Geekbrains

**Разработка портфолио в виде web-страницы (Landing page) – для эффективного поиска работы**

Программа: Разработчик

Специализация: Fullstack

Шестаков Роман Витальевич

г. Тюмень

2025 год

**Содержание**

Введение *1 стр.*

Глава 1. Общие сведения *2 стр.*

1.1 Цели. Концепции

1.2 Проблемы, типовых портфолио

1.3 Обоснование и выводы

Глава 2. Техническое задание *2 стр.*

2.1. Назначение сайта

2.2. Требования к организации языковых версий сайта

2.3. Требования к верстке сайта

2.4. Способы навигации по сайту

2.5. Описание требований к элементам страницы

2.6. Загрузка новых проектов

Глава 3. Работа с макетом сайта *2 стр.*

Заключение (~ 4 стр.)

Список используемой литературы

Приложения

**ВВЕДЕНИЕ**

**Тема проекта:** Разработка web-портфолио в виде Landing page – для эффективного поиска работы

**Цель:** Создать web-портфолио в котором можно эффективно рассказать о себе. С возможностью демонстрации выполненных работ (проектов). Быстрая загрузка проекта в портфолио (пополнение коллекции работ и навыков). Разработчик всегда на связи.

**Какую проблему решает:** Качественная презентация навыков разработчика – в любой момент времени и через любое устройство (*gadget*).

**Задачи:**

1. Составить техническое задание с описанием разделов и их содержанием.
2. Найти шаблон (дизайн) и доработать его - используя, онлаин-сервис Figma
3. Определиться с языками программирования – необходимыми для создания приложения
4. Изучить сборщики проектов – выбрать наиболее подходящий (подготовить его для создания проекта).
5. Поиск контента (фото, изображения, текст – наполнение разделов)
6. Подбор шрифтов – согласно макета
7. Верстка приложения
8. Мониторинг хостинг сервисов (цена, доступность интерфейса) – подбор оптимального
9. Backend – разработка и загрузка приложения на хостинг

**Инструменты:** MicrosoftWord, VSCode, Gulp, Figma, GitHub, Scss, Css, Html, Js, Php, Json, GoogleFonts, Icomoon, Node.js, Npm, W3, Beget, Shadow-generate.

# Глава 1. Общие сведения

* 1. **Цели. Концепции**

Целями проекта является быстрое и качественное донесение информации о компетентности и опыте разработчика в удобном формате для потенциального клиента или работодателя.

К концепциям относится создание web-портфолио в виде Landing page, с описанием ключевых навыков, возможностью демонстрации выполненных работ (проектов) и быстрой связью с разработчиком.

О терминах:

**Портфолио** — это комплект наглядных образцов и кейсов, которые кандидат собрал за время работы.

Главная задача портфолио — показать реальные навыки, наработанный опыт и квалификацию кандидата.

Оно может понадобиться в разных ситуациях, например:

- при трудоустройстве на работу

- для участия в конкурсах и грантах

- для привлечения клиентов и инвесторов

Существует несколько основных видов портфолио, каждый из которых подходит для определённой сферы деятельности:

Творческое портфолио — используется художниками, дизайнерами, фотографами, писателями. Оно включает в себя примеры работ, иллюстрации, отрывки из текстов и так далее.

**Профессиональное портфолио** — чаще всего готовится для выпускников вузов и специалистов различных отраслей. Оно содержит дипломы, сертификаты, отзывы работодателей и примеры проектов.

Ученическое портфолио — предназначено для школьников и студентов. Оно может включать в себя результаты тестов, рефераты, грамоты и дипломы

**Landing page** «посадочная страница» — это одностраничный сайт с краткой информацией о товаре, услуге или мероприятии. Его задача — превращать посетителей в клиентов.

Задача лендинга — на одной странице коротко рассказать о продукте и мотивировать пользователя совершить конкретное целевое действие:

- купить товар;

- подписаться на рассылку;

- оставить заявку на оказание услуги;

- зарегистрироваться на онлайн-мероприятие;

- забронировать билеты на концерт или выступление;

- получить предложение о сотрудничестве.

Пользователи переходят на лендинг по ссылкам с рекламных баннеров, постов из соцсетей, электронных писем или СМС.

**Web** (Всемирная паутина) — распределённая неоднородная компьютерная система коллективного пользования гипермедийными документами, действующая на базе сети Интернет.

