Министерство просвещения Приднестровской Молдавской Республики

Государственное образовательное учреждение среднего профессионального образования «Тираспольский техникум информатики и права»

ИНДИВИДУАЛЬНЫЙ ПРОЕКТ

«Веб-страница «Расписание уроков»»

по учебной дисциплине «Информатика»

по специальности 09.02.07 «Информационные системы и программирование»

Выполнил Кирчу И.И.

обучающийся I курса

специальность 09.02.07 «Информационные системы и программирование»

Руководитель Шандригоз Наталья Николаевна

Преподаватель информатики высшей квалификационной категории

Допущен к защите

«\_\_\_»\_\_\_\_\_\_\_2024 г.

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Тирасполь 2024

**СОДЕРЖАНИЕ**

|  |  |
| --- | --- |
| ВВЕДЕНИЕ | 3 |
| 1. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ СОЗДАНИЯ ВЕБ-СТРАНИЦЫ | 5 |
| 1.1. Виды веб-страниц | 5 |
| 1.2. Основные способы создания веб-страниц (сайтов) | 7 |
| 1.3. Язык гипертекстовой разметки документа HTML и его назначение | 8 |
| 2. ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ ИНДИВИДУАЛЬНОГО ПРОЕКТА | 14 |
| 2.1. Постановка задачи | 14 |
| 2.2. Этапы создания веб-страницы | 14 |
| 2.3. Тестирование и отладка кода веб-страницы | 15 |
| ЗАКЛЮЧЕНИЕ | 23 |
| СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ | 24 |

**ВВЕДЕНИЕ**

Актуальность темы состоит в необходимости оптимизации учебного процесса через автоматизацию составления расписания уроков. В условиях постоянно меняющихся требований к образовательной программе и ограниченных ресурсов, эффективное расписание становится ключевым фактором успешного управления временем и ресурсами образовательных учреждений.

Теоретическая значимость работы заключается в разработке методологии применения веб-технологий для автоматизации процесса составления расписания, что может привести к повышению качества образования и улучшению управления учебными заведениями.

Практическая целесообразность работы проявляется в следующем:

- изучены современные подходы к веб-программированию и разработке приложений.

- разработан прототип веб-приложения для автоматизации составления расписания уроков.

Целью проектной работы является создание веб-приложения для автоматизации составления расписания уроков, что позволит учебным заведениям более эффективно управлять учебными ресурсами.

Задачи исследования

1. определить основные требования и ограничения при составлении расписания уроков;
2. раскрыть возможности веб-технологий в автоматизации учебных процессов;
3. разработать алгоритмы для автоматизированного составления расписания;
4. исследовать пользовательский опыт работы с веб-приложением и дать рекомендации по его улучшению;

Предметом исследования является процесс автоматизации составления расписания уроков с использованием веб-технологий.

Объектом исследования выступают веб-приложения для составления расписания уроков.

Период исследования охватывает разработку и тестирование веб-приложения, начиная с текущего момента и до конца учебного года.

1. **ТЕОРЕТИЧЕСКИЕ АСПЕКТЫ ПО СОЗДАНИЮ ИНДИВИДУАЛЬНОГО ПРОЕКТА**
   1. **Виды веб-страниц**

Веб-страницы можно классифицировать по нескольким признакам: функциональному назначению, характеру взаимодействия с пользователем и способу обработки информации.

По функциональному назначению:

1 Статические страницы содержат фиксированную информацию, не меняющуюся при каждом запросе.

Контент задается в момент создания страницы и остается неизменным до следующего ручного обновления.

Создаются с использованием HTML и CSS, не требуют сложных серверных технологий.

Маленькая нагрузка на сервер**:** Поскольку не требуется динамическая обработка запросов, серверные ресурсы используются более эффективно.

Примеры:Страницы с информацией о компании, контактные данные, биографии, портфолио.

2 Динамические страницы изменяются в зависимости от действий пользователя или данных, поступающих в реальном времени.

Особенности**:**

Контент генерируется сервером или клиентом при каждом запросе или взаимодействии пользователя.

Требуют серверных языков программирования (например, PHP, Python, Ruby) и баз данных (например, MySQL, PostgreSQL).

Часто включают элементы, позволяющие пользователю взаимодействовать с контентом (например, формы, комментарии, обновления в реальном времени).

Высокая нагрузка на сервер**:** Требуют значительных серверных ресурсов для обработки запросов и генерации контента.

Примеры**:** Социальные сети, новостные порталы, интернет-магазины, блоги с комментариями.

