МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

ФАКУЛЬТЕТ АВТОМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

Кафедра автоматизированных систем управления

Эссе

Дисциплина: Системы управления веб-контентом

Задание выполнил:

Ревединский Кирилл Дмитриевич

АВТФ, АПМ-24



#### Преподаватель:

Хворостов Владимир Александрович

Новосибирск, 2025

**ВВЕДЕНИЕ**

В рамках курса веб-разработки были выполнены пять лабораторных работ, которые позволили изучить основы HTML, CSS и принципы создания веб-страниц. Каждая лабораторная работа включала конкретные задания, направленные на освоение важных аспектов вёрстки: от структуры HTML-документа до создания адаптивных блоков и навигационных меню. В ходе выполнения лабораторных работ мы познакомились с базовой структурой HTML-документа, научились использовать CSS для стилизации элементов, а также создали адаптивные веб-страницы. Данный курс позволил закрепить полученные знания на практике и получить навыки, необходимые для создания современных пользовательских интерфейсов.

В данном работе собрана вся информация и выводы, которые были получены мной в процессе выполнения лабораторных работ, а также примеры того, как эти знания могут быть полезны в дальнейшем, включая написание магистерской диссертации.

**Лабораторная работа №1: Основы HTML**

В этой лабораторной мы освоили базовые теги HTML: Основные теги для разметки текста в HTML включают теги для структурной разметки (например, <html>, <head>, <body>), теги для форматирования текста (например, <b>, <i>, <u>, <p>, <br>), теги для заголовков (например, <h1>-<h6>), теги для списков (например, <ul>, <ol>, <li>), теги для таблиц (например, <table>, <tr>, <td>, <th>), теги для ссылок (например, <a>), а также теги для изображений (например, <img>).

Более подробно:

* **<html>:** обозначает корневой элемент HTML-страницы.
* **<head>:** содержит служебную информацию о документе, например, заголовок страницы, мета-теги и стили.
* **<body>:** содержит основной контент страницы, который виден пользователю.
* **<b>, <i>, <u>:** форматируют текст как жирный, курсивный и подчеркнутый, соответственно.
* **<p>:** представляет абзац текста.
* **<br>:** заставляет текст перейти на новую строку.
* **<h1>-<h6>:** определяют заголовки различных уровней.
* **<ul>, <ol>:** представляют немаркированный и нумерованный списки, соответственно.
* **<li>:** представляет элемент списка.
* **<table>, <tr>, <td>, <th>:** используются для создания таблиц.
* **<a>:** создает гиперссылку.
* **<img>:** вставляет изображение.

Первое задание заключалось в установке текстового редактора Notepad++ (но я пользовался WebStorm) и создании HTML-документа с корректной структурой. В документе я использовал обязательные элементы: <!DOCTYPE html>, <html>, <head>, <title>, <meta charset>, а также наполнил тело страницы (<body>) произвольным содержанием. Затем я освоил основные HTML-теги для форматирования текста, такие как абзацы (<p>), заголовки (<h1>–<h6>), списки (<ul>, <ol>, <li>), а также теги для выделения текста (<strong>, <em>, <b>, <i>, <br>, <hr>, <pre>). В рамках задания также была вставка изображения с помощью тега <img> и создание простых форм и таблиц. Я научился работать с тегами <form>, <input>, <select>, <textarea>, <table>, <tr>, <td>. После этого создал второй HTML-документ и связал его с первым при помощи гиперссылок (<a>). Итоговым этапом была проверка валидности HTML-кода через официальный валидатор W3C. Эта лабораторная заложила фундаментальные знания о структуре веб-страниц и базовой семантике HTML.

Выводы:

Понял, насколько важно соблюдать структуру HTML-документа для корректного отображения в браузерах и доступности контента. Семантические теги делают сайт более читаемым не только для людей, но и для поисковых систем.

**Лабораторная работа № 2: Основы CSS**

Вторая лабораторная работа была посвящена изучению CSS. Освоил теги для форматирования текста (<b>, <i>, <u>, <strong>, <em>, <mark> и др.) и создания гиперссылок (<a>). Познакомился с атрибутами href, target, title.

Были изучены каскадные таблицы стилей (CSS) и три способа их подключения к HTML: внешний файл, внутренний стиль в <style> и встроенный стиль через атрибут style. Я реализовал все три способа, а также закрепил различия между стилями, назначенными на тег, класс и id. Особое внимание уделялось пониманию различий между каскадированием — принципом приоритета правил в зависимости от специфичности, и наследованием — автоматическим принятием свойств от родительских элементов. Я отработал эти понятия на практике, написав примеры с перекрывающимися стилями и вложенными элементами. Вторым заданием было создание страницы по образцу, где я применил стилизацию для создания аккуратного, визуально приятного макета, похожего на предложенный пример Эта лабораторная научила меня, как стилизовать HTML-документ, настраивать цвета, шрифты, отступы и прочие визуальные параметры.