* 1. **Проблемы, типовых портфолио**

Основной объем начинающих разработчиков сталкиваются с проблемой не эффективной коммуникацией с потенциальным клиентом или работодателем. Портфолио выполнено формате (doc, pdf), готовые работы (проекты), предоставлены в виде ссылок на репозиторий или web-сайты.

Проблема – для эффективной оценки, клиенту или работодателю нужно просмотреть, большой оббьем информации из различных источников (трата времени клиента = низкая оценка!).

О терминах:

**Формат doc** — это формат файла, используемый Microsoft Word. Он содержит текст, изображения, графики, таблицы.

**Формат pdf** (Portable Document Format) — это универсальный формат файлов, разработанный компанией Adobe Systems в 1993 году. Основная цель создания PDF — обеспечить надёжное и неизменное отображение документов на различных устройствах и платформах

**web-сайт** — одна или несколько логически связанных между собой **web** -страниц, объединённых под одним адресом (доменным именем) и воспринимаемых пользователем как единое целое.

**Репозиторий** (от англ. repository — хранилище) — место, где хранятся и поддерживаются какие-либо данные. Чаще всего данные в репозитории хранятся в виде файлов, доступных для дальнейшего распространения по сети.

* 1. **Обоснование и выводы**

- Web-портфолио позволит быстрее и эффективнее, привлечь клиента или работодателя.

- Каждый новый проект будет загружен в портфолио, что будет увеличивать потенциал разработчика.

- рост потенциала и экономия времени = прибыль!

# Глава 2. Техническое задание

Техническое задание (ТЗ) — это документ с подробным описанием требований заказчика к проекту. В нём указывают характеристики продукта, особенности задачи, дополнительные условия, сроки выполнения

**2.1. Назначение сайта**

- официальный одностраничный сайт-портфолио.

К основным функциям сайта относятся:

- быстрая навигация по разделам;

- просмотр ленты проектов;

- переход по ссылке в проект и обратно на сайт;

- рассказать о разработчике;

- обратная связь с разработчиком;

- быстрая смена языка Ru - En;

Сайт должен быть оптимизирован для корректного отображения в браузерах Google Chrome, Opera, Mozilla Firefox, Яндекс.Браузер, MicrosoftEdge, AppleSafari.

Разработка сайта должна осуществляться при помощи следующих технологий: HTML, CSS, Java-Script, PHP.

**2.2.**  **Требования к организации языковых версий сайта**

Язык сайта: русский, английский.

Оформление и содержание страниц иноязычного сайта аналогично русскоязычному варианту. На страницах иноязычного сайта размещается контент, предоставленный Заказчиком. В случае не предоставления необходимой информации для каких-либо блоков страницы сайта оформляются без учета данных блоков. Смена языка должна осуществляться без перезагрузки страницы.

**2.3.**  **Требования к верстке сайта**

Верстка выполняется под браузеры Microsoft Edge, Mozilla Firefox, Google Chrome, Apple Safari и Opera последних официально выпущенных версий на момент подписания Приложения на этапе верстки, для платформ ПК и MAC. Для устаревших браузеров должно выводиться сообщение, что корректная работа возможна на браузерах, указанных выше.

Верстка выполняется только под 100% масштаб браузера и 100% масштаб операционной системы. Адаптивная верстка должна быть разработана для двух точек излома:

* Смартфоны, вертикальное расположение — 360 px (и до 767 px);
* Планшеты, вертикальное расположение — 768 px (и до 1279 px);
* Персональные компьютеры — на разрешения 1280 пикселей и более используется текущая (ПК-версия) версия макета.
* При открытии файлов верстки на устройствах, ширина экрана которых меньше 1280 пикселей, применяется соответствующая адаптивная версия дизайна. При больших разрешениях применяется основной вариант дизайна сайта. На мобильных устройствах, c целью оптимизации производительности, допускается упрощение верстки, в том числе — изменение размеров изображения, отключение или упрощение анимационных эффектов, видео, скриптов.

Для браузеров, в которых отсутствует поддержка стандартов CSS3, допускается незначительное отличие от макетов, вызванное неполной поддержкой CSS3 с сохранением функциональности.