По характеру взаимодействия с пользователем:

1 Информационные страницы**:**

Основная цель — предоставление информации пользователю.

Особенности**:**

Преимущественно текст, изображения, видео, которые редко меняются.

Пользователи в основном читают или просматривают контент без активного взаимодействия.

Обычно пользователи имеют простую структуру для легкого доступа к информации.

Примеры**:** Новостные статьи, блоги, онлайн-энциклопедии, страницы с документацией.

2 Интерактивные страницы:

Описание**:** Позволяют пользователю взаимодействовать с контентом и выполнять определенные действия.

Особенности**:**

Пользователи могут выполнять действия, такие как заполнение форм, голосование, оставление комментариев.

Контент может изменяться в реальном времени в ответ на действия пользователя.

Часто включают JavaScript и другие технологии для создания интерактивности.

Примеры**:** Регистрационные формы, страницы с поисковыми системами, онлайн-анкеты, панели управления пользователем.

По способу хранения и обработки информации:

1. Клиентские веб страницы **:**

Обработка данных и выполнение скриптов происходит на стороне клиента (в браузере пользователя).

Скрипты (обычно на JavaScript) выполняются в браузере пользователя, что уменьшает нагрузку на сервер.

Обеспечивают мгновенную реакцию на действия пользователя без необходимости обращения к серверу.

2. Серверные веб-страницы:

Обработка данных и выполнение логики происходит на сервере.

Все данные обрабатываются на сервере, что позволяет использовать мощные серверные ресурсы.

Серверная обработка позволяет лучше защищать данные и бизнес-логику.

* 1. **Основные способы создания веб-страниц (сайтов)**

Существует несколько основных способов создания веб-страниц, каждый из которых имеет свои особенности, преимущества и недостатки.

Ручное кодирование подразумевает создание веб-страниц с использованием языков разметки и программирования, таких как HTML, CSS, JavaScript и серверные языки, такие как PHP, Python, Ruby, Java и другие.

HTML: Основной язык разметки для создания структуры веб-страниц.

CSS: Язык стилей, используемый для оформления и позиционирования элементов на веб-странице.

JavaScript: Язык программирования, используемый для добавления интерактивности и динамического поведения на веб-страницах.

Они позволяет разработчику контролировать каждый аспект веб-страницы, от структуры до стилей и поведения.

Подходит для создания сложных и уникальных решений, которые трудно реализовать с помощью шаблонных систем.

Оптимизированный код может обеспечивать высокую производительность и быструю загрузку страниц.

2. Использование систем управления контентом (CMS)

Системы управления контентом позволяют создавать и управлять веб-сайтами без необходимости глубокого знания кода. Примеры популярных CMS включают WordPress, Joomla, и Drupal.

WordPress: Самая популярная CMS, используемая для создания блогов, новостных сайтов, интернет-магазинов и других типов сайтов.

Joomla: Подходит для создания сайтов любой сложности, включая социальные сети и порталы.

Drupal: Предлагает высокую гибкость и мощные функции для создания сложных и масштабируемых сайтов.

Дружественный интерфейс позволяет легко создавать и обновлять контент.

Большое количество готовых шаблонов и плагинов упрощает процесс разработки и добавления новых функций.

* 1. **Язык гипертекстовой разметки документа HTML и его назначение**

HTML: это язык разметки, используемый для создания структуры веб-страниц. Он является основным строительным блоком веб-документов и описывает их содержание и структуру с помощью тегов.

Основные элементы HTML:

Теги: специальные маркеры, которые указывают браузеру, как отображать определенные части содержимого. Пример: <h1>, <p>, <a>.

Атрибуты: дополнительные параметры, которые предоставляют информацию о тегах. Пример: href в теге <a> для указания адреса ссылки.

Семантические элементы: теги, которые имеют специальное значение и улучшают структуру документа. Пример: <header>, <footer>, <article>.

Назначение HTML заключается в создании базовой структуры веб-страницы, на основе которой затем добавляются стили (CSS) и интерактивность (JavaScript). HTML позволяет разработчикам описывать заголовки, параграфы, ссылки, изображения и другие элементы, необходимые для построения полноценного веб-документа.

HTML используется для создания структуры и содержания веб-страниц. Основные задачи и функции HTML включают:

1. Создание структуры документа: HTML позволяет организовывать и структурировать контент на веб-странице, используя заголовки, параграфы, списки, таблицы и другие элементы.

2. Гипертекстовые ссылки: HTML предоставляет возможность создания гиперссылок, позволяющих пользователям переходить между страницами и ресурсами в интернете.

