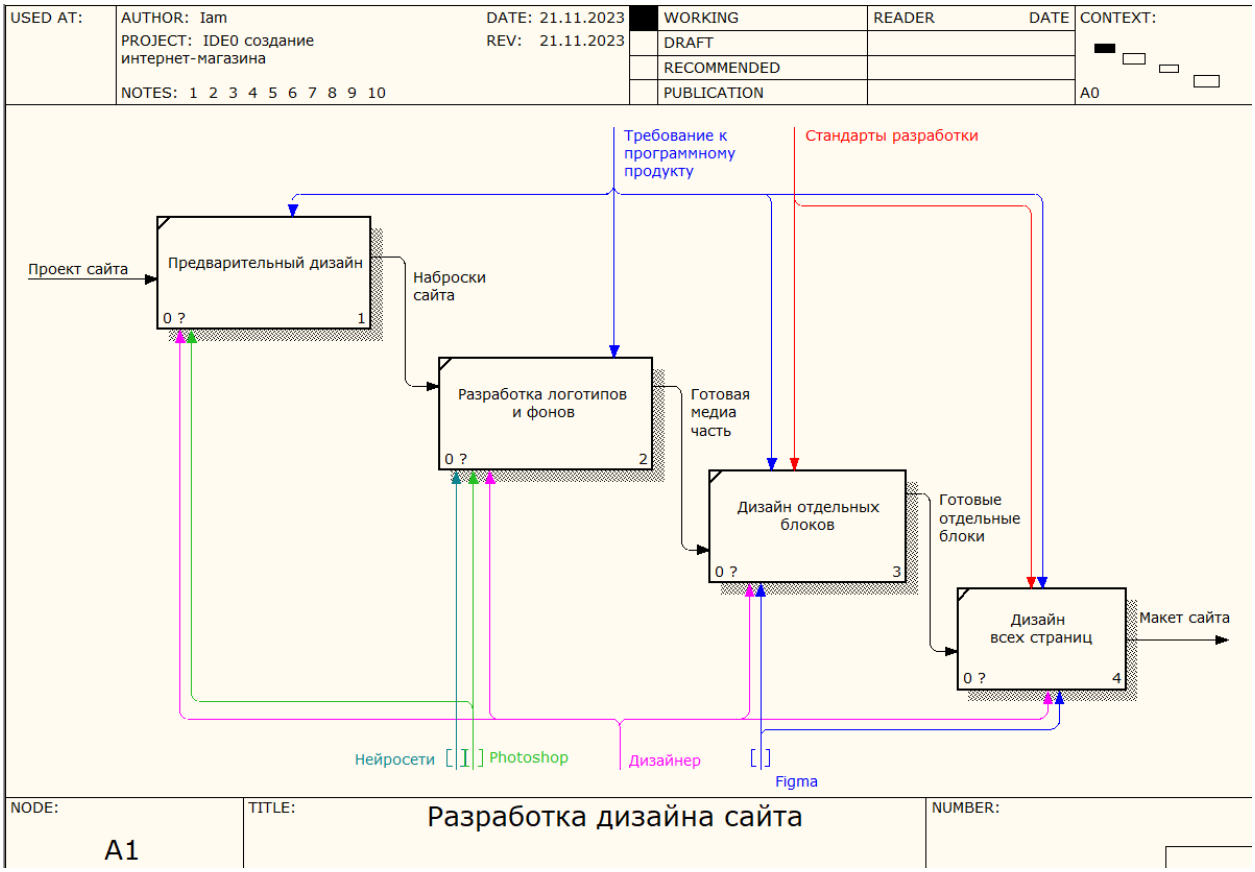


Схема 10— Декомпозиция уровня 1 разработки дизайна сайта

Дальше большая декомпозиция уровня 2 разработка сайта:

Первый модуль — это написание клиентской части, принимает макет сайта, управление производится стандартами разработки и требованием к программному продукту, внешние воздействия — программист, браузер, редактор кода, JavaScript.

Разработка базы данных производится программистом с помощь. MySQL, стандартами разработки и требованиями к программному продукту.

Написание серверной части последний модуль этого уровня, выполняет так же программист при помощи серверного языка PHP, веб-сервера Apache, браузера, MySQL и редактора кода под управлением стандартами разработки и требований к программному продукту.

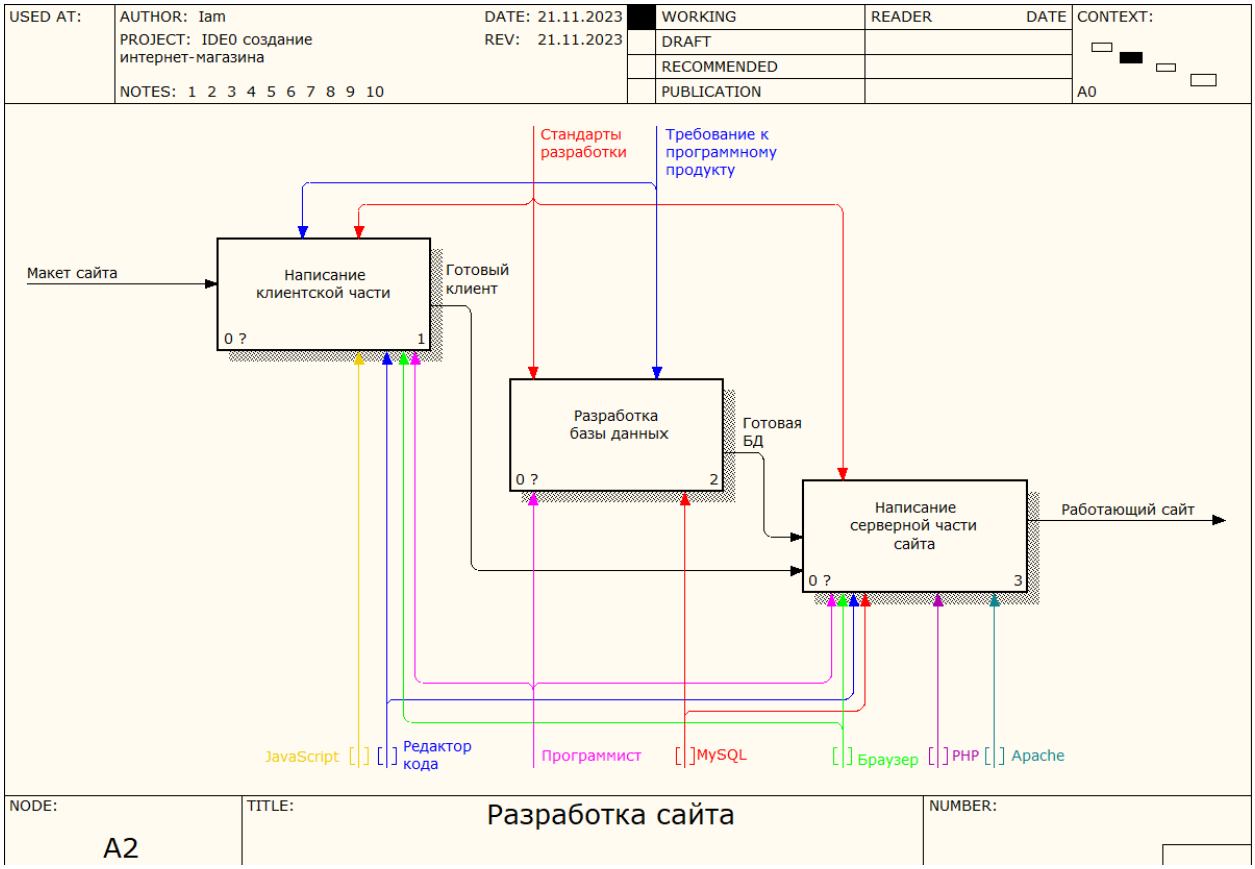


Схема 11 — Декомпозиция уровня 2 разработки сайта

Далее каждый блок этого уровня имеет свою декомпозицию:

Клиентская часть, уровень 2.1 имеет 4 модуля:

Все блоки имеют одни и те же факторы внешнего воздействия — программист, JavaScript, браузер и редактор кода. А вот по управлению они делятся на две группы:

Требование к программному продукту и стандарты разработки:

– Статичная верстка;

– Добавление оставшейся анимации и доработка деталей.

Только стандарты разработки:

– Клиентская маршрутизация;

– Написание запросов к серверу;

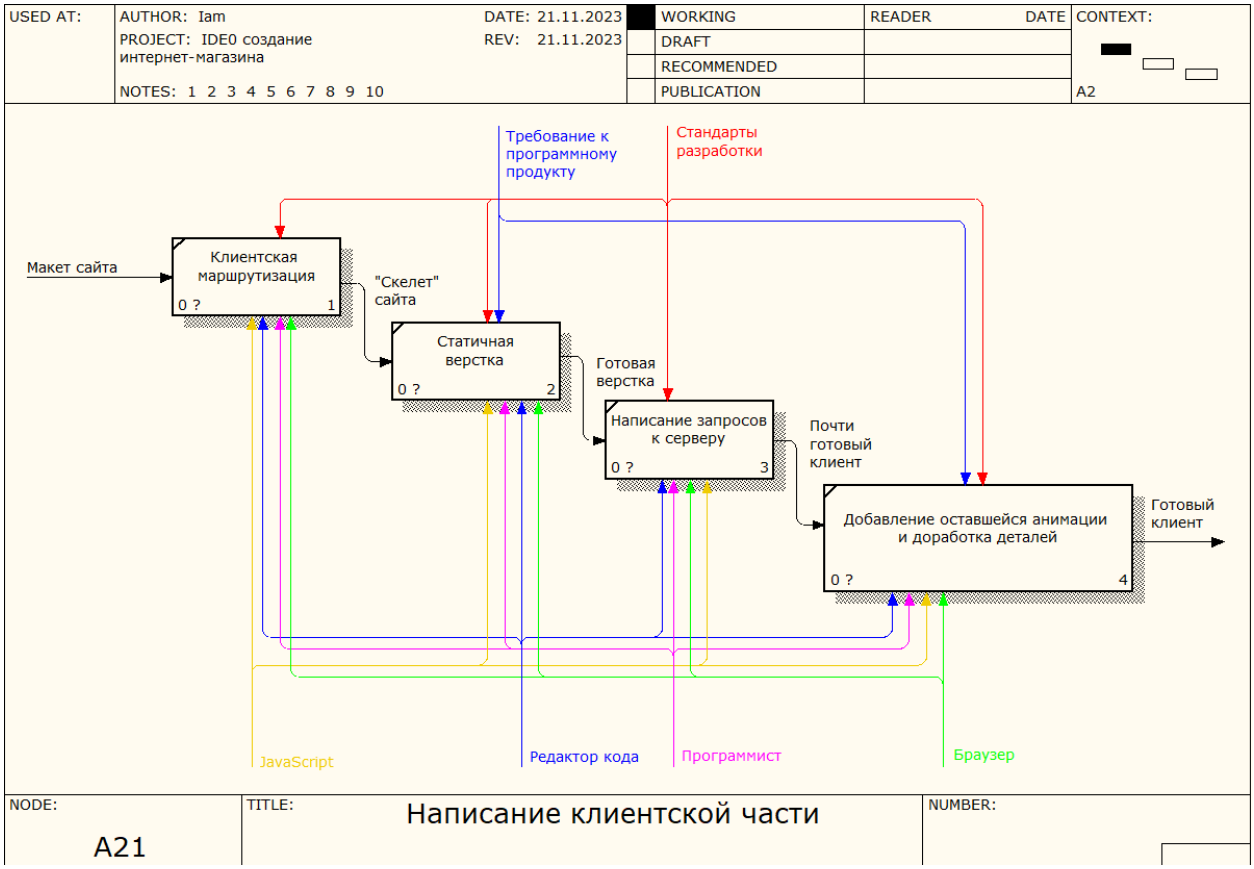


Схема 12 — Декомпозиция уровня 2.1 Клиентская часть

Разработка базы данных, уровень 2.2, имеет 3 модуля:

Создание представления базы данных — производит программист в соответствии с требованием к продукту.

Создание таблиц и расположения данных в них, производится по готовому макету БД программистом с помощью MySQL и стандартами разработки.

И разработка логических связей между таблицами и добавление триггеров производится эквивалентно предыдущему. На выходе готовая БД.

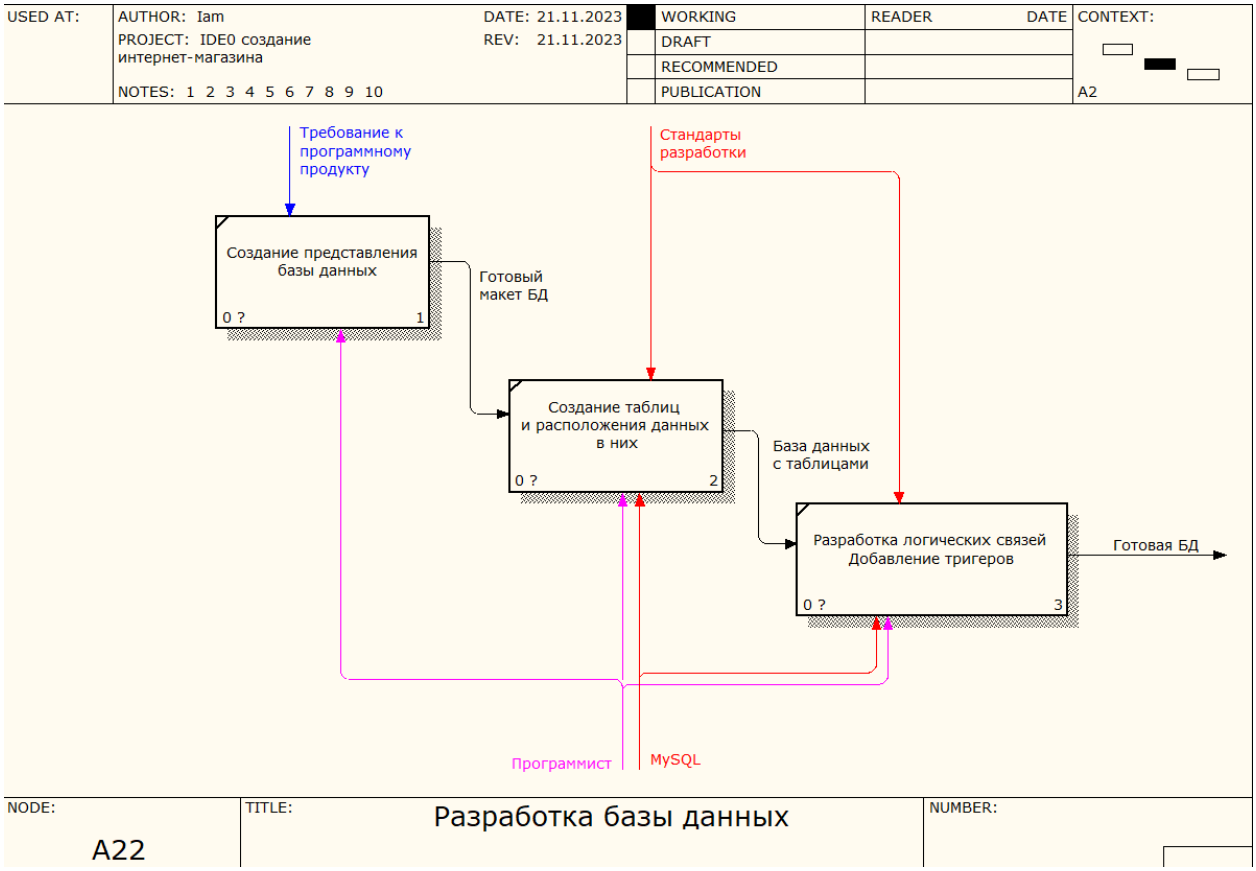


Схема 13 — Декомпозиция уровня 2.2 Разработка базы данных

Для уровня 2.3 написание серверной части сайта понадобится 4 модуля:

Первые 2 модуля — серверная маршрутизация и написание логики обработки запросов с клиентской части, имеют одинаковые воздействия — Браузер, программист, редактор кода, PHP, Apache и одинаковую управляющую часть стандарты разработки. К 3 модулю написание запросов к базе данных добавляется еще MySQL. 4 модуль серверная безопасность управляется стандартами разработки и имеет следующие воздействия — Браузер, программист и Apache.