Под корректной версткой понимается соответствие страниц сайта согласованным макетам в масштабе 100% с учетом особенностей отображения HTML-разметки в операционной системе пользователя, в том числе:

* Отображение и сглаживание шрифтов;
* Масштабирование изображений браузером;
* Пользовательские настройки браузера (шрифты, размеры шрифтов по умолчанию и т.п.).

Верстка должна реализовывать анимацию или иные эффекты, описание которых приведено в настоящем техническом задании.

**2.4.**  **Способы навигации по сайту**

Сайт должен обладать понятной и удобной системой навигации. Вверху страницы отображаются облегченная навигационная панель, которая обеспечивает переход к основным пунктам меню сайта (Главная, Новости и т.д.) и обратно.

**2.5.**  **Описание требований к элементам страницы** (шапка, контентная часть, футер, мини-карточки, кнопки и т.д.).

***Шапка страницы (header)*** – верхняя часть страницы, визуально отделенная от основного контента, содержащая навигаторы и графические элементы оформления.

Шапка страницы содержит:

* перечисление разделов – (Галавная, Обо мне, Знания, Мои работы, Контакты)
* при нажатии происходит плавный переход в раздел по якорной ссылке
* шапка фиксируется при прокрутке
* при размере экрана 425px разделы перемещаются в меню - бургер
* активный раздел выделен

Следующий раздел (под шапкой) страницы содержит:

* + элемент переключателя языковой версии RU/EN;
  + фото разработчика
  + Краткая информация

***Контентная часть (main) –*** середина страницы

* + слайдер – перечисление инструментов и навыков
  + карточки с описанием и ссылками на готовые проекты (показываем 3 последних)
  + в карточке краткое описание проекта + возможность быстрого просмотра основных разделов.
  + кнопка – показывает более поздние проекты

***Подвал страницы (footer)*** – нижняя часть страницы, визуально отделенная от основного контента.

Подвал страницы содержит:

* + телефон;
  + иконки социальных сетей (при нажатии – переход в соцсети)
  + кнопка – Напишите на почту (форма отправки письма)

**2.6.**  **Загрузка новых проектов –** папка с новым проектом добавляется на хостинг в директорию pages. Данные из pages выгружаются на сайт в виде карточек проекта.

 Хостинг – (англ. hosting) — услуга по предоставлению вычислительных мощностей для размещения информации на сервере, постоянно находящемся в сети (обычно Интернет). Хостингом также называется услуга по размещению оборудования клиента на территории провайдера с обеспечением подключения его к каналам связи с высокой пропускной способностью (колокация, от англ. collocation).Обычно хостинг входит в пакет по обслуживанию сайта и подразумевает как минимум услугу размещения файлов сайта на сервере, на котором запущено ПО, необходимое для обработки запросов к этим файлам (веб-сервер). Как правило, в обслуживание уже входит предоставление места для почтовой корреспонденции, баз данных, DNS, файлового хранилища на специально выделенном файл-сервере и т. п., а также поддержка функционирования соответствующих сервисов.

Директории - (файловая система) — механизм организации файлов в файловой системе.

# Глава 3. Работа с макетом

**3.1.** **Описание редактора**.

Figma — это онлайн-графический редактор для создания дизайна интерфейсов, прототипирования и работы с векторными изображениями.

Программа работает прямо в браузере и не требует установки на компьютер, что делает её доступной с любого устройства. Figma также предлагает desktop версию для пользователей, предпочитающих работать с локальным приложением.

Некоторые возможности сервиса:

- рисование и редактирование векторных форм (линии, прямоугольники, эллипсы, полигоны и т. д.);

- инструменты для работы с текстом, включая шрифты, стили и параметры форматирования;

- создание прототипов с интерактивными переходами между экранами и элементами;

- функционал для работы со слоями, группами и фреймами, что упрощает организацию контента;

- встроенная библиотека компонентов, позволяющая быстро добавлять и настраивать стандартные элементы интерфейса;

