**Содержание**

|  |  |
| --- | --- |
| Введение | 4 |
| 1. Постановка задачи | 6 |
| 1.1. Обзор аналогичных решений | 6 |
| 1.2. Техническое задание | 7 |
| 1.3. Выбор средств реализации | 8 |
| 1.4. Вывод | 9 |
| 2. Проектирование проекта | 10 |
| 2.1. Выбор способа верстки | 10 |
| 2.2. Выбор стилевого оформления | 10 |
| 2.3. Выбор шрифтового оформления | 10 |
| 2.4. Разработка логотипа | 10 |
| 2.5. Разработка пользовательских элементов | 11 |
| 2.6. Выводы | 11 |
| 3. Реализация структуры веб-сайта | 12 |
| 3.1. Структура HTML-документа | 12 |
| 3.2. Добавление таблиц стилей SCSS и CSS | 15 |
| 3.3. Использование стандартов XML (SVG) | 15 |
| 3.4. Управление элементами DOM | 16 |
| 3.5. Выводы | 18 |
| 4. Тестирование веб-сайта | 19 |
| 4.1. Адаптивный дизайн веб-сайта | 19 |
| 4.2. Кроссбраузерность веб-сайта | 19 |
| 4.3. Руководство пользователя | 19 |
| 4.4. Выводы | 20 |
| Заключение | 21 |
| Список использованных литературных источников | 22 |
| Приложение А Прототипы веб-страниц | 23 |
| Приложение Б Макет структуры веб-сайта | 28 |
| Приложение В UseCase диаграмма | 33 |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Введение**

В эпоху цифровой трансформации персональные компьютеры превратились в универсальные инструменты, одновременно служащие рабочими станциями, образовательными платформами, творческими студиями и мультимедийными центрами. Их роль в современном мире сложно переоценить: от выполнения профессиональных задач в сферах дизайна, программирования и инженерии до повседневного использования в обучении, развлечениях и удалённой работе. Однако парадокс современного компьютерного рынка заключается в том, что при невероятном изобилии комплектующих и конфигураций процесс создания оптимальной системы остаётся чрезвычайно сложным даже для технически подкованных пользователей. Это обусловлено необходимостью учитывать множество взаимосвязанных параметров: от базовых требований совместимости компонентов до тонкостей теплового баланса, энергопотребления и будущей масштабируемости системы.

Рост рынка ПК и связанных с ним технологий продолжает набирать обороты. По данным аналитиков (IDC, Gartner, 2025), мировой рынок компьютерных комплектующих к 2025 году достигнет $150 млрд, при этом 65% покупателей сталкиваются с трудностями при самостоятельном подборе компонентов. Основные проблемы включают:

* **Сложность выбора компонентов – даже опытные пользователи могут ошибиться при подборе процессора, материнской платы, оперативной памяти и видеокарты из-за постоянно обновляющихся стандартов (DDR5, PCIe 5.0, новые сокеты);**
* **Неочевидные ограничения совместимости – например, физические размеры корпуса и системы охлаждения, требования к блоку питания, поддержка конкретных технологий (Ray Tracing, DLSS, Resizable BAR);**
* **Динамика цен и доступность комплектующих – из-за колебаний спроса, логистических проблем и дефицита некоторых компонентов (например, видеокарт в 2020–2022 гг.) пользователи вынуждены постоянно пересматривать свои конфигурации.**

Современные сервисы по подбору конфигураций ПК сталкиваются с рядом системных проблем, существенно ограничивающих их эффективность. Во-первых, отсутствуют интеллектуальные инструменты проверки совместимости – большинство платформ предлагают лишь базовые фильтры, не учитывающие все возможные конфликты, такие как необходимость обновления BIOS для поддержки новых процессоров или тонкие нюансы взаимодействия компонентов. Во-вторых, технологическая база таких сервисов заметно отстаёт от развития аппаратных стандартов: средний возраст их технологического стека достигает 7 лет, что делает интерфейсы неудобными, а функционал – ограниченным. В-третьих, наблюдается недостаточная мобильная адаптация – только 38% существующих решений полноценно оптимизированы для смартфонов, несмотря на то, что более 60% пользователей ищут информацию о сборке ПК именно с мобильных устройств (StatCounter, 2024). Наконец, ключевой проблемой остаётся отсутствие персонализированных решений: как показывают исследования (PwC, 2023), 82% пользователей хотели бы получать индивидуальные конфигурации, учитывающие их бюджет, специфические задачи (гейминг, монтаж видео, машинное обучение) и персональные предпочтения (уровень шума, дизайн с RGB-подсветкой, компактность корпуса).

Всё это создаёт острую потребность в современном, удобном и интеллектуальном сервисе, который не только автоматизирует подбор комплектующих, но и предлагает оптимальные решения на основе актуальных рыночных данных, тестов производительности и показателей энергоэффективности. Внедрение технологий машинного обучения, а также интеграция с базами данных реальных сборок позволят значительно упростить процесс конфигурирования ПК, снизив порог вхождения для новичков и сэкономив время опытных пользователей. Такой подход сможет устранить существующие недостатки, предлагая пользователям по-настоящему умные, адаптивные и удобные инструменты для создания идеальной системы под их нужды.

Цель проекта заключается в разработке принципиально нового веб-решения "PC\_Configurator", которое выходит за рамки традиционных систем подбора компонентов. Наше приложение призвано стать интеллектуальной платформой, объединяющей глубокий технический анализ с интуитивно понятным интерфейсом. В отличие от существующих аналогов, система будет не просто проверять базовую совместимость деталей, а комплексно оценивать их взаимодействие, учитывая как текущие технические параметры, так и перспективы будущего апгрейда. Особое внимание уделяется созданию адаптивного механизма рекомендаций, который учится на предпочтениях пользователя и рыночных тенденциях, предлагая по-настоящему оптимальные конфигурации под конкретные задачи и бюджет. **Социальная значимость проекта заключается в демократизации доступа к современным компьютерным технологиям, позволяя как новичкам, так и профессионалам создавать идеально сбалансированные системы без необходимости глубокого погружения в технические нюансы. % по сравнению с традиционными методами.**

Разрабатываемый конфигуратор ПК обладает значительным потенциалом для трансформации рынка компьютерных сборок. В перспективе платформа может быть дополнена следующими инновационными функциями:

* **Искусственный интеллект для прогнозирования производительности нейросетевая модель сможет предсказывать реальную производительность системы в конкретных задачах на основе анализа тысяч аналогичных сборок и бенчмарков;**
* **VR-режим визуализации сборки** – **технология виртуальной реальности поможет пользователям предварительно оценить визуальную составляющую будущей системы.**

