**АЗЕРБАЙДЖАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ НЕФТИ И ПРОМЫШЛЕННОСТИ**

**Факультет: Информационные технологии и управление**

**Кафедра: Компьютерная инженерия**

**Предмет: «Веб-системы и технологии»**

Курсовая работа

**Тема** «Разработка системы загрузки программного обеспечения онлайн»

**Группа:** 680.22

**Курc:** 3

**Специальность: 050655** Информационные технологии

**Студент**: Дуньямалыев Джалил

**Руководитель:** **преп. Халилов М. Э.**

**Зав.кафедрой: доц.Рагимова Н.А.**

**Баку – 2025**

**АЗЕРБАЙДЖАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ НЕФТИ И ПРОМЫШЛЕННОСТИ**

**АЗЕРБАЙДЖАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ НЕФТИ И ПРОМЫШЛЕННОСТИ**

**ЗАДАНИЕ ПО КУРСОВОЙ РАБОТЕ**

**ФАКУЛЬТЕТ**: Информационные технологии и управление

**КАФЕДРА**: Компьютерная инженерия

**Группа** 680.22 **курс** III

**Специальность** **050655** Информационные технологии

**Студент** Дуньямалыев Джалил

**Зачетная книжка**

**Руководитель курсовой работы**  **преп. Халилов М. Э.**

**Срок выдачи**

**Дата сдачи**

**Тема курсовой работы** «Разработка системы загрузки программного

обеспечения онлайн »

**Отзыв руководителя курсовой**:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Подпись студента**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Подпись рук.курсовой**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Подпись зав.кафедрой**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Дата защиты курсовой**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**Оценка**\_\_\_\_\_\_\_\_\_\_

**Пред.комиссии**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

**Члены комиссии**: 1.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

2.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

3.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

**СОДЕРЖАНИЕ**

[**ВВЕДЕНИЕ** 3](#_Toc198460114)

[**1.ТЕОРЕТИЧЕСКАЯ ЧАСТЬ** 4](#_Toc198460115)

[**1.1 Web-программирование** 4](#_Toc198460116)

[**1.2 HTML** 5](#_Toc198460117)

[**1.3 Dynamic HTML** 6](#_Toc198460118)

[**1.3.1 Каскадные таблицы стилей** 6](#_Toc198460119)

[**1.3.2 JavaScript** 7](#_Toc198460120)

[**2. ПРАКТИЧЕСКАЯ ЧАСТЬ** 7](#_Toc198460121)

[**2.1 Описание предметной области** 7](#_Toc198460122)

[**2.2 Создание окружения для проекта** 8](#_Toc198460123)

[**2.3 Верстка HTML страницы** 8](#_Toc198460124)

[**2.4 Использование стилей CSS** 24](#_Toc198460125)

[**2.5.Функциональность с помощью JS** 25](#_Toc198460126)

[**3. РЕЗУЛЬТАТ** 28](#_Toc198460127)

[**4. ЛИТЕРАТУРА** 29](#_Toc198460128)

[**5. ПРИЛОЖЕНИЯ** 30](#_Toc198460129)

# **ВВЕДЕНИЕ**

Веб-системы и технологи и — это инструменты, языки программирования и платформы для разработки, внедрения и поддержки веб-приложений. Их эволюция привела к появлению множества языков, фреймворков и СУБД, расширив функционал современных решений.

Ключевым последствием развития веб-технологий стало вовлечение новых пользователей, включая непрофессиональную аудиторию. Это потребовало адаптации интерфейсов, сделав их интуитивными и доступными. Рост интереса к цифровым решениям стал возможен только после упрощения взаимодействия с системами.

Веб-технологии делятся на:

1. Серверные(PHP, CGI) — обеспечивают доступ к данным и обработку запросов.

2. Клиентские (HTML, JavaScript, DHTML) — отвечают за отображение и взаимодействие с пользователем.

Архитектура «клиент-сервер» лежит в основе интернет-сервисов: клиент отправляет запрос, сервер обрабатывает его и возвращает результат или ошибку.

Цель курсовой работы — разработка сайта для загрузки ПО с удобным и безопасным интерфейсом. Этапы:

1. Сбор требований (анализ потребностей клиента).

2. Дизайн (макеты, цветовые схемы, UI/UX).

3. Разработка (HTML, CSS, JavaScript).

4. Тестирование (проверка совместимости и функционала).

Итог: создание интуитивно понятного и надежного ресурса для загрузки программного обеспечения.

# **1.ТЕОРЕТИЧЕСКАЯ ЧАСТЬ**

## **1.1** **Web-программирование**

Веб-программирование — это область программирования, ориентированная на создание веб-приложений, обеспечивающих функционирование динамических сайтов во Всемирной паутине. Оно включает в себя разработку как клиентской, так и серверной части веб-приложений.

**Клиентские языки** выполняются на стороне пользователя, обычно в браузере. К ним относятся HTML, CSS и JavaScript. Эти языки позволяют создавать интерактивные и динамичные элементы на веб-страницах.

**Серверные языки** обрабатываются на сервере и формируют содержимое, которое отправляется пользователю. Примеры таких языков включают PHP, Python, Ruby и Java. Серверные языки часто взаимодействуют с базами данных для хранения и извлечения информации.

Веб-программирование также включает использование различных фреймворков и шаблонов проектирования, для упрощения и структурирования процесса разработки.

## **1.2****HTML**

HTML (HyperText Markup Language) — это язык разметки, используемый для структурирования и отображения веб-страниц и их содержимого. Он определяет, как различные части контента — такие как заголовки, абзацы, списки, изображения и ссылки — должны быть представлены в браузере.

**Основные понятия HTML:**

* **Элементы HTML**: Основные строительные блоки HTML-документа. Каждый элемент состоит из начального тега, содержимого и конечного тега. Например: <p>Это абзац текста.</p>
* **Атрибуты**: Дополнительные параметры, которые предоставляют информацию о элементах. Атрибуты указываются в начальном теге элемента. Например: <img src="image.jpg" alt="Описание изображения">
* **Структура документа**: HTML-документ начинается с декларации <!DOCTYPE html>, за которой следуют элементы <html>, <head> и <body>. Внутри <head> размещаются метаданные, такие как <title> и ссылки на стили, а в <body> — основной контент страницы.
* **Семантические теги**: HTML5 представил теги, которые придают структуре документа смысловую нагрузку, такие как <header>, <footer>, <article>, <section> и <nav>.

## **1.3 Dynamic HTML**

Dynamic HTML или DHTML - так принято называть связку языка HTML, каскадных таблиц стилей, скриптового языка и объектной модели документов. Скриптовым языком может выступать JavaScript или VisualBasic, но именно первый язык получил большую популярность и сегодня используется повсеместно. При помощи DHTML можно создавать интерактивные Web страницы, он позволяет легко и гибко обрабатывать данные запроса и формировать динамический ответ. DHTML может быть использован для реализации интерфейса Drag'n'Drop. На его основе создаются игры и другие интерактивные сервисы. Стоит отметить, что для реализации принципов DHTML достаточно лишь браузера, который будет обрабатывать содержимое страницы. То есть, нет необходимости, например, в обращениях к базе данных.