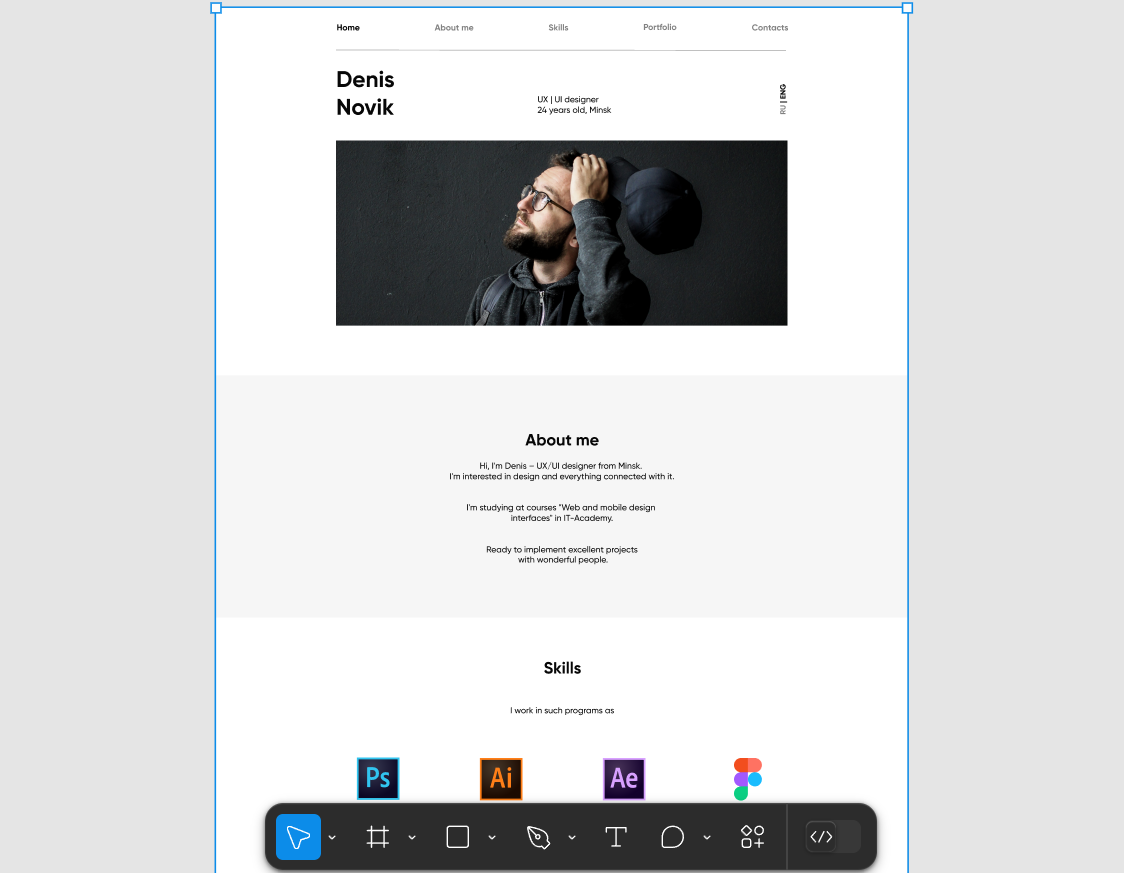
- инструменты для совместной работы в реальном времени, включая комментирование, редактирование и обсуждение проектов внутри команды;