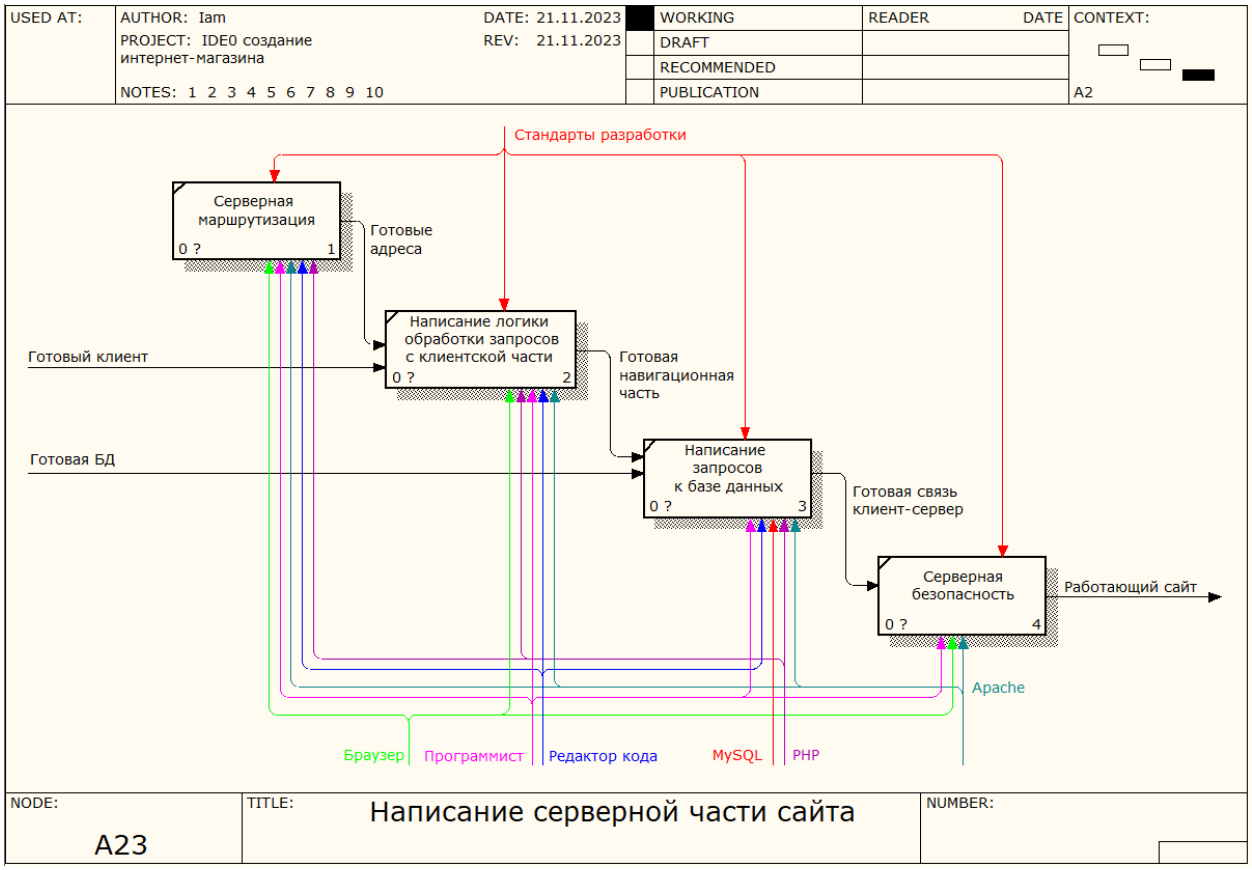


Схема 14 — Декомпозиция уровня 2.3 Написание серверной части

Декомпозиция Тестирования, уровень 3, происходит на 2 модуля:

Тестирование клиентской части и тестирование серверной части. Оба модуля имеют одни и те же источники внешнего воздействия — Программист и Браузер.

Последняя декомпозиция — уровень 4, Размещение сайта на хостинге:

Есть 4 модуля у всех одинаковое управление — требование к программному продукту; и одинаковое внешнее воздействие — программист:

Выбор хостинга;

Загрузка файлов на сервер;

Выбор домена;

Настройка сервера;

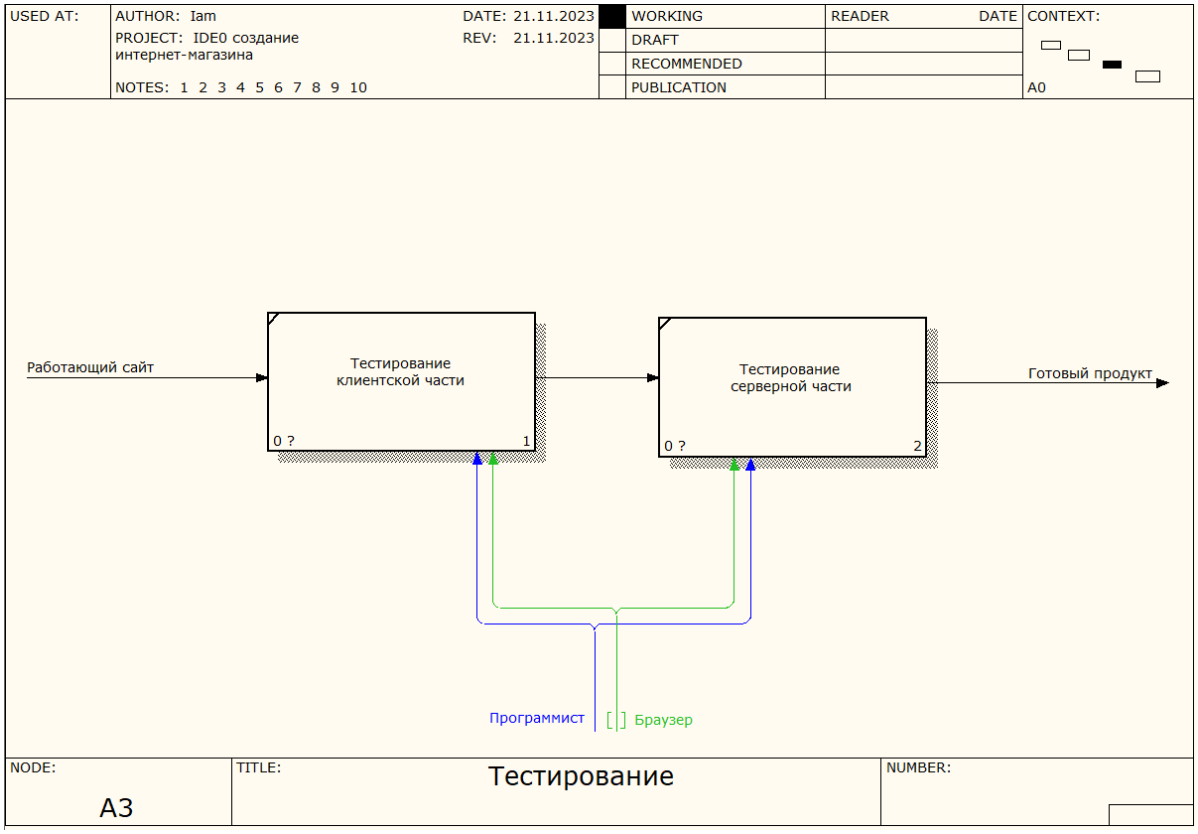


Схема 15 — Декомпозиция уровня 3 Тестирование

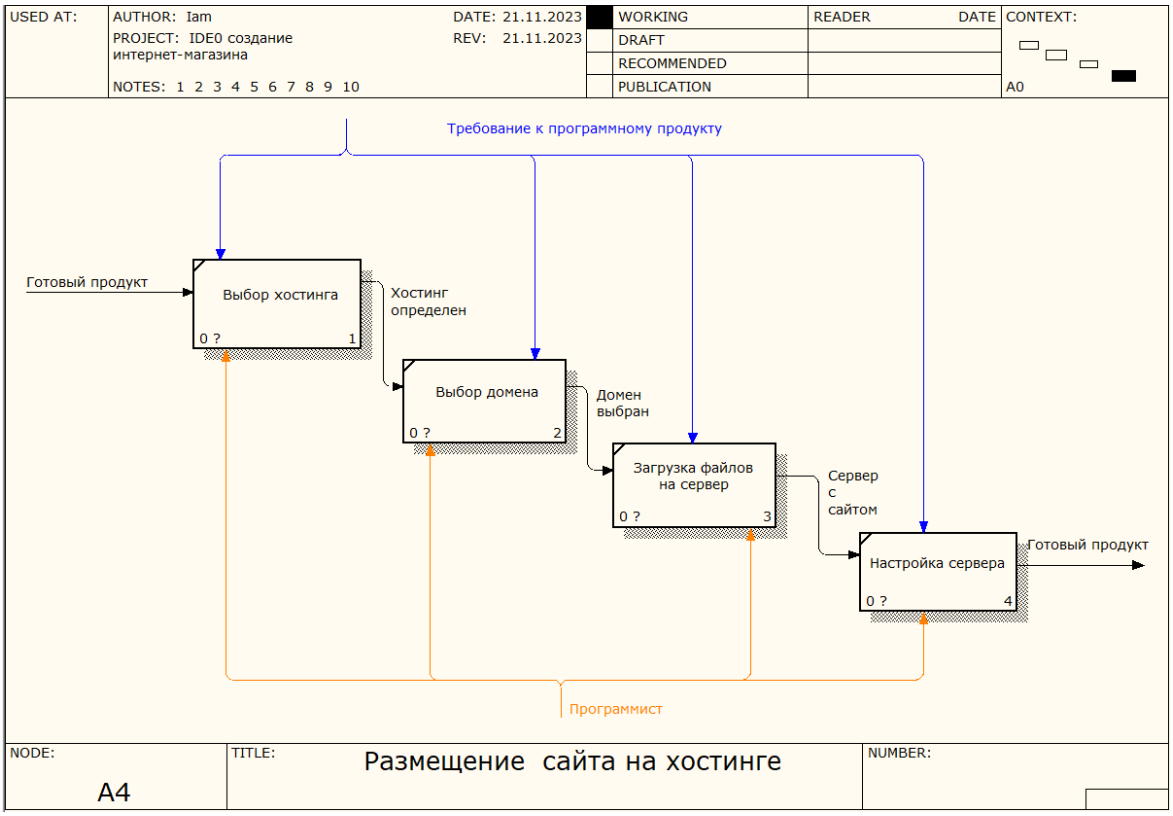


Схема 16 — Декомпозиция уровня 4 Размещение сайта на хостинге

3.1 Моделирование IDEF3

Для описания логики взаимодействия информационных потоков наиболее подходит IDEF3, называемая также workflow diagramming – методологией моделирования, использующая графическое описание информационных потоков, взаимоотношений между процессами обработки информации и объектов, являющихся частью этих процессов.

Основные описательные блоки диаграммы IDEF3:

1. Работы – являются центральными компонентами модели, изображаются прямоугольниками с прямыми углами и имеют имя, обозначающее процесс действия.

2. Связи – показывают взаимоотношение работ.

3. Перекрестки используются для отображения логики взаимодействия стрелок при слиянии и разветвлении или для отображения множества событий, которые могут или должны быть завершены перед началом следующей работы. Типы перекрестков:

Обозначение	Наименование	Смысл в случае слияния стрелок	Смысл в случае разветвления стрелок
	Asynchronous AND	Все предшествующие процессы должны быть завершены	Все следующие процессы должны быть запущены
	Synchronous AND	Все предшествующие процессы завершены одновременно	Все следующие процессы запускаются одновременно
	Asynchronous OR	Один или несколько предшествующих процессов должны быть завершены	Один или несколько следующих процессов должны быть запущены
	Synchronous OR	Один или несколько предшествующих процессов завершены одновременно	Один или несколько следующих процессов запускаются одновременно
	XOR (Exclusive OR)	Только один предшествующий процесс завершен	Только один следующий процесс запускается

Рисунок 1 – Типы перекрестков

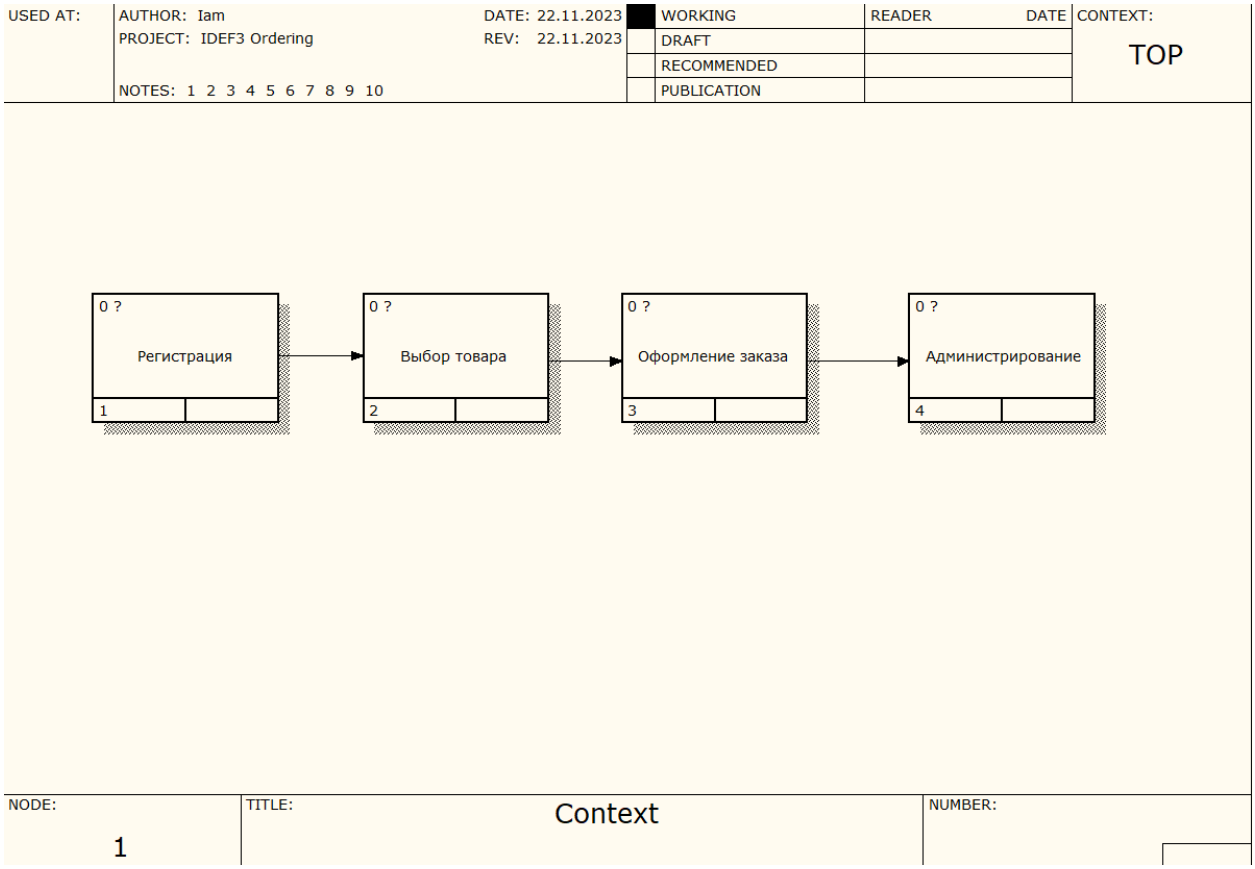


Схема 17 — Контекстная IDEF3 схема оформления заказа

Схема имеет 4 уровня со своими модулями:

Регистрация; Оформление заказа;

Выбор товара; Администрирование.

Декомпозиция 1 уровня Регистрация:

Доступно 2 опции войти или зарегистрироваться — при входе требуется логин и пароль, при регистрации — имя, фамилия, номер, почта, дата рождения. Далее при входе идет проверка на существование пользователя, а при регистрации проверка допустимости всех введённых значений. Далее обе опции разделяются на 2 условия: либо все в порядке и происходит успешная авторизация, либо что-то не так и идет возврат в начало опции.