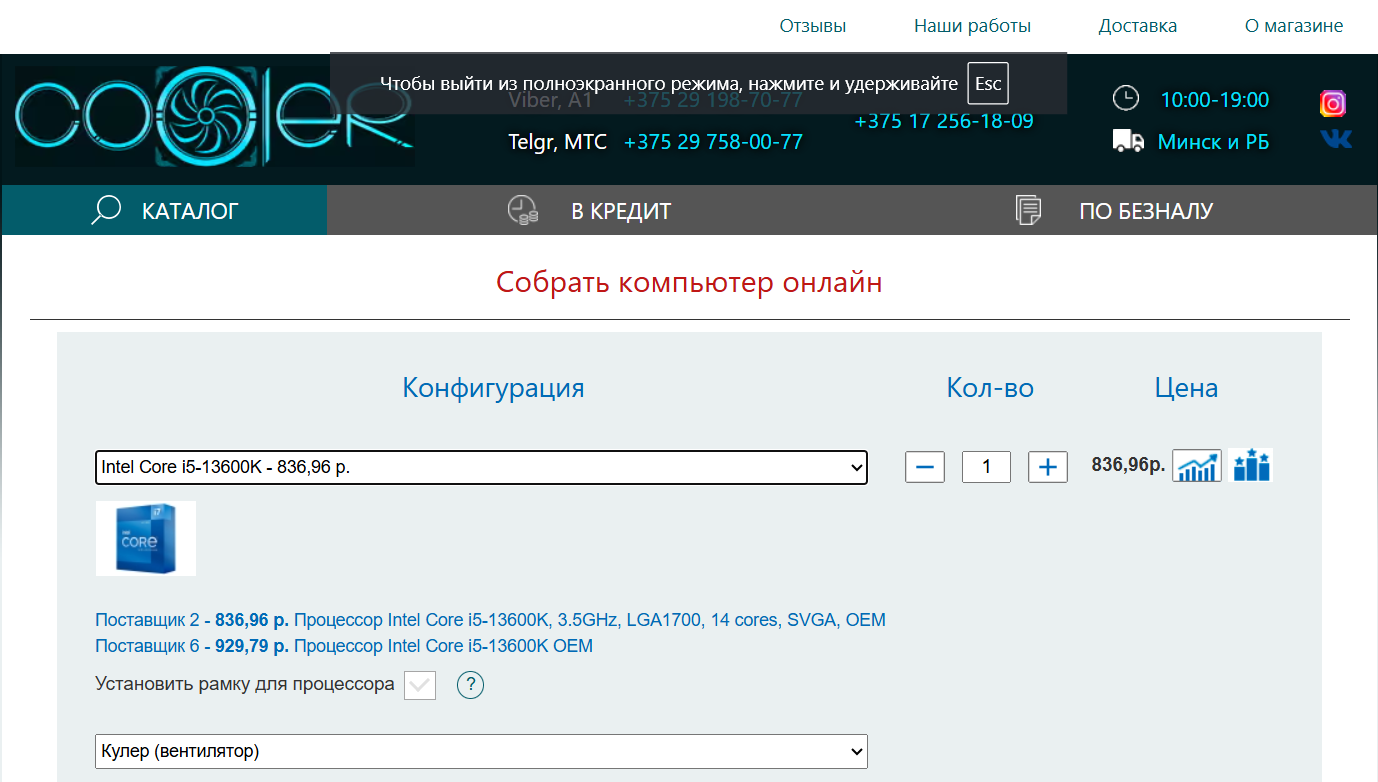
**1. Постановка задачи**

**1.1. Обзор аналогичных решений**

В данном разделе будет представлен анализ существующих решений на рынке конфигураторов ПК, включая их функционал, интерфейсы и пользовательский опыт.

**Ниже буду представлены сайты, схожие с PC\_Configurator.**

**Cooler.by: предлагает базовую проверку совместимости, но интерфейс плохо адаптирован под мобильные устройства, страница конфигуратора сайта представлена на рисунке 1.1:**

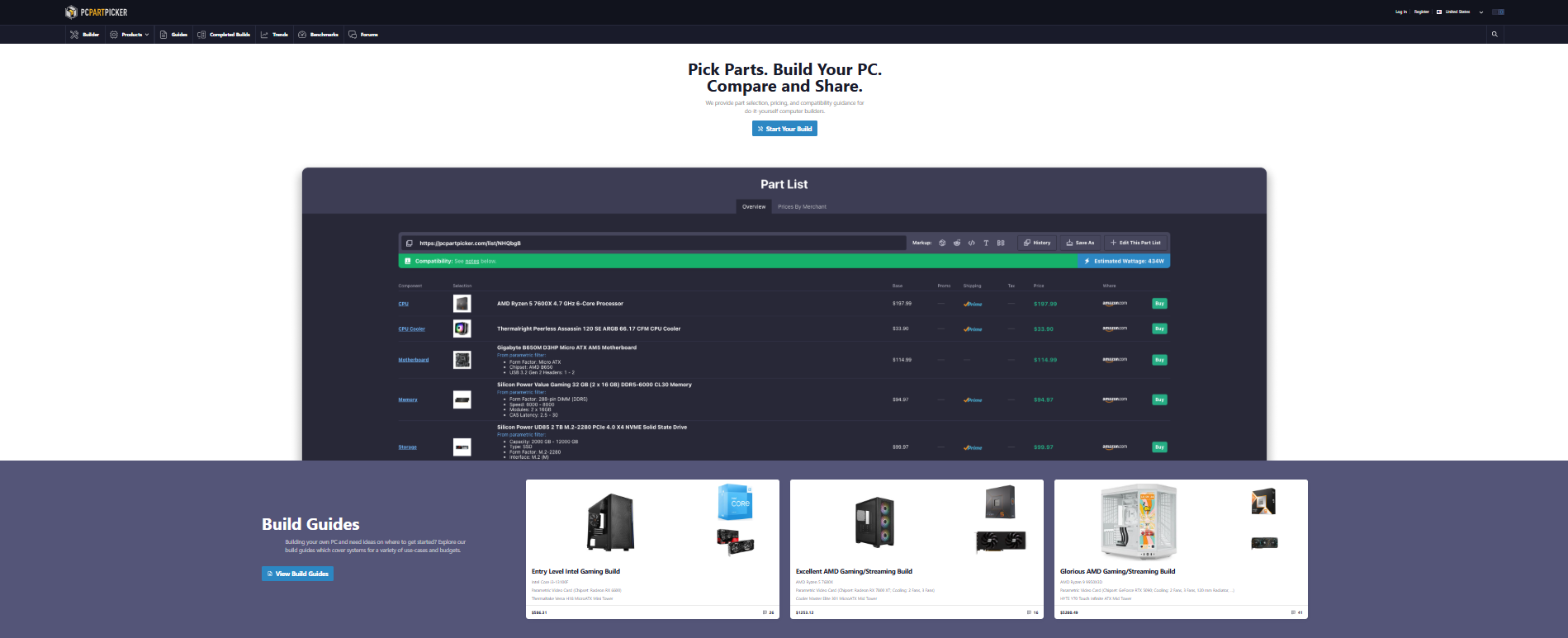
****

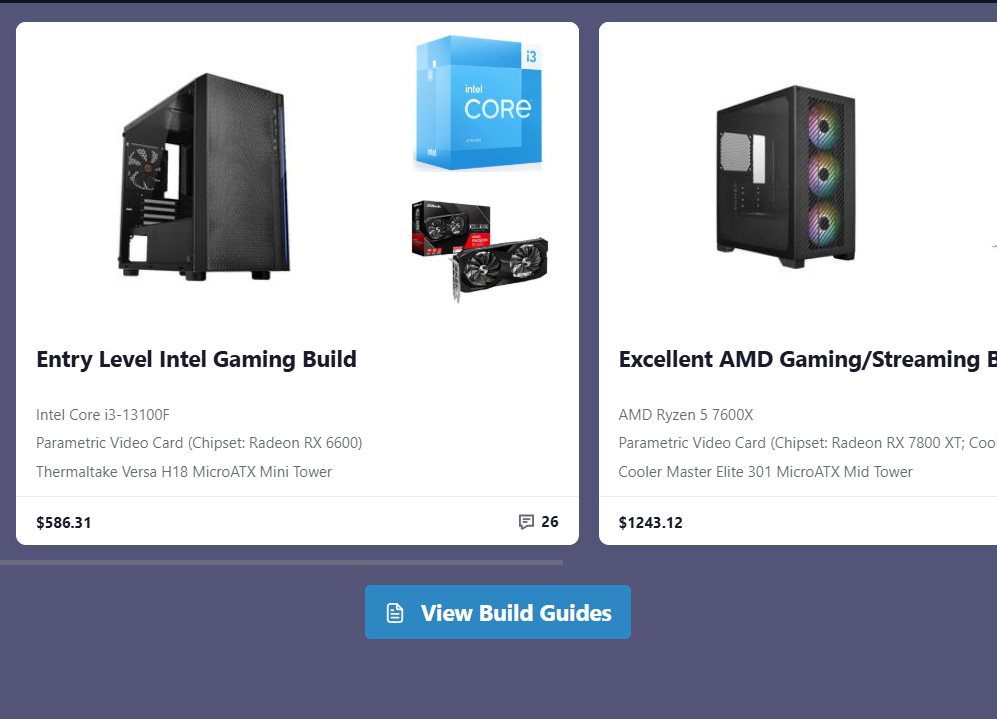
**а б**

а – компьютерная версия, б – мобильная версия;

