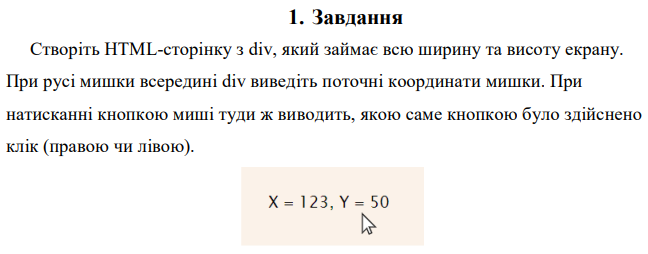
***Лабораторна робота №5***

***Хід роботи:***

***Завдання на лабораторну роботу:***

1. ***Завдання на лабораторну роботу:***

******

***Виконання:***

* ***HTML:***

<!DOCTYPE *html*>  
<html *lang*="en">  
<head>  
 <meta *charset*="UTF-8">  
 <title>Task 1</title>  
 <link *rel*="stylesheet" *href*="./styles/css/task\_1.css">  
 <link *rel*="icon" *href*="../../../../img/icons/racoon\_animal\_icon\_123571.ico">  
</head>  
<body>  
<div *class*="wrapper">  
 <div *class*="container"><p *class*="coordinates"></p><p *class*="mouse\_click"></p></div>  
 <script *src*="./js/task\_1.js"></script>  
</div>  
</body>  
</html>

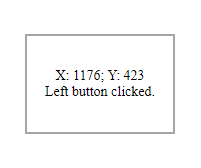
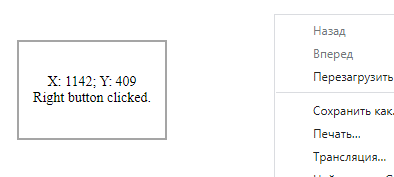
* ***SASS:***

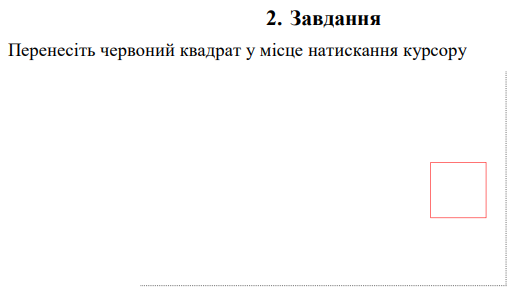
\*  
 margin: 0  
 padding: 0  
 box-sizing: border-box  
  
.*wrapper* width: 100%  
 height: 100vh  
 display: flex  
 align-items: center  
 justify-content: center  
  
.*container* display: flex  
 flex-direction: column  
 justify-content: center  
 align-items: center  
 width: 150px  
 height: 100px  
 font-size: 14px  
 border: 2px solid #a5a5a5

* ***JS:***

'use strict'  
  
*const findOutButtonClick* = (*button*) => {  
 *switch* (*button*) {  
 *case* 0:  
 *return* 'Left button clicked.';  
 *case* 1:  
 *return* 'Middle button clicked.';  
 *case* 2:  
 *return* 'Right button clicked.';  
 *case* 3:  
 *return* 'Back button clicked.';  
 *case* 4:  
 *return* 'Forward button clicked.';  
 *default*:  
 *return* `Unknown button code: ${*button*}`;  
 }  
}  
  
*const* coordinates = document.querySelector('.*coordinates*');  
*const* mouse\_click = document.querySelector('.*mouse\_click*');  
  
document.querySelector('.*wrapper*').addEventListener('mousemove', (*e*) => {  
 *const* coordinateX = *e*.clientX;  
 *const* coordinateY = *e*.clientY;  
 coordinates.innerHTML = `X: ${coordinateX}; Y: ${coordinateY}`;  
});  
  
window.addEventListener('mouseup', (*e*) => {  
 mouse\_click.innerHTML = `${*findOutButtonClick*(*e*.button)}`  
})

***Результат програми:***

******

***Виконання:***

* ***HTML:***

<!DOCTYPE *html*>  
<html *lang*="en">  
<head>  
 <meta *charset*="UTF-8">  
 <title>Task 2</title>  
 <link *rel*="stylesheet" *href*="./styles/css/task\_2.css">  
 <link *rel*="icon" *href*="../../../../img/icons/racoon\_animal\_icon\_123571.ico">  
</head>  
<body>  
<div *class*="wrapper">  
 <div *class*="square"></div>  
 <script *src*="./js/task\_2.js"></script>  
</div>  
</body>  
</html>

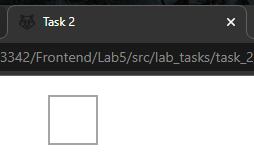
* ***SASS:***

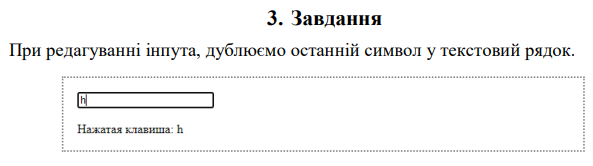
\*  
 margin: 0  
 padding: 0  
 box-sizing: border-box  
  
.*wrapper* width: 100%  
 height: 100vh  
 display: flex  
 align-items: center  
 flex-direction: column  
  