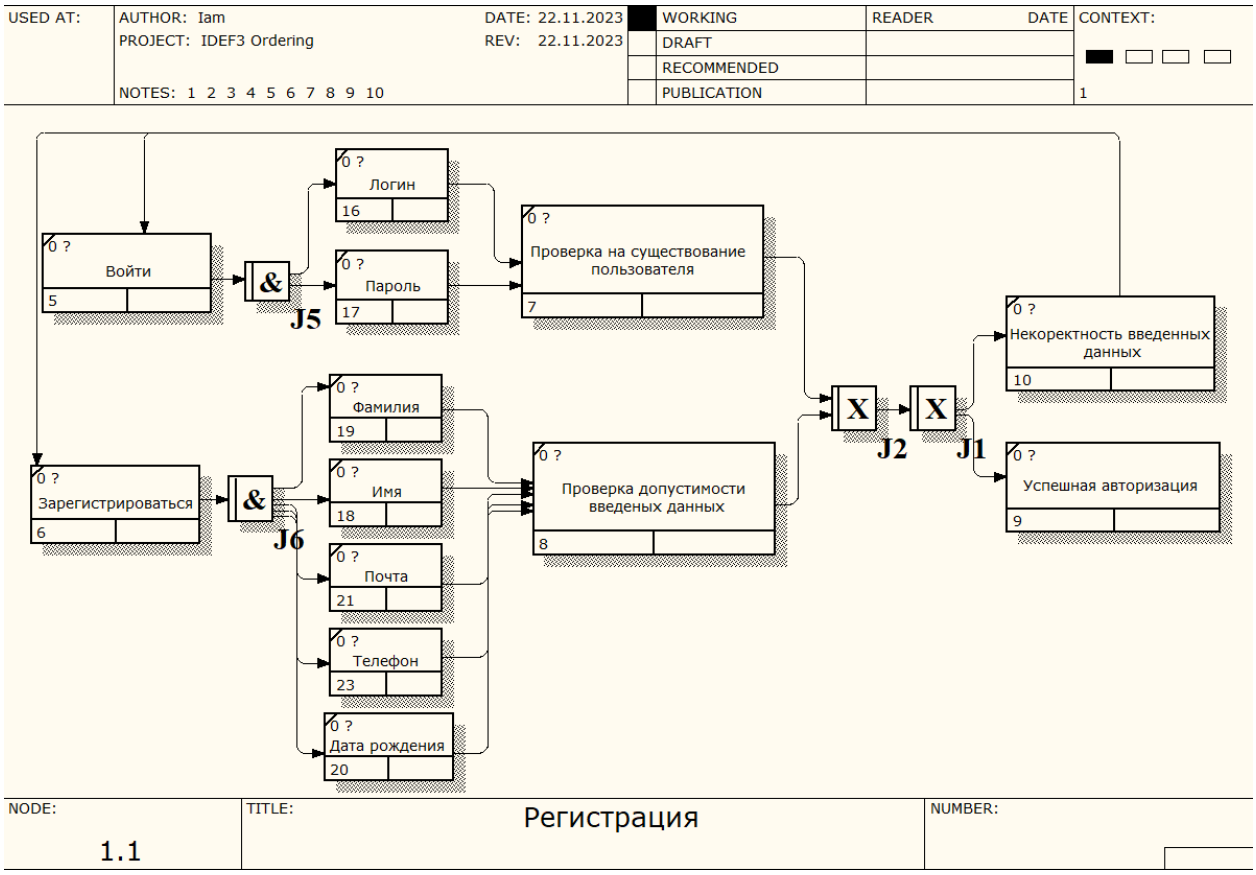


Схема 18 — IDEF3 Декомпозиция уровня 1 Регистрация

Далее декомпозиция уровня 2 Выбор товара:

Есть 4 модуля, первый модуль — это каталог, из него можно попасть в фильтрацию товаров или сразу в выборку товаров, после выборки товаров идет корзина — из нее можно перейти обратно в каталог или продолжить оформление.

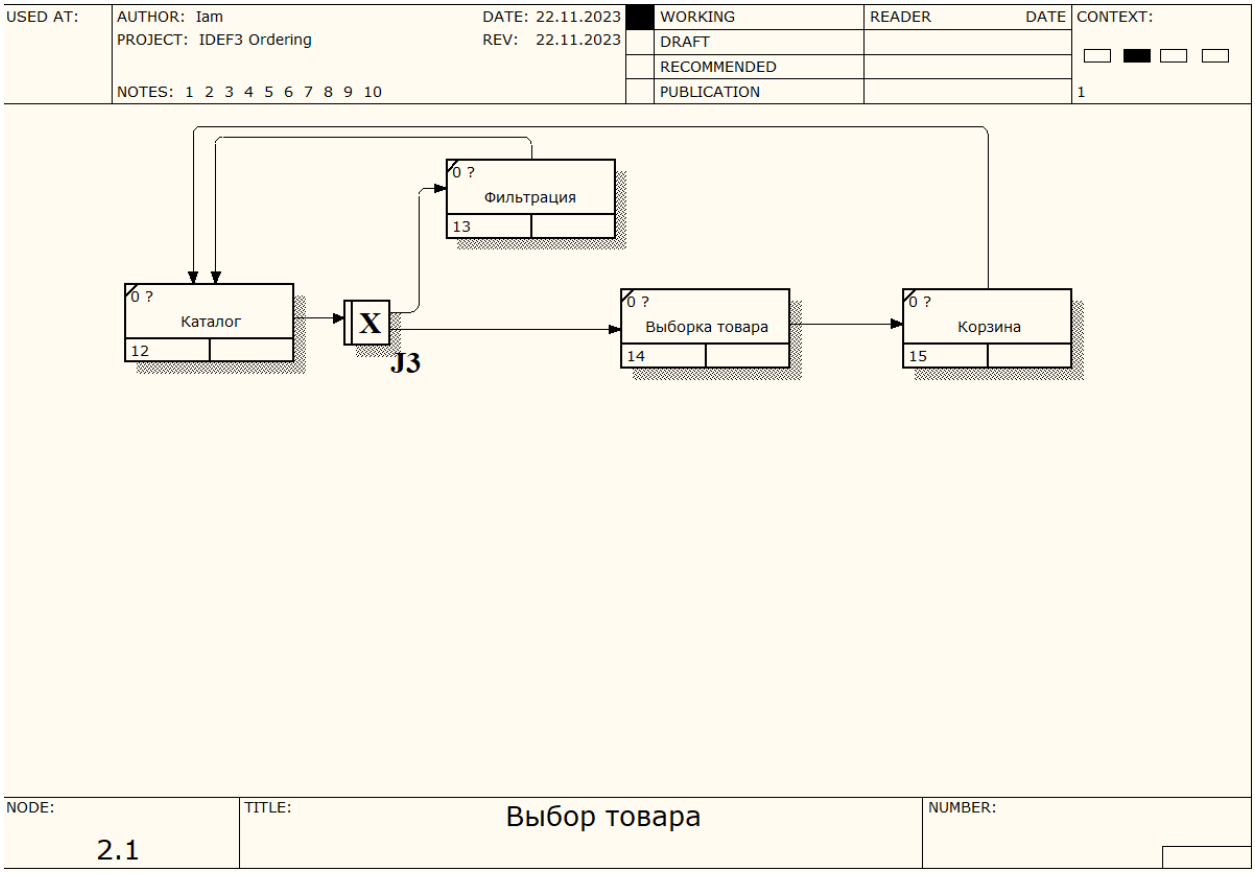


Схема 19 — IDEF3 Декомпозиция уровня 2 Выбор товара

Декомпозиция 3 уровня Оформление заказа:

Есть 3 главных модуля — нажатие на кнопку оформить заказ в корзине, далее форма оформления заказа, в которой обязательно нужны поля: дата доставки, адрес магазина, артикль товаров, общая сумма, данные пользователя. После ввода всех полей и отправки формы создается заказ в базе данных.

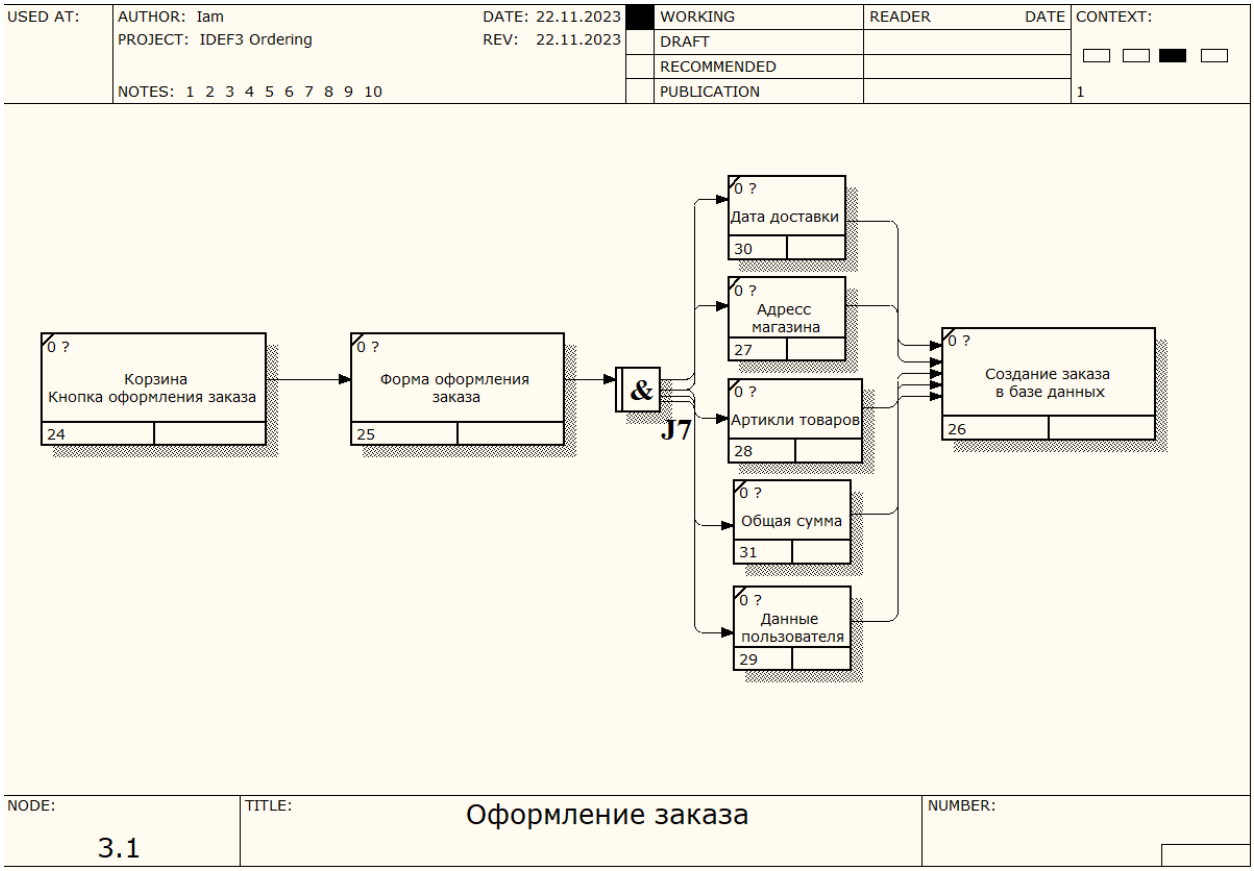


Схема 20 — IDEF3 Декомпозиция уровня 3 Оформление заказа

Декомпозиция делит 4 уровень Администрирования на:

Проверка заказа;

Распределение;

Смена статуса.

Схема 21 — IDEF3 Декомпозиция уровня 3 Администрирование

3.2 Моделирование IDEF1X

Основными элементами модели IDEF1X являются: сущности, атрибуты и отношения.

Сущность — это множество реальных или абстрактных объектов, обладающих общими характеристиками, например, персонал, явления, предметы. Каждый такой объект может быть представлен только одной уникальной сущностью. Сущность имеет уникальное имя. Как правило, ее название обозначается существительным в единственном числе, изображается в виде прямоугольника.

Атрибут — характеристика сущности. Каждый атрибут имеет один или несколько экземпляров атрибутов, т.е. конкретных значений. Таким образом, каждый конкретный экземпляр сущности будет иметь конкретный экземпляр атрибутов.

Отношения — это связи между двумя и более сущностями. Сущность, обладающая некоторыми атрибутами, является типом информации, которая хранится в базе данных. Связь этих типов информации друг с другом показывается в виде отношений между сущностями.

Моделирование IDEF1X проводится программе Microsoft Visio.

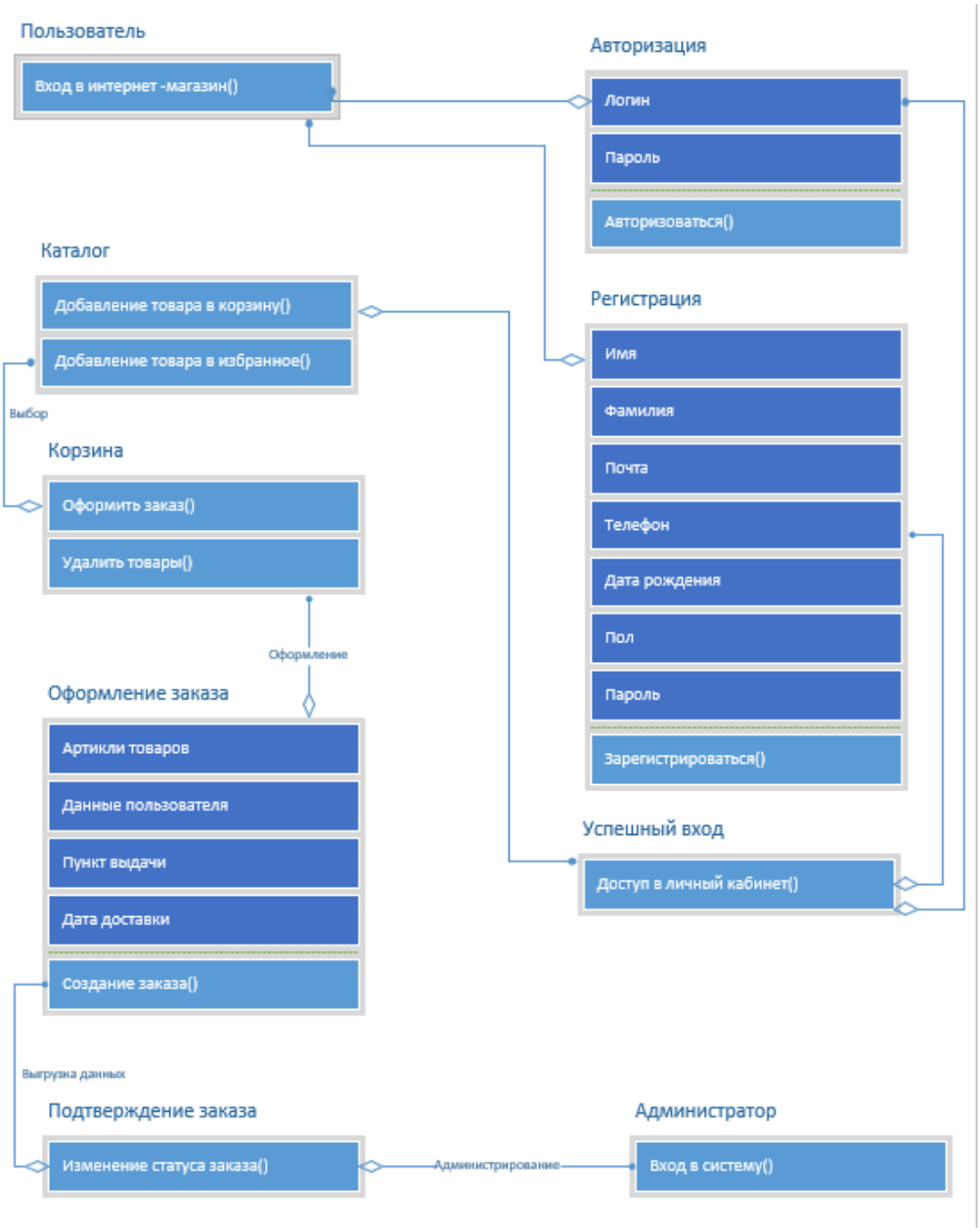


Схема 22 — IDEF1X оформление заказа

4 Объектно-ориентированное проектирование информационной системы

Для описания функционального назначения системы построена диаграмма вариантов использования. Разработанная диаграмма вариантов использования представлена ниже. В системе представлено два актер – пользователь и администратор. У пользователя есть следующие варианты:

Вход и Регистрация — вход включает в себя под сущность: логин и пароль. Регистрация включает в себя следующие под сущности: пароль, имя, фамилия, телефон, почта, дата рождения и пол;

Личный кабинет — зависит от одной из сущностей вход или регистрация. Включает в себя сущности расширение: деавторизация; избранное, в котором есть расширяющая под сущность удалить из избранного; Профиль с включающей под сущности изменить информацию; Заказы с двумя расширяющими под сущностями получить информацию о заказе и отменить заказ.

Каталог — включает в себя 3 включающей под сущности: Просмотр товара; добавить в избранное; добавить товар в корзину. И так же включает 1 расширяющую под сущность — фильтрация товаров.

Корзина — включает 1 включающую под сущность — оформить заказ и две расширяющих — удалить товары и добавить товары.

Сущность Администратор имеет 1 сущность с 2 включающими под сущностями — распределение и изменение статуса.

Диаграмма вариантов использование была построена в Microsoft Visio.