## **1.3.1****Каскадные таблицы стилей**

**CSS (Cascading Style Sheets)** — это технология, предназначенная для оформления HTML-страниц. С помощью CSS можно задавать внешний вид элементов: шрифты, цвета, отступы, расположение и другие параметры. Основная цель CSS — отделить содержимое (HTML) от оформления, что упрощает разработку, поддержку и изменение веб-документов.

Стили могут быть подключены разными способами:

* **Встроенные стили (embedded)** — CSS-код размещается в блоке <style> внутри HTML-документа.
* **Внешние файлы (external)** — стили хранятся в отдельном .css-файле и подключаются через тег <link>.
* **Inline-стили** — стили задаются напрямую в атрибуте style элемента.

CSS-правило состоит из **селектора** и **блока свойств**:

селектор {

свойство: значение;

}

Когда одно и то же свойство задано в нескольких местах, применяется правило с более высоким приоритетом — это и называется **каскадом**.

## **1.3.2 JavaScript**

**JavaScript** — объектно-ориентированный скриптовый язык программирования, активно используемый в веб-разработке и являющийся важной частью DHTML. JavaScript позволяет создавать динамические элементы на веб-страницах, взаимодействовать с HTML и CSS, обрабатывать события, управлять формами и изменять содержимое страниц без перезагрузки.

**Особенности JavaScript:**

* Встроенный язык, запускаемый в браузере;
* Похож на язык C, но с динамической типизацией и автоматическим управлением памятью;
* Переменные могут включать буквы, цифры, символы \_ и $, но не начинаться с цифры.

**Подключение JavaScript:**

* **Встроенный скрипт**:

<script type="text/javascript">

alert('Hello, World!');

</script>

* **Внешний файл**:

<script type="text/javascript" src="script.js"></script>

JavaScript позволяет:

* Заполнять и проверять формы;
* Управлять стилями и содержимым страницы;
* Реализовывать интерактивное поведение веб-приложений.

# **2. ПРАКТИЧЕСКАЯ ЧАСТЬ**

## **2.1 Описание предметной области**

**Предметная область** — онлайн-информационная система для загрузки ПО онлайн. Это часть реального мира, подлежащая исследованию и автоматизации.

**Архитектура сайта** — структура информации и навигации, обеспечивающая удобный доступ к данным. Грамотное проектирование упрощает поиск, улучшает пользовательский опыт и способствует росту популярности ресурса. При разработке важно учитывать влияние структуры на продвижение товаров и услуг: правильная подача информации увеличивает посещаемость и акцентирует внимание на ключевых разделах.

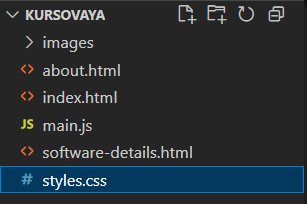
**Интерфейс** — важный элемент сайта, определяющий взаимодействие с пользователем. Чёткое и удобное расположение информационных блоков позволяет быстро находить нужные данные.

Используемые технологии:

* HTML — язык разметки для создания структуры веб-страниц.
* CSS — язык для оформления внешнего вида элементов.
* JavaScript — язык программирования, позволяющий реализовать интерактивность на сайте.

## **2.2** **Создание окружения для проекта**

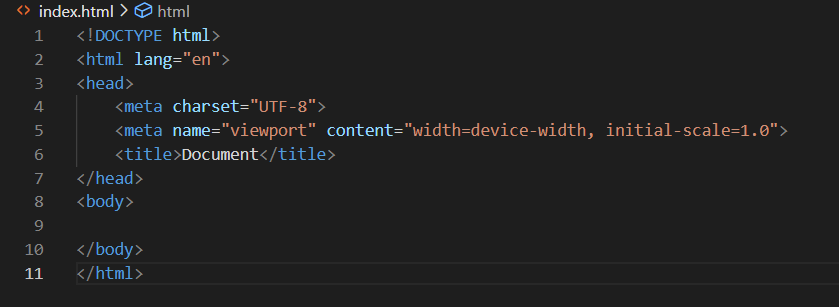
Проект создаётся в Visual Studio Code в папке “kursovaya”. Включает три страницы (index.html, about.html, software-details.html), стили (styles.css), скрипт (main.js) и папку для изображений.



*Рис. 2.1- Итоговый вид окружения*

## **2.3 Верстка HTML страницы**

Открываем файл “index.html” и пишем стандартный HTML5 код (рис. 2.2).



*Рис. 2.2*

Элемент **<!DOCTYPE>** указывает браузеру тип HTML-документа и помогает правильно интерпретировать его содержимое.Тег **<html>** является контейнером для всего содержимого страницы, включая разделы <head> и <body>, и рекомендуется к использованию.Элемент **<head>** содержит метаинформацию, такую как заголовок страницы, стили и скрипты.Элемент **<body>** включает основное содержимое страницы и используется один раз в документе.

Подключение внешних файлов, таких как таблицы стилей, осуществляется с помощью тега <link> внутри элемента <head> (см. рис. 2.3).

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

*Рис. 2.3*

Через rel="shortcut icon" ставим иконку в названии страницы :

**<link rel="shortcut icon" href="./images/Picture8-removebg-preview.png">**

Получаем такой результат(рис. 2.4):



*Рис. 2.4*

**Описание шапки сайта (Header)**

Шапка сайта (<header>) представляет собой верхнюю навигационную панель, обеспечивающую быстрый доступ к основным разделам веб-ресурса.

**1. Логотип и название сайта**

В левой части шапки расположен логотип с текстовым названием **«СофтПортал»**, который является кликабельной ссылкой на главную страницу (index.html).Логотип реализован с помощью элемента <a class="logo">.

        <a href="index.html" class="logo">

          <span class="logo-text">СофтПортал</span>

        </a>

**2. Основное навигационное меню**

Навигационное меню (<nav class="main-nav">) включает следующие пункты:

* **Главная**
* **Категории**
* **Новинки**
* **Популярное**

<nav class="main-nav">

        <a href="index.html" class="nav-link active">Главная</a>

      </nav>

**3. Поисковая строка**

* В правой части шапки (<div class="header-right">) расположен элемент поиска.
* Поле ввода (<input type="search">) с плейсхолдером *«Поиск программ...»* подсказывает пользователю его назначение.
* Иконка лупы (<img src="./images/search-icon.png">) визуально обозначает функцию поиска.