3. Встраивание мультимедиа: HTML позволяет встраивать изображения, видео, аудио и другие мультимедийные элементы на веб-страницу.

4. Формы ввода данных: HTML предоставляет элементы для создания форм, которые позволяют пользователям вводить данные и отправлять их на сервер.

5. Семантическая разметка: HTML5 ввел семантические теги, которые улучшают структуру и читаемость документа как для людей, так и для поисковых систем.

HTML является основой для дальнейшего применения стилей (CSS) и интерактивности (JavaScript), что позволяет создавать полнофункциональные, красивые и интерактивные веб-страницы.

Теоретические аспекты проекта по созданию расписания уроков на языке HTML могут быть представлены следующим образом:Теоретические основы HTML HTML— это основной язык разметки для создания веб-страниц. Он позволяет разработчикам структурировать контент и представлять его в интернете. Таблицы в HTML используются для организации данных, что делает их идеальным инструментом для создания расписаний уроков.

Структура таблицы в HTML Таблица в HTML создается с помощью тега <table> и может содержать строки (<tr>), ячейки (<td>), заголовки (<th>), а также объединение ячеек по горизонтали (colspan) или вертикали (rowspan). Эти элементы позволяют создать четкую и понятную структуру расписания.

Вот пример с данным тегом и как он выглядит на странице:

<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Расписание 113 группы</title>

<style>

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

}

table {

width: 100%;

border-collapse: collapse;

font-size: 12px;

}

th, td {

border: 1px solid black;

padding: 5px;

text-align: left;

}

th {

background-color: #f2f2f2;

}

h1 {

font-size: 18px;

text-align: center;

margin: 10px 0;

}

</style>

</head>

<body>

<h1>Расписание 113 группы</h1>

<table>

<thead>

<tr>

<th>День недели</th>

<th>Время</th>

<th>№</th>

<th>Дисциплина</th>

<th>Преподаватель</th>

<th>Кабинет</th>

</tr>

</thead>

<tbody>

<tr><td rowspan="4">Понедельник</td><td>8:30 – 9:50</td><td>1</td><td>ОПД.08 Химия</td><td>Пименова Г.М.</td><td>15</td></tr>

<tr><td>10:00 – 11:20</td><td>2</td><td>ОДП.01 Математика</td><td>Романенко Т.В.</td><td>44</td></tr>

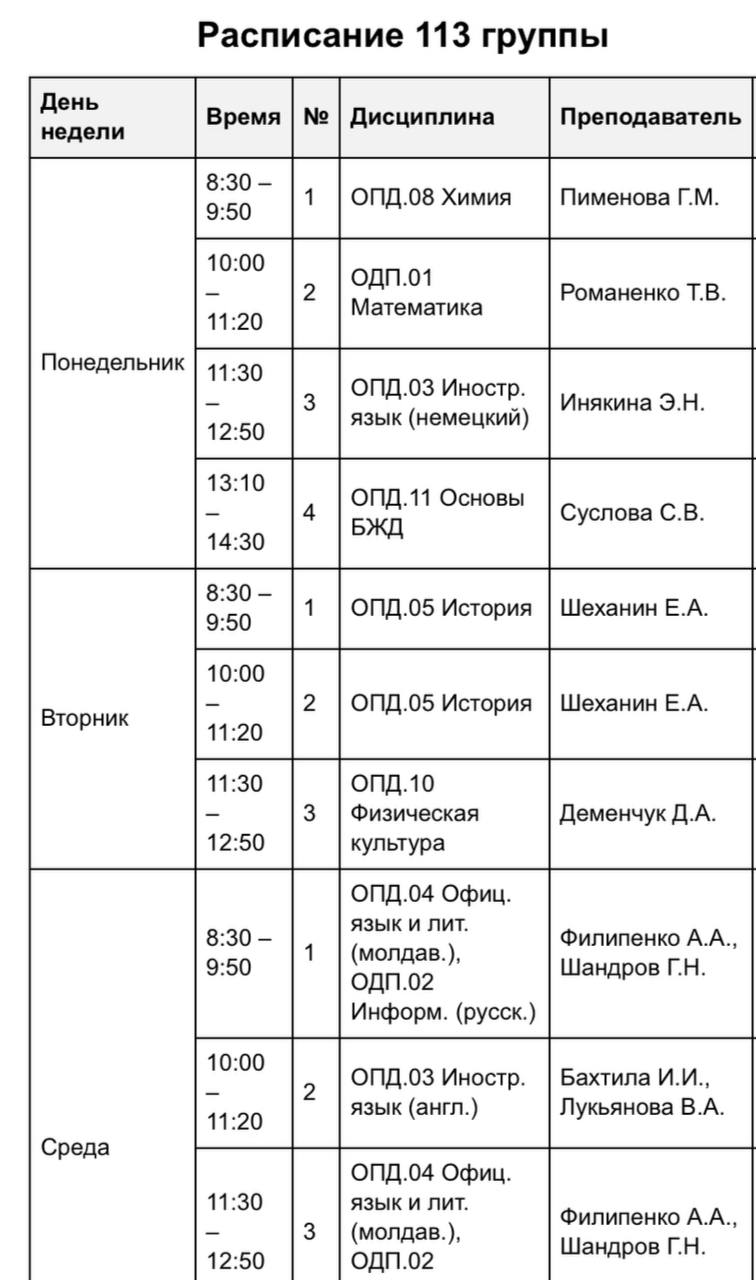
<tr><td>11:30 – 12:50</td><td>3</td><td>ОПД.03 Иностр. язык (немецкий)</td><td>Инякина Э.Н.</td><td>23/16</td></tr>