.*square* display: block  
 border: 2px solid #a5a5a5  
 width: 50px  
 height: 50px

* ***JS:***

'use strict';  
  
*const* wrapper = document.querySelector('.*wrapper*');  
*const* square = document.querySelector('.*square*');  
  
wrapper.addEventListener('click', (*e*) => {  
 *const* coordinateX = *e*.clientX;  
 *const* coordinateY = *e*.clientY;  
  
 square.style.position = 'absolute';  
 square.style.left = coordinateX + 'px';  
 square.style.top = coordinateY + 'px';  
});

***Результат програми:***

******

***Виконання:***

* ***HTML:***

<!DOCTYPE *html*>  
<html *lang*="en">  
<head>  
 <meta *charset*="UTF-8">  
 <title>Task 3</title>  
 <link *rel*="stylesheet" *href*="./styles/css/task\_3.css">  
 <link *rel*="icon" *href*="../../../../img/icons/racoon\_animal\_icon\_123571.ico">  
</head>  
<body>  
<div *class*="wrapper">  
 <div *class*="container">  
 <section *class*="section">  
 <table *class*="section\_\_table">  
 <tr>  
 <td>  
 <label *for*="inputText" *class*="table\_\_label\_text">Enter text:</label>  
 </td>  
 <td>  
 <input *type*="text" *name*="inputText" *id*="inputText" *class*="table\_\_input\_text" *maxlength*="40">  
 </td>  
 </tr>  
 <tr>  
 <td *colspan*="2">  
 <div *class*="table\_\_output\_text"></div>  
 </td>  
 </tr>  
 </table>  
 </section>  
 </div>  
 <script *src*="./js/task\_3.js"></script>  
</div>  
</body>  
</html>

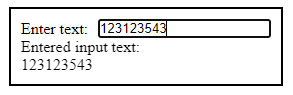
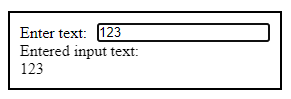
* ***SASS:***

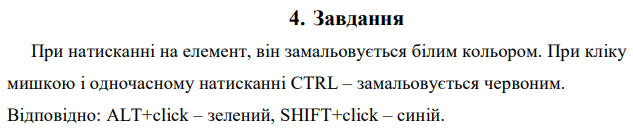
\*  
 margin: 0  
 padding: 0  
 box-sizing: border-box  
  
.*wrapper* width: 100%  
 height: 100vh  
 display: flex  
 align-items: center  
 justify-content: center  
 flex-direction: column  
  
.*section* border: 2px solid black  
 margin: 10px 0  
 padding: 10px  
  
.*section\_\_table* border-collapse: collapse  
  
.*table\_\_label\_text* margin-right: 10px

* ***JS:***

'use strict';  
  
*const* outputField = document.querySelector('.*table\_\_output\_text*');  
*const* inputText = document.querySelector('#inputText');  
  
inputText.addEventListener('input', (*event*) => {  
 outputField.innerHTML = `<p>Entered input text:</p><p>${*event*.target.value}</p>`;  
})

***Результат програми:***

****** ******

******

***Виконання:***

* ***HTML:***

<!DOCTYPE *html*>  
<html *lang*="en">  
<head>  
 <meta *charset*="UTF-8">  
 <title>Task 4</title>  
 <link *rel*="stylesheet" *href*="./styles/css/task\_4.css">  
 <link *rel*="icon" *href*="../../../../img/icons/racoon\_animal\_icon\_123571.ico">  
</head>  
<body>  
<div *class*="wrapper">  
 <div *class*="container">  
 <section *class*="section"></section>  
 </div>  
 <script *src*="./js/task\_4.js"></script>  
</div>  
</body>  
</html>

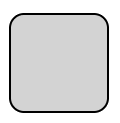
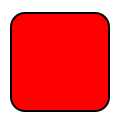
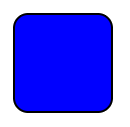
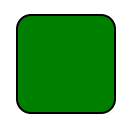
* ***SASS:***

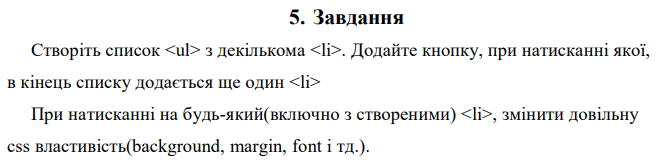
\*  
 margin: 0  
 padding: 0  
 box-sizing: border-box  
  
.*wrapper* width: 100%  
 height: 100vh  
 display: flex  
 align-items: center  
 justify-content: center  
 flex-direction: column  
  
.*section* border: 2px solid black  
 border-radius: 15px  
 width: 100px  
 height: 100px  
 cursor: pointer

* ***JS:***

'use strict';  
  
*const* section = document.querySelector('.*section*');  
  
section.addEventListener('click', (*event*) => {  
 *if* (*event*.button === 0)  
 section.style.backgroundColor = 'lightgrey';  
  
 *if* ((*event*.ctrlKey || *event*.metaKey) && *event*.isTrusted)  
 section.style.backgroundColor = 'red';  
  