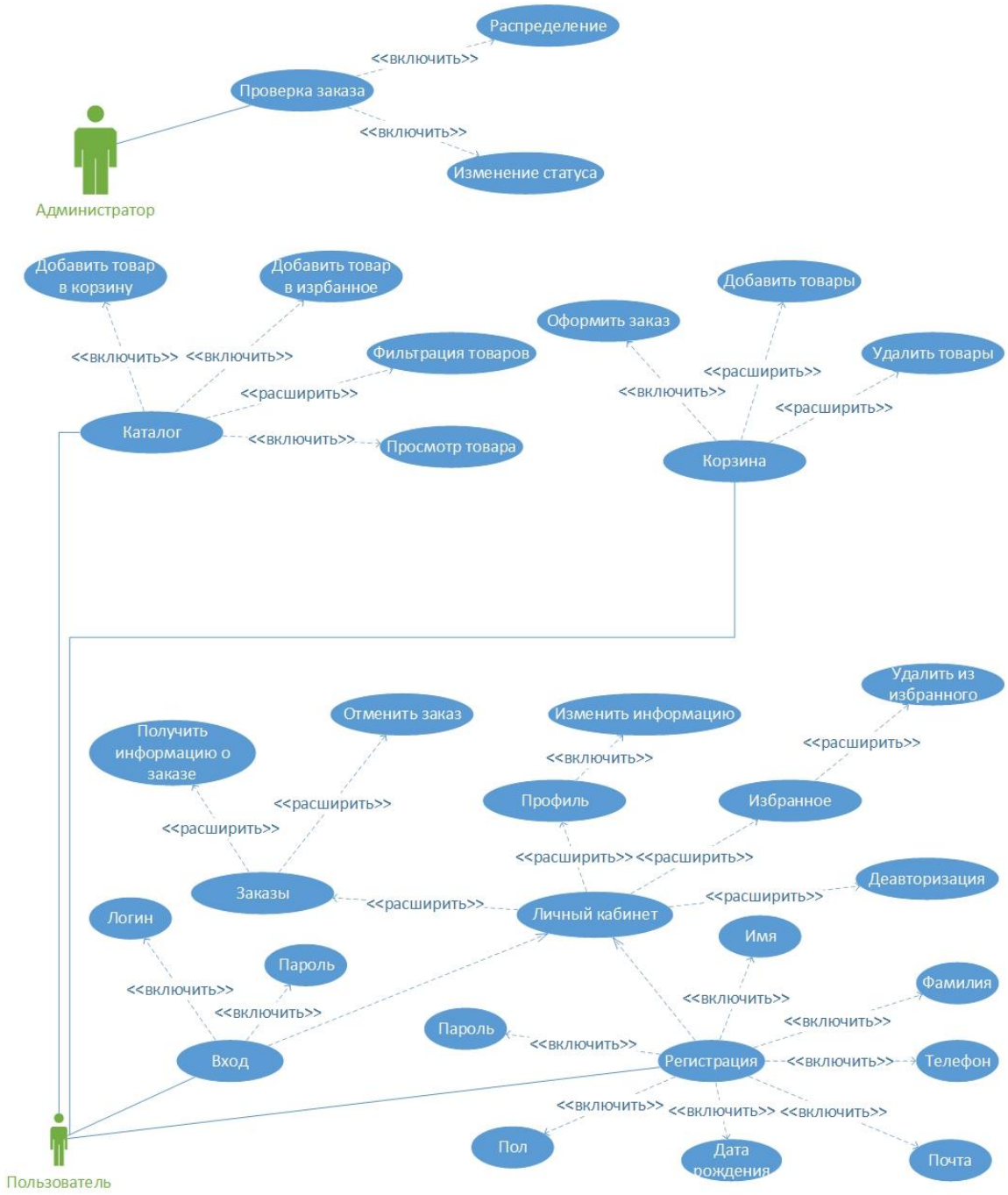


Схема 23 — Диаграмма вариантов использования

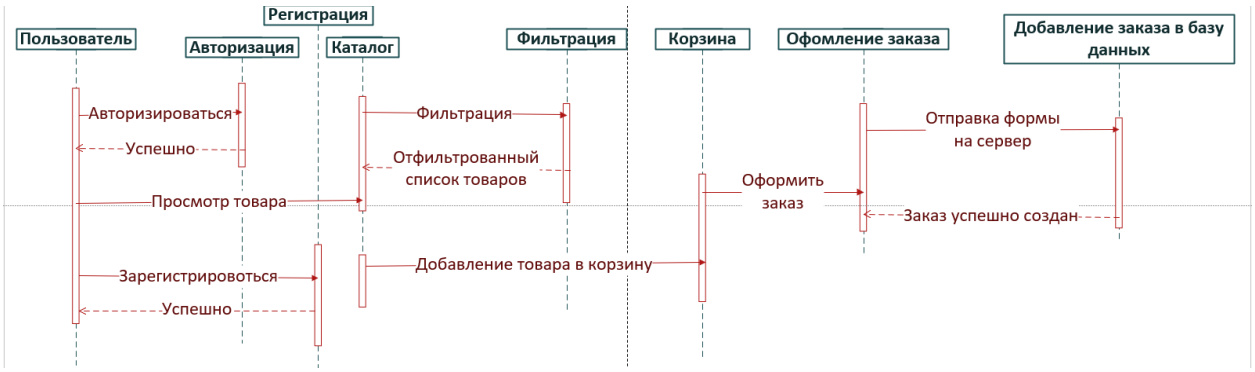


Схема 24 — Диаграмма последовательности действий

5 Проектирование и дизайн интернет-магазина

5.1 Построение структуры сайта

Интернет-магазин будет содержать следующие блоки (страницы):

1. Шапка и подвал сайта — статическая обертка, которая будет доступна на каждой странице для удобной навигации. Шапка будет содержать в себе следующие ссылки: на просмотр товаров категории мужчинам/женщинам/детям; на страничку информации; главная; корзина и личный кабинет. Подвал будет содержать в себе различные ссылки на блоки в страничке информации, ссылку на политику конфиденциальности и быструю форму на подписку.

2. Главная страница — на ней будет представлен блок с актуальными новостями, и двумя слайдерами — популярные товары и скидки.

3. Страничка информации — будет содержать несколько статей: о нас; как заказать товар; соц. Сети и связь. Так же там будет форма обратной связи.

4. Страничка каталог — содержит в себе фильтры и карточки с товарами.

5. Страничка товара — информация о товаре, выбор размера, добавление в корзину, слайдер с картинками, отзывы, информация и слайдер с похожими товарами.

6. Корзина — информация о добавленных товарах в корзину, возможность прибавить/убавить число товара, удалить товар, добавить в избранное, очистить корзину и оформить заказ.

7. Личный кабинет — личный профиль пользователя и его редактирование, заказы пользователя, избранные товары и кнопка деавторизации.

8. Страничка авторизации — содержит блок авторизации и регистрации.

Так же мне логотип, иконки, шрифт и различные эскизы для оформления дизайна, для этого я буду использовать нейросети: Fusion Brain (Kandinsky), pixelbin, cleanup.pictures. Инструмент Photoshop. Интернет-ресурсы: icons.getbootstrap, fonts.google (шрифт Raleway).

5.2 Дизайн страниц

Проектирование и дизайн страниц я буду делать в интернет-приложении — Figma. Так же следует добавить, что это не окончательный дизайн и после самой верстки страниц он будет выглядеть еще лучше.

Дизайн шапки и подвала. Слева в шапке будут располагаться 4 ссылки на каталог мужской/женский/детский и на страницу информации; По середине будет ссылка с логотипом сайта ведущая на главную; справа две кнопки-ссылки ведущие на корзину и на личный кабинет. В подвале сайта снизу будет располагаться ссылка-логотип ведущая на главную; так же всю область будут занимать ссылки ведущие на различные блоки страницы информации, так же политика конфиденциальности и форма быстрой подписки на новости.

Дизайн главной страницы. Сверху будет располагаться слайдер с новостями — слева текст и управление слайдером, справа изображение к данной новости. Ниже идут два одинаковый по структуре мини-слайдера — популярные товары и главные скидки. Управление слайдером находится сверху справа.

Дизайн страницы каталога. В самом вверху будет заголовок категории товаров, ниже будут располагаться фильтры и далее сами карточки товаров.

Дизайн страницы товара. Первый блок делится на 2 части — справа слайдер с изображениями товара, слева вся информация и действия. Вверху первой части находится наименование товара, и его тип, ниже идет артикул, цена, цвет, другие цвета, блок выбора размера и кнопка добавление в корзину. Во втором блоке странице находится вкладки с описание и отзывами. Третий блок содержит мини-слайдер с похожими товарами.

Дизайн страницы информации. Первый блок будет содержать информацию о магазине, текст, приведенный в макете для примера. Далее идет информация о том, как сделать заказ, связь с магазином и форма обратной связи.

Дизайн корзины покупок. Страница делится на два блока: слева панель управления и общая информация о товарах; справа все карточки со всеми добавленными товарами, в карточке содержится информация об товаре и меню управления.

Дизайн страницы авторизации. Страница сделана в виде вкладок, сверху находится переключатель снизу контент.

Дизайн личного кабинета. Страница разделена на 2 части, слева меню управления контентом, справа содержимое. Есть 3 вкладки, профиль, избранное и заказы. Так же есть кнопка деавторизации. В целях экономии места продемонстрирована только вкладка заказов.

6 Разработка интернет-магазина

6.1 Написание клиентской части

6.1.1 Клиентская маршрутизация

Так как это не просто сайт интернет-магазина, а веб-приложение, то помимо серверной маршрутизации требуется еще и клиентская. Маршрутизация будет осуществляться с помощью библиотеки **React Router**, а само приложение будет написано на JavaScript библиотеки — **React**. Схема клиентской маршрутизации, которая написана в приложении:

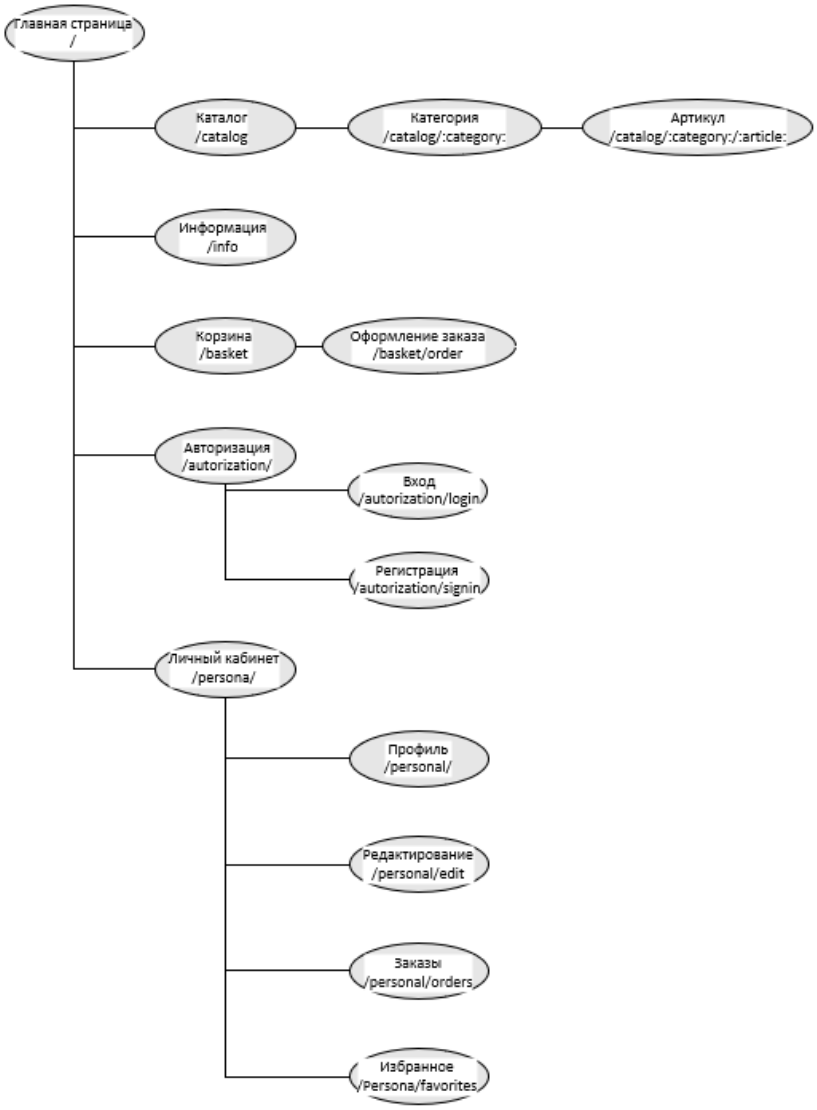


Схема 25 — Клиентская маршрутизация

6.1.2 Статичная верстка страниц

Верстка будет осуществляется с использованием **JSX (JavaScriptXML)** на библиотеки **React**, в качестве препроцессора для стилей будет использоваться **SCSS**.

Первое с чего надо начать это верстка статичных элементов — шапка и подвал, они будут на каждой странице, то есть все следующие страницы будут помещаться между двумя этими блоками. Далее нужно реализоваться карточку товара.

Следующая идет главная страница. В ней немного изменил слайдер и мини-слайдер, а именно элементы управления. Такие элементы как слайдеры, вкладки, активные ссылки, переключатели, кнопки и т.п. будут просто сверстаны и неживыми, ими я займусь в следующей части. Результат верстки главной страницы:

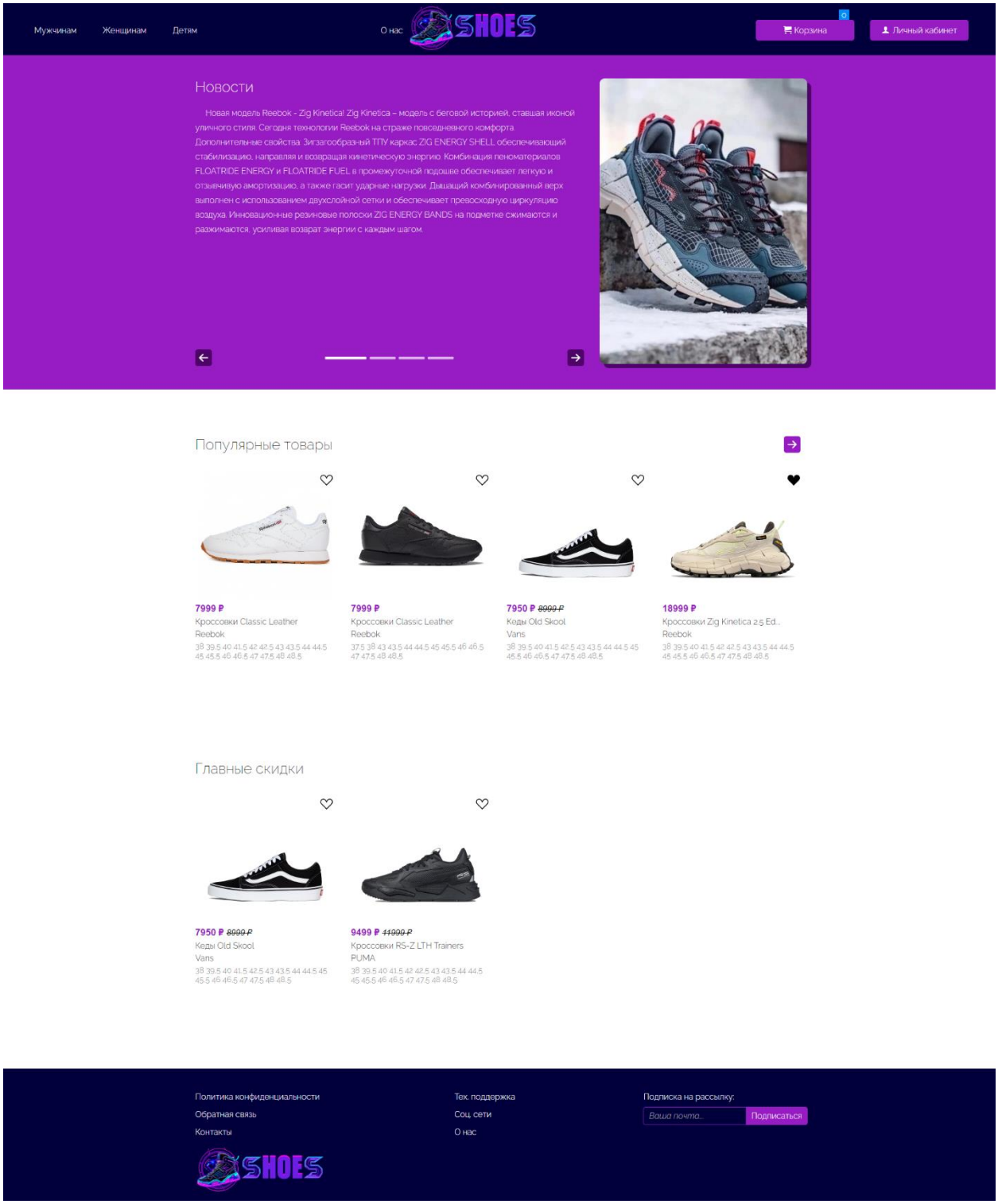


Рисунок 2 — Главная страница сайта

Далее верстка страницы каталога и блока фильтрации.