Рисунок 1.1 - Страница конфигуратора сайта**Cooler.by**

**PCpartpicker.com: присутствует функционал выбора компонентов, но интерфейс имеет недостатки, связанные с размерами шрифта и относительно устаревшим дизайном. Пример главной страницы сайта изображён на рисунке 1.2:**

****

****

**а б**

а – компьютерная версия, б – мобильная версия;

Рисунок 1.2 – Главная страница сайта**PCpartpicker.com**

**1.2. Техническое задание**

Конфигуратор ПК представляет собой интуитивно понятный инструмент, который шаг за шагом ведёт пользователя через процесс сборки компьютера, предлагая оптимальные варианты комплектующих на каждом этапе. Помимо базовых компонентов, система также учитывает дополнительные факторы, такие как тепловыделение процессора и рекомендуемую систему охлаждения, что помогает избежать перегрева и других технических проблем. В случае выбора несовместимых деталей, конфигуратор не только выделяет ошибку, но и предлагает альтернативные варианты, объясняя причины замены. Готовые сборки можно не только сохранять в корзину, но и делиться ими с другими пользователями или экспортировать в виде спецификации для консультации со специалистами.

Каталог товаров разработан с учётом потребностей как новичков, так и опытных сборщиков, предоставляя расширенные фильтры, включая мощность блока питания, поддерживаемые стандарты оперативной памяти и даже уровень шума компонентов. Функция сравнения товаров дополнена интерактивными графиками производительности, помогающими наглядно оценить разницу между моделями. Детальные карточки товаров содержат не только технические характеристики, но и реальные отзывы пользователей, рейтинги надёжности и советы по эксплуатации. Корзина адаптируется под разные сценарии покупки: можно сохранить несколько конфигураций для сравнения или постепенно дополнять сборку новыми компонентами. При оформлении заказа система анализирует выбранные товары и предлагает сопутствующие услуги, такие как профессиональная сборка, настройка ПО или расширенная гарантия.One-Click оформление интегрировано с популярными платёжными системами, что ускоряет процесс, а также позволяет использовать сохранённые данные для повторных покупок. Для корпоративных клиентов и постоянных покупателей предусмотрены дополнительные опции, включая отложенный платёж и персонализированные скидки на комплектующие.

**Требования к интерфейсу:**

**Дизайн:**

* **Стиль: киберпанк с неоновыми акцентами (#00f7ff, #b400ff);**
* **Адаптивная верстка (Mobile First подход);**
* **Интерактивные элементы;**
* **Анимированные кнопки и карточки товаров;**
* **Подсветка совместимых компонентов;**
* **3D-эффекты при наведении;**
* **Типографика;**
* **Заголовки: шрифт Orbitron (700);**
* **Основной текст: Share Tech Mono (400);**
* **Кнопки и акценты: Audiowide (400);**

**Технические требования:**

**Frontend-разработка:**

**Для реализации клиентской части приложения используются современные веб-технологии, обеспечивающие высокую производительность, отзывчивый интерфейс и удобство разработки.**

**Основные технологии:**

**HTML5 с обязательным использованием семантической разметки. Это обеспечивает правильную структуру документа, улучшает SEO-оптимизацию и делает контент более доступным для вспомогательных технологий.**

**SCSS (Sass) в качестве препроцессора CSS с модульной структурой стилей. Такой подход позволяет организовать код более эффективно, использовать переменные, миксины и вложенные правила, что значительно упрощает поддержку и масштабирование стилей.**

**JavaScript (ES6+) для реализации всей бизнес-логики приложения. Современные возможности языка (стрелочные функции, деструктуризация, async/await) помогут создать чистый и поддерживаемый код.**

**Инструментарий разработки:**

* **Figma** – **основной инструмент для создания прототипов интерфейса и дизайн-системы. Позволяет быстро тестировать UX-решения и согласовывать их с командой.**
* **Git** – **система контроля версий для командной работы. Использование веток (Git Flow) обеспечит стабильность разработки и возможность отката изменений при необходимости.**

**1.3. Выбор средств реализации программного продукта**

**Для разработки конфигуратора ПК были выбраны современные и надежные технологии, обеспечивающие высокую производительность, удобство поддержки и масштабируемость проекта. Основной упор сделан на использование проверенных решений, позволяющих создавать интуитивно понятный интерфейс, эффективную бизнес-логику и стабильную работу системы.**

**Frontend-разработка:**

**Клиентская часть приложения реализована с применением следующих технологий:**

* **HTML5 с семантической разметкой — обеспечивает четкую структуру документа, улучшает индексацию поисковыми системами (SEO) и повышает доступность для пользователей с ограниченными возможностями.**
* **CSS3 + SCSS (Sass) — используется для стилизации интерфейса с применением препроцессора, что позволяет:**
* **Организовать код в модули для лучшей поддержки;**
* **Использовать переменные (например, для цветовых схем, включая неоновые элементы);**
* **Применять миксины и вложенные правила, сокращая дублирование кода;**
* **Легко масштабировать стили при развитии проекта;**
* **JavaScript (ES6+) — основной язык для реализации клиентской логики, включая:**

1. **Динамическое взаимодействие с пользователем (фильтрация комплектующих, расчет стоимости);**
2. **Валидацию данных и обработку событий.**

Такой набор технологий обеспечивает оптимальный баланс между производительностью, скоростью разработки и возможностью дальнейшего расширения функционала. Кроме того, выбранные решения обладают активным сообществом и обширной документацией, касательно новых комплектующих, что упрощает поддержку проекта.

**1.4. Выводы**

**Конфигуратор ПК — это интеллектуальная система, которая превращает сложный процесс подбора комплектующих в простой и интуитивно понятный диалог между пользователем и технологией. За счет глубокой интеграции с актуальными базами данных компонентов, сервис в режиме реального времени анализирует сотни технических параметров, автоматически исключая несовместимые комбинации и предлагая оптимальные альтернативы.**

**Особое внимание уделено адаптивности интерфейса — система динамически подстраивает уровень детализации информации под опыт пользователя. Новички получают лаконичные подсказки с базовыми характеристиками, в то время как опытные сборщики могут активировать расширенный режим с отображением таких специализированных параметров, как напряжение питания RAM, поддержка конкретных версий PCIe или даже совместимость с экзотическими системами охлаждения.**

**Функция визуализации компонентов реализована через технологию дополненной реальности, позволяющую "примерить" выбранные комплектующие к реальному окружению. Это особенно полезно при оценке физических размеров компонентов — пользователь может убедиться, что мощная видеокарта поместится в его рабочем пространстве или что выбранный кулер не будет конфликтовать с боковой панелью корпуса.**

**Для бизнеса такой подход означает не только снижение нагрузки на службу поддержки, но и возможность собирать уникальные аналитические данные о поведении пользователей. Система фиксирует типичные ошибки при сборке, популярные комбинации комплектующих и даже "болевые точки", где пользователи чаще всего прерывают процесс конфигурации.**

**результатов работы собранного ПК в различных сценариях — от профессиональных рабочих нагрузок до требовательных игр.**

**Такой комплексный подход превращает конфигуратор из простого инструмента подбора в интеллектуальную платформу, которая не только помогает совершить покупку, но и обучает пользователей, формируя новое кач**ество сервиса в сегменте компьютерных компонентов.

