СОДЕРЖАНИЕ

Введение

Учебная практика направлена на закрепление и углубление теоретической подготовки обучающегося, приобретение им практических навыков и компетенций, а также опыта самостоятельной работы в выбранной профессиональной деятельности. Целью освоение учебной практики является получение первичных профессиональных умений и навыков. Прохождение учебной практики предполагает выполнение следующих задач:

− закрепление теоретических знаний и умений, приобретенных в предшествующий период теоретического обучения;

− овладение профессиональными навыками решения практических задач;

− развитие первичных профессиональных умений по направлению и профилю подготовки;

На современном этапе развития информационных технологий все больше людей и компаний предпочитают использовать web-сервисы для размещения объявлений. Это связано с необходимостью эффективного продвижения товаров и услуг, а также поиском нужной информации для конечных пользователей. Однако существующие платформы не всегда удовлетворяют все потребности пользователей, особенно в части удобства использования и функциональности. В связи с этим становится актуальной задача создания современного web-сервиса для размещения объявлений.

Основной научно-технической проблемой, которая требует решения, является создание web-сервиса, способного обеспечить удобство использования и эффективное размещение объявлений для различных категорий пользователей. Это включает в себя разработку удобного интерфейса, а так же функциональности web-сервиса.

Целью данной учебной практики является разработка и реализация web-сервиса для размещения объявлений.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Изучение и выбор технологий создания web-сервисов, а также анализ функциональных требований к web-сервису;

2. Проектирование удобного интерфейса и функциональности web-сервиса;

3. Разработка механизмов размещения и просмотра информации;

1 Теоретические основания разработки автоматизированного программного обеспечения

1.1 Выбор технологии реализации автоматизированного программного обеспечения

В качестве технологии для создания автоматизированного ПО был выбран комплекс технологий веб-разработки, включающий в себя HTML, CSS и JavaScript.

HTML (Hyper Text Markup Language) - это основной язык разметки веб-страниц. С помощью него определяется структура содержимого веб-страницы, такая как заголовки, абзацы, списки, ссылки, изображения и другие элементы.

CSS (Cascading Style Sheets) - это язык таблиц стилей, используемый для оформления внешнего вида веб-страниц. Он определяет стилизацию элементов HTML.

JavaScript - это мощный язык программирования, применяемый для создания интерактивных веб-страниц и придания им динамической функциональности. JavaScript позволяет добавлять изменяемое поведение на веб-страницах, обрабатывать события, создавать анимации и многое другое.

История языков

HTML был создан Тимом Бернерсом-Ли в 1991 году в ЦЕРНе в Швейцарии. Изначально он представлял из себя простой язык разметки для описания структуры документов, но с течением времени стал основой для создания веб-страниц и веб-приложений.

CSS был разработан Хаконом Ли и Бертом Босом в конце 1996 года. CSS был создан для оформления внешнего вида веб-страниц, отделив структуру документа (HTML) от его стилизации, что позволило значительно упростить процесс создания и поддержания веб-сайтов.

JavaScript был разработан Бренданом Эйхом в 1995 году за короткий период времени в Netscape Communications Corporation. Он был создан для добавления динамической функциональности к веб-страницам, что позволило

внедрять интерактивные элементы на веб-сайтах и создавать более сложные пользовательские интерфейсы.

Особенности и преимущества языков:

1. Кроссплатформенность: HTML, CSS, JavaScript может выполняться на различных платформах и устройствах, включая браузеры, серверы и мобильные устройства;

2. Легковесность: Эти технологии обладают низкими системными требованиями, что делает их идеальным для разработки веб-приложений с небольшими нагрузками;

3. Простота: Эти технологии обладают низкими системными требованиями, что делает их идеальным для разработки веб-приложений с небольшими нагрузками;

4. Асинхронность: JavaScript позволяет выполнять асинхронные операции, такие как запросы к серверу, без блокировки пользовательского интерфейса, что улучшает производительность веб-приложений;