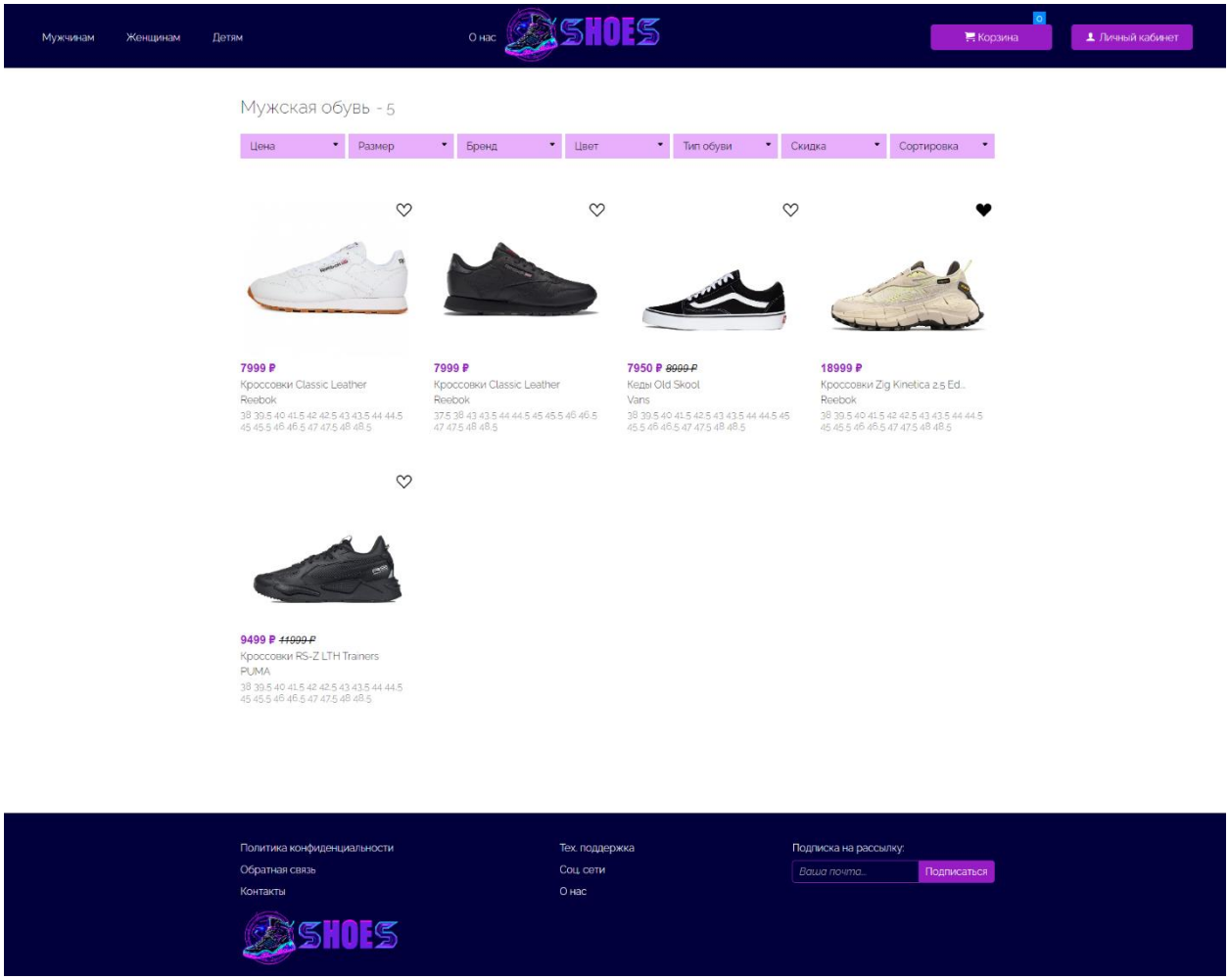


Рисунок 3 — Страница каталога

Верстка страницы товара. Изменил слайдер, положение некоторых элементов, добавил несколько деталей.

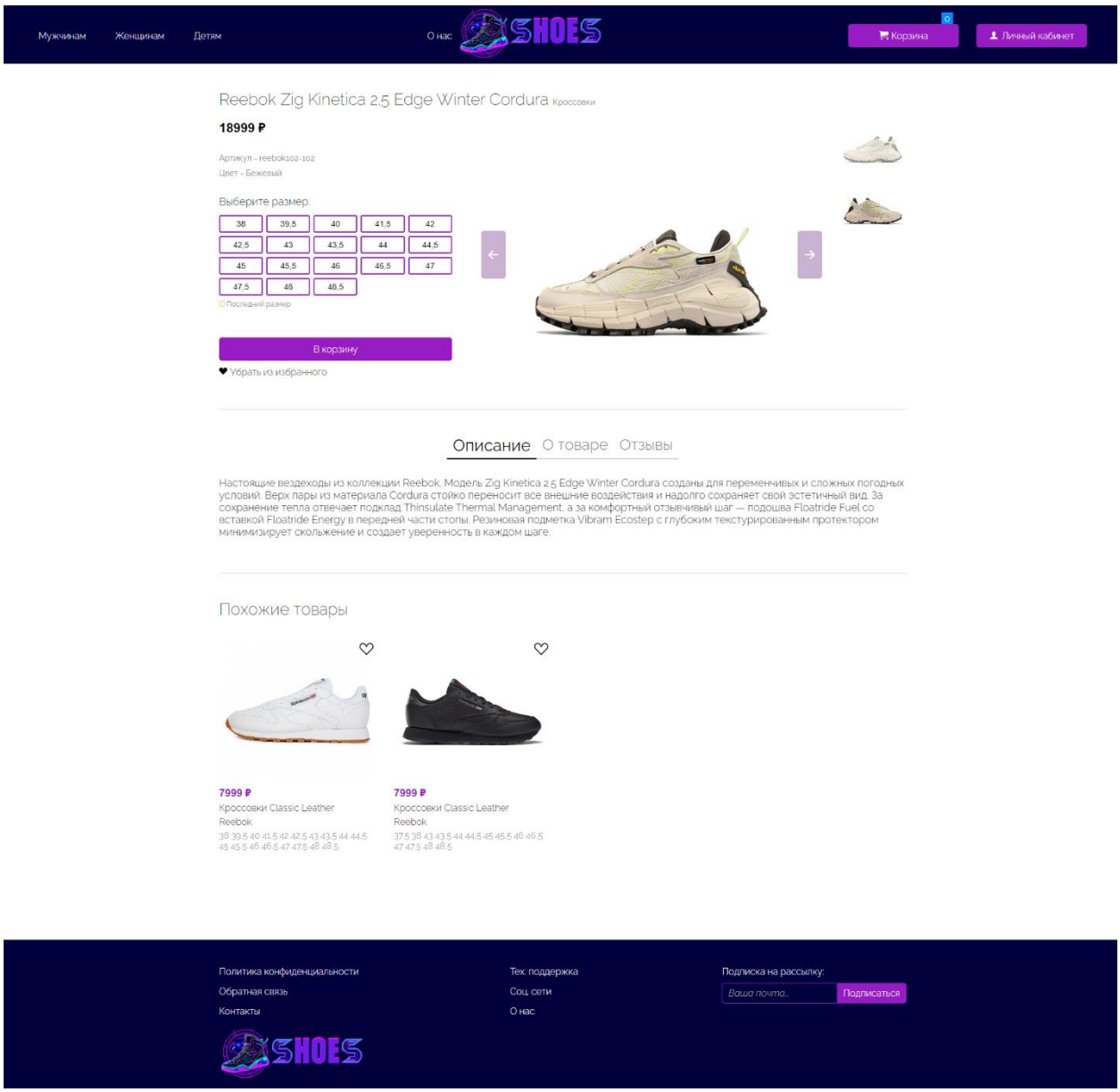


Рисунок 4 — Страница товара

Свёрстанная страница информации:

Рисунок 5 — Страница информации

Страница корзины покупок. Здесь я слегка изменил карточку товара.

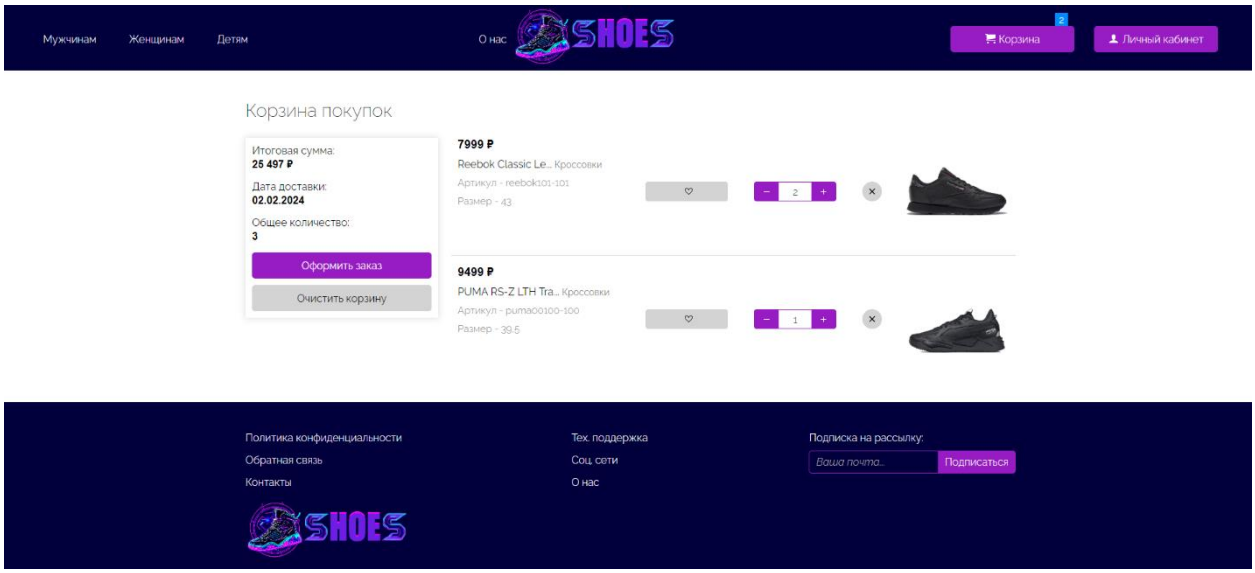


Рисунок 6 — Страница корзины покупок

Верстка страниц личного кабинета. Изменил поля информации профиля, так же поля редактирования профиля, блок избранного переделан, добавил туда еще фильтры (теперь это как отдельный каталог), в блоке заказов слегка улучшил карточку заказа, и подправил кнопку деавторизации.