 *if* ((*event*.altKey) && *event*.isTrusted)  
 section.style.backgroundColor = 'green';  
  
 *if* (*event*.shiftKey && *event*.isTrusted)  
 section.style.backgroundColor = 'blue';  
})

***Результат програми:***

******

***Виконання:***

* ***HTML:***

<!DOCTYPE *html*>  
<html *lang*="en">  
<head>  
 <meta *charset*="UTF-8">  
 <title>Task 5</title>  
 <link *rel*="stylesheet" *href*="./styles/css/task\_5.css">  
 <link *rel*="icon" *href*="../../../../img/icons/racoon\_animal\_icon\_123571.ico">  
</head>  
<body>  
<div *class*="wrapper">  
 <div *class*="container">  
 <section *class*="section">  
 <table *class*="section\_\_table">  
 <tr>  
 <td>  
 <label *for*="olList" *class*="table\_\_label\_ol\_list"  
 *id*="labelOlList">Ol elements list:</label>  
 </td>  
 </tr>  
 <tr>  
 <td>  
 <ol *class*="table\_\_ol\_list" *id*="olList">  
 <li *class*="ol\_list\_\_li\_element">1</li>  
 <li *class*="ol\_list\_\_li\_element">2</li>  
 <li *class*="ol\_list\_\_li\_element">3</li>  
 <li *class*="ol\_list\_\_li\_element">4</li>  
 <li *class*="ol\_list\_\_li\_element">5</li>  
 </ol>  
 </td>  
 </tr>  
 <tr>  
 <td>  
 <input *type*="button" *value*="Add element to ol list"  
 *id*="buttonAddElemList" *class*="table\_\_button\_add\_elem\_list">  
 </td>  
 </tr>  
 </table>  
 </section>  
 </div>  
 <script *src*="./js/task\_5.js"></script>  
</div>  
</body>  
</html>

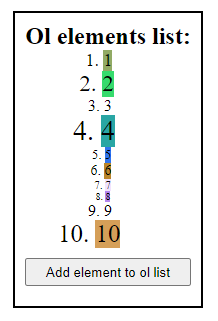
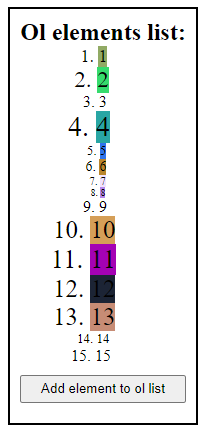
* ***SASS:***

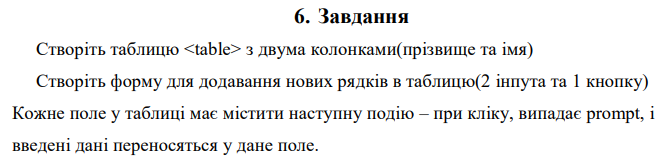
\*  
 margin: 0  
 padding: 0  
 box-sizing: border-box  
  
.*wrapper* width: 100%  
 height: 100vh  
 display: flex  
 align-items: center  
 justify-content: center  
 flex-direction: column  
  
.*section* border: 2px solid black  
 margin: 10px 0  
 padding: 10px  
 text-align: center  
  
.*section\_\_table* border-collapse: collapse  
  
.*table\_\_label\_ol\_list* font-size: 24px  
 font-weight: bold  
  
.*table\_\_ol\_list* display: flex  
 flex-direction: column  
 align-items: center  
  
.*ol\_list\_\_li\_element* cursor: pointer  
  
.*table\_\_button\_add\_elem\_list* margin: 10px 0  
 width: 100%  
 height: 28px

* ***JS:***

'use strict';  
  
*const* getRandomInt = (fromNumber, toNumber) => Math.round((Math.random() \* (toNumber - fromNumber) + fromNumber));  
*const* generateRandomColor = () => Math.floor(Math.random() \* 16777215).toString(16);  
  
*function* addEventListenersToList(nodeList) {  
 *for* (*let* i = 0; i < nodeList.length; i++) {  
 nodeList[i].addEventListener('click', (event) => {  
 *const* chosenCss = getRandomInt(0, 2);  
 *const* target = event.target;  
 console.log(chosenCss);  
 *switch* (chosenCss) {  
 *case* 0:  
 target.style.backgroundColor = '#' + *generateRandomColor*();  
 *break*;  
 *case* 1:  
 target.style.fontSize = *getRandomInt*(8, 28) + 'px';  
 *break*;  
 *case* 2:  
 target.style.padding = `${getRandomInt(0, 25)}px, ${getRandomInt(0, 25)}px, ${getRandomInt(0, 25)}px, ${getRandomInt(0, 25)}px`;  
 *break*;  
 }  
 })  
 }  
}  
  
*const* buttonAddElemList = document.querySelector('#buttonAddElemList');  
*const* olList = document.querySelector('#olList');  
*let* olListElements = document.querySelectorAll('.*ol\_list\_\_li\_element*');  
  