5. Обширные библиотеки и фреймворки: Существует множество библиотек и фреймворков, таких как React, Angular и Vue.js, которые упрощают разработку веб-приложений на JavaScript;

6. Широкое применение: JavaScript используется не только для веб-разработки, но также для создания мобильных приложений, десктопных приложений и даже для разработки серверных приложений (через Node.js);

Основы синтаксиса

Синтаксис языка HTML основывается на тэгах, которые определяют структура документа. Тэги чаще всего используются в парах, которые состоят из открывающего и закрывающего тэга(<h1></h1>), но некоторые тэги могут быть и одиночными(
).

CSS использует селекторы для выбора элементов HTML, к которым будут применять стили. Каждое правило CSS состоит из селектора и объявления стилей в фигурных скобках. Например, p { background-color: red;}

JavaScript использует синтаксис, близкий к синтаксису языка C, что включает в себя операторы, условия, циклы и функции.

Структуры данных

JavaScript предлагает широкий набор встроенных структур данных, которые обеспечивают гибкость и удобство при разработке программ. Основные структуры данных включают списки (Array), объекты (Object), множества (Set) и словари (Map). Списки позволяют хранить изменяемые последовательности элементов, объекты обеспечивают хранение последовательностей ключ-значение, множества используются для хранения уникальных элементов, а словари предлагают ассоциативные массивы для хранения пар ключ-значение(ключ может быть любым типом данных, а не только строкой, как в Object). Эти структуры данных являются мощными инструментами, которые упрощают управление данными и позволяют эффективно решать разнообразные задачи программирования.

Функции

Функции в JavaScript являются основным механизмом для организации и повторного использования кода. Они позволяют группировать логически связанные инструкции в блоки, которые можно вызывать по имени, передавая аргументы и получая результаты. Функции поддерживают параметры по умолчанию, передачу произвольного количества аргументов, что делает их гибкими и удобными для различных сценариев использования. В Python также поддерживаются стрелочные функции для создания более компактных функций. Благодаря функциям, код становится более структурированным, читабельным и легко поддерживаемым.

1.2 Сравнительный обзор инструментов языков программирования HTML, CSS и JavaScript

При создании web-сервиса для размещения электронных объявлений используются только встроенные функции языков. Такие как, разметка web-страницы с помощью HTML, добавление стилей, благодаря CSS, функции и возможность создания интерактивных web-страниц и динамической функциональности web-сайта с помощью JavaScript.

Обычно сайт делится на три части: основной макет(HTML), определение стилей(CSS) и автоматизированные скрипты(JavaScript).

HTML (Hyper Text Markup Language):

1. Цель: Структурирование и организация контента веб-страницы;
2. Инструменты: Теги (например, `<p>`, `<h1>`, `<img>`, `<a>`) для определения элементов и их свойств;
3. Преимущества: Простой в изучении. обеспечивает основу для всех веб-страниц, поддерживает широкий спектр медиа-контента (текст, изображения, видео, аудио);
4. Недостатки: Не предназначен для стилизации или динамического поведения, достаточно ограничен в функциональности;

Таблица 1 – Основные тэги языка HTML:

* *	Является блочным элементом и предназначен для выделения фрагмента документа
<button><button>	Создаёт кликабельную кнопку, которая может быть использована в формах или в любом другом месте документа
<form></form>	Устанавливает форму на веб-странице, предназначена для обмена данными.
<script></script>	Предназначен для описания скриптов, может содержать ссылку на программу или её текст на определённом языке
<b>	Текст внутри <b> отображается жирным начертанием

Продолжение таблицы 1

* * * *	Представляют из себя заголовки разделов
* *	Устанавливает перевод строки
<input>	Определяет пользовательское поле для ввода информации
<textarea><textarea>	Представляет собой элемент формы для создания области, в которую можно вводить несколько строк текста

CSS (Cascading Style Sheets):