<tr><td>13:10 – 14:30</td><td>4</td><td>ОПД.11 Основы БЖД</td><td>Суслова С.В.</td><td>3</td></tr>

<tr><td rowspan="3">Вторник</td><td>8:30 – 9:50</td><td>1</td><td>ОПД.05 История</td><td>Шеханин Е.А.</td><td>41</td></tr>

<tr><td>10:00 – 11:20</td><td>2</td><td>ОПД.05 История</td><td>Шеханин Е.А.</td><td>41</td></tr>

<tr><td>11:30 – 12:50</td><td>3</td><td>ОПД.10 Физическая культура</td><td>Деменчук Д.А.</td><td>спортзал</td></tr>



1. **ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ ИНДИВИДУАЛЬНОГО ПРОЕКТА**
   1. **Постановка задачи**

Цель проекта: Создать веб-страницу, отображающую расписание уроков, которая будет удобной и функциональной для всех пользователей.

Задачи проекта:

1. Определить целевую аудиторию.

2. Составить список требований к функциональности веб-страницы.

3. Создать макет пользовательского интерфейса, учитывая удобство и интуитивную понятность.

4. Реализовать отображение расписания в удобной форме.

5. Выбрать технологии для разработки.

6. Провести тестирование всех функций веб-страницы для проверки корректности работы.

7. Исправить обнаруженные ошибки и недочеты.

* 1. **Этапы создания веб-страницы**

Создание веб-страницы для расписания уроков включает несколько ключевых этапов, каждый из которых важен для достижения конечной цели проекта. Важной частью каждого этапа является работа с кодом. Ниже представлены основные этапы реализации проекта и их связь с программированием.

Этап 1:

1.Определение целей и задач проекта:

Цель: Создание веб-страницы для отображения и управления расписанием уроков.

Задачи: Сбор требований, разработка функционала, тестирование.

2.Анализ требований целевой аудитории:

Выявление пользователей: Определение целевой аудитории.

Сбор требований: Определение требований к функциональности и интерфейсу веб-страницы.

Этап 2:

1.Определение основных элементов интерфейса:

Таблица расписания: Определение структуры и дизайна таблицы.

Этап 3: Выбор технологий и инструментов

1. Определение стека технологий:

HTML, CSS, JavaScript.

2. Функциональное тестирование:

Проверка функций: Тестирование работы всех функций.

3.Кроссплатформенное тестирование:

Проверка на разных устройствах: Тестирование веб-страницы на различных устройствах и в разных браузерах.

4.Отладка:

Исправление ошибок: Устранение выявленных проблем и багов.

Оптимизация производительности.

**2.3Тестирование и откладка кода веб-страницы**

Общая характеристика объекта исследования:

Объектом исследования является веб-приложение, отображающее расписание уроков для группы 113 специальности “Информационные системы и программирование”. Целью данного проекта является создание удобного интерфейса для студентов и преподавателей, позволяющего быстро и легко просматривать расписание занятий.

Характеристика отдельных структурных элементов объекта исследования:

 1. Основной интерфейс веб-приложения:

 Веб-приложение состоит из двух основных частей: панели выбора дня недели и области отображения расписания.

 Панель выбора дня недели расположена в левой части экрана и содержит кнопки для каждого дня недели, выполненные в виде блоков с рамками и стилизованные с помощью CSS.

 Область отображения расписания расположена в правой части экрана и выводит информацию о выбранном дне в виде структурированного списка.