*let* elemsList = []  
*for* (*let* i = 0; i < olListElements.length; i++) {  
 elemsList.push(olListElements[i].textContent);  
}  
addEventListenersToList(olListElements);  
  
buttonAddElemList.addEventListener('click', () => {  
 *const* element = document.createElement('li');  
 element.innerHTML = `${elemsList.length + 1}`;  
 element.classList.add('ol\_list\_\_li\_element');  
  
 elemsList.push((elemsList.length + 1).toString());  
 olList.appendChild(element);  
  
 olListElements = document.querySelectorAll('.*ol\_list\_\_li\_element*');  
 addEventListenersToList(olListElements);  
});

***Результат програми:***

******

***Виконання:***

* ***HTML:***

<!DOCTYPE *html*>  
<html *lang*="en">  
<head>  
 <meta *charset*="UTF-8">  
 <title>Task 6</title>  
 <link *rel*="stylesheet" *href*="./styles/css/task\_6.css">  
 <link *rel*="icon" *href*="../../../../img/icons/racoon\_animal\_icon\_123571.ico">  
</head>  
<body>  
<div *class*="wrapper">  
 <div *class*="container">  
 <section *class*="section">  
 <table *class*="section\_\_table">  
 <tr>  
 <td>  
 <table *class*="table\_\_output\_table" *id*="outputNameSurnameTable">  
 <tr>  
 <td>  
 Surname  
 </td>  
 <td>  
 Name  
 </td>  
 </tr>  
 </table>  
 </td>  
 </tr>  
 <tr>  
 <td>  
 <input *type*="button" *value*="Add element to table" *id*="buttonAddElemTable" *class*="table\_\_button\_add\_elem\_table">  
 </td>  
 </tr>  
 </table>  
 </section>  
 </div>  
 <script *src*="./js/task\_6.js"></script>  
</div>  
</body>  
</html>

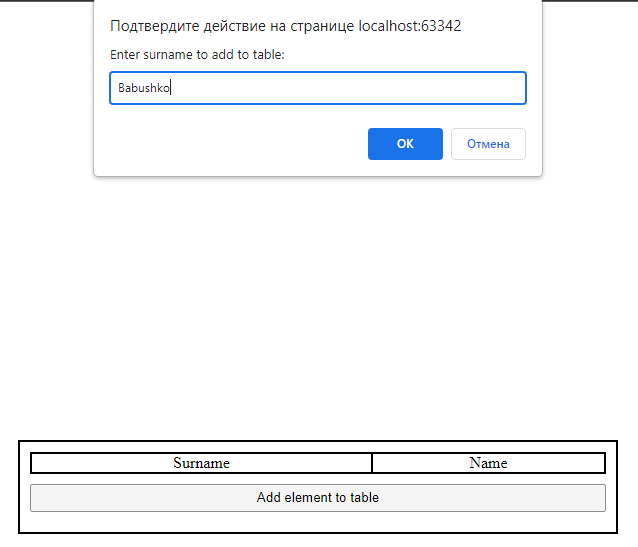
* ***SASS:***

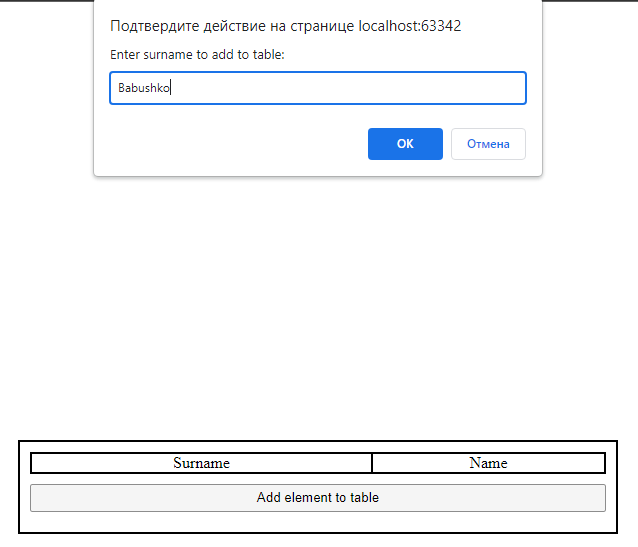
\*  
 margin: 0  
 padding: 0  
 box-sizing: border-box  
 *// outline: 1px solid red*.*wrapper* width: 100%  
 height: 100vh  
 display: flex  
 align-items: center  
 justify-content: center  
 flex-direction: column  
  
.*section* border: 2px solid black  
 margin: 10px 0  
 padding: 10px  
 text-align: center  
 width: 600px  
 max-width: 60vw  
  
.*section\_\_table* width: 100%  
 border-collapse: collapse  
  
.*table\_\_button\_add\_elem\_table* margin: 10px 0  
 width: 100%  
 height: 28px  
 cursor: pointer  
  
.*table\_\_output\_table* border-collapse: collapse  
 width: 100%  
 tr  
 &:*first-child* td  
 border: 2px solid black  
 cursor: auto  
 td  
 border: 1.5px solid black  
 cursor: pointer

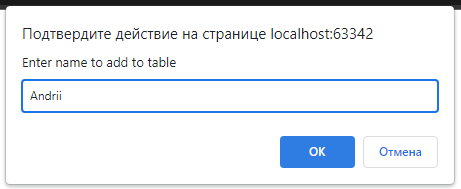
* ***JS:***

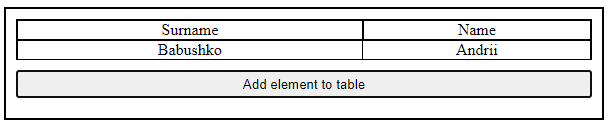
'use strict';  
  
