Специальность 09.02.07 Информационные системы и программирование

Выполнил  
Обучающийся 3 курса

группы ИП-1 Калинин Алексей Викторович

преподаватель Кирилов Назари Андреевич

Курсовая работа защищена с оценкой

г. Жуковский, 2024г.

**Введение**

В эпоху цифровых технологий наличие онлайн-присутствия стало неотъемлемой частью успешного ведения бизнеса и личного брендинга. Сайты-визитки представляют собой компактные и информативные веб-страницы, которые позволяют пользователям быстро ознакомиться с основными данными о компании или индивидууме. Они служат эффективным инструментом для привлечения клиентов, предоставления информации о предлагаемых услугах и установления контакта. Среди множества инструментов и технологий, доступных для веб-разработки, фреймворк Tailwind CSS выделяется своей уникальной концепцией "utility-first". Этот подход позволяет разработчикам использовать готовые классы для стилизации элементов, что значительно ускоряет процесс разработки и упрощает создание адаптивных интерфейсов. Tailwind CSS предоставляет разработчикам возможность сосредоточиться на дизайне и функциональности, минимизируя время, затрачиваемое на написание кастомных стилей.

# Глава 1. Теоретическая часть

* 1. ***Обзор***

Tailwind CSS — это утилитарный CSS-фреймворк, который предоставляет разработчикам возможность создавать пользовательские дизайны без необходимости писать собственные стили. Он основан на концепции "utility-first", что означает, что вместо написания CSS-классов для каждого элемента, разработчики могут использовать готовые классы для стилизации.

***Преимущества использования Tailwind CSS***

*Быстрота разработки*:- Благодаря готовым классам можно быстро создавать интерфейсы, что особенно важно в условиях ограниченных сроков.

*Адаптивность*- Tailwind позволяет легко создавать адаптивные дизайны с помощью медиазапросов, что делает сайт удобным для пользователей на различных устройствах.

*Настраиваемость*- Фреймворк легко настраивается под нужды проекта, позволяя разработчикам добавлять свои собственные стили и компоненты.

Сообщество и поддержка- У Tailwind CSS большое сообщество, что облегчает поиск решений и примеров, а также предоставляет доступ к множеству готовых компонентов и шаблонов.

На рис.1.1 вёрстка с помощью Tailwind

На рис 1.2 тот же результат с использованием классического CSS

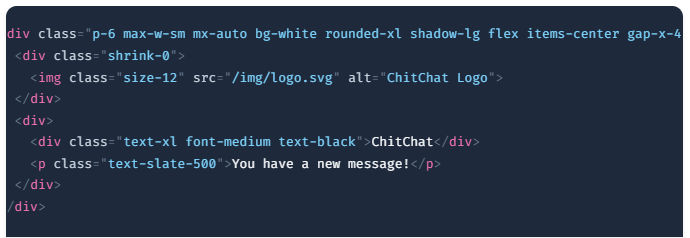
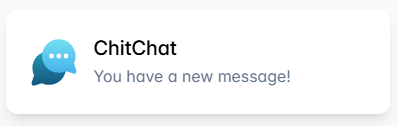
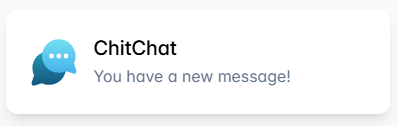
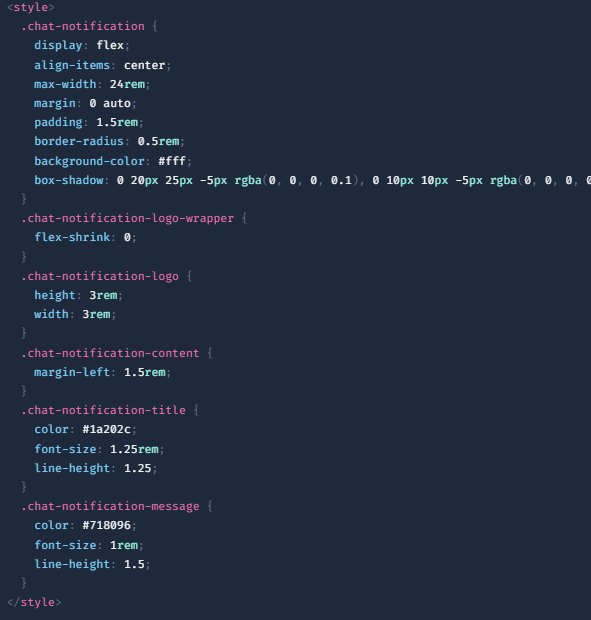