 2. Панель выбора дня недели:

 Включает кнопки для каждого дня недели (Понедельник, Вторник, Среда, Четверг, Пятница, Суббота, Воскресенье).

 При нажатии на кнопку соответствующего дня, информация о расписании для этого дня загружается и отображается в области расписания.

 3. Область отображения расписания:

 Представляет собой блок с рамкой, внутри которого выводится список занятий для выбранного дня.

 Каждый элемент списка содержит время занятия, предмет, имя преподавателя и номер кабинета.

Порядок деятельности и функционирования

 1. Инициализация веб-страницы:

При загрузке веб-страницы отображается заголовок “Расписание уроков 113 группы”.

 В панели выбора дня недели отображаются кнопки для каждого дня.

 2. Выбор дня недели:

 Пользователь нажимает на кнопку соответствующего дня недели.

Вызванная функция JavaScript загружает соответствующее расписание из предопределенного объекта.

 3. Отображение расписания:

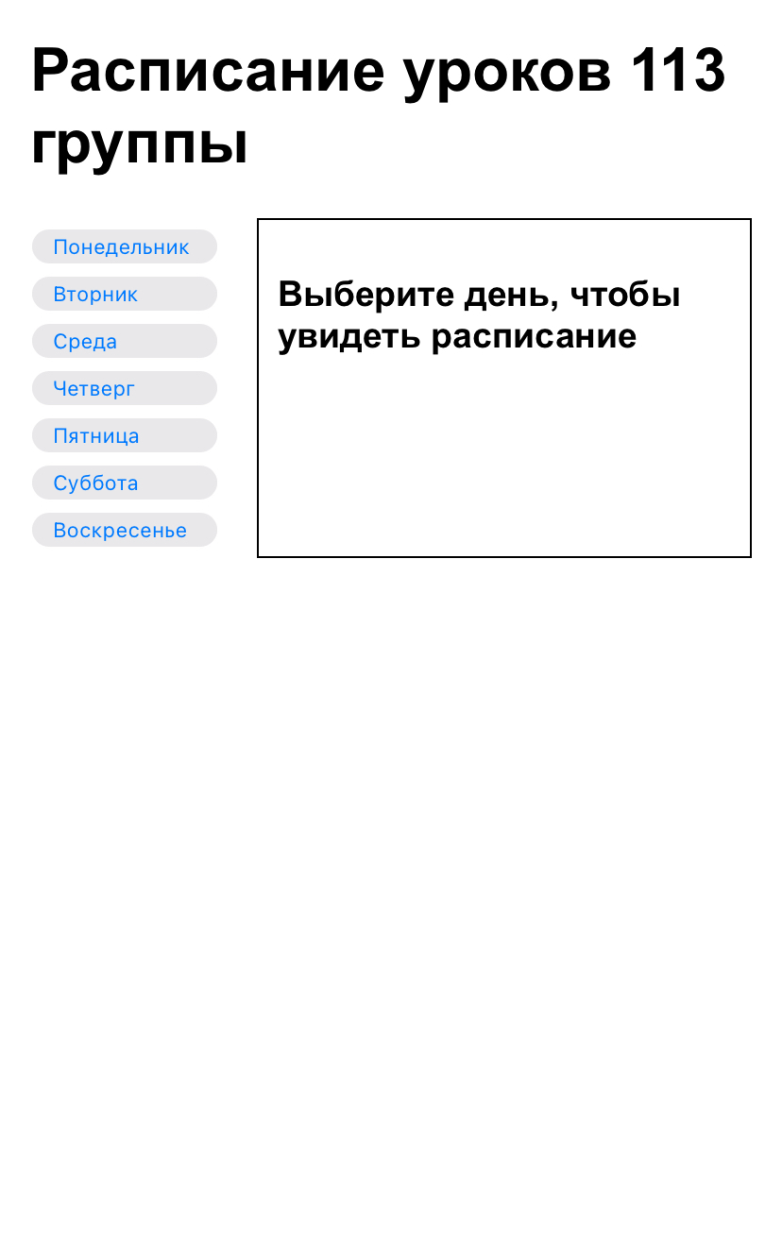
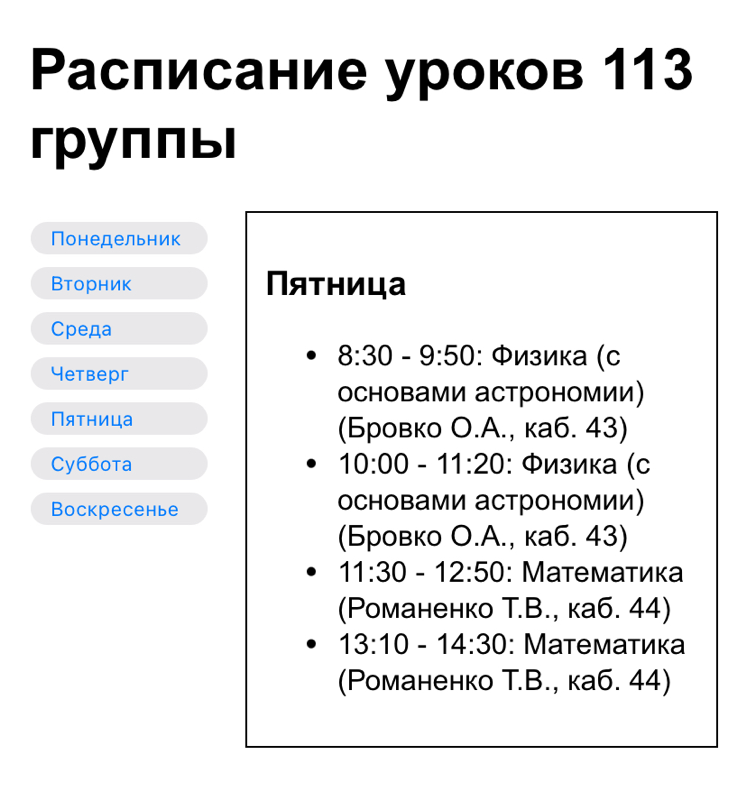
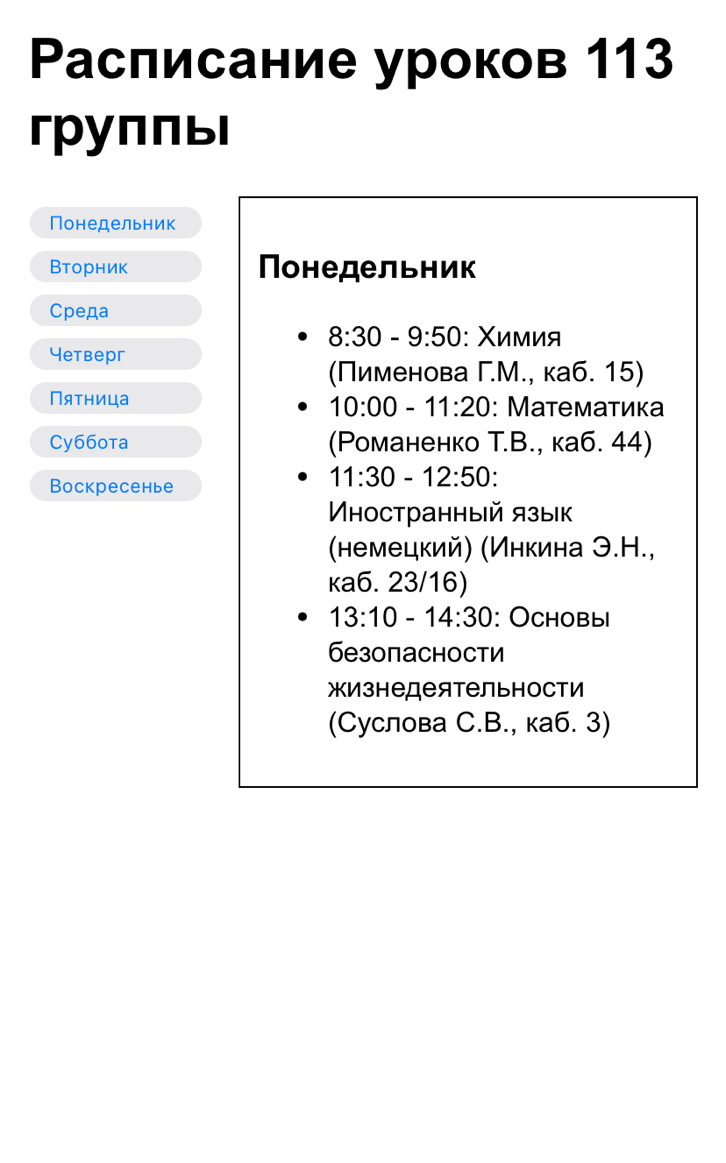
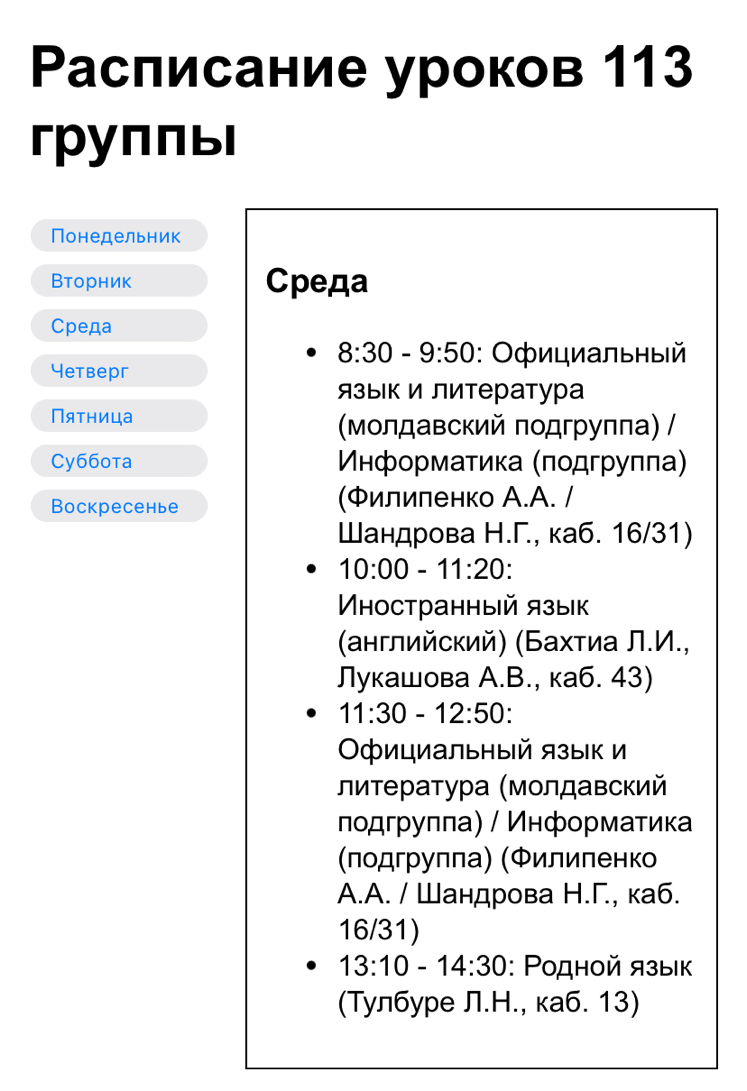
 Область отображения расписания обновляется содержимым, соответствующим выбранному дню.