<div class="header-right">

      <div class="search-container">

         <input type="search" class="search-input" placeholder="Поиск программ...">

          <img src="./images/search-icon.png" class="search-icon">

      </div>

</div>

После использования CSS стилей навбар примет вид(см рис. 2.5):

*Рис. 2.5*

**Описание секции (Hero Section)**

Секция (<section class="hero">) является ключевым элементом главной страницы, выполняющим маркетинговую и навигационную функции. Этот блок призван привлечь внимание пользователя и побудить его к взаимодействию с сайтом.

**1. Структура и содержимое**

Секция состоит из двух основных частей:

* **Текстовый блок (**<div class="hero-text">**)**:
  + Заголовок <h1> с основным предложением: "Скачивайте надежное программное обеспечение"
  + Описательный абзац <p> с уточнением преимуществ: "Безопасные и проверенные программы..."
  + Группа кнопок-призывов к действию:
    - Основная кнопка "Просмотреть категории" (класс primary)
    - Вторичная кнопка "Популярные программы" (класс outline)
* **Графический блок (**<div class="hero-image">**)**:
  + Изображение программного обеспечения (<img src="./images/logo2.png">)
  + Стилизовано с закругленными углами (border-radius: 30px)

<section class="hero">

      <div class="container">

        <div class="hero-content">

          <div class="hero-text">

            <h1>Скачивайте надежное программное обеспечение</h1>

            <p>Безопасные и проверенные программы для всех ваших потребностей. Быстрая загрузка, без вирусов.</p>

            <div class="hero-buttons">

              <a href="#" class="button primary">Просмотреть категории</a>

              <a href="#featured" class="button outline">Популярные программы</a>

            </div>

          </div>

          <div class="hero-image">

            <img src="./images/logo2.png" alt="Software Download Hero" style="border-radius: 30px;">

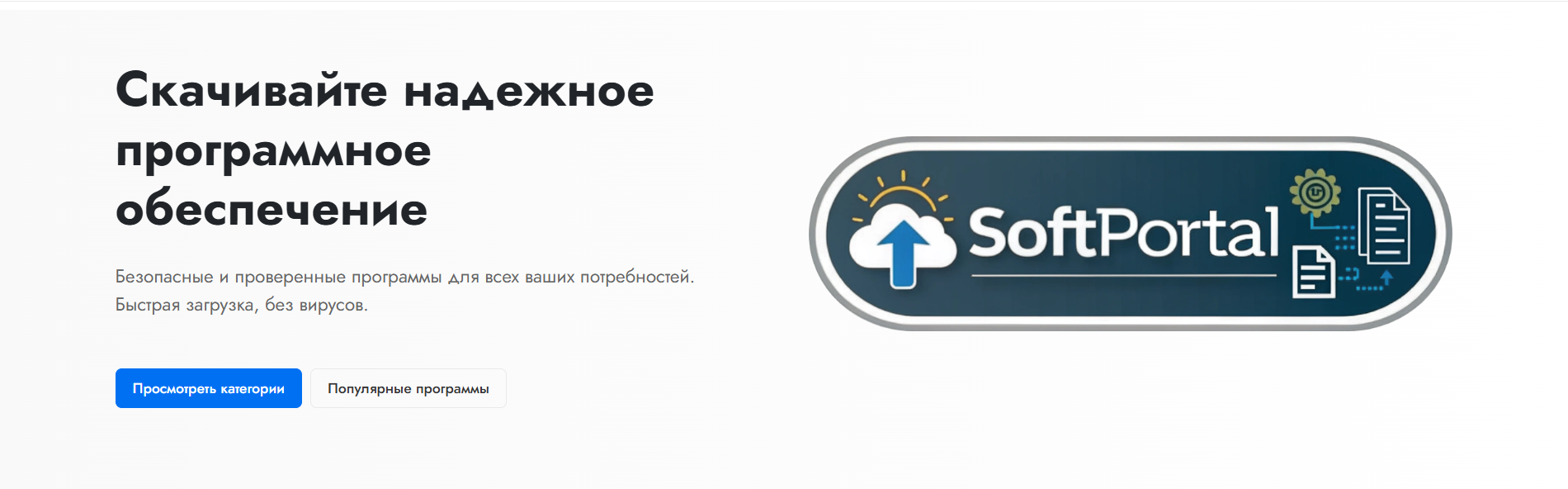
          </div>

      </div>

      </div>

    </section>

В конце она будет выглядеть так(см рис.2.6):



*Рис. 2.6*

**Пошаговое создание поискового блока на сайте**

Добавляем новый раздел сайта с помощью тега <section class="search-section">.

<section class="search-section">...</section>

Внутри секции размещаем контейнер <div class="container">. Он ограничивает ширину содержимого и выравнивает элементы по центру страницы, обеспечивая аккуратный и читаемый внешний вид.

Внутри контейнера создаём блок <div class="search-content">, который включает в себя:

* Заголовок второго уровня <h2>, приглашающий пользователя начать поиск: **"Найдите нужное программное обеспечение"**.
* Краткое описание в виде абзаца <p>.

Под описанием добавляем блок <div class="main-search">, внутри которого находится поле ввода поиска.

<section class="search-section">

      <div class="container">

        <div class="search-content">

          <h2>Найдите нужное программное обеспечение</h2>

          <p>Более 10,000 программ для Windows, Mac и Linux</p>

          <div class="main-search">

            <input type="search" class="main-search-input" placeholder="Поиск программ...">

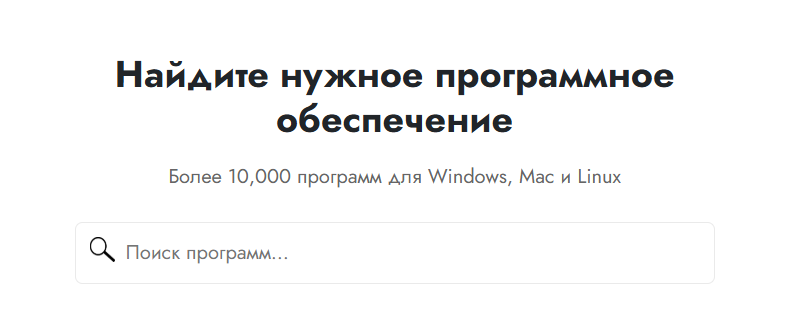
            <img src="./images/search-icon.png" class="main-search-icon">

          </div>

        </div>

      </div>

    </section>



*Рис. 2.7*

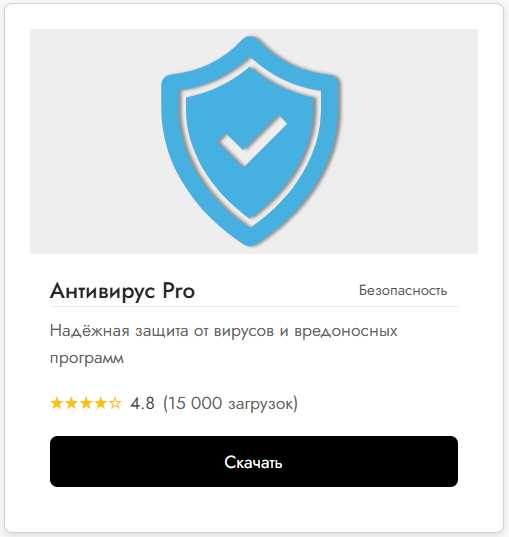
**Создание секции популярных программ на сайте**

Создаётся секция <section id="featured" class="featured-section"> с подборкой популярных программ. Внутри размещается контейнер <div class="container"> для центрирования контента и заголовок <h2> — "Популярные программы".