Рис.1.1



Рис.1.2

**1.2 Установка**

**Установка через npm**

Если вы планируете использовать Tailwind CSS в более сложном проекте, рекомендуется установить его через npm. Это позволяет вам настраивать Tailwind и использовать его в сочетании с другими инструментами сборки, такими как Webpack или Parcel. Шаги: Убедитесь, что у вас установлен Node.js и npm. Вы можете проверить это, выполнив команды node -v и npm -v в терминале. Создайте новую папку для вашего проекта и перейдите в нее:

mkdir my-project

cd my-project

*Инициализируйте новый проект npm:*

npm init –y

*Установите Tailwind CSS:*

npm install tailwindcss

*Создайте файл конфигурации Tailwind*:

npx tailwindcss init

*Это создаст файл****tailwind.config.js****, который вы можете использовать для настройки Tailwind.*

*Создайте файл CSS, например styles.css, и добавьте в него следующие строки:*

@tailwind base;

@tailwind components;

@tailwind utilities;

*Скомпилируйте ваш CSS с помощью Tailwind. Вы можете использовать команду:*

npx tailwindcss build styles.css -o output.css

# Глава 2. Практическая часть

**Проектирование сайта-визитки**

Перед началом разработки необходимо спроектировать структуру сайта-визитки она включает следующие разделы:

*Навигация*- для более удобного использования сайта

*Главная* - краткая информация о вас или вашей компании

*Навыки*- описание предоставляемых услуг.

*Контакты*- форма обратной связи и контактная информация.

**1.Навигация**





Обертка <header>: Используется элемент <header> для обозначения верхней части страницы, содержащей навигацию.

Классы bg-black, shadow-md и text-white задают черный фон, добавляют тень и устанавливают белый цвет текста соответственно. Это создает контрастный и современный вид.

Контейнер: Внутри <header> находится <div> с классами container mx-auto p-4 flex justify-between items-center.

Класс container ограничивает ширину содержимого и центрирует его на странице.

mx-auto добавляет автоматические отступы по бокам, что также помогает в центрировании. p-4 добавляет внутренние отступы, чтобы текст не прилипал к краям.

flex justify-between items-center делает контейнер флекс-контейнером, выравнивая элементы по горизонтали с равными промежутками между ними и выравнивая их по центру по вертикали.

Заголовок: Элемент <h1> содержит название сайта "Сайт визитка".

Классы text-2xl и font-bold увеличивают размер шрифта и делают его жирным, что подчеркивает важность заголовка.

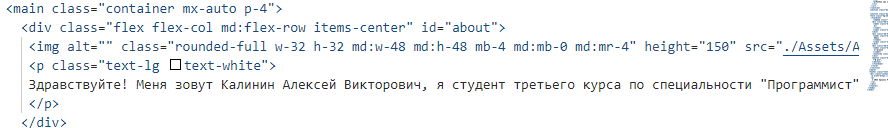
Навигация: Элемент <nav> содержит список навигационных ссылок, представленных в виде ненумерованного списка <ul>.

Класс flex делает список флекс-контейнером, а space-x-4 добавляет горизонтальные отступы между элементами списка.

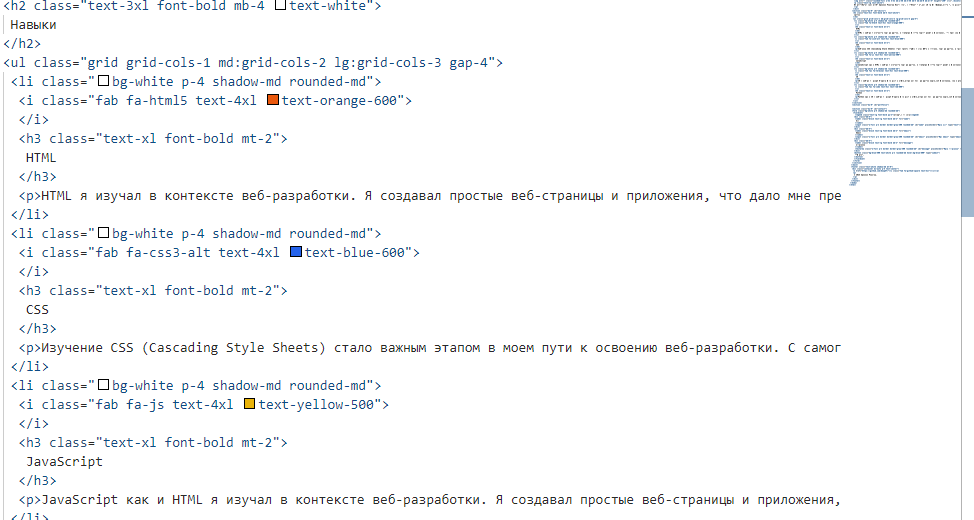
Ссылки: Каждая ссылка <a> ведет к соответствующему разделу страницы (например, "Обо мне", "Навыки", "Связь").