**2. Проектирование страниц веб-сайта**

**2.1. Выбор способа верстки**

Для достижения высокой гибкости макета и обеспечения корректного отображения интерфейса на различных устройствах была выбрана адаптивная верстка с использованием CSS Grid и Flexbox. Эти технологии позволяют выстраивать сетки, адаптирующиеся под размеры экрана, и задавать поведение элементов без использования медиа-запросов в большинстве типовых сценариев. Применяется методология Mobile First, при которой дизайн создается в первую очередь для мобильных устройств, а затем масштабируется для планшетов и десктопов.

**2.2. Выбор стилевого оформления**

Цветовая схема оформлена в стиле киберпанк с доминирующими цветами: неоново-голубой (#00f7ff) и фиолетовый (#b400ff) на темном фоне (#0f0f1a). Такое оформление создает эффект современного высокотехнологичного интерфейса. Также используются полупрозрачные слои, свечения, рамки и градиенты, подчеркивающие стиль интерфейса. Особое внимание уделено контрастности и читаемости, чтобы обеспечить удобство восприятия.

**2.3. Выбор шрифтового оформления**

Для отображения текста в интерфейсе применяются следующие шрифты:

Заголовки: **Orbitron**, шрифт с геометрической формой символов, идеально подходящий для футуристического дизайна. Основной текст: **Share Tech Mono**, моноширинный шрифт. Подключение шрифтов осуществляется через Google Fonts с последующей настройкой веса и межстрочного интервала для повышения читаемости.

**2.4. Разработка логотипа**

Логотип создан вручную в редакторе Figma. Он представляет собой стилизованное изображение названия сайта с неоновыми дорожками, образующими аббревиатуру «PC». Элементы логотипа выполнены с использованием линейного градиента и свечения по контуру. Логотип адаптируется по размеру в зависимости от разрешения экрана и поддерживает svg-формат для максимальной четкости. Пример логотипа изображён на рисунке 2.1:



Рисунок 2.1 **— Логотип сайта**

**2.5. Разработка пользовательских элементов**

При разработке пользовательских элементов интерфейса был реализован комплексный подход, обеспечивающий как визуальную согласованность, так и высокий уровень юзабилити. Все интерактивные компоненты - кнопки, поля ввода, выпадающие списки и переключатели - создавались с единой дизайн-системой, где особое внимание уделялось интерактивным состояниям: плавным hover-эффектам с неоновым свечением, четко выраженным active-состояниям с тактильной обратной связью и заметным focus-индикаторам для клавиатурной навигации. Техническая реализация основана на модульной системе SCSS с использованием переменных для управления размерами, цветовой палитрой (включая основные акцентные цвета #00f7ff и #b400ff), радиусами скругления и эффектами теней, что обеспечивает легкую поддержку и масштабируемость стилей. Для мобильных устройств предусмотрены увеличенные области касания (не менее 48×48px), оптимизированные touch-события и упрощенные формы ввода, адаптированные под экранные клавиатуры. Особый акцент сделан на доступности интерфейса - все компоненты соответствуют стандартам WCAG 2.1 по контрастности, снабжены ARIA-атрибутами для скринридеров и поддерживают альтернативные способы взаимодействия, что в совокупности создает инклюзивную среду для всех категорий пользователей.

**2.7. Выводы**

Визуальная часть приложения разрабатывалась с упором на удобство использования и современный дизайн. Адаптивный интерфейс корректно отображается на любых устройствах благодаря продуманной верстке. Темная цветовая схема с неоновыми элементами создает технологичный образ, сохраняя удобство восприятия информации. Интерфейс реагирует на действия пользователя плавными анимациями, помогая ориентироваться в процессе сборки. Все элементы быстро загружаются и работают даже на слабых устройствах. Дизайн сочетает минимализм с необходимыми техническими деталями, постепенно раскрывая сложные параметры по мере углубления в конфигурацию. Простота навигации и понятная структура делают процесс подбора комплектующих доступным для пользователей с разным уровнем подготовки. Визуальные подсказки и интерактивные элементы направляют пользователя, помогая принимать обоснованные решения при выборе компонентов.

**3. Реализация структуры веб-сайта**

**3.1. Структура HTML-документа**

HTML-документ веб-приложения "Конфигуратор ПК" соответствует современным стандартам семантической разметки и адаптивности. Ниже представлены ключевые части документа с пояснениями.

**На листинге 3.1 изображена базовая структура html-документа, состоящая из самых необходимых структурных тегов.**

|  |
| --- |
| <!DOCTYPE html>  <html lang="ru">  <head>  <meta charset="UTF-8">  <meta name="viewport"content="width=device-width,  initial-scale=1.0">  <title>PC Config | Конфигуратор ПК</title>  <!-- Подключение стилей и шрифтов -->  </head>  <body class="cyber-body">  <!-- Основное содержимое страницы -->  <script src="..."></script><!-- Подключение скриптов -->  </body>  </html> |

Листинг 3.1 **—** Базовый каркас

HTML-страницы также содержат ссылки, которые подключают внешние ресурсы с помощью тега link, такие как CSS и Bootstrap. Пример подключения внешних файлов показан на листинге 3.2:

|  |
| --- |
| <head>  ...  <link href="<!--Подключение Bootstrap-->"rel="stylesheet"> <link rel="stylesheet"href=<!-- Подключение CSS-файлов -->">  <link href="<!--Подключение шрифтов-->"rel="stylesheet">  </head> |

**Листинг 3.2 — Подключение внешних ресурсов**

В данном листинге представлен тег main, который является основным контейнером для содержимого веб-страницы, посвященной сборке персональных компьютеров. Этот блок имеет класс cyber-main, указывающий на киберстилистику оформления. Внутри него расположены два ключевых раздела: герой-секция и блок с функциональными возможностями. Герой-секция содержит эффектный заголовок с градиентным и мерцающим текстом ("Собери свой идеальный ПК"), под которым находится интерактивная область сборки ПК (cyber-pc-animation) с визуализацией компонентов (процессор, видеокарта и другие). Ниже размещена секция возможностей, состоящая из карточек (cyber-feature-card). Каждая карточка включает иконку (например, иконку процессора из библиотеки Bootstrap Icons), заголовок и

|  |
| --- |
| <main class="cyber-main">  <div class="cyber-hero">  <h1class="cyber-title">  <span class="cyber-text-gradient">Собери</span>свой  <span class="cyber-text-flicker">идеальный ПК</span>  </h1>  <div class="cyber-pc-animation">  <div class="pc-component cpu"></div>  <div class="pc-component gpu"></div>  ...  </div>  </div>    <section class="cyber-features">  <div class="cyber-feature-card">  <div class="cyber-feature-icon"><iclass="bi bi-cpu"></i></div>  <h3>Автоподбор компонентов</h3>  <p>Умная система подберет совместимые комплектующие.</p>  </div>  ...  </section>  </main> |

краткое описание функции.