Если выбран Воскресенье, выводится сообщение “Выходной”.

Выводы и предложения

Анализ проведенного исследования показал, что созданное веб-приложение успешно выполняет свою основную задачу,предоставляет пользователям удобный и интуитивно понятный способ просмотра расписания уроков.

Общий вид сайта:



Общий вид кода:

<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Расписание уроков 113 группы</title>

<style>

body {

font-family: Arial, sans-serif;

}

.container {

display: flex;

}

.days {

margin-right: 20px;

}

.days button {

display: block;

margin: 5px 0;

padding: 10px;

width: 100px;

text-align: left;

cursor: pointer;

}

.schedule {

border: 1px solid #000;

padding: 10px;

width: 400px;

}

</style>

</head>

<body>

<h1>Расписание уроков 113 группы</h1>

<div class="container">

<div class="days">

<button onclick="showSchedule('понедельник')">Понедельник</button>

<button onclick="showSchedule('вторник')">Вторник</button>

<button onclick="showSchedule('среда')">Среда</button>

<button onclick="showSchedule('четверг')">Четверг</button>

<button onclick="showSchedule('пятница')">Пятница</button>

<button onclick="showSchedule('суббота')">Суббота</button>

<button onclick="showSchedule('воскресенье')">Воскресенье</button>

</div>

<div class="schedule" id="schedule">

<h3>Выберите день, чтобы увидеть расписание</h3>

</div>

</div>

<script>

const schedules = {

понедельник: `

<h3>Понедельник</h3>

<ul>

<li>8:30 - 9:50: Химия (Пименова Г.М., каб. 15)</li>

<li>10:00 - 11:20: Математика (Романенко Т.В., каб. 44)</li>

<li>11:30 - 12:50: Иностранный язык (немецкий) (Инкина Э.Н., каб. 23/16)</li>

<li>13:10 - 14:30: Основы безопасности жизнедеятельности (Суслова С.В., каб. 3)</li>

