Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования «Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых» (ВЛГУ)

Кафедра ФиПМ

ЛАБОРАТОРНАЯ РАБОТА № 4

по дисциплине: «Технология разработки веб-приложений»

на тему: «Независимые и пользовательские объекты в JavaScript/ECMAScript.»

Выполнил:

студент группы ИТ-122

Овчинникова В.А.

Проверил:

Болачков А. В.

Владимир 2024

**Цель работы.**

Практическое освоение возможностей, предоставляемых стандартизованными объектами языка ECMAScript, а также возможностей по созданию собственных типов объектов.

**Задание.**

Реализовать «Записную книжку - органайзер». На странице отображается в табличном виде некоторый месяц некоторого года. Между месяцами можно переходить (например, нажимая стрелки рядом с названием месяца). Текущая дата выделена красным фоном. При щелчке на некоторой дате открывается список дел на эту дату (если ничего не запланировано, то выводить фразу «Планов нет»). Для выбранной даты можно добавить какое-то дело. Даты, для которых список дел не пустой, обведены синей рамкой.

**Код HTML-файла**

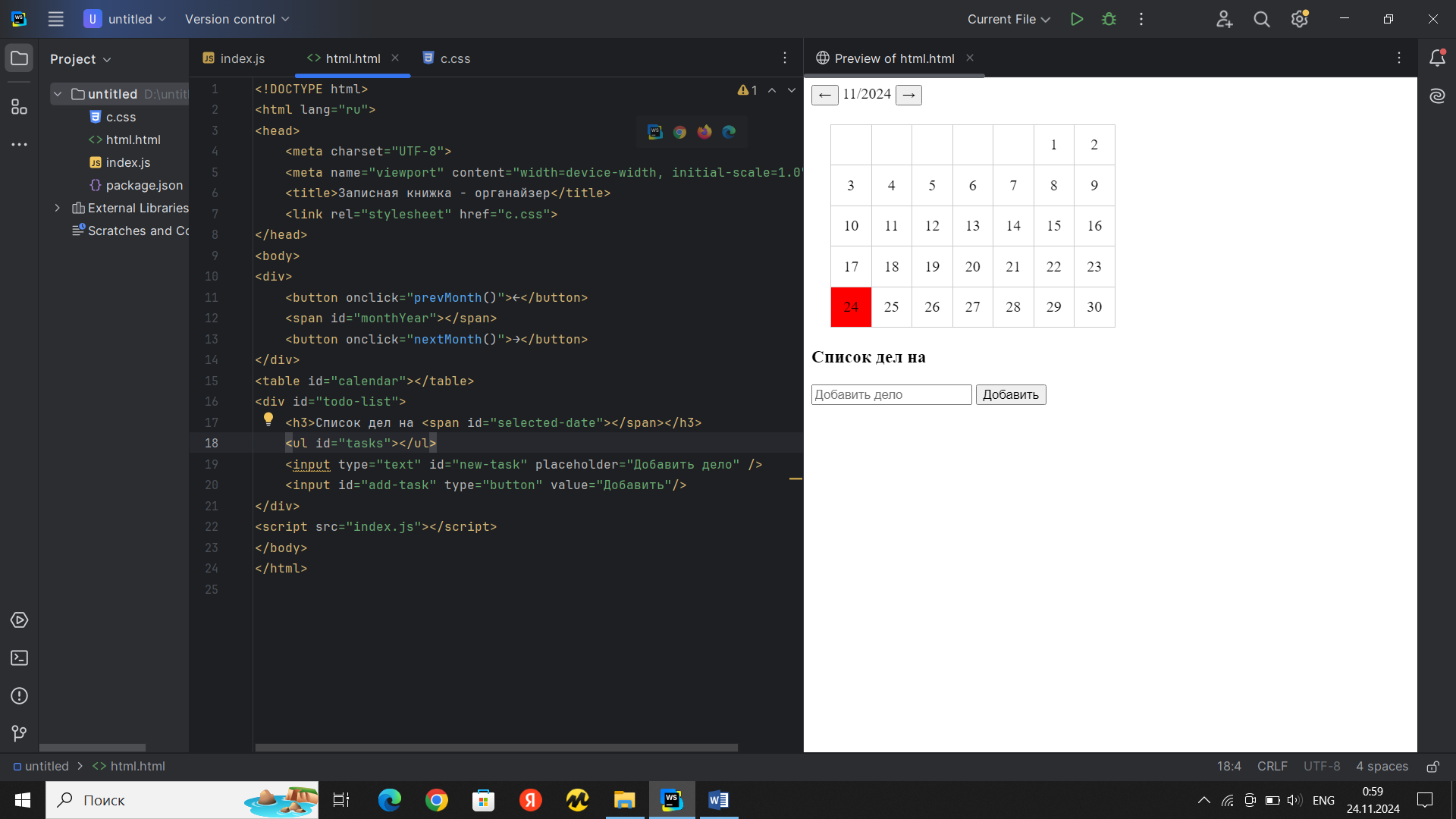
<!DOCTYPE html>  
<html lang="ru">  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
 <title>Записная книжка - органайзер</title>  
 <link rel="stylesheet" href="c.css">  
</head>  
<body>  
<div>  
 <button onclick="prevMonth()">←</button>  
 <span id="monthYear"></span>  
 <button onclick="nextMonth()">→</button>  
</div>  
<table id="calendar"></table>  
<div id="todo-list">  
 <h3>Список дел на <span id="selected-date"></span></h3>  
 <ul id="tasks"></ul>  
 <input type="text" id="new-task" placeholder="Добавить дело" />  
 <input id="add-task" type="button" value="Добавить"/>  
</div>  
<script src="index.js"></script>  
</body>  
</html>