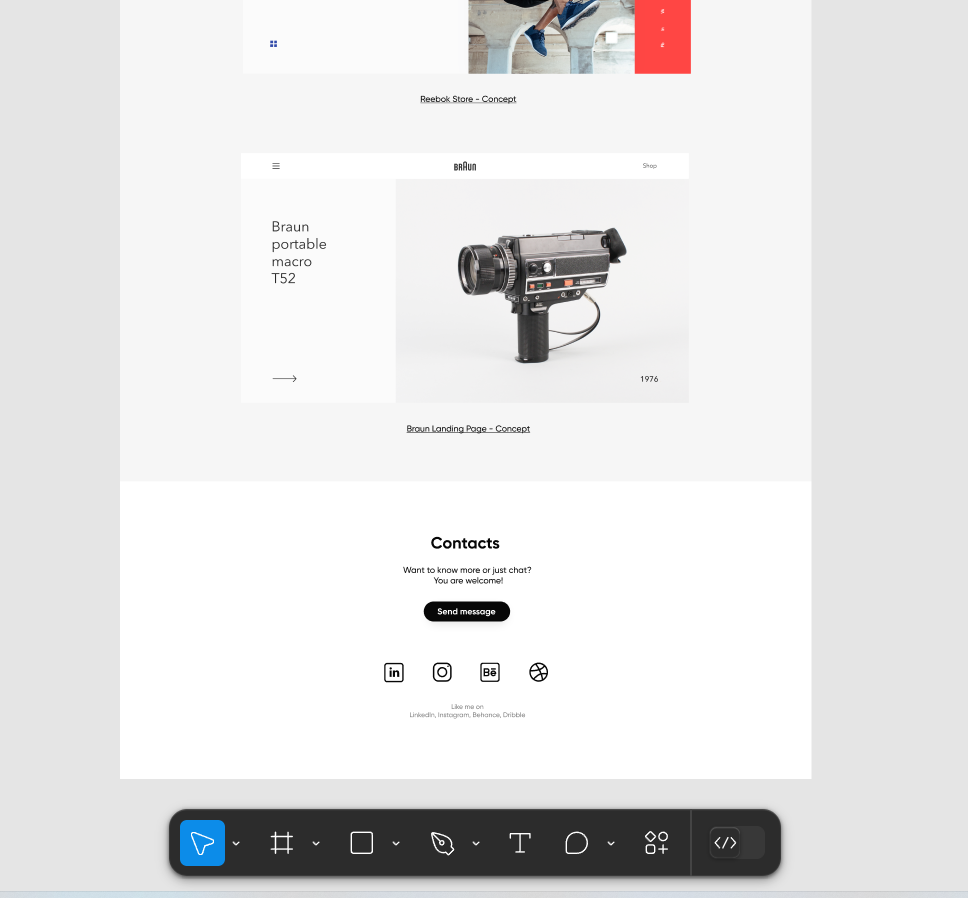
- интеграция с другими приложениями, такими как Slack, Jira, и другими, для удобства управления проектами.

Плюсом **Figma -** является ее большая популярность, поэтому существует много сообществ. Большой выбор специалистов и готовых работ.

**3.2. Подготовка макета.**

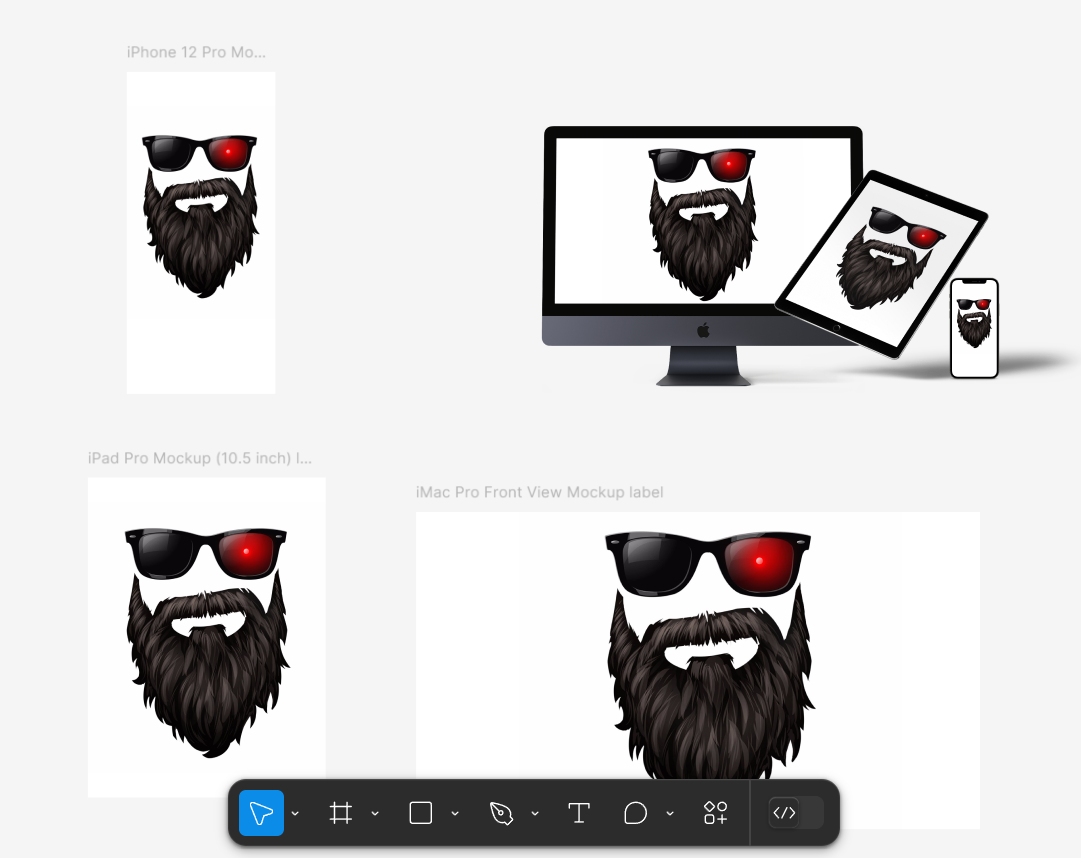
Макет выбран из готовых работ согласно - Технического задания. Бесплатный, находится в открытом доступе.