*function addEventListenersToTable*(*tableNodeList*) {  
 *for* (*let* i = 0; i < *tableNodeList*.length; i++) {  
 *tableNodeList*[i].addEventListener('click', (*event*) => {  
 *const* target = *event*.target;  
 *const* surname = target.parentElement.firstChild.textContent;  
 *const* name = target.parentElement.lastChild.textContent;  
 alert(`Clicked row:\nSurname: '${surname}'; Name: '${name}'`);  
 });  
 }  
}  
  
*const* buttonAddElemTable = document.querySelector('#buttonAddElemTable');  
  
buttonAddElemTable.addEventListener('click', (*event*) => {  
 *const* surname = *prompt*('Enter surname to add to table:');  
 *const* name = *prompt*('Enter name to add to table');  
  
 *const* outputNameSurnameTable = document.querySelector('#outputNameSurnameTable')  
  
 *const* newTableRow = document.createElement('tr');  
 *const* newSurnameColumn = document.createElement('td');  
 *const* newNameColumn = document.createElement('td');  
  
 newSurnameColumn.innerHTML = surname;  
 newNameColumn.innerHTML = name;  
  
 newTableRow.appendChild(newSurnameColumn);  
 newTableRow.appendChild(newNameColumn);  
 outputNameSurnameTable.appendChild(newTableRow);  
  
 *const* outputTableElems = document.querySelectorAll('#outputNameSurnameTable tr');  
 console.log(outputTableElems);  
 *addEventListenersToTable*(outputTableElems);  
});

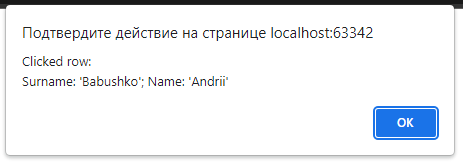
***Результат програми:***

******

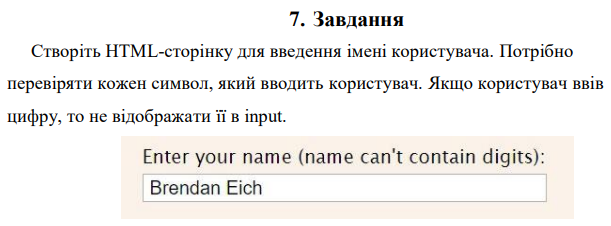




******

******

1. ***Домашнє завдання:***

******

***Виконання:***

* ***HTML:***

<!DOCTYPE *html*>  
<html *lang*="en">  
<head>  
 <meta *charset*="UTF-8">  
 <title>Task 7</title>  
 <link *rel*="stylesheet" *href*="./styles/css/task\_7.css">  
 <link *rel*="icon" *href*="../../../../img/icons/racoon\_animal\_icon\_123571.ico">  
</head>  
<body>  
<div *class*="wrapper">  
 <div *class*="container">  
 <section *class*="section">  
 <table *class*="section\_\_table">  
 <tr>  
 <td>  
 <label *for*="inputName" *id*="labelName" *class*="table\_\_label\_name">Enter name(can't contain digits):</label>  
 </td>  
 <td>  
 <input *type*="text" *name*="inputName" *id*="inputName" *class*="table\_\_input\_name" *required*>  
 </td>  
 </tr>  
 </table>  
 </section>  
 </div>  
 <script *src*="./js/task\_7.js"></script>  
</div>  
</body>  
</html>

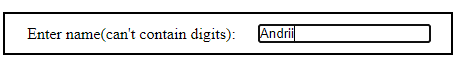
* ***SASS:***

\*  
 margin: 0  
 padding: 0  
 box-sizing: border-box  
*// outline: 1px solid red*.*wrapper* width: 100%  
 height: 100vh  
 display: flex  
 align-items: center  
 justify-content: center  
 flex-direction: column  
  
.*section* border: 2px solid black  
 margin: 10px 0  
 padding: 10px  
 text-align: center  
 width: 450px  
 max-width: 60vw  
  
.*section\_\_table* width: 100%  
 border-collapse: collapse

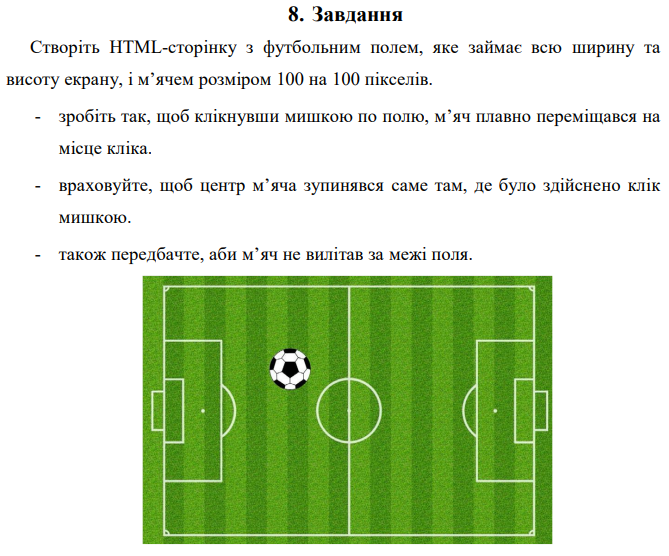
* ***JS:***

'use strict';  
  