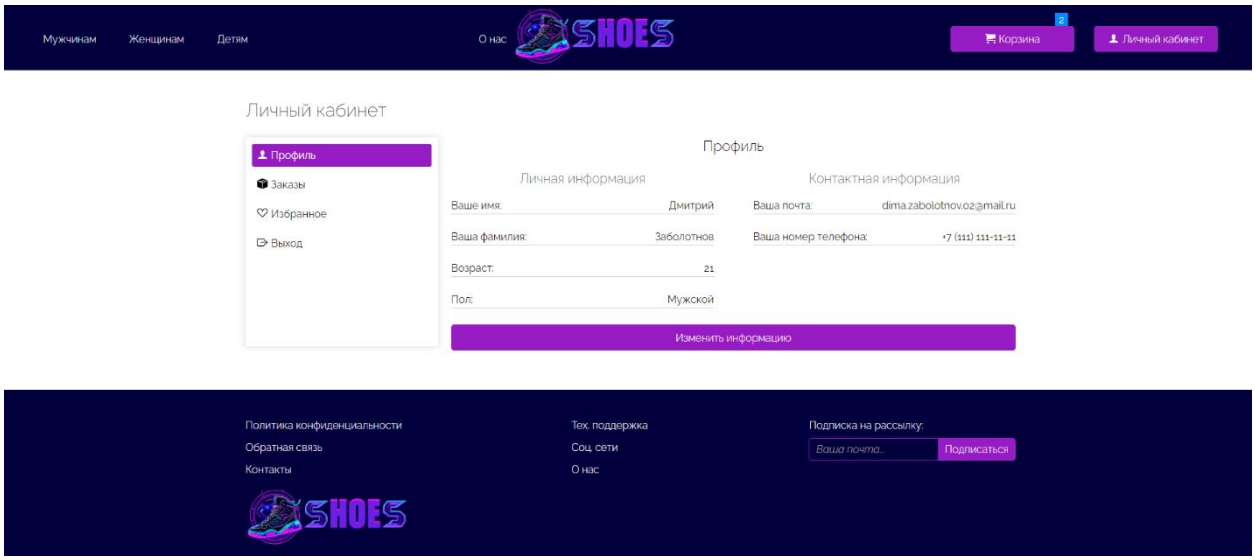


Рисунок 7 — Страница профиля: Профиль

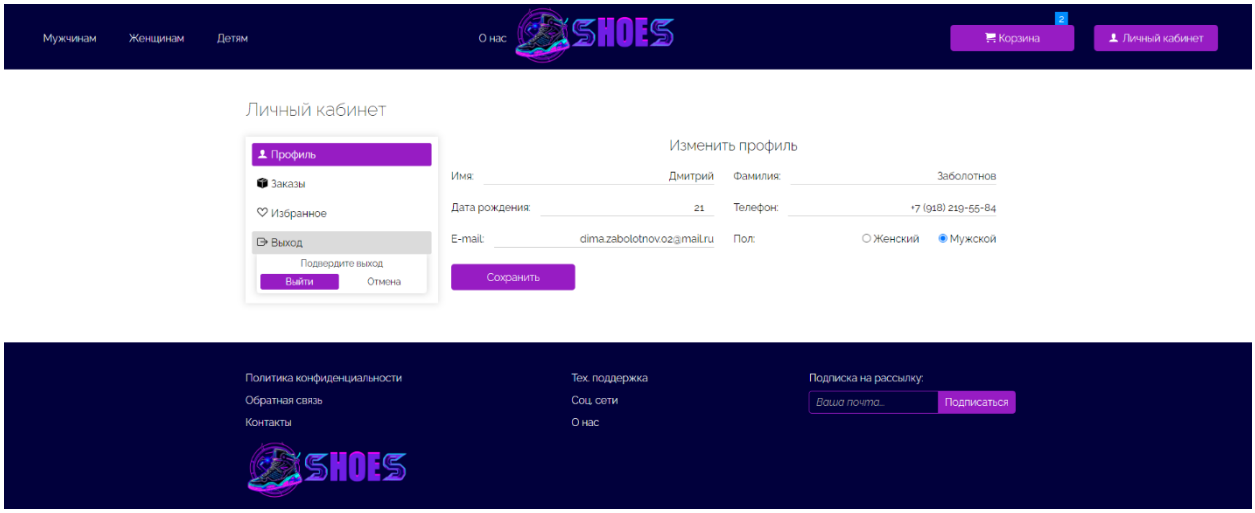


Рисунок 8 — Страница профиля: Редактирование

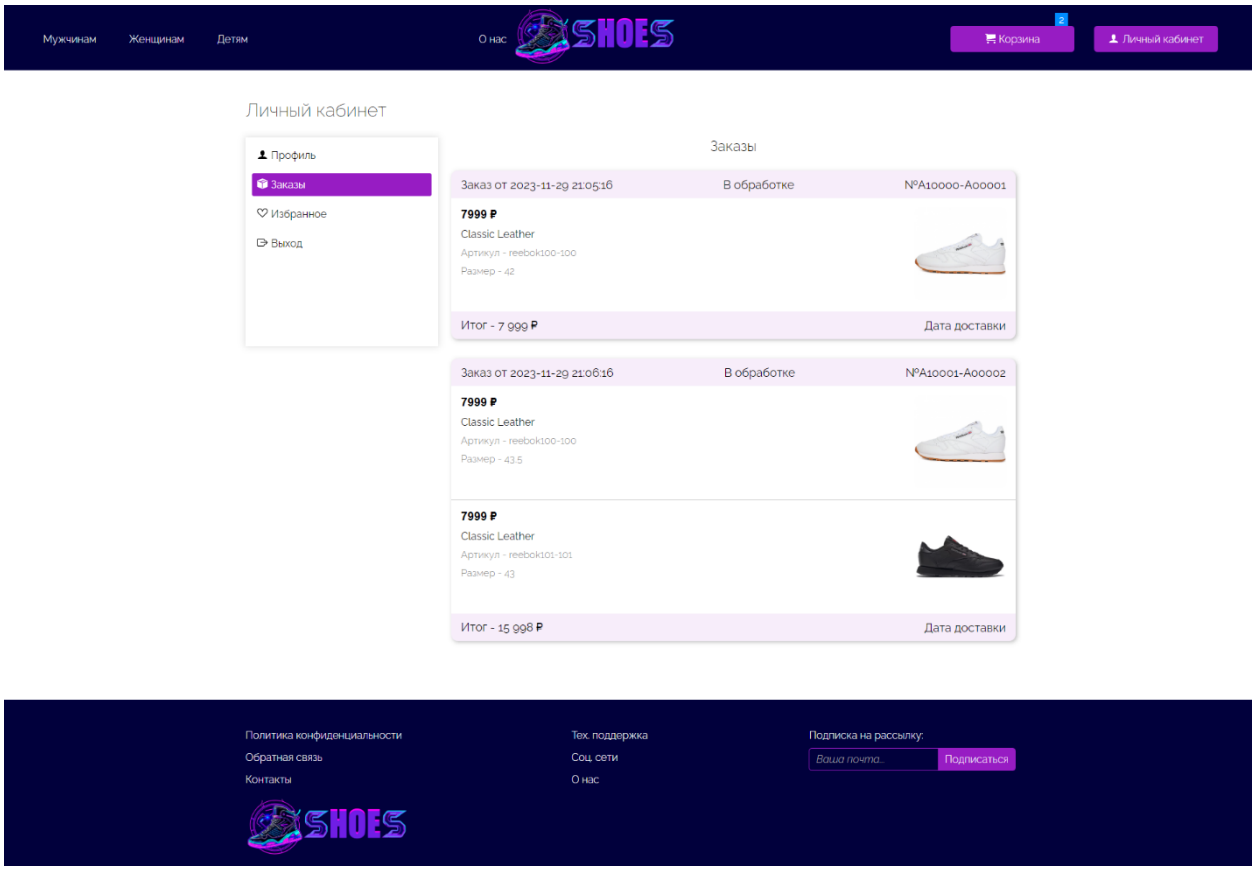


Рисунок 9 — Страница профиля: Заказы

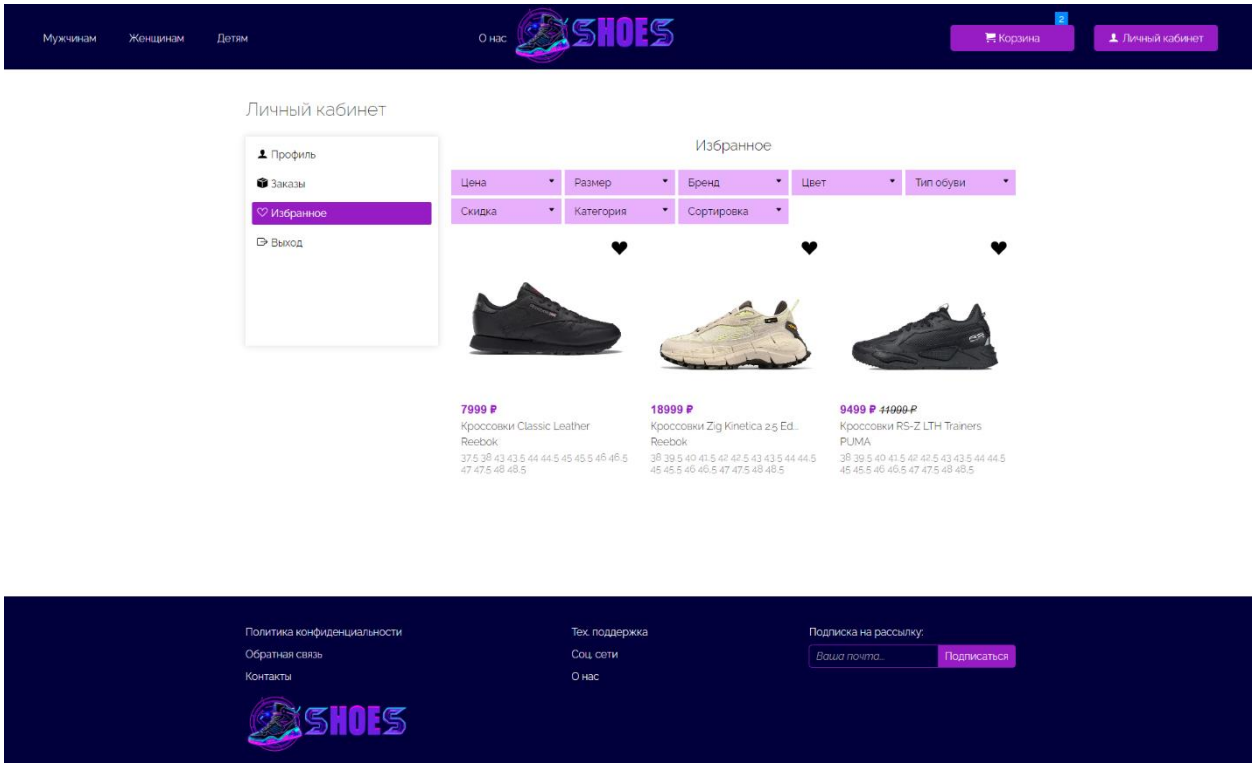


Рисунок 10 — Страница профиля: Избранное

Верстка страницы регистрации/авторизации. Слегка изменил форму регистрации.

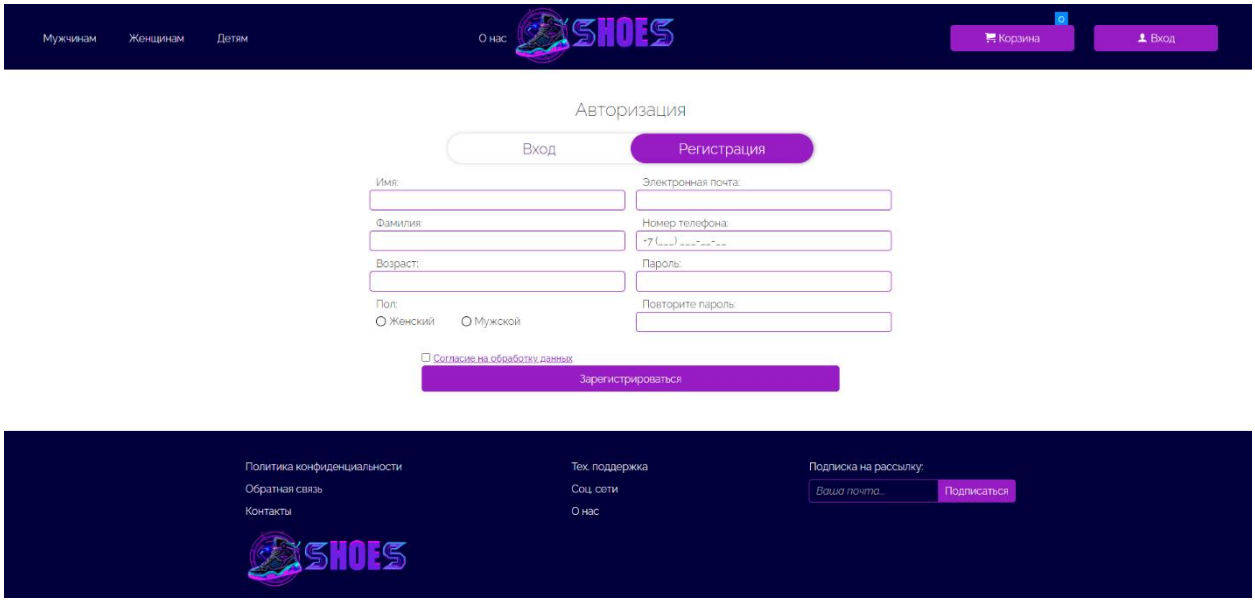


Рисунок 11 — Страница авторизации: Регистрация

6.1.2 Написание запросов к серверу и логики сайта

Для написания запросов к серверу и формирования логики от них я буду использовать, к добавке **React**, библеотеку контроля состояний **Redux** и ту же библиотеку **React Router**.

Для главной страницы нужна информация: об слайдере — текста и картинки; информация о двух мини-слайдеров — 2 массива содержащие обьекты с информацией о каждом товаре. Далее эти слайдеры нужно оживить.

Информацию на странице информации я оставлю статичной, а вот для формы нужно написать запрос на отпраку данных формы заполненой пользователем.

Так же в подвале сайта есть форма которой то же нужен запрос на сервер.

Страница каталога на нее можно попасть через категорию man, woman, kids или просто по пути /catalog. Первое нужно написать логику фильтров, коротко при клике на определенную вкладку фильтров будет появляться меню с вариантами данного фильтра, при выборе варианта будет отправлятся запрос на сервер с информацией об выбранных фильтрах и будет возвращать каталог товаров отфильтрованных по переданым фильтрам, каталог товаров будет составляться из этого запроса.

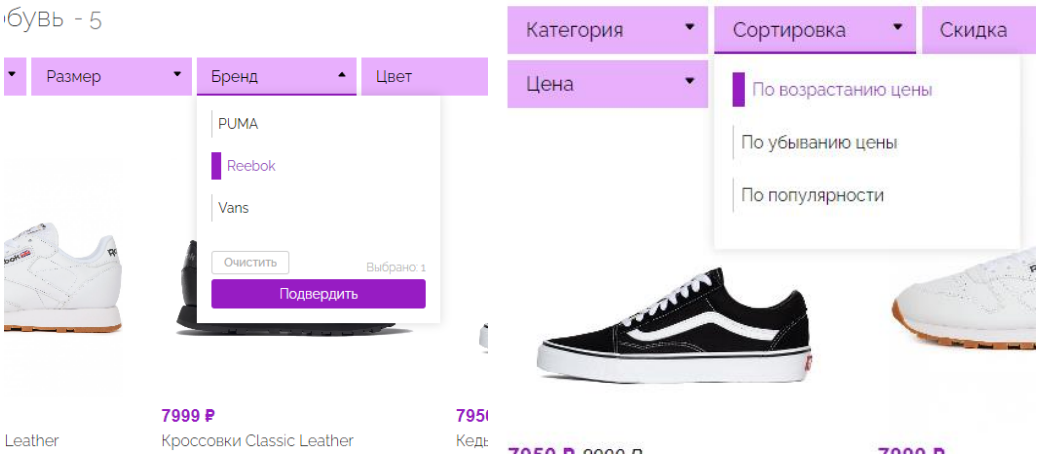


Рисунок 12 и 13 — Готовые фильтры

Написание логики авторизации, логика будет такая что если пользователь авторизован ему будет доступна страница личного кабинета и не доступна страница авторизации, если пользователь не авторизован, то наоборот. Для этого при входе на сайт будет отправляться запрос на сервер для получения информации об авторизации пользователя.

Логика корзины покупок, при входе на сайт будет отправляться запрос о получении корзины покупок для показа на главной странице количества продуктов в корзине если пользователь добавлял в нее товары. Так же нужен запрос для добавления товара в корзину, он будет доступен из карточки товара и из страницы товара, при нажатии на кнопку добавить в корзину и обязательного выбора размера. Так же на странице корзины в карточках товара нужно сформировать запросы на убавление и добавление количества товара, удаление товара из корзины и очистки корзины. Так же есть запрос на добавление товара в избранное, он будет доступен только авторизованным пользователям.

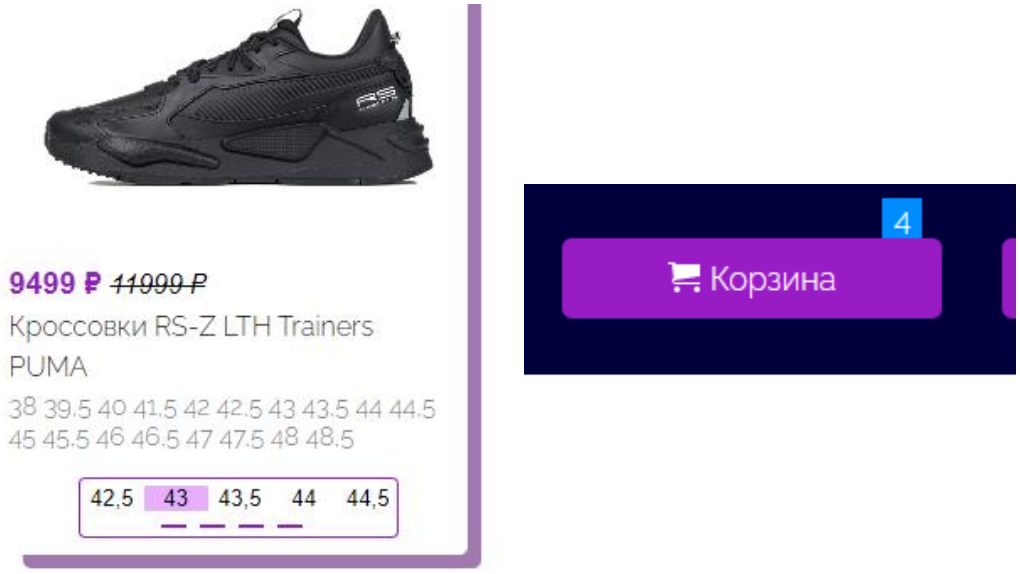


Рисунок 14 и 15 — Выбор размера товара и показ количества товара в корзине

Авторизация и личный кабинет. Страница авторизации будет включать два запроса для авторизации и для регистрации, при успешном ответе с сервера будет происходить перезагрузка сайта и переход на главную, то есть

при перезапуске снова отправляется запрос на информацию об авторизации и если на сервере обработалось все правильно, и пользователь авторизован, то вернется положительный ответ. В личном кабинете справа модуль управления будет статичный, то есть на каждой странице в личной кабинете, при клике на вкладку происходит переход на выбранную страницу и справа появляется выбранный контент. Личный кабинет включает в себя единый запрос на получение главной информации об пользователе, запрос на изменение информации пользователя, при переходе на страничку избранного отправляется запрос на получение избранных товаров пользователя, на странице заказов отправляется запрос на получение всех заказов пользователя, так же при клике по вкладке выход появляется саб-меню с кнопкой выйти или отменить, при клике на кнопку выход отправляется запрос на деавторизацию.

Для страницы товара отправляется запрос для получения обьекта с информацией об выбраном товаре: название, цена, тип, изображения, доступные размеры и т.п. Так же к этому обьекту добавляется массив с похожими товарами. Так же на этой странице присутствует слайдер и виджеты которые нужно оживить.

Далее добавление некоторой анимации и подправка остальных деталей.

6.2 Разработка базы данных

6.2.1 Проектирование базы данных

База данных должна содержать следующие таблицы:

Таблица с информацией обо всех товарах (goods), в ней должна быть информация об названии, бренда, типа, изображения, изображения слайдеров, категорию, артикул, цена, старая цена, описание и является ли товар хитом;

Таблица со схожими товарами (goods_related) — содержит артикул товара и артикул его схожего товара;

Таблица с вариациями товара (goods_variations) — содержит артикул товара и артикул его схожего товара;

Таблица категорий (categories) — название категории и ее код (название на англ.);

В таблице пользователей (users) должен быть имя, фамилия, телефон, почта, возраст, пол и пароль;

Таблица отзывов (feedbacks) — уникальный индификатор пользователя, артикул товара, оценка, заголовок и текст;

Таблица избранного (favorites) — уникальный индификатор пользователя и артикул товара;

Таблица заказов (orders) — артикул заказа, уникальный индификатор пользователя, статус, дата оформления, дата изменения, дата доставки и коментарий.

Под таблица заказов, товары в заказах (orders_goods) — уникальный индификатор пользователя, уникальный индификатор заказа и размер;

Таблица для главного слайдера (news) — текст, изображение;

Таблица фильтров (filters) — наименование фильтра, код фильтра и тип (мульти выбор, одиночный и диапазон значений);

Под таблица фильтров, значения фильтров (filters_values) — значение, код и уникальный индификатор фильтра;

Под таблица значения фильтров, товары подходящие под эти фильтры (filters_goods) — артикул товара и уникальный индификатор значения фильтра.