**Листинг 3.3 — Структура тега main**

**3.2. Добавление таблиц стилей SCSS и CSS**

Использование **SCSS** в проекте обеспечивает гибкость и поддерживаемость стилей за счёт:

* **Модульности (разделение на части: переменные, миксины, компоненты);**
* **Переменных (цвета, шрифты, отступы);**
* **Миксинов (переиспользуемые блоки кода);**
* **Вложенных правил (упрощение структуры).**

SCSS-миксины обеспечивают стилизацию элементов интерфейса. Первый создает градиентный текст, используя linear-gradient с обрезкой по контуру текста. Второй добавляет анимацию при наведении: плавное смещение элемента вверх на 5px и неоновую тень. Оба миксина поддерживают киберпанк-стиль проекта, используют современные CSS-свойства и оптимизированы для производительности. Применяются для заголовков, кнопок и интерактивных элементов. Пример показан на листинге 3.4:

|  |
| --- |
| @mixin gradient-text($color1, $color2) {  background: linear-gradient(90deg, $color1, $color2);  -webkit-background-clip: text;  background-clip: text;  color: transparent;  }  @mixin hover-effect{  transition: transform0.3s, box-shadow0.3s;  &:hover{  transform: translateY(-5px);  box-shadow: 010px20pxrgba($cyber-cyan, 0.2);  }  } |

**Листинг 3.4 — Пример создания миксинов**

Данный SCSS-код определяет стили для киберпанк-кнопок с двумя вариантами оформления. Основная кнопка использует диагональный градиент с эффектом свечения при наведении за счет тени с 50% прозрачностью. Код отличается компактной записью с использованием вложенности SCSS и &:hover-псевдоклассов для описания состояний. Пример изображён на листинге 3.5:

|  |
| --- |
| .btn-cyber{  &-primary{  background: linear-gradient(135deg, $cyber-cyan, $cyber-blue);  border: none;  &:hover{  box-shadow: 08px20pxrgba($cyber-cyan, 0.5);  }  }  &-outline{  border: 2pxsolid$cyber-cyan;  background: transparent;  &:hover{  background: $cyber-cyan;  }  }  } |

Листинг 3.5 **— Использование псевдоклассов**

Следующий SCSS-код реализует адаптивную верстку для двух ключевых компонентов интерфейса. Для заголовка используется прогрессивное увеличение отступов - базовый отступ на мобильных устройствах удваивается на экранах шире 768px, что создает более сбалансированную композицию на десктопах. Сетка каталога (.catalog-grid) построена по принципу mobile-first: изначально одноколоночный макет на мобильных устройствах трансформируется в двухколоночный на планшетах (от 576px) и трехколоночный на десктопах (от 992px). Реализация использует современные подходы: CSS Grid для гибкого размещения элементов, SCSS-переменные для единообразия стилей и медиа-запросы с четко определенными контрольными точками (576px, 768px, 992px), что в совокупности обеспечивает плавную адаптацию интерфейса под различные разрешения экранов и оптимальное представление контента на всех типах устройств. Пример показан на листинге 3.6:

|  |
| --- |
| .cyber-header{  padding: $spacing-md;  @media(min-width: 768px) {  padding: $spacing-md\*2;  }  }  .catalog-grid{  grid-template-columns: 1fr;  @media(min-width: 576px) {  grid-template-columns: repeat(2, 1fr);  }  @media(min-width: 992px) {  grid-template-columns: repeat(3, 1fr);  }  } |

Листинг 3.6 **— Использование псевдоклассов**

**3.3. Использование стандартов XML (SVG)**

Этот SVG-код создает иконку корзины покупок с перечеркнутым товаром. В одном компактном path-элементе объединены: символ "Х" (перечеркнутый товар), сама корзина и два круга-колеса. Размеры 100x100 px с масштабируемым viewBox 24x24 обеспечивают четкое отображение на любых экранах. Минималистичный дизайн без лишних элементов позволяет легко менять цвет и размер через CSS. Идеально подходит для интерфейсов интернет-магазинов, особенно в современных лаконичных дизайнах. Пример SVG - картинки изображён на листинге 3.7:

|  |
| --- |
| <svg width="100"height="100"viewBox="0 0 24 24">  <path d="M22.73 22.73L2.77 2.77 2 2l-.73-.73L0 2.54l4.39 4.39 2.21 4.66-1.35 2.45c-.16.28-.25.61-.25.96 0 1.1.9 2 2 2h7.46l1.38 1.38c-.5.36-.83.95-.83 1.62 0 1.1.89 2 1.99 2 .67 0 1.26-.33 1.62-.84L21.46 24l1.27-1.27zM7.42 15c-.14 0-.25-.11-.25-.25l.03-.12.9-1.63h2.36l2 2H7.42zm8.13-2c.75 0 1.41-.41 1.75-1.03l3.58-6.49c.08-.14.12-.31.12-.48 0-.55-.45-1-1-1H6.54l9.01 9zM7 18c-1.1 0-1.99.9-1.99 2S5.9 22 7 22s2-.9 2-2-.9-2-2-2z"/>  </svg> |

Листинг 3.7 **— SVG-изображение корзины**

**Пример SVG-иконки предупреждения, показано на листинге 3.8:**

|  |
| --- |
| <svg width="50"height="50"viewBox="0 0 24 24">  <path d="M12 2C6.48 2 2 6.48 2 12s4.48 10 10 10 10-4.48 10-10S17.52 2 12 2zm1 15h-2v-2h2v2zm0-4h-2V7h2v6z"fill="#e74c3c"/>  </svg> |

Листинг 3.8 **— SVG-иконка предупреждения**

Данный XML-документ представляет собой стандартизированный формат для хранения информации о компонентах компьютерной системы. Корневой элемент <components> служит контейнером для всего перечня комплектующих. Каждый отдельный компонент (в примере - материнская плата) описывается собственным тегом с уникальным идентификатором в атрибуте id. На листинге 3.9 показан пример xml-структуры компонента:

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <components>  <motherboard id="mb1">  <name>PRIME B660M-K</name>  <brand>ASUS</brand>  <price>350</price>  <socket>LGA1700</socket>  <formFactor>Micro-ATX</formFactor>  </motherboard>  </components> |

Листинг 3.9 **— Данные о компоненте motherboard**

**3.4. Управление элементами DOM**

Этот JavaScript-код реализует функционал мобильного меню (бургер-меню). Сначала код находит два элемента на странице: иконку бургер-меню с классом .burger-menu и навигационное меню с классом .nav, сохраняя их в переменные burgerMenu и nav соответственно. При клике на бургер-меню срабатывает обработчик события, который переключает класс active как у самой иконки (что обычно меняет её внешний вид, например, превращает в крестик), так и у навигационного меню (что обычно приводит к его появлению или исчезновению с экрана). Весь механизм работы основан на добавлении и удалении класса active, а конкретные визуальные изменения при этом определяются в CSS-стилях. Реализация бургер-меню показана на листинге 3.10:

|  |
| --- |
| **constburgerMenu=document.querySelector('.burger-menu');**  **constnav=document.querySelector('.nav');**  **if(burgerMenu&&nav) {**  **burgerMenu.addEventListener('click', function() {**  **this.classList.toggle('active');**  **nav.classList.toggle('active');**  **});**  **}** |

Листинг 3.10 **— Бургер-меню**

|  |
| --- |
| functionupdateCartCount() {  constcart=JSON.parse(localStorage.getItem('cart')) ||[]; constcartCountElements=document.querySelectorAll('.cart-count');  if(cartCountElements.length>0) {  consttotalItems=cart.reduce((sum, item) =>sum+item.quantity, 0); cartCountElements.forEach(element=>{  element.textContent=totalItems; element.style.display=totalItems>0?'inline-block':'none';  });  }  } |

Следующий JavaScript-код реализует функцию обновления счетчика товаров.Функция получает текущее содержимое корзины из локального хранилища localStorage, преобразуя JSON-строку в массив объектов, или использует пустой массив, если данные отсутствуют. Затем она находит все элементы на странице с классом cart-count, которые отвечают за отображение количества товаров. Если такие элементы существуют, функция вычисляет количество товаров в корзине, суммируя значения quantity для каждого товара. Значения записываются в текст каждого элемента-счетчика, после чего их видимость изменяется - счетчики отображаются только при наличии товаров в корзине, в противном случае скрываются. Пример работы с локальным хранилищем показан в листинге 3.11:

Листинг 3.11 **— Фунция-счётчик цены**

**3.5. Выводы**

**Разработка структуры веб-сайта создала надежную основу для реализации как визуальной составляющей, так и функциональных возможностей приложения. Применение современных технологий, включая SCSS для стилизации, SVG для графики и семантической верстки, обеспечило комплексный подход к разработке, гарантирующий высокую степень масштабируемости и удобство дальнейшего сопровождения проекта. Использование SCSS позволило организовать модульную систему стилей с переменными и миксинами, что упрощает поддержку и единообразие дизайна. Векторная графика в формате SVG обеспечивает четкое отображение на любых устройствах с поддержкой интерактивности и анимации. Семантическая верстка улучшила доступность контента и SEO-оптимизацию, а также облегчила работу с кодом. Продуманная архитектура проекта отличается модульностью компонентов, четким разделением логики и представления, что значительно упрощает процесс тестирования и дальнейшего расширения функциональности. Такой подход не только отвечает текущим требованиям, но и создает прочный фундамент для будущего развития приложения, включая адаптацию под новые платформы и интеграцию дополнительных сервисов, сохраняя при этом гибкость и производительность системы на всех этапах жизненного цикла проекта.**

**4. Тестирование веб-сайта**

**4.1. Адаптивный дизайн веб-сайта**

**Была проведена комплексная проверка отображения и взаимодействия с веб-приложением на различных устройствах:**

1. **Смартфоны с диагональю экрана от 4.5" до 7": адаптивная сетка, изменяемая структура колонок, скрытие лишних элементов;**
2. **Планшеты: интерфейс переключается на двухколонный режим, крупные элементы управления;**

**Применяются media-запросы, относительные единицы измерения (em, rem, %) и грид-система на Flexbox + CSS Grid. Протестированы основные breakpoint'ы: 320px, 480px, 768px, 1024px, 1280px.**

**4.2. Кроссбраузерность веб-сайта**

В рамках тестирования веб-приложения была проведена комплексная проверка кросс-браузерной совместимости в актуальных версиях популярных браузеров, включая Google Chrome 124.0 и выше, Mozilla Firefox 124.0 и новее, а также Microsoft Edge начиная с версии 118.0. Особое внимание уделялось корректности визуального отображения всех элементов интерфейса, в частности, правильности рендеринга векторной SVG-графики и системных шрифтов во всех тестируемых окружениях.

Функциональное тестирование охватило все ключевые JavaScript-механизмы, включая систему переключения между блоками контента и работу скриптов навигации, что подтвердило стабильность выполнения клиентских сценариев независимо от браузерной платформы. Визуальные эффекты и интерактивные элементы, реализованные средствами SCSS, такие как плавные анимации и реакции на действия пользователя, также демонстрировали ожидаемое поведение во всех протестированных средах.

Результаты тестирования подтвердили, что выбранные технологии и подходы к разработке обеспечивают согласованное и предсказуемое отображение контента, а также стабильную работу всех интерактивных компонентов в современных браузерных средах. Проверка выявила равнозначное качество визуального представления и одинаковый уровень производительности независимо от используемого пользователем браузера, что соответствует требованиям к кросс-платформенной совместимости современного веб-приложения.

**4.3. Руководство пользователя**

**Навигация по сайту построена вокруг пяти основных разделов. На главной странице вы найдете общее описание сервиса, примеры популярных сборок и быстрые ссылки для перехода к основным функциям. Центральным элементом является Конфигуратор - профессиональный инструмент для подбора совместимых компонентов ПК. В Каталоге представлен полный ассортимент комплектующих с возможностью их покупки. Все выбранные товары попадают в Корзину, где можно оформить заказ. В разделе Контакты собрана вся необходимая информация для связи с нашей службой поддержки.**

**Работа с конфигуратором начинается с выбора процессора - этот ключевой компонент определяет совместимость всех остальных частей системы. После выбора ЦПУ, система автоматически предложит вам подходящие варианты материнских плат. Далее следует последовательно подобрать:**

* **Оперативную память (рекомендуемый минимум — 16 ГБ для современных систем);**
* **Видеокарту (особенно важна для игровых конфигураций);**
* **Накопители (SSD значительно ускоряют работу системы);**
* **Блок питания и систему охлаждения.**

**Проверка совместимости выполняется автоматически нашим интеллектуальным конфигуратором, который:**

1. **Анализирует технические параметры всех компонентов;**
2. **Визуально выделяет несовместимые элементы;**
3. **Предлагает альтернативные варианты при обнаружении проблем;**
4. **Активирует кнопку оформления заказа только для корректных сборок;**

**Каждая карточка товара содержит основные параметры и кнопку добавления в корзину. Для получения полной технической информации просто нажмите на интересующий вас товар.**

**Оформление заказа происходит в Корзине, где вам доступны:**

* **Изменение количества товаров**
* **Расчет итоговой стоимости с учетом доставки**

**Для завершения оформления потребуется указать ваши контактные данные и выбрать удобный способ получения заказа.**

**4.4. Выводы**

**Веб-приложение прошло всестороннее тестирование по ключевым метрикам — адаптивность, кроссбраузерность, удобство использования. Это позволило устранить технические недоработки до запуска и гарантировать положительный пользовательский опыт.**

**Заключение**

В ходе выполнения курсового проекта был разработан веб-сайт «PC», который сочетает в себе функциональность инструмента для подбора комплектующих и полноценного интернет-магазина. Проект успешно решает поставленные задачи, предоставляя пользователям удобный и интуитивно понятный интерфейс для сборки компьютера, проверки совместимости компонентов и оформления заказов.

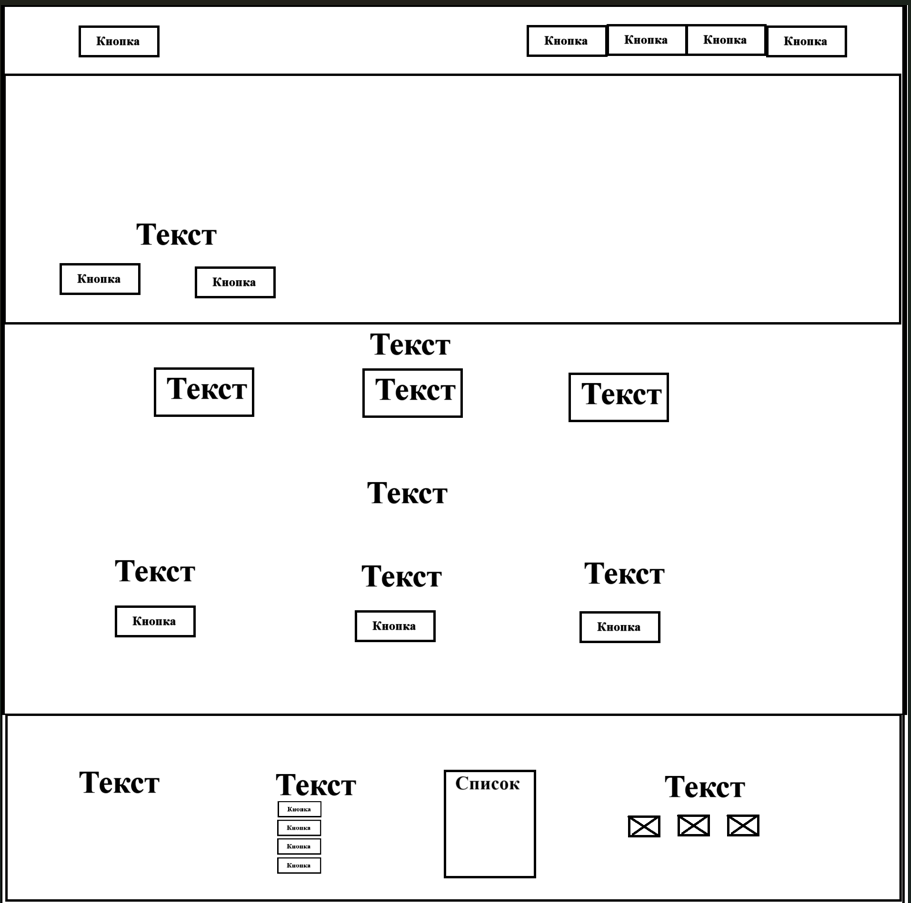
Разработанный веб-сайт «Конфигуратор ПК» представляет собой современное и конкурентоспособное решение, которое не только упрощает процесс сборки компьютера, но и предоставляет пользователям все возможности интернет-магазина. Проект успешно сочетает инновационные технологии, удобный интерфейс и высокую производительность, что делает его востребованным на рынке. Дальнейшее развитие проекта позволит расширить его функционал и укрепить позиции среди аналогов.