*const containsNumbers* = (*str*) => /\d/.test(*str*);  
  
*const* inputName = document.querySelector('#inputName');  
  
inputName.addEventListener('input', (*event*) => {  
 *if* (*containsNumbers*(*event*.target.value)){  
 console.log('CONTAIN DIGITS!')  
 console.log(`Target value: ${*event*.target.value}`);  
 *event*.target.value = *event*.target.value.replace(/\d/g, '');  
 }  
});

***Результат програми:***





******

***Виконання:***

* ***HTML:***

<!DOCTYPE *html*>  
<html *lang*="en">  
<head>  
 <meta *charset*="UTF-8">  
 <title>Task 8</title>  
 <link *rel*="stylesheet" *href*="./styles/css/task\_8.css">  
 <link *rel*="icon" *href*="../../../../img/icons/racoon\_animal\_icon\_123571.ico">  
</head>  
<body>  
 <div *class*="wrapper">  
 <div *class*="football\_stadium">  
 <img *src*="./img/football\_ball.png" *alt*="football ball 2d" *class*="football\_stadium\_\_ball" *id*="ball">  
 </div>  
 </div>  
 <script *src*="./js/task\_8.js"></script>  
</body>  
</html>

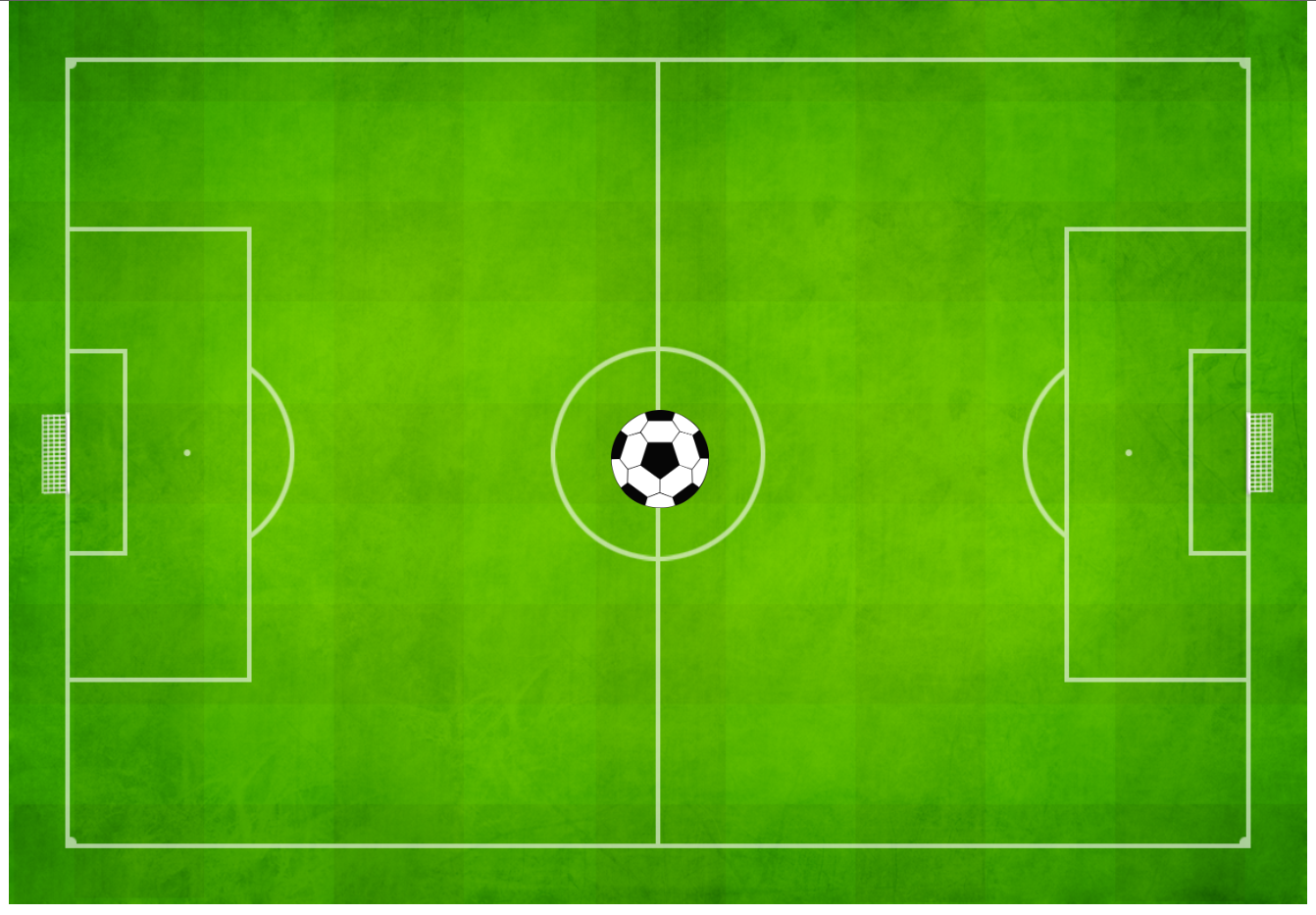
* ***SASS:***

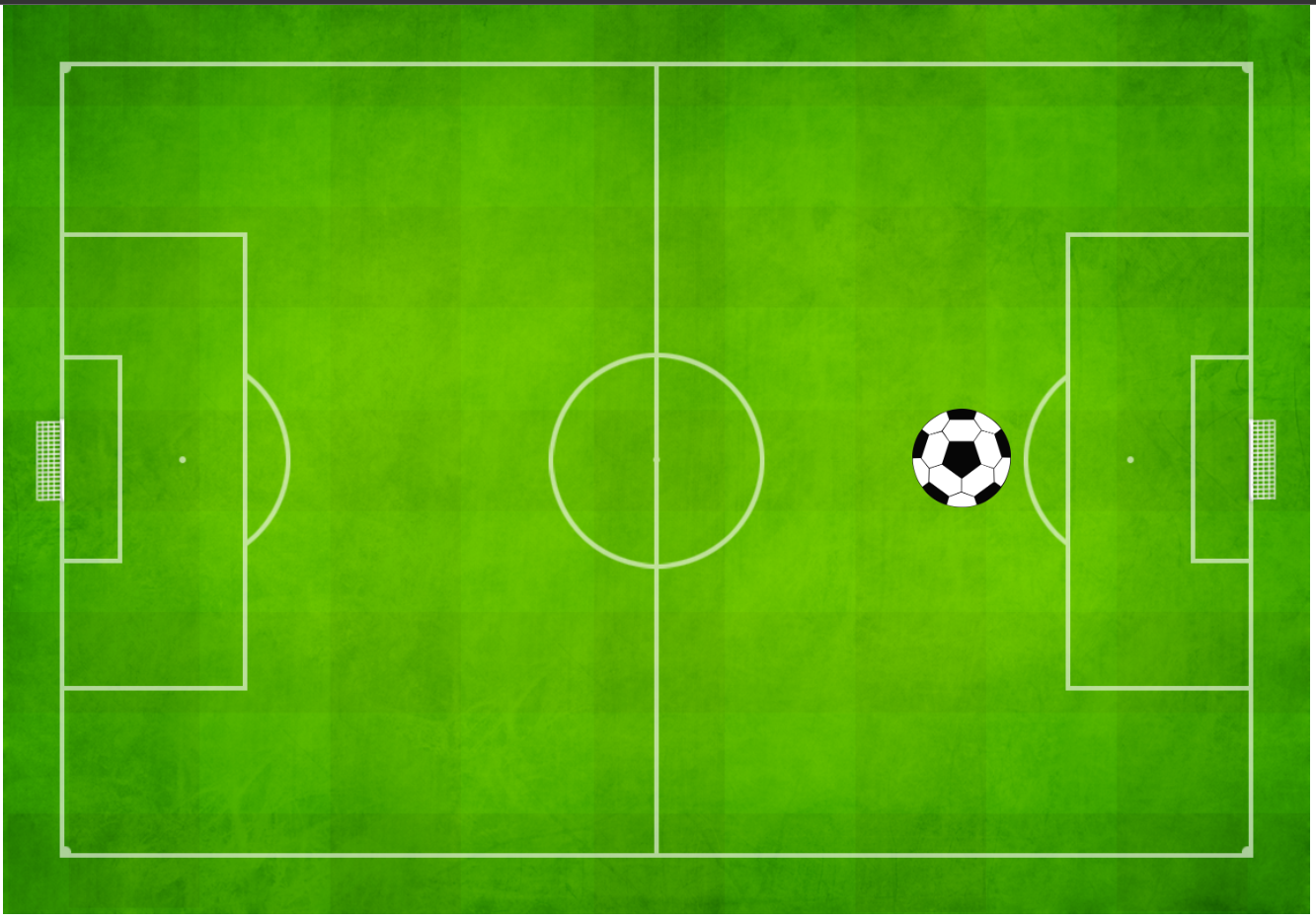
\*  
 margin: 0  
 padding: 0  
 box-sizing: border-box  
 *// outline: 1px solid red*body  
 overflow-y: hidden  
  
.*wrapper* width: 100%  
 height: 100vh  
 display: flex  
 justify-content: center  
  
.*football\_stadium* background: url("../../img/football\_stadium\_2d.png") no-repeat  
 background-size: contain  
 width: 70%  
 height: 95%  
 cursor: pointer  
  
.*football\_stadium\_\_ball* width: 100px  
 height: 100px  
 position: absolute  
 top: 43%  
 left: 47%  
 transition: top .5s ease, left .5s ease