Ниже добавляется блок <div class="software-grid" id="featured-software"> — сетка карточек. Каждая карточка (<div class="card">) включает:

* изображение программы,
* название (<h3>),
* категорию (<span class="tag">),
* описание (<p class="description">),
* рейтинг (звёзды, оценка, загрузки),
* кнопку скачивания (возможно в ссылке).

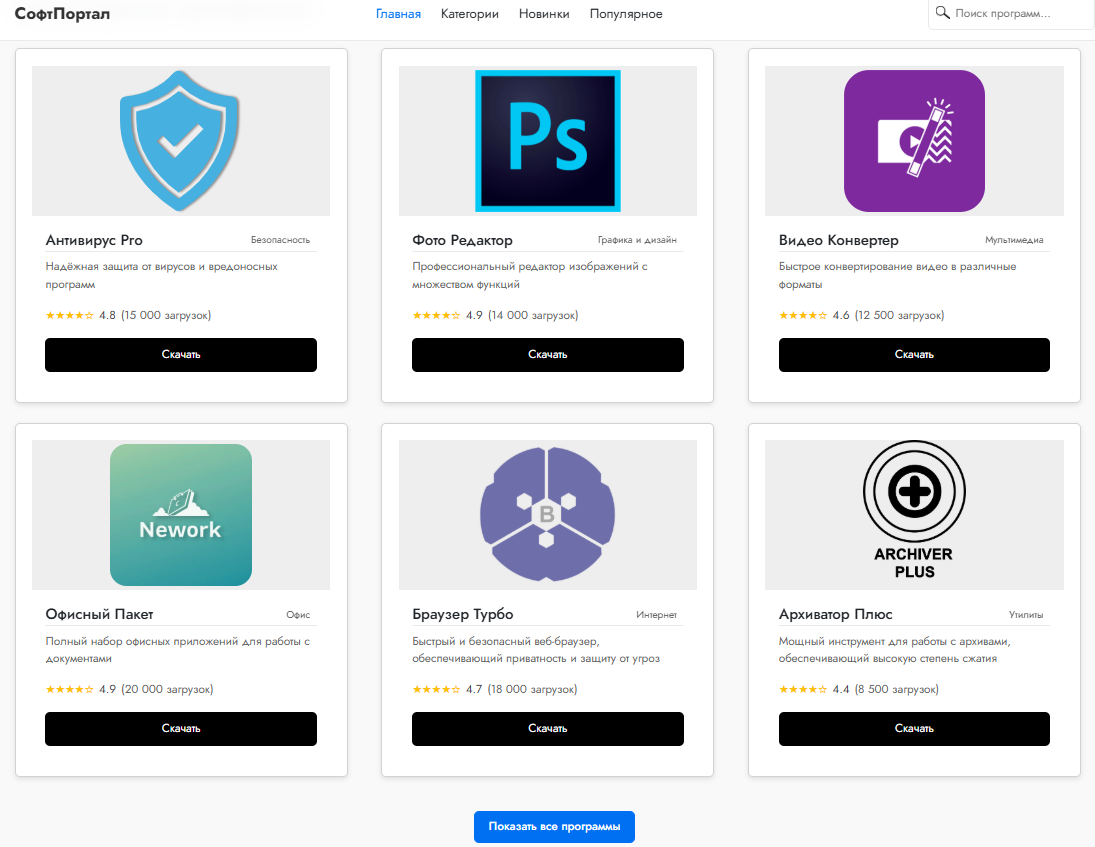
Результат одной карточки (см рис. 2.8):



*Рис. 2.8*

В конце секции добавляется кнопка <a href="categories.html" class="button primary">Показать все программы</a>,позволяющая пользователю перейти к полному списку программ на сайте.

После задания стилей получаем(см рис. 2.9):



*Рис. 2.9*

**Создание подвала сайта**

Создаём элемент <footer class="footer">.

Внутри подвала создаём контейнер <div class="container footer-container">.

В блоке <div class="footer-left"> размещаем:

* Логотип сайта в виде ссылки <a href="index.html" class="logo">.
* Текст авторских прав:

<p style="margin-bottom: 0px;">&copy; <span id="current-year"></span> СофтПортал...</p>

Здесь тег <span id="current-year"></span> будет динамически подставлять актуальный год.

В блоке <div class="footer-links"> размещаем навигационные ссылки:

* "О нас" — ссылка на страницу с информацией о проекте.
* "Контакты" — форма обратной связи или контактные данные.
* "Конфиденциальность" — политика обработки персональных данных.
* "Условия использования" — пользовательское соглашение.

Все ссылки оформлены стандартным тегом <a>.

Получаем(см рис. 2.10):



*Рис. 2.10*

Здесь наша первая страница завершена, позже мы зададим ему стили.

Начинаем верстку второй страницы, а именно файл “about.html”.Переход к нему возможен через нажатие на “О нас” в подвале прошлой страницы:

Шапку сайта создаем как на первой странице.

Далее создаем навбар:

<nav class="breadcrumbs">

                <ol class="breadcrumbs-list">

                    <li><a href="index.html">Главная</a></li>

                    <li>/</li>

                    <li class="current">О нас</li>

                </ol>

            </nav>

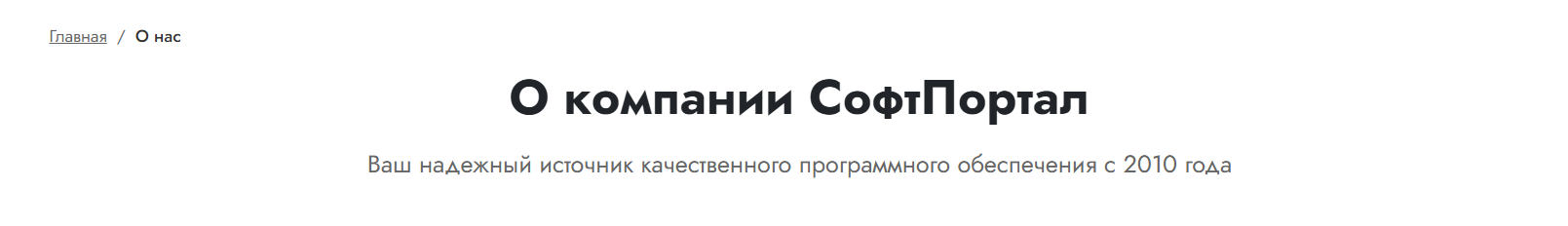
И предоставляем общую информацию пользователю:

 <div class="about-header">

                <h1>О компании СофтПортал</h1>

                <p class="about-subtitle">Ваш надежный источник ...</p>

            </div>



*рис. 2.11*

**Cоздание информационных секций "О нас"**

Создаем контейнер для контента с классом «about-content»

Каждая текстовая секция оформлена в отдельном элементе <section class="about-section">.