Наследование и каскадирование:

• Наследование: некоторые свойства (цвет, шрифт) передаются от родителя к потомку автоматически.

• Каскадирование: если для одного элемента применяются несколько правил, побеждает то, что:

o более специфично (например, #id > .class > tag);

o находится ниже в коде (если специфичность равна);

o использует !important (имеет наивысший приоритет).

Выводы:  
 Форматирование текста позволяет улучшить визуальное восприятие информации, а ссылки — сделать сайт интерактивным и связанным с другими ресурсами.

**Лабораторная работа № 3: Блочная модель и позиционирование**

Третья лабораторная была посвящена блочной модели и компоновке. В первом задании я изучил свойства, управляющие размерами, отступами, границами и положением блоков: width, height, margin, padding, border, а также position (со значениями static, relative, absolute), display (inline, block, inline-block) и float. Я продемонстрировал отличия между способами позиционирования на примерах, а также визуально показал, как работают различные типы отображения и обтекание. Вторым заданием была вёрстка макета “Идеи для подарков” — с применением всех перечисленных свойств. В процессе я научился выравнивать элементы, размещать изображения и текст по макету, использовать блочную структуру и отступы, чтобы добиться визуального соответствия заданному примеру.

Выводы:  
 Списки и таблицы — инструменты для представления информации в структурированном виде. Я научился использовать вложенные списки и объединение ячеек.

**Лабораторная работа № 4: Фоны и Flexbox**

Заключительная лабораторная работа была направлена на стилизацию элементов интерфейса, использование фонов и взаимодействие с пользователем. В первом задании я освоил свойства background-color, background-image, background-position и комбинированное свойство background. Также я создал ссылку, которая меняет цвет при наведении с помощью псевдокласса :hover, и реализовал изменение фона блока при наведении. Затем я перешёл к изучению Flexbox — мощного инструмента для гибкой вёрстки. Я использовал Flexbox для адаптации макета из предыдущей лабораторной и убедился, насколько проще и удобнее становится компоновка элементов. В последнем задании я создал выпадающее меню с использованием вложенных списков и псевдокласса :hover. Эта лабораторная показала, как можно оживить интерфейс с помощью CSS и добавить интерактивности, не прибегая к JavaScript.

**Лабораторная работа № 5. Основы JS и git**

В пятой лабораторной работе основной акцент был сделан на практическом применении системы контроля версий Git и платформы GitHub, а также на более сложной веб-разработке с использованием сторонних библиотек и логики взаимодействия с пользователем.

Первым заданием было знакомство с системой Git. Я создал новый публичный репозиторий специально для этой лабораторной. Затем с помощью Git и WebStorm я клонировал репозиторий на локальную машину, связал локальную директорию с удалённой и освоил базовые команды работы с Git в командной строке: git clone, git add, git commit, git push. Эта часть лабораторной дала представление о том, как отслеживать изменения в коде, делать коммиты с комментариями, управлять версионностью и публиковать свою работу в удалённом хранилище. Я также самостоятельно установил Git на свой компьютер по инструкции с официального сайта.

Во втором задании требовалось реализовать веб-страницу со слайдером изображений. Я изучил несколько популярных JavaScript-библиотек и выбрал подходящую для задачи. Слайдер был интегрирован в HTML-документ, стилизован с помощью CSS, и обеспечивал прокрутку изображений, создавая интерактивную и привлекательную галерею. Я добавил адаптивность, чтобы слайдер корректно отображался на разных устройствах. По завершению проекта я выполнил коммит и отправил изменения в репозиторий GitHub.

Третье задание было самым интересным с точки зрения логики: создание калькулятора вкладов. Я разработал страницу, в которой пользователь мог выбрать вид вклада из выпадающего списка — "Пополняемый" или "Срочный", после чего динамически подгружались соответствующие сроки. Для каждого срока была задана своя процентная ставка. После заполнения формы (сумма вклада, вид, срок) и нажатия на кнопку "Рассчитать", пользователь получал итоговое сообщение с рассчитанной суммой по вкладу. Я реализовал базовую валидацию данных, чтобы исключить пустые поля и некорректный ввод. Вся логика была реализована на JavaScript, интерфейс — на HTML и CSS. Завершив задание, я также сделал коммит и пуш проекта в GitHub, удостоверившись в актуальности и прозрачности всех изменений.

Эта лабораторная работа стала логическим продолжением предыдущих, дополнив навыки веб-разработки умением использовать систему контроля версий и реализовывать более сложные формы взаимодействия с пользователем. Я приобрёл уверенность в публикации проектов в интернете, работе с внешними библиотеками и в разработке интерактивных элементов интерфейса.

**Заключение**

Все лабораторные работы дали мне прочное представление о том, как создаются современные веб-страницы. Я научился писать корректный HTML, применять CSS для оформления и построения макетов, работать с позиционированием и компоновкой элементов, а также использовать Flexbox и псевдоклассы для создания интерактивных элементов. Эти знания заложили базу для дальнейшего изучения фронтенд-разработки и более сложных технологий.