1. Цель: Определение внешнего вида и стиля веб-страницы (отступы, цвета, шрифты, макеты);
2. Инструменты: Селекторы для выбора элементов, свойства для определения стилей;
3. Преимущества: Отделяет стиль от структуры (HTML), обеспечивает единый стиль для всех элементов, позволяет создавать адаптивные макеты для различных устройств;
4. Недостатки: Может быть сложным для изучения, особенно с использованием более продвинутых свойств, не предназначен для взаимодействия с пользователем;

Таблица 2 – Основные свойства CSS:

display: ;	**Свойство display** определяет, как элемент будет взаимодействовать с другими элементами на странице
grid-template-areas: ;	Определяет именованные области сетки
grid-area: ;	Задает элементу имя, на которое можно сослаться при определении шаблона сетки, созданного с помощью свойства grid-template-areas
grid-template-columns: ;	Задает количество и ширину столбцов, которые будет занимать элемент в сетке
grid-template-rows: ;	Задает количество и высоту столбцов, которые будет занимать элемент в сетке
margin: ;	Задаёт размер внешнего отступа вокруг элемента

Продолжение таблицы 2

padding; ;	Задаёт размер внутреннего отступа вокруг элемента
justify-content: ;	Определяет, как браузер распределяет пространство вокруг флекс-элементов вдоль главной оси контейнера
align-items: ;	Определяет, как браузер распределяет пространство вокруг флекс-элементов вдоль поперечной оси контейнера
height: ; *width: ;*	Устанавливают высоту и ширину элементов
border-radius: ;	Свойство, добавляющее элементам скругление углов
background-color: ;	Устанавливает цвет фона элемента

JavaScript:

1. Цель: Добавление интерактивности и динамического поведения на веб-страницы;
2. Инструменты: Переменные, операторы, циклы, функции, объекты, события, встроенные объекты, библиотеки и фреймворки;
3. Преимущества: Широкие возможности для создания интерактивных интерфейсов, позволяет создавать веб-приложения с богатой функциональностью, активно развивается и имеет огромную экосистему;
4. Недостатки: Может быть сложным для начинающих, требует глубокого понимания принципов программирования;

Таблица 3 – Основные возможности JavaScript:

let	Позволяет объявить локальную переменную с областью видимости, ограниченной текущим блоком кода
for () {}	Оператор в JavaScript для создания цикла, который состоит из трёх необязательных условий
document	Модель документа, которая представляет всё содержимое страницы
getElementById(value)	Выбирает элемент, у которого атрибут id равен value

Продолжение таблицы 3

addEventListener()	Используется для прослушивания событий на элементах DOM-дерева
getElementsByClassName(value)	Выбирает все элементы, которые имеют класс value, возвращает список
event.preventDefault()	Позволяет отменить действия браузера по умолчанию
querySelector(value)	Выбирает первый элемент, который соответствует css-селектору value
lst.push()	Добавляет один или более элементов в конец массива и возвращает новую длину массива
innerText/innerHTML	Позволяет получить содержимое элемента в виде строки
lst.length	Возвращает длину или количество элементов некоторого списка

2 Практическая реализация автоматизированного программного обеспечения

2.1 Проектирование алгоритма реализации web-сервиса программными средствами языков программирования HTML, CSS, JavaScript

Проектирование алгоритма реализации web-сервиса для размещения электронных объявлений программными средствами языков программирования HTML, CSS, JavaScript требует систематического подхода к разработке. Важно начать с анализа требований к сервису и определения основных функций, которые он должен выполнять. Затем следует определить структуры данных, которые будут использоваться для хранения информации об одном объявлении и всех объявлений.

После этого необходимо приступить к разработке алгоритмов для основных функций сервиса, таких как отображение списка объявлений, возможность добавления нового объявления и просмотр уже созданных объявлений.

Важно предусмотреть возможность расширения функциональности сервиса путем добавления новых функций. Для этого следует создать гибкую архитектуру, которая позволит легко добавлять новый функционал без необходимости переписывать большую часть кода.