Создаем интерактивный макет изображения для карточки проекта. Изображение создается путем загрузки в figma - скриншот страницы проекта с разрешением (телефон, планшет, компьютер)

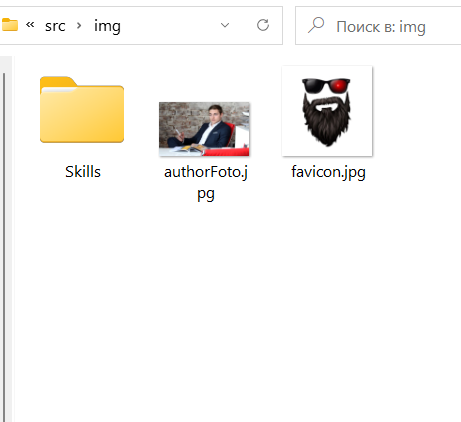
Скриншот - это снимок (фотография) содержимого экрана на компьютере, телефоне или планшете. Он фиксирует визуальную информацию, создавая изображение текста, графики или любых элементов, присутствующих на дисплее.



**3.3. Работа с контентом.**

Создаем основную директорию проекта – portfolio. В portfolio создаем директорию source (src) – в ней храниться исходные файлы проекта.

В src создаем директорию image (img) – в которой сохраняем все картинки и иконки из макета.



* 1. Скрипт - программа в результате которой формируются динамические веб-страницы или выполняются какие-то функции сайта.
  2. CSS - (англ. Cascading Style Sheets — каскадные таблицы стилей), формальный язык описания внешнего вида документа, написанного с использованием языка разметки.
  3. Html - язык гипертекстовой разметки (Hyper Text Markup Language), позволяет готовить веб-страницу с помощью простых редакторов.
  4. Plug-in - (англ. plug-in, от plug in «подключать») — независимо компилируемый программный модуль, динамически подключаемый к основной программе и предназначенный для расширения и/или использования её возможностей. Плагины обычно выполняются в виде разделяемых библиотек.
  5. TCP/IP - (Transmission Control Protocol/ Internet Protocol) стандартный протокол Интернета.
  6. PHP- (англ. PHP: Hyper text Preprocessor — «PHP: препроцессор гипертекста»; первоначально Personal Home Page Tools — «Инструменты для создания персональных веб-страниц»; произносится пи-эйч-пи) — скриптовый язык программирования общего назначения, интенсивно применяемый для разработки веб-приложений. В настоящее время поддерживается подавляющим большинством хостинг- провайдеров и является одним из лидеров среди языков программирования, применяющихся для создания динамических веб-сайтов.
  7. FTP - (File Transfer Protocol) протокол передачи файлов, который подразумевает передачу файлов в так называемом оперативном, или online-режиме.
  8. Web-сервер – это постоянно подключённый к Интернету компьютер, который передаёт эти странички по запросу пользователей.
  9. Gulp — таск-менеджер для автоматического выполнения часто используемых задач при разработке приложений с использованием платформы Node.js. Он позволяет, например, формировать файлы стилей, объединять и унифицировать файлы, оптимизировать изображения, транспилировать код.
  10. Node.js — это бесплатная кросс-платформенная среда выполнения JavaScript с открытым исходным кодом. Она создана на движке V8 от Google, который используется для преобразования JavaScript в компьютерный код
  11. SCSS (Sassy Cascading Style Sheets) — это синтаксис препроцессора Sass, который расширяет стандартные возможности CSS с помощью новых синтаксических конструкций, таких как миксины, циклы, переменные и другие.
  12. JSON (JavaScript Object Notation) — текстовый формат для хранения и обмена структурированными данными. Он основан на синтаксисе объектов в JavaScript, но не зависит от него. Данные представляются в виде пар «ключ-значение», в которых ключи — всегда строки, а значения могут быть представлены различными типами: числовыми, строковыми, логическими.
  13. JavaScript — это высокоуровневый, интерпретируемый язык программирования, используемый в основном для создания интерактивных веб-страниц.