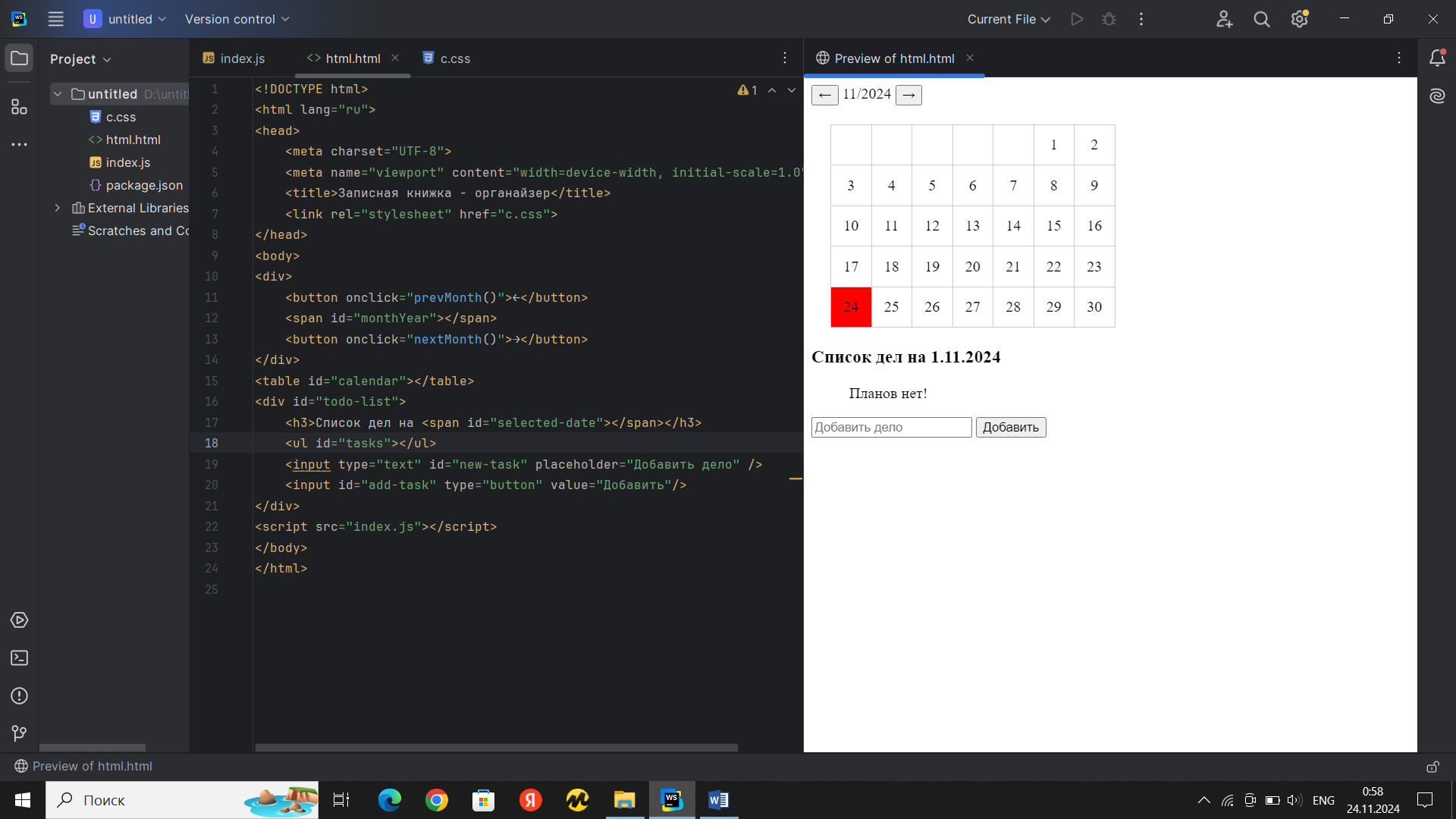
**Код CSS-файла**

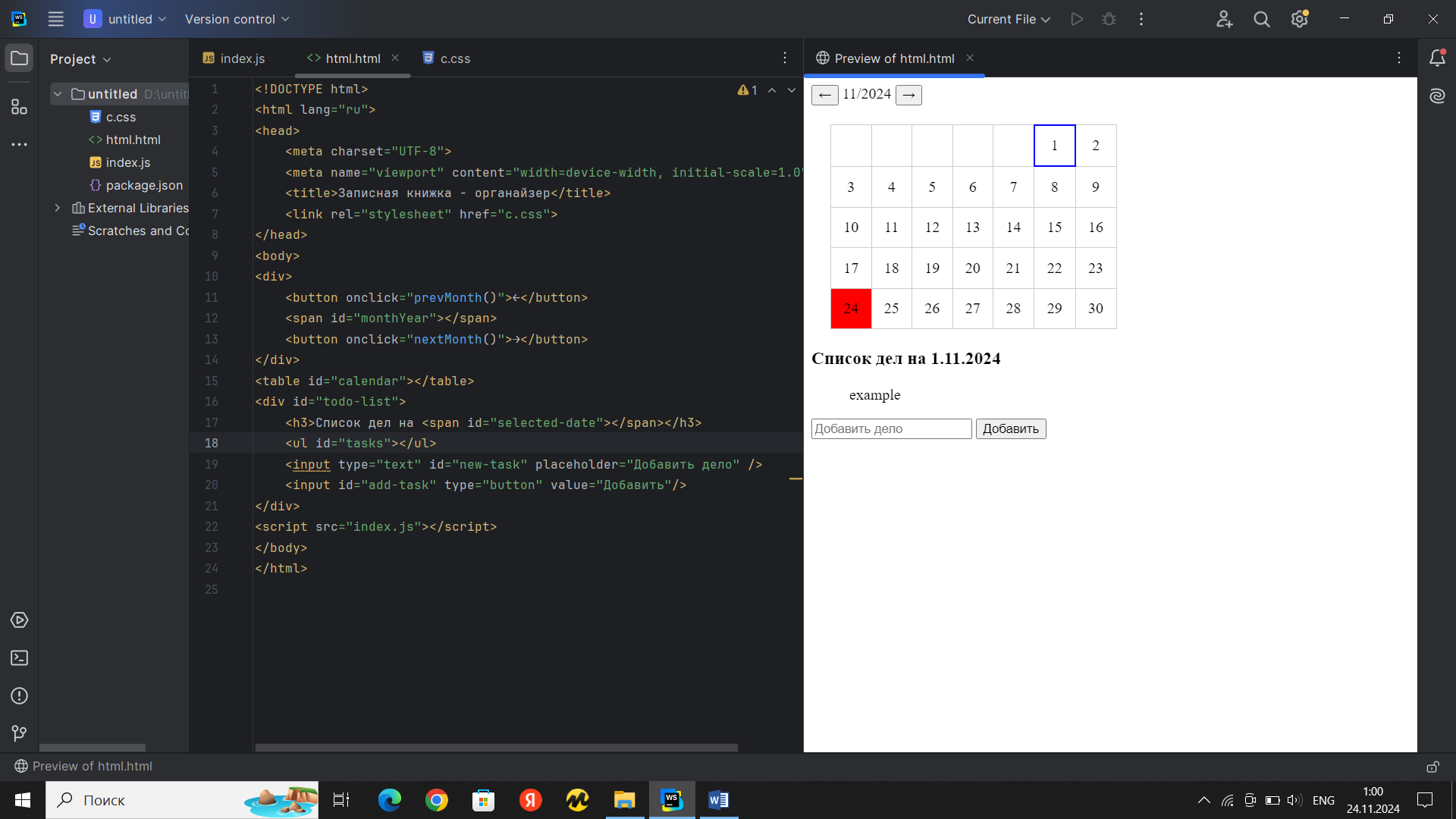
table { border-collapse: collapse; margin: 20px; }  
td { border: 1px solid #ccc; width: 40px; height: 40px; text-align: center; cursor: pointer; }  
.today { background: red; }  
.has-tasks { border: 2px solid blue;  
}