6.2.2 Связи таблиц

База данных разработа в СУБД MySQL при помощи веб-приложения для администрирования MySQL — phpMyAdmin. Связи будут представлены в виде таблицы составленной автоматически в phpMyAdmin:

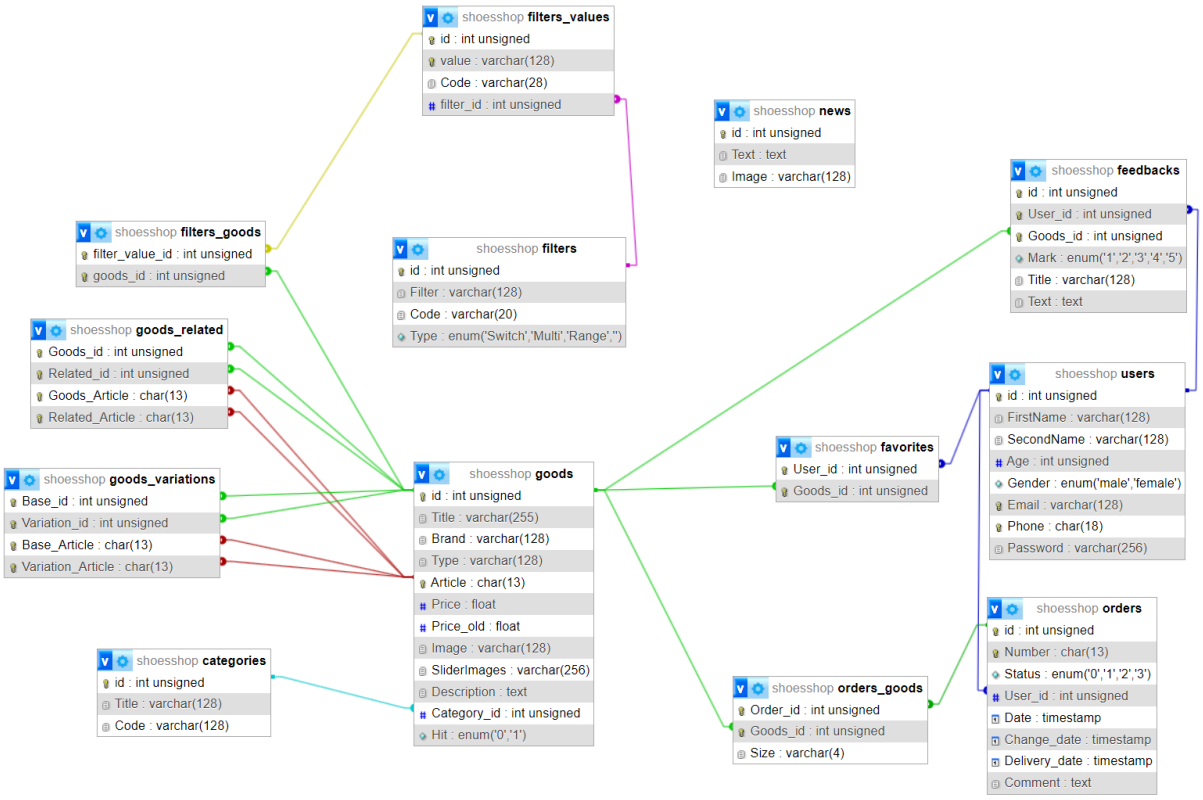


Рисунок 16 — Связи в базе данных

6.3 Серверная часть

6.3.1 Создание базовой структуры сервера

Серверная часть сайта была написана на серверном языке программирования PHP с использование библеотеки Composer и была развернута на веб-сервере Apache.

Разработка базовой структуры сайта (движка), начинается с расположения папок и файлов в каталоге.

6.3.2 Написание серверной маршрутизации

Серверная маршрутизация будет следующая:

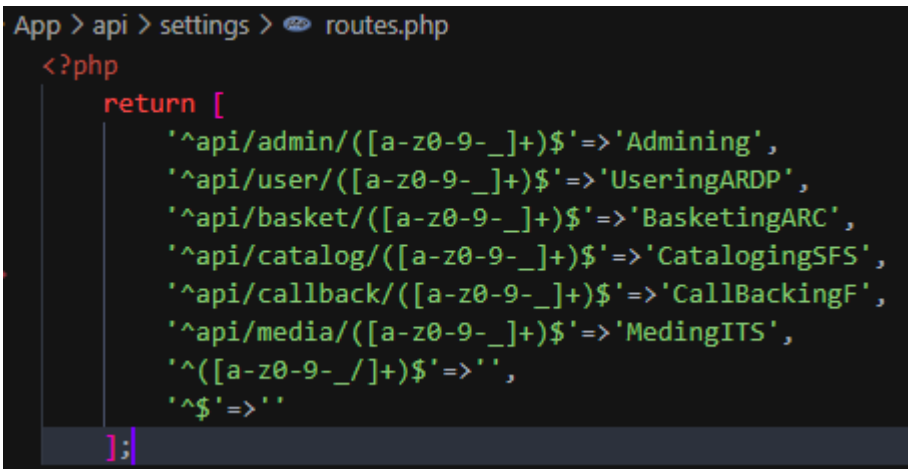


Рисунок 17 — Серверная маршрутизая

Все запросы начинающиеся с api — означают обращения с клиентской части.

admin — запросы поступающие от админской части;

user — запросы авторизации или требующие авторизации;

basket — запросы косающиеся корзины покупок;

catalog — запросы на получения каталога и все что касается товаров;

callback — запросы включающие отправку формы;

media — запросы на получения медиа файлов или какого информационного контента.

6.3.3 Написание логики обработки запросов с клиентской части

Логика ответа на поступающие запросы следующая: все запросы поступают в главный файл и далее идет распределение через маршрутизатор в другие файлы-классы в зависимости от строки запроса.

Логика авторизованных запросов — при попадание в класс user в функции-конструкторе идет проверка на авторизацию пользователя, от этого значения идет дальнейшая логика ответа из вызванной функции.

Далее все файлы-классы по работе стандарты — при поподании в существующий класс, вызывается вызваная функция, если не переданы все нужные аргумены то сервер возвращает ошибку 403 Bad Request и завершает работу скрипта.

Далее есть файл functions.php в котором содержаться некоторые повторяющиеся функции.

6.3.4 Написание запросов к базе данных и реализация сессий

Сессии являются простым способом хранения информации для отдельных пользователей с уникальным идентификатором сессии. Это может использоваться для сохранения состояния между запросами страниц. Идентификаторы сессий обычно отправляются браузеру через сессионный cookie и используются для получения имеющихся данных сессии.

Сессия сможет содержать максимум два обьекта: Обьект user для пользователя, при успешной авторизации/регистрации создается обьект user с информацией об пользователе; Объект basket создается сразу при заходе на пользователя на сайт и при всех маниупуляциях с корзиной пользователем корзина перезаписывается.

Файл-класс user:

Запрос на регистрацию пользователя, полученные данные проходят фильтрацию, если все в порядке, то выполняется следующий код:

```
Db::getPreparedQuery("INSERT INTO `users`(`Age`, `FirstName`, `Gender`, `Email`, `Phone`, `Password`, `SecondName`)
                      VALUES (?, ?, ?, ?, ?, ?, ?)", $attributes);
$userId = Db::getQuery("SELECT LAST_INSERT_ID()", false, true);
$_SESSION['user']['id'] = $userId;
foreach($userData as $k=>$v) {
    if($k == "password") continue;
    $_SESSION['user'][$k] = $v;
}
```

Рисунок 18 — Фрагмент кода блока регистрации

Где знаки вопроса есть переданные значение пользователем. Потом делается запрос на получение последнего вставленного значения, и вся информация передается в сессию пользователя.

Запрос на авторизацию пользователя, метод требует номер телефона и пароль, идет запрос на получения данных пользователя с указаным номером, и идет сравнение паролей, если они совпадают то создается сессия пользователя.

```
ry {
    $result = Db::getPreparedQuery("SELECT * FROM `users` WHERE `Phone` = ?", $attributes);
    if(empty($result)) {header("HTTP/1.0 403 Bad request"); exit(json_encode("User not found"));}
    if(password_verify($userData['password'], $result['Password'])) {
        foreach($result as $k=>$v) {
            $k = lcfirst($k);
            if($k == "password") continue;
            if($k == "phone") $k = "number";
            if($k == "email") $k = "mail";
            $_SESSION['user'][$k] = $v;
        }
```

Рисунок 19 — Фрагмент кода блока авторизации

Запрос на редактирование информации пользователя, отправляется запрос на изменения данных по указанному уникальному индификатору взятому из сессии.

```
{
    Db::getPreparedQuery("UPDATE `users` SET `Age` = ?, `FirstName` = ?, `Gender` = ?, `Email` = ?,
                          `Phone` = ?, `SecondName` = ?
                          WHERE `Phone` = '" .$_SESSION['user']['number'] . "'", $attributes);
    foreach($userData as $k=>$v) {
        if($k == "password") continue;
        $_SESSION['user'][$k] = $v;
    }
```

Рисунок 20 — Фрагмент кода блока редактирования

Запрос на получение зарегистрирован ли пользователь возвращает результат проверки на существование сессии пользователя.

Запрос на деавторизацию удаляет текущую сессию пользователя.

Запрос на получение пользователя возвращает сессию пользователя.

Запрос на изменение статуса избранного товара, принимает артикул товара и действие — set или unset. Далее взависимости от переданого значения действия составлется фрагмент запроса и контатенируется с основным. Если же запрос будет сделан из корзины, то следует проверить есть ли данный товар в корзине и если да изменить его статус избранного и в ответ вернутьсессию корзины.

Рисунок 21 — Фрагмент кода блока изменения избранного

Запрос на получение заказов пользователя, отправляет запрос на получение информации об заказе и об товарах в заказе, где пользователя определяется по уникальному индификатору.

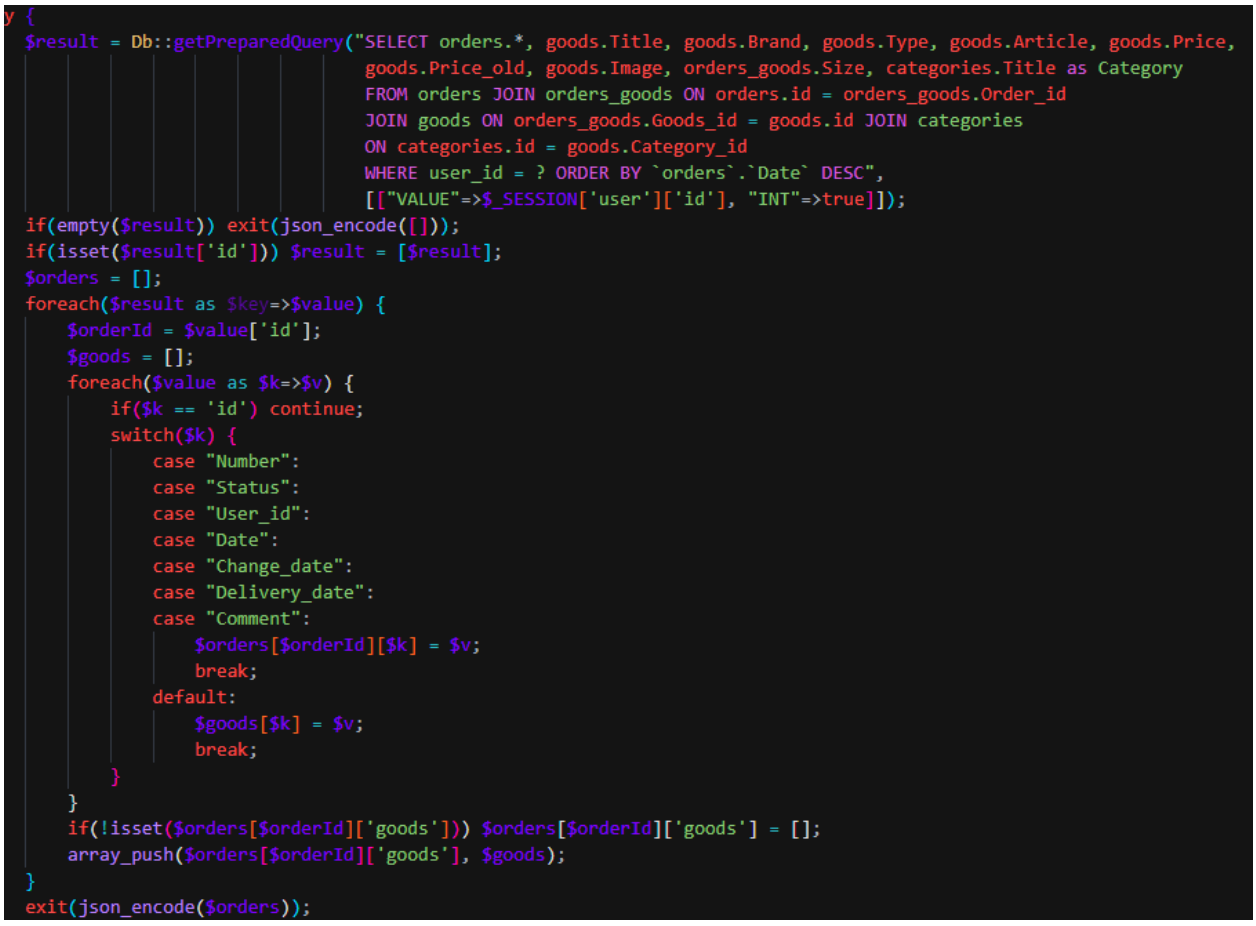


Рисунок 22 — Фрагмент кода блока получения заказов

Файл-класс basket:

Он содержит функцию конструктор, которая проверят на существование сессии basket и если ее нет, то создает ее.

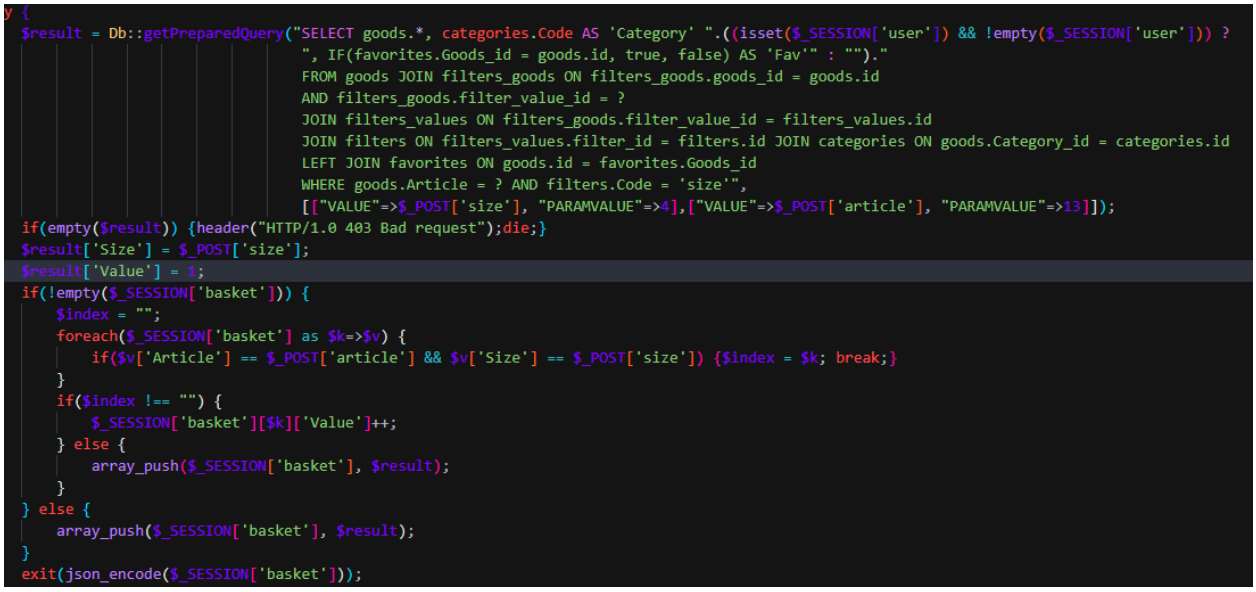
Метод полной очисти корзины, очищает все сессию корзины и возвращает пустой массив.

Метод удаления товара из корзины, если существует товар с переданым артикулам то удаляет его из корзины.

Метод установки количества выбранного товара, в зависимости от переданного действия — уменьшить (less) или прибавить (more), убавляет или прибавляет количество товара. Если товар в едином числе и передано его уменьшение, то товар удаляется.

Метод получения корзины, возвращает сессию корзины.