**Разметка первой секции — "Наша миссия"**

В первой секции размещены:

* Блок с изображением <div class="about-image">, в котором содержится картинка офиса сайта:
* <img src="./images/ofis2.jpg" alt="Офис СофтПортал">
* Блок с текстом <div class="about-text">, содержащий:
  + Заголовок <h2>Наша миссия</h2>.
  + Два абзаца <p>, подробно описывающие цели проекта и принципы проверки программ.

Таким образом, текст и изображение располагаются рядом.

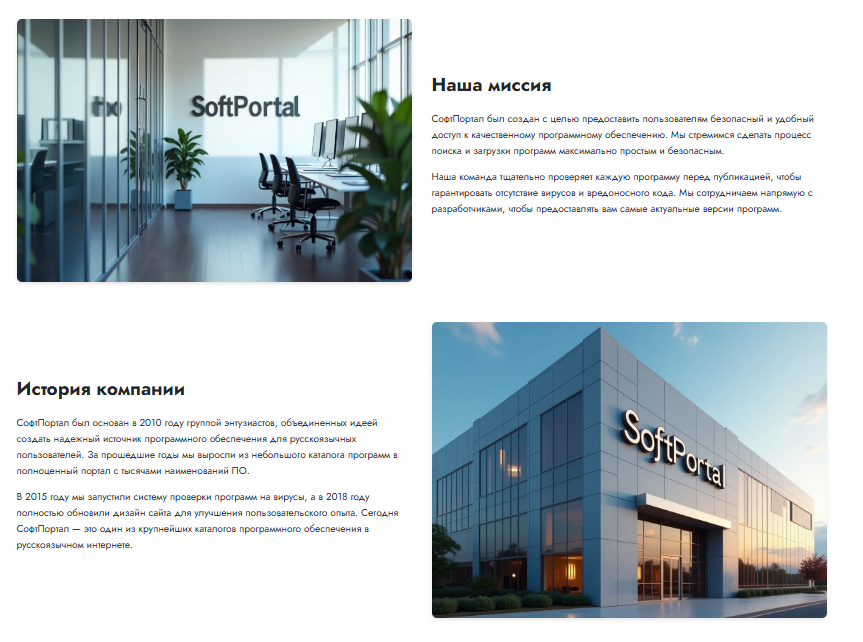
**Разметка второй секции — "История компании"**

Вторая секция имеет тот же принцип построения, но с добавлением класса reverse, который позволяет через CSS менять порядок отображения элементов (например, сначала текст, потом картинка или наоборот).

Внутри второй секции:

* Блок <div class="about-image"> с картинкой команды:
* <img src="./images/building.jpg" alt="Команда СофтПортал">
* Блок <div class="about-text"> с:
  + Заголовком <h2>История компании</h2>.
  + Двумя абзацами <p>, рассказывающими об основаниях и развитии проекта СофтПортал.

**Итог(см рис. 2.12):**



**Рис. 2.12**

**Пошаговое создание секций "Наши ценности", "Наша команда" и "СофтПортал в цифрах"**

**Секция 1. Наши ценности**

**Основная структура**

* Вся секция обёрнута в <section class="about-values">

**Заголовок секции**

* Добавлен заголовок второго уровня <h2>Наши ценности>, который сразу сообщает о содержании.

**Сетка карточек**

* Карточки ценностей размещены в контейнере <div class="values-grid">.
* Для каждой ценности создана отдельная карточка <div class="value-card">, внутри которой:
  + Иконка в блоке <div class="value-icon"> с изображением.
  + Подзаголовок <h3> с названием ценности.
  + Короткий текст в абзаце <p>.

Таким образом, каждая ценность красиво и компактно оформлена.

**Секция 2. Наша команда**

**Основная структура**

* Секция начинается с <section class="about-team">.

**Заголовок**

* Заголовок второго уровня <h2>Наша команда> представляет секцию.

**Сетка участников**

* Используется контейнер <div class="team-grid">, чтобы аккуратно разместить профили сотрудников в ряд.

**Карточки участников**

* Каждый член команды представлен в блоке <div class="team-member">, содержащем:
  + Фотографию в <div class="team-photo">.
  + Имя участника <h3>.
  + Должность <p class="team-role">.
  + Краткое описание задач <p class="team-description">.

Карточки визуально унифицированы, что делает блок "Наша команда" аккуратным и привлекательным.

**СофтПортал в цифрах**

**Основная структура**

* Вся секция помещена в <section class="about-stats">.

**Заголовок**

<h2>СофтПортал в цифрах</h2> привлекает внимание к статистике проекта.

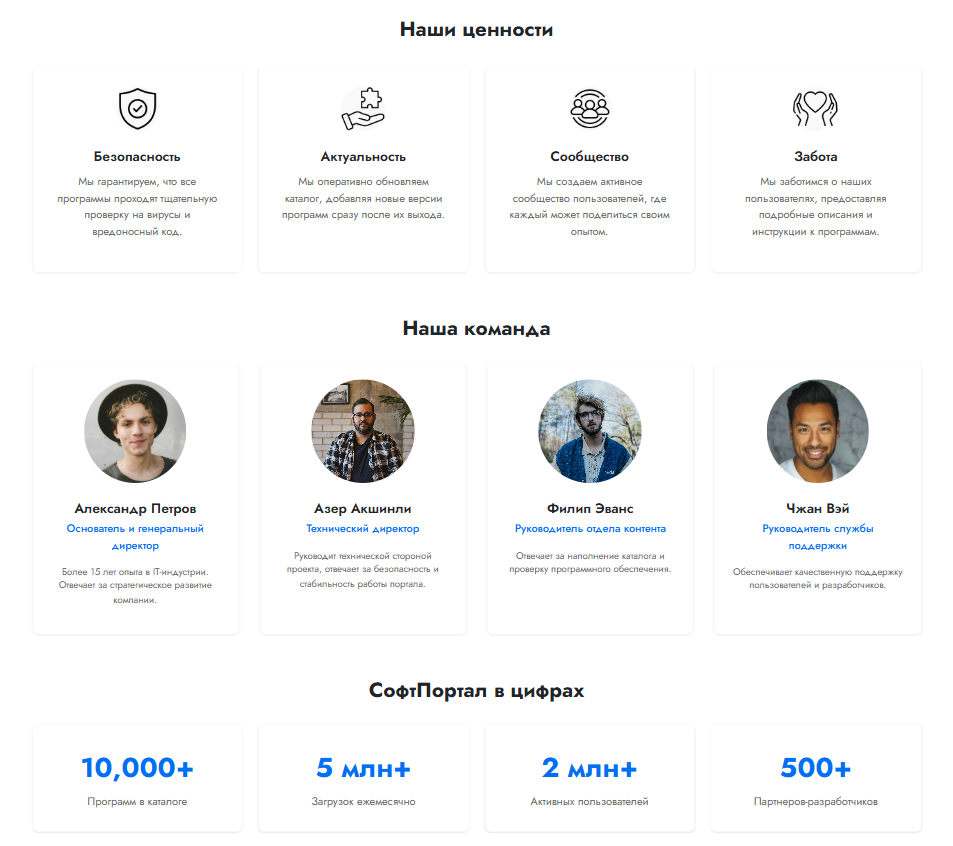
**Сетка статистики**

* Блок <div class="stats-grid"> организует статистические карточки.