**Код JS-файла**

const *tasks* = {};  
let *currentDate* = new *Date*();  
function renderCalendar() {  
 const monthYear = *document*.getElementById("monthYear");  
 const calendar = *document*.getElementById("calendar");  
 const month = *currentDate*.getMonth();  
 const year = *currentDate*.getFullYear();  
 monthYear.textContent = `${month + 1}/${year}`;  
 calendar.innerHTML = '';  
 const firstDay = new *Date*(year, month, 1);  
 const lastDay = new *Date*(year, month + 1, 0).getDate();  
 const startDay = firstDay.getDay();  
 let date = 1;  
 for (let i = 0; i < 5; i++) {  
 let row = calendar.insertRow();  
 for (let j = 0; j < 7; j++) {  
 const cell = row.insertCell();  
 if (i === 0 && j < startDay) {  
 cell.textContent = '';  
 } else if (date > lastDay) {  
 cell.textContent = '';  
 } else {  
 cell.textContent = date;  
 const dateString = new *Date*(year, month, date).toDateString();  
 cell.onclick = function () {  
 const day = cell.textContent;  
 *document*.getElementById('selected-date').innerText = `${day}.${month + 1}.${year}`;  
 showTasks(dateString);  
 };  
 if (*tasks*[dateString]) cell.classList.add('has-tasks');  
 if (dateString === new *Date*().toDateString()) cell.classList.add('today');  
 date++;  
 }  
 }  
 }  
}  
function showTasks(dateString) {  
 const taskList = *document*.getElementById("tasks");  
 const taskArray = *tasks*[dateString] || [];  
 taskList.innerHTML = taskArray.length ? taskArray.join(', ') : "Планов нет!";  
 const input = *document*.getElementById('new-task');  
  
 *document*.getElementById('add-task').onclick = function() {  
 const newTask = input.value;  
 if(newTask){  
 *tasks*[dateString] = *tasks*[dateString] || [];  
 *tasks*[dateString].push(newTask);  
 renderCalendar();  
 }  
 input.value ="";  
  
 }  
}  
function prevMonth() {  
 *currentDate*.setMonth(*currentDate*.getMonth() - 1);  
 renderCalendar();  
}  
function nextMonth() {  
 *currentDate*.setMonth(*currentDate*.getMonth() + 1);  
 renderCalendar();  
}  
renderCalendar();







**Вывод:** в ходе лабораторной работы рассмотрели возможности, предоставляемые стандартизованными объектами языка ECMAScript, а также возможностей по созданию собственных типов объектов.

Различия JavaScript и ECMAScript. JavaScript - реализация, а ECMAScript - стандарт. Общие черты: синтаксис, особенности. Ключевые изменения включают поддержание классов, модулей и асинхронных функций. Тенденции: расширение функциональности, улучшение производительности и удобства работы.

В JavaScript функция является объектом. Преимущество в том, что функции могут иметь свойства и методы, быть переданными как аргументы, возвращаться из других функций. Это позволяет использовать функциональный стиль программирования, создавать функции высшего порядка и замыкания.