Он позволяет создавать богатые интерактивные пользовательские интерфейсы, обрабатывать события, управлять мультимедийными ресурсами, выполнять валидацию форм и работать с данными в реальном времени.

Некоторые области применения JavaScript:

* Веб-сайты и веб-приложения. Практически на каждом современном сайте используют код, написанный на JS.
* Расширения для браузера. Небольшие простые скрипты, которые добавляют дополнительный функционал — блокируют рекламу, позволяют сохранять аудио, отправляют уведомления о новых письмах или меняют цветовую схему сайта.
* Мобильные приложения. Например, интерфейс для работы с облачным хранилищем, его можно написать на JavaScript и собрать в приложение с помощью специальных инструментов.
* Серверная часть сайтов и программ. Язык программирования JavaScript можно использовать для написания любых сервисов: чатов, компьютерных программ и даже нейросетей. Для этого к нему нужно подключить движок Node.js.
* Игры. На JS можно писать несложные браузерные игры.
  1. WebP — это современный формат изображений, разработанный компанией Google в 2010 году. Он предназначен для того, чтобы обеспечивать высокое качество при меньшем размере файла по сравнению с традиционными форматами, такими как JPEG и PNG.

WebP используется в веб-разработке для оптимизации изображений на сайтах и в социальных сетях, а также в мобильных приложениях, где важна скорость загрузки.

* 1. Figma — онлайн-сервис для разработки интерфейсов и прототипирования с возможностью организации совместной работы в режиме реального времени. Используется для создания упрощённых прототипов интерфейсов, а также для детальной проработки дизайна интерфейсов мобильных приложений, веб-сайтов, корпоративных порталов.

# Глава 2. Термины и определения

**Основная часть** составляет 80% от всего объёма написанного. В нём последовательно рассказывается, как были решены вопросы, перечисленные во вступлении, какие цели достигнуты и с каким результатом.

Содержит 2 главы — теоретическую и практическую, где автор выражает свою позицию относительно гипотез.

В практической части студент должен пошагово описать:

* как он делал проект;
* какие инструменты использовал в работе;
* на каком этапе использовал инструменты и как они помогли в достижении цели работы.

В основной части должны быть практические примеры, результаты и выводы после каждого элемента исследования.

# Заключение

В заключение необходимо включить следующее:

1. Краткие и ёмкие теоретические и практические выводы, которые были получены во время анализа теоретической базы и практического исследования.
2. Оценка проведённого исследования, описание его результатов.
3. Практическая значимость работы, рекомендации и планы на дальнейшие исследования.
4. Общий итог — достижение цели, выполнение задач, доказательство гипотезы.
5. Предложения по совершенствованию объекта исследования.

# Список используемой литературы

Здесь нужно будет указатьсписок используемой литературы, ссылки на все ресурсы, которые нужны были для создания проектной работы.

Основные правила оформления использованной литературы и ресурсов:

1. Каждый источник упоминается единожды, независимо от того, насколько часто на него ссылаются.
2. Список литературы оформляется в алфавитном порядке по фамилии автора, сначала русскоязычная литература, затем иностранная, далее интернет-сайты.
3. Библиографическая запись обязательно включает:
   * Фамилию автора или фамилии их группы, инициалы (при наличии).
   * Название статьи, книги, справочника, закона, иного документа.
   * Населённый пункт, в котором был издан источник, наименование издательства.
   * Год публикации.
   * Число страниц.

## Пример

* *Книга: Автор. Название книги. Город: Издательство, Год.*
* *Статья: Автор. "Заголовок статьи." Название журнала Том, номер (Год): страницы.*

# Приложения

В **приложения** обычно входят артефакты, получившиеся в процессе создания проекта:

1. Объёмные графики и таблицы, которые не помещаются на лист А4.
2. Длинные математические формулы и расчёты по ним.
3. Характеристики аппаратуры, которая использовалась для проведения исследования.
4. Авторские методики.
5. Вспомогательный материал: тесты, карточки, схемы, рисунки.
6. Материалы, полученные на предприятии: отчёты, прочие документы.