В процессе разработки следует учитывать принципы модульности и чистоты кода, чтобы обеспечить легкость поддержки и дальнейшего развития проекта. Разделение функционала на небольшие модули и использование понятных имен переменных и функций помогут сделать код более понятным и удобным для работы.

Подзадачи, которые необходимо решить для реализации основной задачи:

1. Создать основной макет web-сервиса с помощью grid-сетки;

2. Создать форму для добавления объявлений, включающая в себя заголовок объявления, описание, имя и телефон для связи;

3. Создать функцию открытия формы для добавления объявлений;

4. Создать функцию закрытия формы для добавления объявлений;

5. Определить структуру данных для хранения всех объявлений;

6. Реализовать функцию добавления данных из формы в структуру данных, используя обработчик события;

7. Реализовать функцию отображения всех объявлений;

8. Создать функцию открытия объявления;

9. Создать функцию закрытия объявления;

Перед завершением разработки необходимо провести тестирование web-сервиса на наличие ошибок и неполадок, а также получить обратную связь от пользователей для улучшения функционала и исправления выявленных проблем.

2.2 Реализация web-сервиса для размещения электронных объявлений. Обоснование его применения в профессиональной деятельности

Реализация прикладного программного обеспечения web-сервис для размещения электронных объявлений начинается с создания основного макета проекта с помощью HTML и CSS.

Далее создается форма для добавления новых объявлений на сайт. Она должна включать в себя ввод данных пользователя, заголовок объявления и его описание, а так же кнопку для отправки формы.

Для более удобного взаимодействия с формой создания объявлений, необходимо создать две управляющие кнопки – кнопка открытия формы и кнопка закрытия.

Для этого создаем две функции с помощью JavaScript – openAddForm() и closeAddForm(), в которых будут скрываться и показываться отдельные части основного макета, а также изменяться заголовок сайта.

Для хранения всей информации об объявлениях создадим переменную списка.

Чтобы добавлять новые данные в список необходимо добавить обработчик события отправки формы и функцию, обрабатывающую добавление – getFormValue(event). В этой функции мы будем получать введенные пользователем данные из формы, создавать структуру для хранения этих данных и добавлять эту структуру в список всех объявлений.

Затем необходимо создать функцию отображения всех объявлений – update(). В ней все элементы списка объявлений будут оборачиваться в специальный макет и загружаться во внутренний HTML основного макета с помошью свойства innerHTML и for.

Для реализации открытия каждого поста необходимо, чтобы каждый элемент списка являлся нажимаемой кнопкой.

Для того, чтобы реализовать открытие и закрытие объявление создадим две функции – openPost(number) и closePost(). В первой функции необходим

параметр, чтобы при открытии получать только необходимые данные из списка всех объявлений, помимо этого в ней будут показываться и форматироваться данные открываемого объявления.

Обоснование его применения в профессиональной деятельности.

Применение прикладного программного обеспечения web-сервиса для размещения электронных объявлений может быть обосновано в различных сферах профессиональной деятельности:

1. Доступность и охват: Web-сервисы предоставляют широкую аудиторию потенциальных клиентов, что позволяет максимально эффективно распространять информацию о ваших услугах или товарах;
2. Эффективность: создание и размещение объявлений в онлайн-режиме значительно экономит время и силы по сравнению с традиционными методами, такими как печать листовок или размещение объявлений в газетах;
3. Стоимость: размещение объявлений в онлайн-режиме, как правило, дешевле, чем традиционные методы рекламы;
4. Гибкость: вы можете легко редактировать или удалять объявления в любое время;
5. Доступность: web-сервисы доступны 24/7, что позволяет вам взаимодействовать с потенциальными клиентами в любое время;
6. Создание бренда: хорошо оформленные объявления могут повысить узнаваемость вашего бренда;

Таким образом, web-сервис для размещения электронных объявлений является эффективным инструментом для продвижения профессиональной деятельности. Он обеспечивает доступность, эффективность, целевую аудиторию, аналитику и гибкость, что делает его ценным ресурсом для любого специалиста, стремящегося расширить свой клиентский круг и повысить узнаваемость бренда.