**Карточки статистики**

* Каждая карточка оформлена в <div class="stat-card">, внутри которой:
  + Крупное число или показатель <div class="stat-number">.
  + Подпись к числу <div class="stat-label">.

Статистика визуально подчеркивает масштаб и успех проекта.



*Рис. 2.13*

**Секция «Свяжитесь с нами»**

**Шаг 1. Основная структура**

* Секция обёрнута в тег <section class="about-contact">.
* Это позволяет отдельно оформить фон, отступы и выравнивание элементов в этой части страницы.

**Шаг 2. Заголовок секции**

* Добавлен заголовок <h2>Свяжитесь с нами>.
* Заголовок сразу дает понять пользователю, о чём эта часть сайта.

**Шаг 3. Основное содержимое: сетка контактов**

Контент разбит на 2 колонки с помощью контейнера <div class="contact-grid">:

**Левая часть — контактная информация:**

* Внутри блока <div class="contact-info"> перечислены способы связи:
  + **Email** с двумя почтовыми адресами.
  + **Телефон** с основным и бесплатным для Азербайджана номерами.
  + **Адрес** офиса с встроенной картой Google Maps (<iframe>).

Для каждого контакта используется:

* Блок <div class="contact-item">, в который входят:
  + Название контакта (<h3>) и описание (<p>).

**Вставка карты:**

* Для адреса в контактах подключена карта через тег <iframe>.

**Шаг 4. Правая часть — форма обратной связи**

* В блоке <div class="contact-form"> размещена форма <form> для отправки сообщений.
* Форма содержит:
  + Поле для имени (input type="text").
  + Поле для email (input type="email").
  + Поле для темы сообщения (input type="text").
  + Большое поле для текста сообщения (<textarea>).

Каждое поле обёрнуто в <div class="form-group"> для удобной стилизации.

* Кнопка отправки формы оформлена через <button type="submit" class="button primary">Отправить сообщение</button>.

**Описание структуры подвала сайта**

**1. Основная обёртка**

* Подвал размещен внутри тега <footer class="footer">.
* Внутри — контейнер <div class="container footer-container">, чтобы контент был аккуратно выровнен и имел одинаковые отступы по ширине сайта.

**2. Левая часть подвала — логотип и копирайт**

В <div class="footer-left"> размещено:

* **Логотип сайта**:

<a href="index.html" class="logo">

<span class="logo-text">СофтПортал</span>

</a>

Ссылка на главную страницу с текстом "СофтПортал".

* **Копирайт**:

<p style="margin-bottom: 0px;">&copy; <span id="current year"></span>СофтПортал...</p>

Используется тег <span id="current-year">, чтобы при помощи JavaScript автоматически подставлять текущий год.

**3. Правая часть подвала — навигационные ссылки**

В <div class="footer-links"> находятся ссылки:

* **О нас** — с классом active, чтобы подсвечивать её как текущую страницу (если пользователь находится на about.html).
* **Контакты**, **Конфиденциальность**, **Условия использования** — пока ведут на заглушки (#), но при желании можно будет подключить настоящие страницы.

Теперь приступим к третьей странице, название которого – “software-details.html”.

Шапка у него такая же как в первых двух как подвал и навбар, поэтому рассмотрим внутреннюю часть.

Создаем контейнер с классом “container software-details-container” и размещаем внутри него элементы.

**Описание раздела страницы программного обеспечения**

**1. Навигация «хлебные крошки» (breadcrumbs)**

<nav class="breadcrumbs">

<ol class="breadcrumbs-list">

<li><a href="index.html">Главная</a></li>

<li>/</li>

<li><a href="#">Категории</a></li>

</ol>

</nav>

**2. Заголовок программы + основная информация (software-header)**

**Левая часть — изображение программы:**

<div class="software-image">

<img src="./images/Antivirus2.svg" alt="Антивирус Pro">

</div>

Показывается иконка/картинка приложения, чтобы пользователь узнал продукт.

**Правая часть — текстовая информация:**

<div class="software-info">

**Внутри**:

* **Название и категория программы**:

<div class="software-title-container">

<h1>Антивирус Pro</h1>

<span class="badge">Безопасность</span>

</div>

* **Рейтинг и количество загрузок**:

<div class="software-rating">

<div class="stars">★★★★☆</div>

<span class="rating-value">4.8</span>

<span class="downloads">(15,000 загрузок)</span>

</div>

* **Описание программы**:

<p class="software-description">

Надежная защита от вирусов и вредоносных программ...

</p>

* **Кнопки действий**:

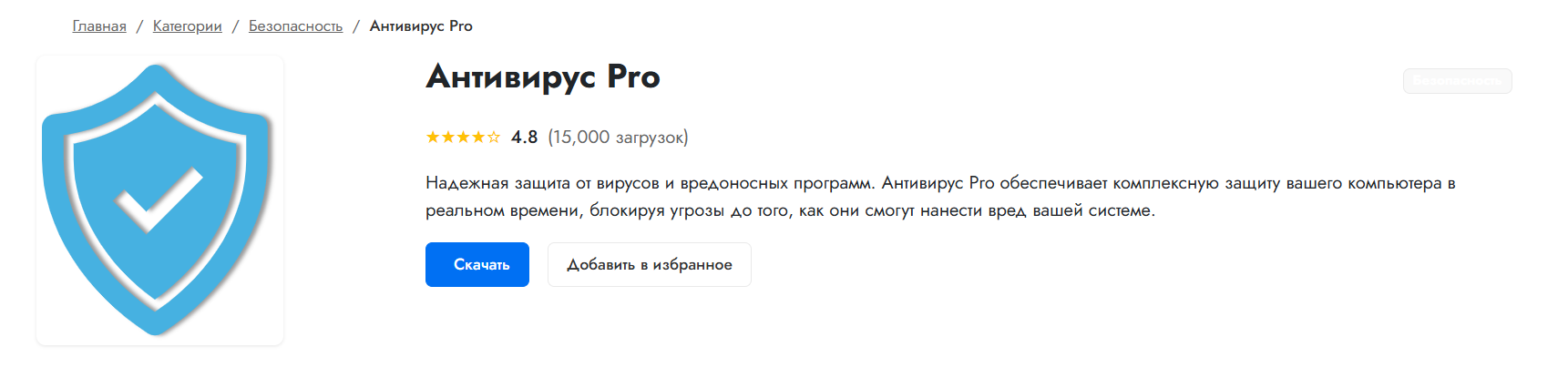
<div class="software-actions">

<button class="button primary download-button">Скачать</button>

<button class="button outline">Добавить в избранное</button>

</div>

Результат(см рис. 2.14):