Метод добавления товара в корзину, передается артикул товара и выбранный размер по ним делается запрос на получение данного товара из базы данных, так же если существует сессия пользователя, к нему добавляется столбец является ли товар избранным. Результат записывается в сессию корзины, если такой товар с таким размером уже есть, то у него просто прибавляется число.



```
y {
$result = Db::getPreparedQuery("SELECT goods.*, categories.Code AS 'Category' ".((isset($_SESSION['user']) && !empty($_SESSION['user'])) ?
                                ", IF(favorites.Goods_id = goods.id, true, false) AS 'Fav'" : "")."
                                FROM goods JOIN filters_goods ON filters_goods.goods_id = goods.id
                                AND filters_goods.filter_value_id = ?
                                JOIN filters_values ON filters_goods.filter_value_id = filters_values.id
                                JOIN filters ON filters_values.filter_id = filters.id JOIN categories ON goods.Category_id = categories.id
                                LEFT JOIN favorites ON goods.id = favorites.Goods_id
                                WHERE goods.Article = ? AND filters.Code = 'size'",
                                [["VALUE"=>$_POST['size'], "PARAMVALUE"=>4],["VALUE"=>$_POST['article'], "PARAMVALUE"=>13]]);
if(empty($result)) {header("HTTP/1.0 403 Bad request");die;}
$result['Size'] = $_POST['size'];
$result['Value'] = 1;
if(!empty($_SESSION['basket'])) {
    $index = "";
    foreach($_SESSION['basket'] as $k=>$v) {
        if($v['Article'] == $_POST['article'] && $v['Size'] == $_POST['size']) {$index = $k; break;}
    }
    if($index !== "") {
        $_SESSION['basket'][$k]['Value']++;
    } else {
        array_push($_SESSION['basket'], $result);
    }
} else {
    array_push($_SESSION['basket'], $result);
}
exit(json_encode($_SESSION['basket']));
```

Рисунок 23 — Фрагмент кода блока добавления товара в корзину

Файл-класс catalog:

Метод получения каталога, совмещает в себе две отдельных функции вызывающие два больших запроса. Первая функция — получение каталога по переданым параметерам фильтрации, в зависимости от них будет прибавляться новые блоки запроса. Так же запрос на получение избранных

товаров будет выполнять эта же функция, но включая проверку на только избранные товары. Вторая функция — получение всех фильтров и их значений, но тех только тех значений под которые есть доступные товары. Далее результаты этих двух функций складываются в один массив и отправляются клиенту.

Максимально большой запрос на получение каталога при выборе всех фильтров будет следующий:

```
SELECT goods.*, GROUP_CONCAT(filters_goods.filter_value_id SEPARATOR ',') AS Size,
IF(favorites.Goods_id = goods.id, "true", "false") AS "Fav" FROM (SELECT goods.id
AS "ID" FROM goods JOIN filters_goods ON filters_goods.goods_id = goods.id JOIN
filters_values ON filters_values.id = filters_goods.filter_value_id JOIN
filters ON filters.id = filters_values.filter_id WHERE
(COALESCE(filters_values.Code, filters_values.id) = "10" OR
COALESCE(filters_values.Code, filters_values.id) = "70") AND filters.Code =
"brand" GROUP BY goods.id) AS BRANDS, (SELECT goods.id AS "ID" FROM goods JOIN
filters_goods ON filters_goods.goods_id = goods.id JOIN filters_values ON
filters_values.id = filters_goods.filter_value_id JOIN filters ON filters.id =
filters_values.filter_id WHERE (COALESCE(filters_values.Code, filters_values.id) =
"65" OR COALESCE(filters_values.Code, filters_values.id) = "65") AND filters.Code
= "type" GROUP BY goods.id) AS TYPES, (SELECT goods.id AS "ID" FROM goods JOIN
filters_goods ON filters_goods.goods_id = goods.id JOIN filters_values ON
filters_values.id = filters_goods.filter_value_id JOIN filters ON filters.id =
filters_values.filter_id WHERE (COALESCE(filters_values.Code, filters_values.id) =
"White" OR COALESCE(filters_values.Code, filters_values.id) = "Black") AND
filters.Code = "color" GROUP BY goods.id) AS COLORS, (SELECT goods.id AS "ID" FROM
goods JOIN filters_goods ON filters_goods.goods_id = goods.id JOIN filters_values
ON filters_values.id = filters_goods.filter_value_id JOIN      filters ON
filters.id = filters_values.filter_id WHERE (COALESCE(filters_values.Code,
filters_values.id) = "43" OR COALESCE(filters_values.Code, filters_values.id) =
"42") AND filters.Code = "size" GROUP BY goods.id) AS SIZES, filters_goods JOIN
goods JOIN filters_values ON filters_goods.goods_id = goods.id AND
filters_goods.filter_value_id = filters_values.id JOIN filters ON
filters_values.filter_id = filters.id AND filters.Code = "size" JOIN categories ON
goods.Category_id = categories.id LEFT JOIN favorites ON goods.id =
favorites.Goods_id WHERE goods.id = BRANDS.id AND goods.id = TYPES.id AND goods.id
= COLORS.ID AND goods.id = SIZES.ID AND (categories.Code = "Man" OR
categories.Code = "All") AND goods.Price <= 7999 AND goods.Price >= 7950 GROUP BY
goods.id ORDER BY `goods`.`Price` DESC;
```

Рисунок 24 — Запрос на получение католага со всеми фильтрациями

Запрос на получение всех фильтров:

```
SELECT filters.Filter, filters.Code, filters.Type, filters_values.id,
IF(filters.Type = 'Multi' AND filters_goods.filter_value_id = filters_values.id,
filters_values.value, IF(filters.Type = 'Multi', null, IF(filters.Code = 'price',
null, COALESCE(filters_values.value, categories.Title)))) AS Name, IF(filters.Type
= 'Multi' AND filters_goods.filter_value_id = filters_values.id,
COALESCE(filters_values.Code, filters_values.id), IF(filters.Type = 'Multi', null,
IF(filters.Code = 'price', CONCAT(MIN(goods.Price), ',', MAX(goods.Price)),
COALESCE(filters_values.Code, filters_values.id, categories.Code)))) AS ValueCode
FROM filters JOIN categories LEFT JOIN filters_values ON filters.id =
filters_values.filter_id JOIN goods LEFT JOIN filters_goods ON filters_values.id =
filters_goods.filter_value_id AND goods.id = filters_goods.goods_id AND
categories.id = goods.Category_id AND (categories.Code = 'Man' OR categories.Code
= 'All') JOIN favorites ON favorites.Goods_id = goods.id WHERE IF(filters.Code =
'price', goods.Category_id = categories.id AND (categories.Code = 'Man' OR
categories.Code = 'All'), filters.Code is NOT null) GROUP BY Name, Code ORDER BY
Type, id ASC, Name DESC;
```

Рисунок 25 — Запрос на получение всех фильтров

Метод получения отдельного товара, включает в себя так же два немальниких запроса: первый запрос делает выборку товара, его размеры, доступные цвета (другие товары); второй запрос делает выборку массива схожих товаров, и у каждого товара должны быть указаны доступные размеры. Далее два этих результата записываются в массив и отправляются клиенту.

```
$product = Db::getPreparedQuery("SELECT goods.*, Base.value AS Color, categories.Code AS Category,
                                 ".((isset($_SESSION['user']) && !empty($_SESSION['user'])) ? " IF(favorites.Goods_id = goods.id, true, false) AS Fav, " : "").
                                 GROUP_CONCAT(filters_values.value SEPARATOR '.') AS Size, Mods.Colors
                                 FROM (SELECT GROUP_CONCAT(CONCAT(goods.Article, '#', categories.Code, '#', goods.Image) SEPARATOR ',') AS Colors
                                 FROM goods_variations JOIN goods ON (goods_variations.Base_id = goods.id OR goods_variations.Variation_id = goods.id)
                                 JOIN categories ON goods.Category_id = categories.id
                                 WHERE (goods_variations.Base_Article = ? OR goods_variations.Variation_Article = ?)) AS Mods,
                                 (SELECT filters_values.value FROM goods JOIN filters_goods ON filters_goods.goods_id = goods.id
                                 JOIN filters_values ON filters_goods.filter_value_id = filters_values.id JOIN filters ON filters_values.filter_id = filters.id
                                 AND filters.Code = 'color' WHERE goods.Article = ?) AS Base,
                                 goods JOIN categories ON goods.Category_id = categories.id JOIN filters_goods ON filters_goods.goods_id = goods.id
                                 JOIN filters_values ON filters_goods.filter_value_id = filters_values.id
                                 JOIN filters ON filters_values.filter_id = filters.id AND filters.Code = 'size'
                                 LEFT JOIN favorites ON favorites.Goods_id = goods.id AND favorites.User_id = 5 WHERE goods.Article = ?",
                                 $prepaired);
$related = Db::getPreparedQuery("SELECT goods.id, goods.Title, goods.Brand, goods.Type, goods.Article, goods.Price, goods.Price_old,
                                 goods.Image, goods.SliderImages, categories.Code AS Category,
                                 ".((isset($_SESSION['user']) && !empty($_SESSION['user'])) ? " IF(favorites.Goods_id = goods.id, true, false) AS Fav, " : "").
                                 GROUP_CONCAT(filters_values.value SEPARATOR '.') AS Size FROM goods_related
                                 JOIN goods ON goods_related.Goods_id = goods.id OR goods_related.Related_id = goods.id
                                 JOIN filters_goods JOIN filters_values JOIN filters ON filters_values.filter_id = filters.id
                                 AND filters.Code = 'size' AND filters_goods.goods_id = goods.id
                                 AND filters_goods.goods_id = goods.id AND filters_goods.filter_value_id = filters_values.id
                                 JOIN categories ON goods.Category_id = categories.id
                                 LEFT JOIN favorites ON goods.id = favorites.Goods_id
                                 ".((isset($_SESSION['user']) && !empty($_SESSION['user'])) ? " AND favorites.User_id = '".$_SESSION['user']['id']."' " : "").
                                 WHERE (goods_related.Goods_Article = ? OR goods_related.Related_Article = ?)
                                 AND goods.Article != ? GROUP BY goods.id",
                                 $prepaired);
if(isset($related['id'])) $related = [$related];
exit(json_encode(["product"=>$product,"related"=>$related]));
```

Рисунок 26 — Фрагмент кода блока получения товара

Файл-класс media:

В нем существует только один метод на получение данных для главного слайдера.

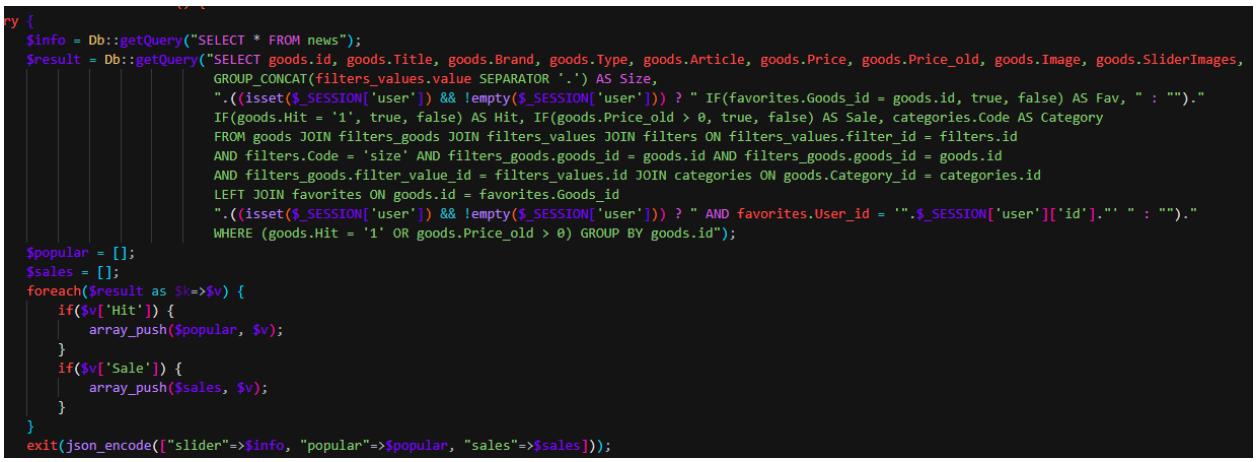


Рисунок 27 — Фрагмент кода блока получения информации об слайдере

Файл-класс callback:

Содержит функцию конструктор, котрая проверят на пустоту массива метода передачи данных POST, если он пуст выкидывается ошибка 403 Bad Request.

В нем есть два метода для получение формы обратной связи и формы быстрой подписки, так как продолжения куда эту информацию отправлять нет, то логика завершается на проверке существования полей и возврата сообщения об успешном выполнении запроса.

6.3.5 Настройка серверной безопасности

Для того что бы любой пользователь не смог свободно путешествовать по каталогу сервера, нужно дать инструкции серверу куда доступ запрещен. Для этого нужно использовать файл настройки .htaccess для Apache. Стандартный файл .htaccess который будет распологаться в корне каталога, который переадресовывает все запросы в один главный php-файл.

```
AddDefaultCharset utf-8
RewriteEngine On
Options -Indexes
RewriteCond %{SCRIPT_FILENAME} !-d
RewriteCond %{SCRIPT_FILENAME} !-f
RewriteRule (.*) index.php?$1 [L,QSA]
```

Рисунок 28 — Корневой .htaccess

Следующий файл .htaccess будет располагаться в корне папки app, там лежит вся логика сервера, структура файла коротка и понятна — запретить доступ всем.

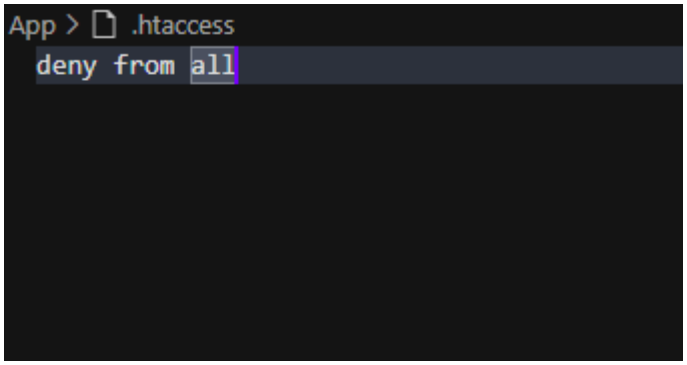


Рисунок 29 — App .htaccess

6.4 Тестирование

Все тестирование будет проходить в ручном режиме. Тестируются следующие возможности сайта:

1. Вход, регистрация, доступ к страницам;
2. Личный кабинет пользователя, проверка корректности работы валидации переданной информации пользователм, проверка изменения данных пользователя;
3. Тестирование живых элементов — слайдеров, кнопок, элементы управления, виджеты и т.п.;
4. Тестирование получения корректного контента в слайдеры;

5. Тестирование каталога, правильное отображение контента при выбранной фильтрации и сортировки, проверка корректности отображения фильтров и информации в карточке товара;

6. Проверка верно переданой информации и доп. Контента для отдельной страницы товара;

7. Тестирование функционала корзины покупок — добавление товара из разных элементов, уменьшение/прибавление количества, удаление, очистка и добавление в избранное;

8. Тестирование отправок форм, проверка корректности работы валидации форм;

9. Тестирование доступа и безопасности сервера.

Заключение

В ходе выполнения курсового проекта был спроектирован и разработан веб-приложение интернет-магазина по продаже обуви. Функциональное проектирование проводилось с помощью диаграмм нотации IDEF. Объектно-ориентированное проектирование проводилось с помощью диаграмм нотации UML. Перед началом проектирования были определены основные функциональные требования, которые в ходе работы были занесены в Техническое задание.

Разработка программного продукта проводилась в редакторе исходного кода Visual Studio Code на языках программирования: JavaScript с использование библетек React, React Router и Redux; PHP с использованием библеотеки Composer. Для разработки дизайна использовалось веб-приложение Figma. Для разбраотки базы данных использовалась СУБД MySQL и веб-приложение phpMyAdmin.

Было проведено описание структуры интернет-магазина, кода клиентской и серверной части, базы данных, связей базы данных с интернет-магазином и пользовательского интерфейса в целом.

В результате выполнения проекта были получены новые знания, новый опыт, а также готовый интернет-магазин, полностью соответствующий поставленным требованиям и готовый к будущим улучшениям.

Список используемых источников

1. https://dev.mysql.com/doc/ — интернет документация MySQL;
2. https://developer.mozilla.org/ru/docs/Web — Руководство по веб-разработки и веб-технологиям, документация к языку программирования JavaScript;
3. https://fonts.google.com/ — Шрифты Google;
4. https://fusionbrain.ai/editor/ — Нейросеть для создания изображений;
5. https://httpd.apache.org/docs/2.0/ — Документация к HTTP серверу Apache;
6. https://icons.getbootstrap.com/ — Иконки Bootstrap;
7. https://reactrouter.com/en/main — интернет документация библеотеки React Router;
8. https://redux-toolkit.js.org/ — интернет документация библеотеки Redux;
9. https://ru.legacy.reactjs.org/ — интернет документация библеотеки React;
10. https://www.php.net/manual/ru/ — руководство по языку программирования PHP;
11. Динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 6-е изд. | Никсон Робин, 2023г., 1200 стр.
12. Гагарина Л.Г., Кокорева Е.В., Сидорова-Виснадул Б.Д. Технология разработки программного обеспе-чения: учеб. пособие [Электронный ресурс]. – М.: ИНФРА-М, 2019. – 400 с. Режим доступа: http://znanium.com/catalog/product/1011120.
13. Зубкова, Т. М. Технология разработки программного обеспечения : учебное пособие / Т. М. Зубкова. — Санкт-Петербург : Лань, 2022. — 324 с. — ISBN 978-5-8114-3842-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/206882 (дата обращения: 10.12.2023). — Режим доступа: для авториз. пользователей.