Заключение

В ходе учебной практики был изучен комплекс технологий веб-разработки, включающий в себя такие языки программирования, как HTML, CSS и JavaScript, разработан алгоритм и программное обеспечение для реализации web-сервиса размещения электронных объявлений с использованием современных информационных технологий и программных средств языков программирования HTML, CSS и JavaScript. Задание выполнено в полном объеме. При изложении изученного материала были использованы основные положения правил по содержанию и оформлению технической документации. Для защиты отчета была подготовлен доклад, обосновывающий применение web-сервиса размещения электронных объявлений в профессиональной деятельности.

Применение web-сервиса размещения электронных объявлений позволяет сократить временные затраты на выполнение задач, снизить вероятность человеческих ошибок и улучшить общую эффективность работы. Благодаря возможности расширения функциональности web-сервиса размещения электронных объявлений, сайт становится гибким инструментом, способным адаптироваться к различным потребностям и требованиям.

Список использованных источников

1. *Руководство по JavaScript - URL:* https://metanit.com/web/javascript/

2. *Руководство по HTML5 и CSS3 - URL:* https://metanit.com/web/html5/

3. *Современный учебник по JavaScript - URL:* https://learn.javascript.ru/js

Приложение

Приложение 1. Реализация web-сервиса для размещения электронных объявлений.

```
<!DOCTYPE HTML>
<html lang="ru">
<head>
   <meta charset = "UTF-8">
   <title>CSS Grid</title>
   <style>
      .header {
         grid-area: header;
         display: flex;
         justify-content: center;
         align-items: center;
         font-size: 3vh;
      }
      .cont {
         grid-area: cont;
         height: 95vh;
         overflow-y: auto;
         /* margin-top: -1vh; */
      }
      .openAddForm, .closeAddForm, .closePost {
         display: flex;
         border-radius: 100%;
         margin-left: 35%;
         height: 3vh;
         width: 3vh;
         justify-content: center;
         align-items: center;
      }
      .closeAddForm {
         display: none;
      }
      .closePost {
         display: none;
      }
      .addForm {
         display: none;
```

```
    grid-area: cont;
}
.main {
    grid-area: main;
    background-color: rgb(245,245,245);
    border-radius: 1vh*1vw;
    display: grid;
    grid-template-columns: 100%;
    grid-template-rows: 3% 97%;
    grid-template-areas:
                    "header"
                    "cont";
}
input, .description {
    width: 30vw;
    margin: 5px;
}
.description {
    height: 10vh;
}
.container {
    display: grid;
    grid-template-columns: 100%;
    grid-template-rows: 96vh 1vh;
    column-gap: 15px;
    row-gap: 15px;
    grid-template-areas:
                    "main"
                    "footer";
    justify-content: center;
}
.post {
    border: none;
    background-color: rgb(255, 255, 255);
    margin: 5px;
    padding: 5px;
    border-radius: 1vw;
    width: 98%;
    height: fit-content;
    align-items: center;
```

```
        }
        .footer {
            grid-area: footer;
            display: flex;
            align-items: flex-end;
            justify-content: flex-end;
        }
        .openedPost {
            display: none;
            padding: 10px;
        }
    </style>
</head>
    <body>
        <div class="container">
            <div class="main">
                <div class="header">
                    <b id="header" style="margin-left: 40vw;">Объявления</b>
                    <button class="openAddForm" onclick="openAddForm()">+</button>
                    <button class="closeAddForm" onclick="closeAddForm()">x</button>
                    <button class="closePost" onclick="closePost()">x</button>
                </div>
                <div class="openedPost"></div>
                <div class="addForm">
                        <form id="addForm" class="addFormCont">
                            <input type="text" name="postHeader" placeholder="Введите заголовок
объявления"/><br>