</ul>

`,

вторник: `

<h3>Вторник</h3>

<ul>

<li>8:30 - 9:50: История (Шеханин Е.А., каб. 41)</li>

<li>10:00 - 11:20: История (Шеханин Е.А., каб. 41)</li>

<li>11:30 - 12:50: Физическая культура (Деменчук Д.А., спортзал)</li>

</ul>

`,

среда: `

<h3>Среда</h3>

<ul>

<li>8:30 - 9:50: Официальный язык и литература (молдавский подгруппа) / Информатика (подгруппа) (Филипенко А.А. / Шандрова Н.Г., каб. 16/31)</li>

<li>10:00 - 11:20: Иностранный язык (английский) (Бахтиа Л.И., Лукашова А.В., каб. 43)</li>

<li>11:30 - 12:50: Официальный язык и литература (молдавский подгруппа) / Информатика (подгруппа) (Филипенко А.А. / Шандрова Н.Г., каб. 16/31)</li>

<li>13:10 - 14:30: Родной язык (Тулбуре Л.Н., каб. 13)</li>

</ul>

`,

четверг: `

<h3>Четверг</h3>

<ul>

<li>8:30 - 9:50: Начальная военная подготовка (Дроздов В.А., каб. 21)</li>

<li>10:00 - 11:20: Родная литература (Горюк А.Д., каб. 33)</li>

<li>11:30 - 12:50: Биология (Пименова Г.М., каб. 15)</li>

<li>13:10 - 14:30: Родная литература (Горюк А.Д., каб. 33)</li>

</ul>

`,

пятница: `

<h3>Пятница</h3>

<ul>

<li>8:30 - 9:50: Физика (с основами астрономии) (Бровко О.А., каб. 43)</li>

<li>10:00 - 11:20: Физика (с основами астрономии) (Бровко О.А., каб. 43)</li>

<li>11:30 - 12:50: Математика (Романенко Т.В., каб. 44)</li>

<li>13:10 - 14:30: Математика (Романенко Т.В., каб. 44)</li>

</ul>

`,

суббота: `

<h3>Суббота</h3>

<ul>

<li>8:30 - 9:50: Обществознание (включая экономику и право) (Золотухина И.В., каб. 16)</li>

<li>10:00 - 11:20: Официальный язык и литература (украинский подгруппа) (Мицула А.А., каб. 43)</li>

</ul>

`,

воскресенье: `

<h3>Воскресенье</h3>

<p>Выходной</p>

`

};

function showSchedule(day) {

document.getElementById('schedule').innerHTML = schedules[day];

}

</script>

</body>

</html>

ЗАКЛЮЧЕНИЕ

Существуют различные способы создания веб страниц,одним из которых является применение языка разментки HTML и CSS.

В ходе выполнения проектной работы разработана веб страница для расписания учебных занятий с использованием HTML и CSS.

В коде использованны различные техналогии для постраения веб страницы:теги ссылок,форматирование,маркированные и нумерованные списки.

В результате выполнения проектной работы получилось сделать нагядную веб страницу,простую в управлении пользователем;раскрыть тему «Расписание уроков»,показав что расписание можно всегда легко найти.

Разработаная веб страница успешно протестирована на примере расписания 113 группы.

Использование мобильных устройств в наше время,показывает что электронные сайты могут значительно упрощать работу или учебу,правильно распределять время и всегда знать где находится пара или когда её не будет.Расписание  занятий в веб формате это удобно,поэтому и простая в оформлении страница идеально подходит для этого.

СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ

1. Семакин И.Г. Информатика. Углубленный уровень: учебник для 10 класса: в 2 ч. Ч. 1/ И.Г.Семакин, Т.Ю.Шеина, Л.В.Шестакова. – М.:БИНОМ. Лаборатория знаний, 2014. – 184 с.

2. Семакин И.Г. Информатика. Углубленный уровень: учебник для 10 класса: в 2 ч. Ч. 2/ И.Г.Семакин, Т.Ю.Шеина, Л.В.Шестакова. – М.:БИНОМ. Лаборатория знаний, 2014. – 232 с.

3. Семакин И.Г. Информатика. Углубленный уровень: учебник для 11 класса: в 2 ч. Ч. 1/ И.Г.Семакин, Е.К.Хеннер, Л.В.Шестакова. – М.:БИНОМ. Лаборатория знаний, 2014. – 176 с.

4. Семакин И.Г. Информатика. Углубленный уровень: учебник для 11 класса: в 2 ч. Ч. 2/ И.Г.Семакин, Е.К.Хеннер, Л.В.Шестакова. – М.:БИНОМ. Лаборатория знаний, 2014. – 216 с.

5. https://metanit.com/web/html5/ - Сайт о прагроммирование