**Список использованных источников**

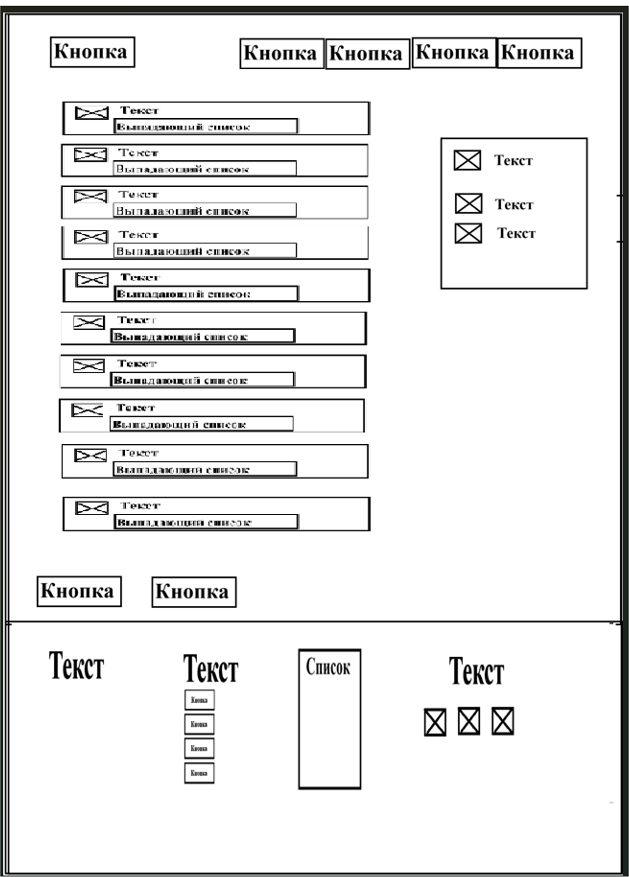
1. Учебник XML для начинающих [Электронный ресурс]. – Режим доступа: <https://msiter.ru/tutorials/uchebnik-xml-dlya-nachinayushchih> - Дата доступа 07.04.2025
2. Учебник Основы JavaScript [Электронный ресурс]. – Режим доступа: <https://learn.javascript.ru/first-steps> - Дата доступа: 11.04.2025.
3. HTML5 и CSS3. Разработка сайтов для любых браузеров и устройств [Фрэйн, 2017] [Электронный ресурс]. – Режим доступа: [https://www.k0d.cc/storage/books/HTML%20+%20CSS/HTML5%20и%20CSS3.%20Разработка%20сайтов%20для%20любых%20браузеров%20и%20устройств%20[Фрэйн,%202017].pdf](https://www.k0d.cc/storage/books/HTML%20+%20CSS/HTML5%20и%20CSS3.%20Разработка%20сайтов%20для%20любых%20браузеров%20и%20устройств%20%5bФрэйн,%202017%5d.pdf) – Дата доступа 17.04.2023

**Приложение А**

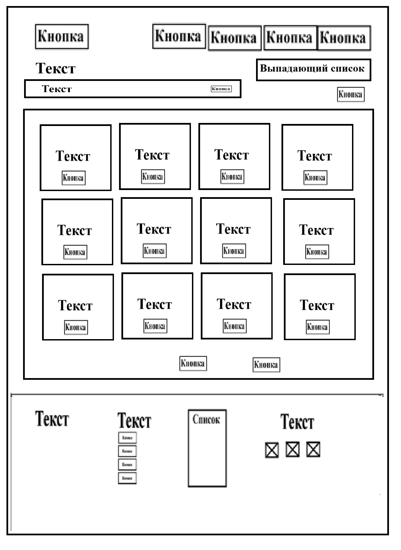
**Прототипы веб-страниц**



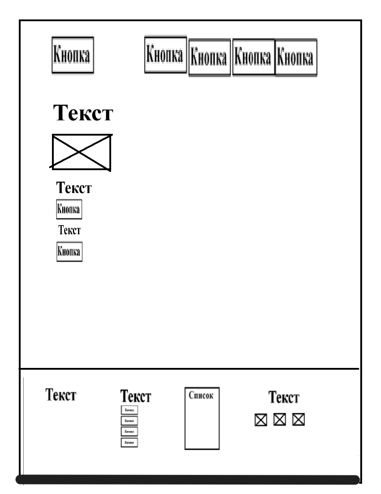
Раздел «Главная»



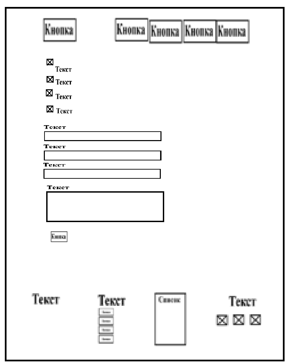
Раздел «Конфигуратор»



Раздел «Каталог»



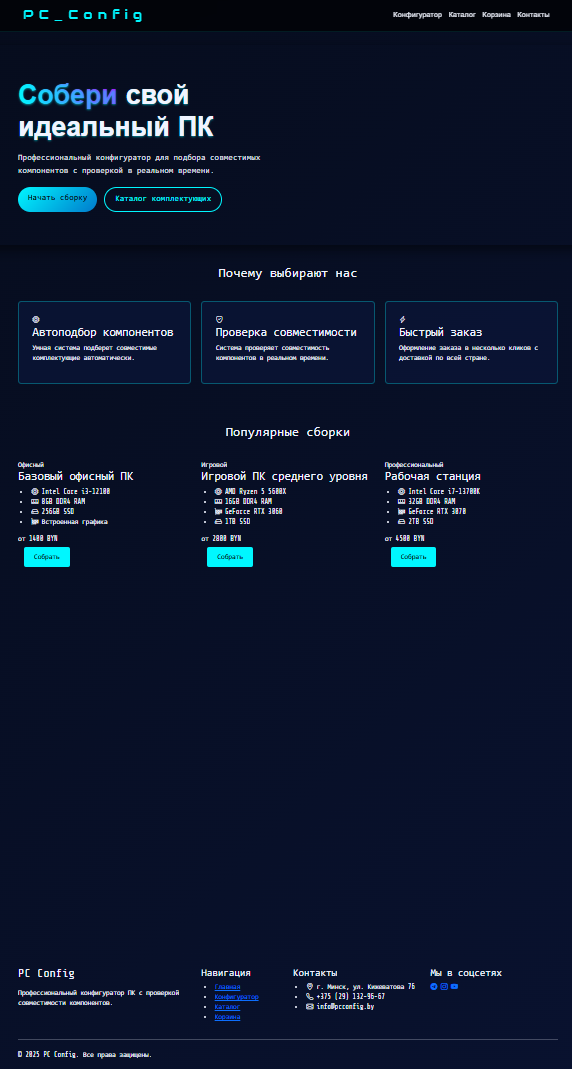
Раздел «Корзина товаров»



Раздел «Контакты»