*рис. 2.14*

Описание компонента "Вкладки" для страницы программного обеспечения

**1. Общая структура**

Компонент представляет собой систему вкладок для отображения различной информации о программном обеспечении. Состоит из двух основных частей:

- Панель переключения вкладок (таб-бар)

- Контентные области для каждой вкладки

**2. Элементы управления (таб-бар)**

<div class="tabs-list">

<button class="tab-button active" data-tab="details">Детали</button>

</div>

- Четыре кнопки-вкладки с соответствующими названиями

- Атрибут `data-tab` связывает кнопку с контентом

- Класс `active` отмечает текущую активную вкладку ("Детали" по умолчанию)

**3. Контентные области**

Вкладка "Детали" (активная по умолчанию)

Содержит три блока информации:

1. Карточка "Информация:

- Версия программы (12.5.1)

- Разработчик (Security Solutions Inc.)

- Дата выпуска (2023-10-15)

- Размер дистрибутива (85 MB)

- Тип лицензии (Платная с пробным периодом)

2. Карточка "Поддерживаемые ОС:

- Windows 10/11

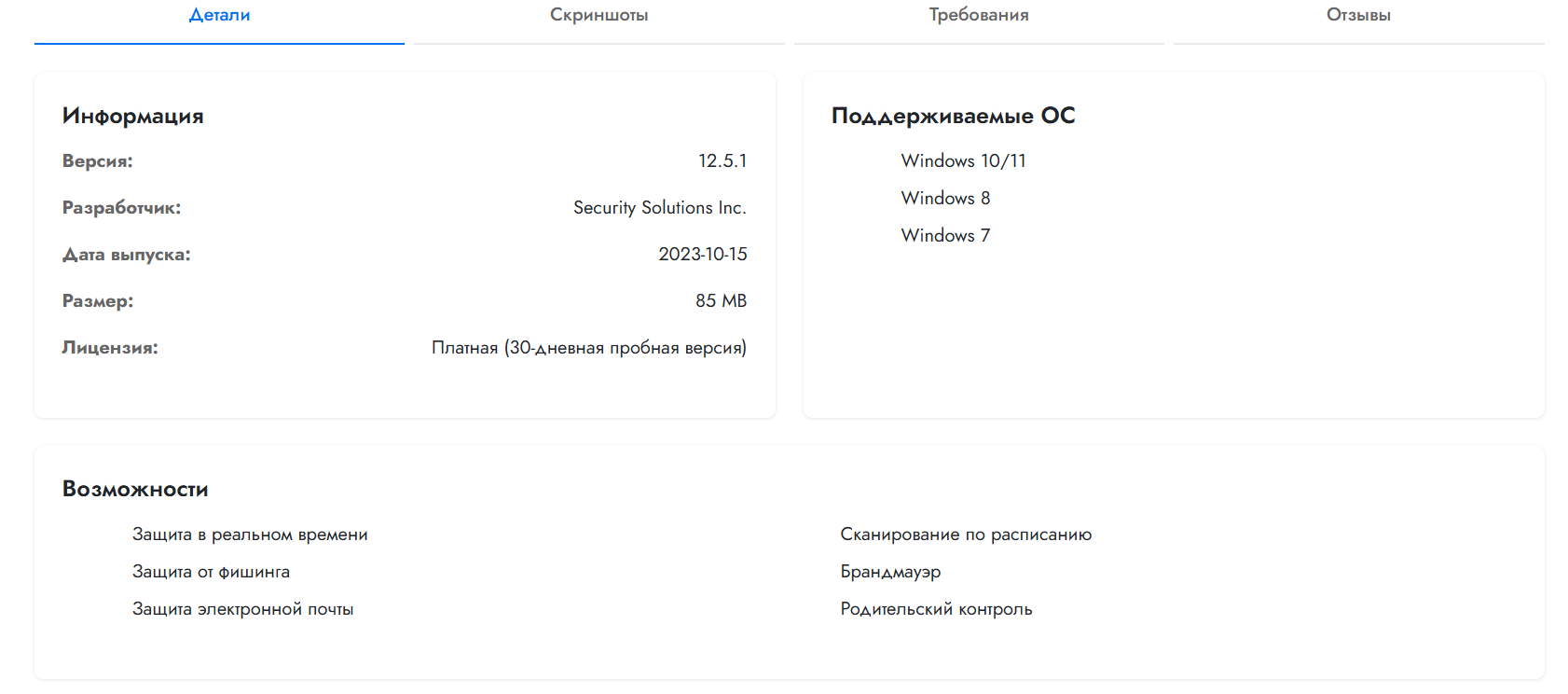
- Windows 8

- Windows 7

3. Карточка "Возможности:

- 6 ключевых функций антивируса

- Каждая функция отмечена иконкой подтверждения

Результат(см рис. 2.15):

*рис. 2.15*

## **2.4 Использование стилей CSS**

**Описание стилей в CSS файле (styles.css)**

**Базовые стили**

- Цветовая схема: Используются CSS-переменные для цветов (`--primary-color`, `--text-color` и др.).

- Шрифты: Основной шрифт - `Inter` с запасными вариантами.

- Общие стили: Сброс отступов, `box-sizing: border-box`, гибкая структура (`flexbox`) для макета.

**Шапка (`header`)**

- Фиксированная позиция: Прилипает к верху страницы.

- Адаптивность: Навигация скрыта на мобильных устройствах (`display: none` при `< 768px`).

- Поиск: Поле поиска с иконкой, появляется только на десктопах.

**Главная страница (`index.html`)**

- Герой-секция: Градиентный фон, адаптивный макет (колонки на десктопе, стопка на мобильных).

- Карточки программ: Сетка (`grid`), адаптируется от 1 до 3 колонок в зависимости от ширины экрана.

- Кнопки: Стили для `.primary` (основная) и `.outline` (контурная).

**Подвал (`footer`)**

- Гибкий макет: Колонки на мобильных, строка на десктопах.

- Ссылки: Стилизованы, активная ссылка выделена цветом.

**Страница "О нас" (`about.html`)**

- Сетки: Используется `grid` и `flexbox` для ценностей, команды и статистики.

- Адаптивность: Колонки меняются от 1 до 4 в зависимости от ширины экрана.

- Карточки: Единый стиль для ценностей, членов команды и статистики.