Класс hover:text-blue-500 изменяет цвет текста на синий при наведении курсора, что улучшает взаимодействие с пользователем и делает навигацию более интерактивной.

**2.Главная**

Раздел "Обо мне" является ключевым элементом сайта-визитки, так как он предоставляет посетителям информацию о вас, ваших целях и профессиональном опыте. В данном разделе представлена следующая информация:

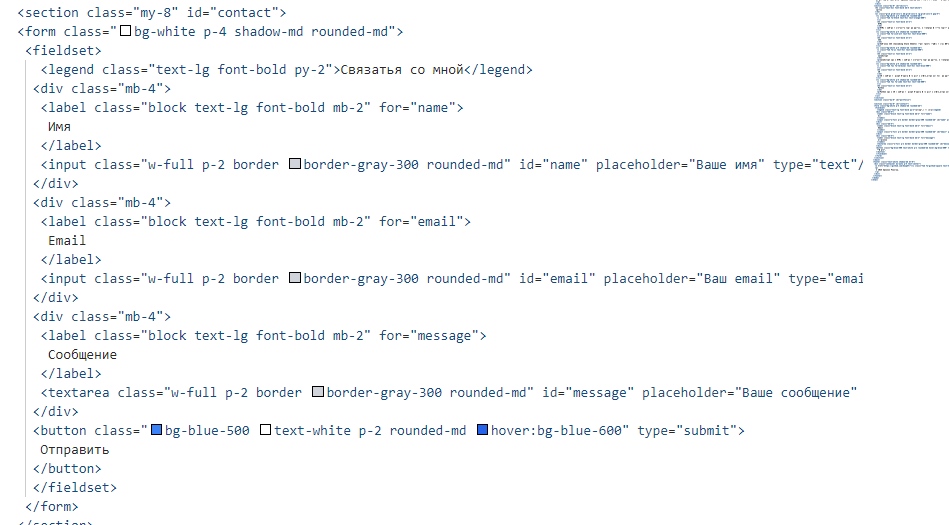
**3.Навыки**

Раздел "Навыки" на вашем сайте-визитке представляет собой важный элемент, который демонстрирует ваши профессиональные компетенции и технические умения. Этот раздел позволяет потенциальным работодателям, клиентам и коллегам быстро оценить ваши способности и уровень подготовки в области программирования и веб-разработки.

#### **Структура раздела**

1. **Заголовок**: Раздел начинается с четкого заголовка "Навыки", который выделяется благодаря большому шрифту и жирному начертанию. Это помогает привлечь внимание и сразу обозначает содержание раздела.
2. **Список навыков**: Навыки представлены в виде списка, где каждый пункт включает:
   1. **Иконка**: Использование иконок из библиотеки Font Awesome для каждого навыка (например, HTML, CSS, JavaScript и т.д.) делает раздел визуально привлекательным и легко воспринимаемым. Иконки помогают быстро идентифицировать технологии и языки программирования.
   2. **Название навыка**: Каждое название навыка выделено заголовком второго уровня, что подчеркивает его важность.
   3. **Описание**: Под каждым названием навыка представлено краткое описание, в котором вы делитесь своим опытом и пониманием технологии.

**4.Контакты**

Раздел "Контакты" на вашем сайте-визитке является важным элементом, который предоставляет посетителям возможность связаться с вами. Этот раздел играет ключевую роль в установлении коммуникации и может быть решающим фактором для потенциальных работодателей, клиентов или коллег, которые заинтересованы в вашем опыте и навыках.

#### **Структура раздела**

1. **Заголовок**: Раздел начинается с четкого заголовка "Связь со мной", который сразу указывает на его содержание и цель. Заголовок выделен с помощью жирного шрифта, что привлекает внимание.
2. **Форма обратной связи**: Основной элемент раздела — это форма, которая позволяет пользователям отправлять вам сообщения. Форма включает следующие поля:
   1. **Имя**: Поле для ввода имени пользователя, что помогает сделать общение более личным.
   2. **Email**: Поле для ввода адреса электронной почты, на который вы сможете ответить. Это критически важный элемент, так как он позволяет вам поддерживать связь.
   3. **Сообщение**: Текстовое поле для ввода сообщения, где пользователи могут задать вопросы, оставить комментарии или выразить интерес к сотрудничеству.
3. **Кнопка отправки**: Внизу формы расположена кнопка "Отправить", которая выделяется цветом и стилем. Это позволяет пользователям легко завершить процесс отправки сообщения.