**Приложение Б**

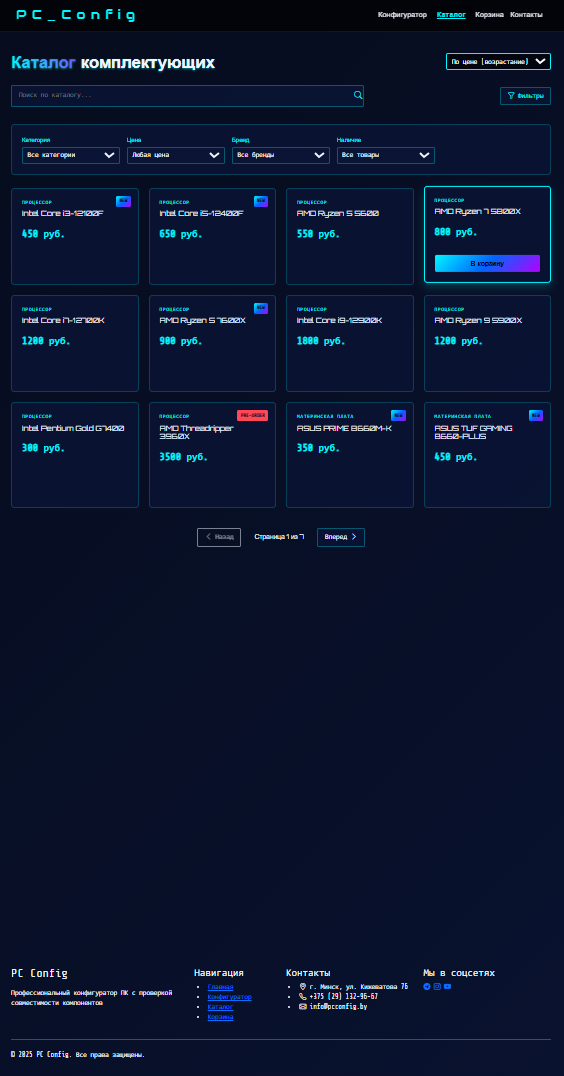
**Макет структуры веб-сайта**

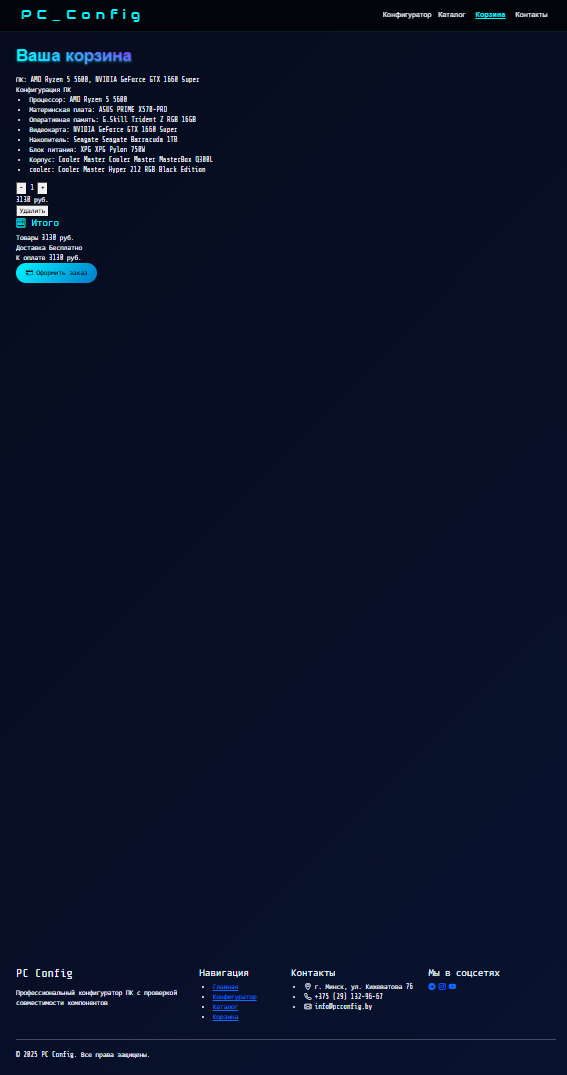


Раздел «Главная»

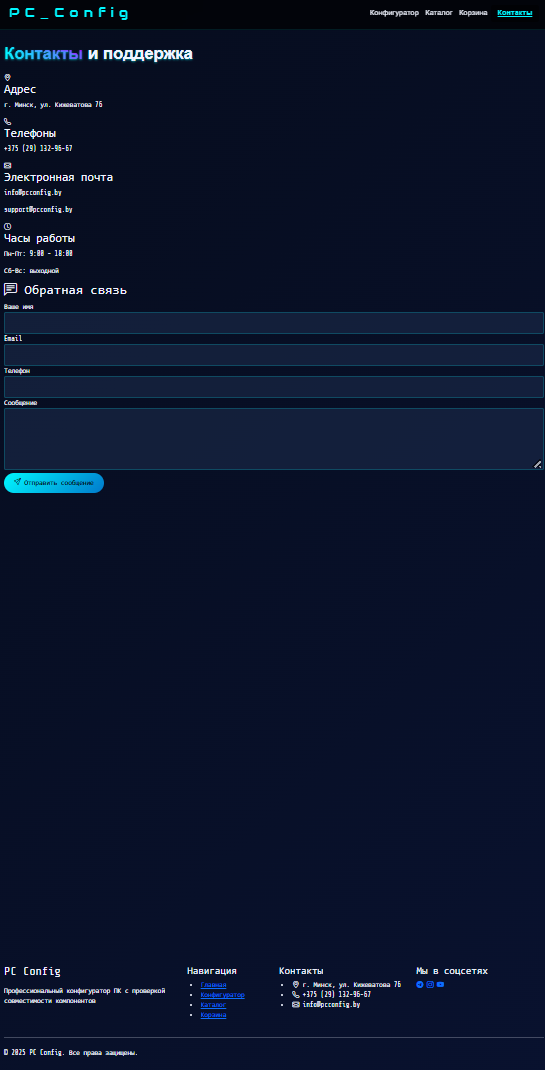
****

Раздел «Конфигуратор»

Раздел «Каталог»

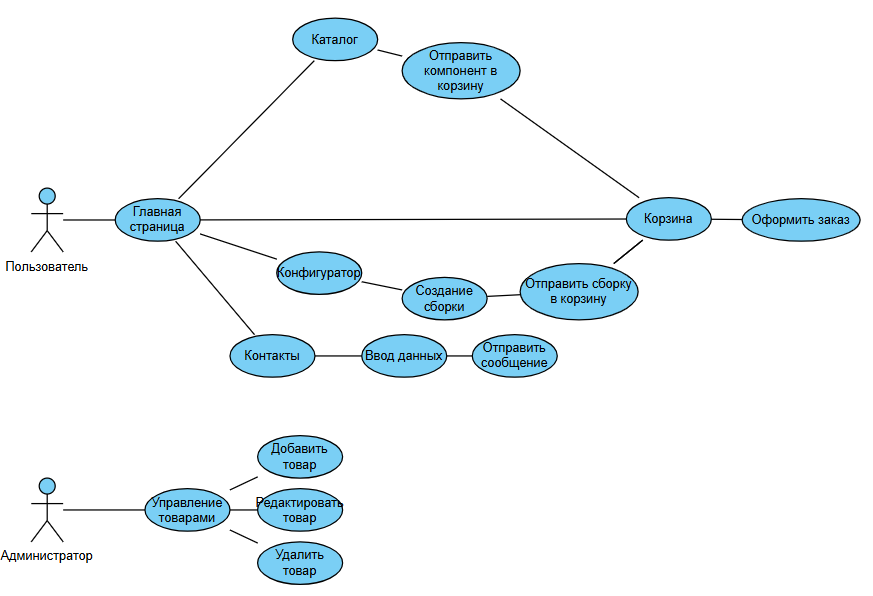


Раздел «Корзина»



Раздел «Контакты»

**Приложение Б**

**UseСase диаграмма**