**Страница программы (`software-details.html`)**

- Вкладки: Стили для переключения между контентом.

- Детали: Адаптивная сетка для информации и системных требований.

**Общие компоненты**

- Хлебные крошки: Стили для навигационной цепочки.

- Рейтинг: Звезды (`★★★★☆`) с цветом `#ffc107`.

- Адаптивные изображения: `max-width: 100%`, `height: auto`.

**Медиазапросы**

- Адаптация под разные устройства: мобильные (`< 640px`), планшеты (`768px`), десктопы (`1024px`).

Особенности

- Тени: Используются для карточек (`box-shadow`).

- Закругления: `border-radius` для кнопок, карточек, изображений.

- Анимации: Плавные переходы для hover-эффектов (`transition`).

## **2.5.Функциональность с помощью JS**

Файл `main.js` содержит логику для работы веб-сайта СофтПортал, включая:

- Поиск программ в реальном времени

- Переключение вкладок

- Автоматическое обновление года в подвале

**1. Данные о программах (`softwareProducts`)**

Массив объектов с информацией о ПО:

const softwareProducts = [{

id: '1',

title: 'Антивирус Pro',

description: 'Надежная защита от вирусов и вредоносных программ',

category: 'Безопасность',

image: './images/Antivirus2.svg'

}];

**2. Поиск в реальном времени**

**Инициализация поиска (`initRealTimeSearch`)**

- Находит все поля поиска на странице.

- Добавляет обработчики событий:

- `input` – динамический поиск при вводе текста.

- `keypress` – обработка нажатия `Enter`.

- Создает контейнер для результатов поиска.

**Фильтрация и отображение (`performRealTimeSearch`)**

- Фильтрует программы по:

- Названию (`title`)

- Описанию (`description`)

- Категории (`category`)

- Показывает результаты в выпадающем списке.

**Пример вывода:**

<ul class="search-results-list">

<li class="search-result-item">

<a href="software-details.html?id=1">

<img src="./images/Antivirus2.svg" alt="Антивирус Pro">

<div>

<div class="search-result-title">Антивирус Pro</div>

<div class="search-result-category">Безопасность</div>

</div>

</a>

</li>

</ul>

**3. Переключение вкладок**

Инициализация вкладок (`initTabs`)

- Обрабатывает клики по кнопкам вкладок.

- Показывает/скрывает соответствующий контент.

**Динамическое создание контента**

Вкладки "Требования", "Отзывы" и "Скриншоты" отсутствуют в HTML, они создаются через JS:

**Пример для вкладки "Требования":**

function createRequirementsTab() {

const requirementsTab = document.createElement("div");

requirementsTab.className = "tab-content";

requirementsTab.id = "requirements-tab";

requirementsTab.innerHTML = `...`; // HTML с системными требованиями

return requirementsTab;}

**4. Автоматическое обновление года в подвале**

var footerYear = document.getElementById('current-year')

footerYear.innerHTML = new Date().getFullYear();

**5. Запуск всех функций при загрузке страницы**

document.addEventListener("DOMContentLoaded", () => {

initRealTimeSearch(); - Поиск

initTabs(); - Вкладки

updateYear(); - Год в футере

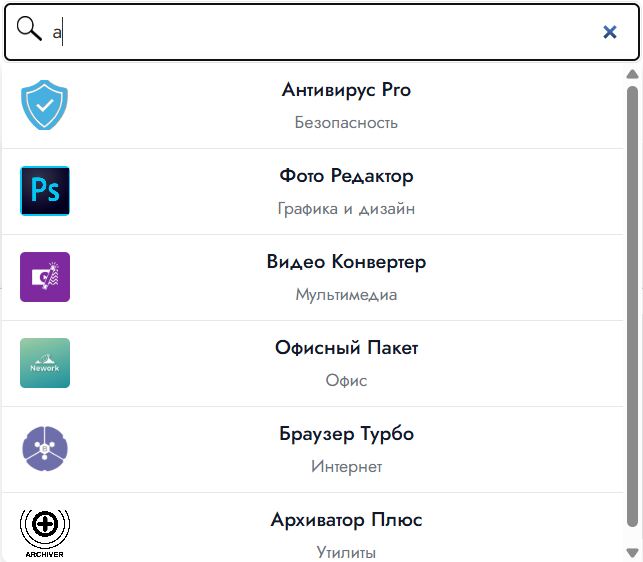
});

**Итог:**

Файл `main.js` обеспечивает:

* Динамический поиск с подсказками.
* Переключение вкладок без перезагрузки страницы.
* Автоматическое обновление года в подвале.

Результат JavaScript-кода :



*Рис. 2.16*

Изображение выглядит как снимок экрана, текст, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

*Рис. 2.17*

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

*Рис. 2.18*

Изображение выглядит как текст, снимок экрана, Шрифт, число

Контент, сгенерированный ИИ, может содержать ошибки.

*Рис. 2.19*

# **3. РЕЗУЛЬТАТ**

За время выполнения курсовой работы были изучены принципы разработки web-сайта:

1. Мы создали веб-страницу используя HTML.
2. Стили выполнены с использованием CSS
3. Создана адаптивная веб-страница с использованием медиа-запросов.
4. Добавлена функциональность с помощью Javascript

В результате была создана современная, функциональная и эстетичная веб-страница, которая удовлетворяет требованиям заказчиков, удобна для пользователей и соответствует современным трендам в веб-дизайне.

Ссылка на веб-страницу: <https://idyllic-chebakia-ee4616.netlify.app/>

# **4. ЛИТЕРАТУРА**

* <https://ru.wikipedia.org/wiki/HTML>
* <https://www.litres.ru/static/or3/view/or.html?art_type=4&bname=%25D0%259E%25D1%2581%25D0%25BD%25D0%25BE%25D0%25B2%25D1%258B%2520%25D0%25B2%25D0%25B5%25D0%25B1-%25D0%25BF%25D1%2580%25D0%25BE%25D0%25B3%25D1%2580%25D0%25B0%25D0%25BC%25D0%25BC%25D0%25B8%25D1%2580%25D0%25BE%25D0%25B2%25D0%25B0%25D0%25BD%25D0%25B8%25D1%258F&art=68904438&user=0&uuid=149b73eb-b22e-11ed-8166-ac1f6b4dd332&cover=%2Fpub%2Fc%2Fcover%2F68904438.jpg&uilang=ru&trial=1>
* <https://sky.pro/wiki/html/chto-takoe-veb-programmirovanie/?utm_source>
* <https://developer.mozilla.org/ru/docs/Learn_web_development/Getting_started/Your_first_website/Styling_the_content>
* <https://developer.mozilla.org/ru/docs/Web/JavaScript>

# **5. ПРИЛОЖЕНИЯ**

Репозиторий:

* <https://github.com/celil-dunya2005/Kursovaya-veb>

Ссылка на сайт:

* <https://idyllic-chebakia-ee4616.netlify.app/>