                            <textarea class="description" name="postDesc" placeholder="Введите
описание"></textarea><br>
                            <input type="text" name="postName" placeholder="Введите своё имя"/><br>
                            <input type="tel" name="postPhone" placeholder="Введите ваш
номер"/><br>
                            <input type="submit">
                        </form>
                </div>
                <div class="cont"> Объявления отсутствуют </div>
            </div>
            <div class="footer"> Разработал Соловьёв Л.А. </div>
        </div>
        <script>
```

```
let lst = [];
for (let i = 0; i < 1; i++) {
    const data = {
        header: 'Ищу рабочего на дачный участок',
        name: 'Леонид',
        desc: 'Ищу рабочего для работ по уходу на дачном участке. Требуется поливать, окучивать, выдергивать сорняки и пр.',
        phone: '+7 777 777 777 777'
    };
    lst.push(data);
}
update();
let addForm = document.getElementById('addForm');
addForm.addEventListener('submit', getFormValue);

function update() {
    document.getElementsByClassName('cont')[0].innerHTML = '';
    for (let i = 0; i < lst.length; i++) {
        let pre = '<br><button class="post" onclick="openPost(' + String(i) + ')">';
        let post = '</button>'
        elem = lst[i];
        let headerPost = 'Заголовок: ' + elem.header + '<br>';
        let namePost = 'Автор: ' + elem.name + '<br>';
        let phonePost = 'Телефон: ' + elem.phone + '<br>';

        document.getElementsByClassName('cont')[0].innerHTML += (pre + headerPost + namePost + phonePost + post);
    }
}

function openPost(number) {
    elem = lst[number];
    let content = '<div><h1>' + elem.header + '</h1>' + elem.desc + '<h3> Телефон для звонка: ' + elem.phone + '<br>' + 'Моё имя: ' + elem.name + '</h3>'+ '</div>';
    document.getElementsByClassName("openedPost")[0].innerHTML = content;
    document.getElementsByClassName("openedPost")[0].style.display = "block";
    document.getElementsByClassName("openAddForm")[0].style.display = "none";
    document.getElementsByClassName("closePost")[0].style.display = "flex";
    document.getElementById("header").innerText = 'Объявление';
    document.getElementsByClassName("cont")[0].style.display = "none";
```

```
}
function closePost() {
    document.getElementsByClassName("openedPost")[0].style.display = "none";
    document.getElementsByClassName("openAddForm")[0].style.display = "flex";
    document.getElementsByClassName("closePost")[0].style.display = "none";
    document.getElementById("header").innerText = 'Объявления';
    document.getElementsByClassName("cont")[0].style.display = "block";
}

function getFormValue(event) {
    event.preventDefault();
    closeAddForm();
    let name = addForm.querySelector('[name="postName"]');
    let header = addForm.querySelector('[name="postHeader"]');
    let desc = addForm.querySelector('[name="postDesc"]');
    let phone = addForm.querySelector('[name="postPhone"]');
    const data = {
        header: header.value,
        desc: desc.value,
        name: name.value,
        phone: phone.value
    };
    header.value = '';
    desc.value = '';
    name.value = '';
    phone.value = '';
    lst.push(data);
    update();
}

function openAddForm() {
    document.getElementsByClassName("addForm")[0].style.display = "grid";
    document.getElementsByClassName("openAddForm")[0].style.display = "none";
    document.getElementsByClassName("closeAddForm")[0].style.display = "flex";
    document.getElementById("header").innerHTML = ' Добавить  ';
    document.getElementsByClassName("cont")[0].style.display = "none";
}
function closeAddForm() {
    document.getElementsByClassName("addForm")[0].style.display = "none";
    document.getElementsByClassName("openAddForm")[0].style.display = "flex";
```

```
            document.getElementsByClassName("closeAddForm")[0].style.display = "none";
            document.getElementById("header").innerText = 'Объявления';
            document.getElementsByClassName("cont")[0].style.display = "block";
        }
    </script>
</body>
</html>
```