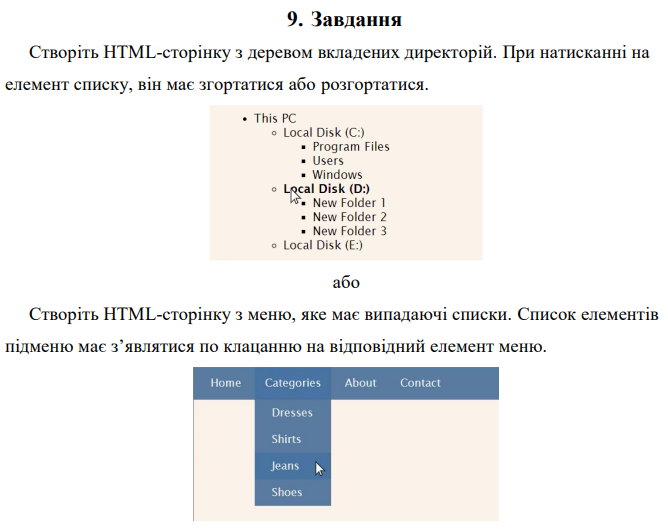
* ***JS:***

'use strict';  
  
*const* footballStadium = document.querySelector('.*football\_stadium*');  
*const* ball = document.querySelector('#ball');  
  
*const* ballTiming = {  
 duration: 2000,  
 iterations: 1,  
}  
  
footballStadium.addEventListener('click', (*event*) => {  
 *const* FinishX = *event*.clientX;  
 *const* FinishY = *event*.clientY;  
 *const* footballStadiumRect = footballStadium.getBoundingClientRect()  
 *if* (FinishY < footballStadiumRect.height + window.innerHeight \* 0.05 && FinishX < footballStadiumRect.width + window.innerWidth \* 0.2) {  
 ball.style.top = FinishY - 50 + 'px';  
 ball.style.left = FinishX - 50 + 'px';  
 }  
}, *false*);

***Результат програми:***





******

***Виконання:***

* ***HTML:***

<!DOCTYPE *html*>  
<html *lang*="en">  
<head>  
 <meta *charset*="UTF-8">  
 <title>Task 9</title>  
 <link *rel*="stylesheet" *href*="./styles/css/task\_9.css">  
 <link *rel*="icon" *href*="../../../../img/icons/racoon\_animal\_icon\_123571.ico">  
</head>  
<body>  
<div *class*="wrapper">  
 <div *class*="nav\_bar">  
 <div *class*="nav\_bar\_\_item" *id*="home">Home</div>  
 <div *class*="nav\_bar\_\_item" *id*="categories">Categories  
 <div *class*="nav\_bar\_\_sub\_nav\_bar">  
 <div *class*="sub\_nav\_bar\_item">Dresses</div>  
 <div *class*="sub\_nav\_bar\_item">Shirts</div>  
 <div *class*="sub\_nav\_bar\_item">Jeans</div>  
 <div *class*="sub\_nav\_bar\_item">Shoes</div>  
 </div>  
 </div>  
 <div *class*="nav\_bar\_\_item" *id*="about">About</div>  
 <div *class*="nav\_bar\_\_item" *id*="contact">Contact</div>  
 </div>  
</div>  
<script *src*="./js/task\_9.js"></script>  
</body>  
</html>

* ***SASS:***

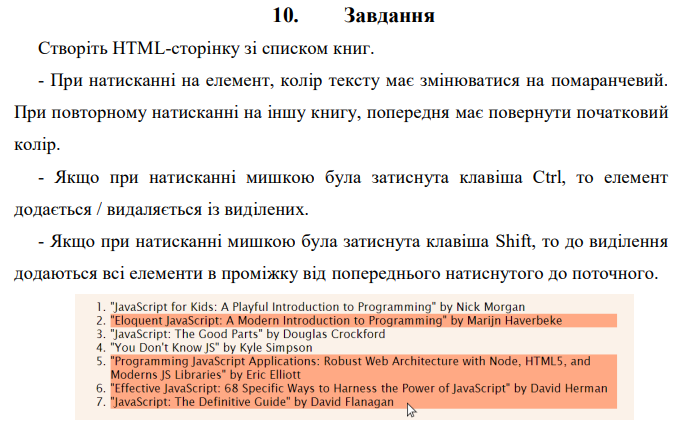
\*  
 margin: 0  
 padding: 0  
 box-sizing: border-box  
*// outline: 1px solid red*.*wrapper* width: 100%  
 height: 100vh  
 display: flex  
 justify-content: center  
  
.*nav\_bar* width: 30%  
 height: 40px  
 background-color: steelblue  
 display: flex  
 justify-content: space-evenly  
  
.*nav\_bar\_\_item*, .*sub\_nav\_bar\_item* font-size: 18px  
 padding: 10px  
 width: 100%  
 color: white  
 background-color: steelblue  
 display: flex  
 align-items: center  
 justify-content: center  
 cursor: pointer  
  
 &:*hover* animation-name: nav\_item\_color  
 animation-duration: 3s  
 animation-timing-function: ease-in-out  
  
*@keyframes* nav\_item\_color  
 *from* background-color: steelblue  
 50%  
 background-color: #6b94b7  
 *to* background-color: steelblue  
  
.*nav\_bar\_\_sub\_nav\_bar* &.*active* display: flex  
 position: relative  
 display: none  
 flex-direction: column  
 justify-content: space-evenly  
 align-items: center  
 width: 90px  
  
.*nav\_bar\_\_sub\_nav\_bar* position: absolute  
 top: 4.1%  
 background-color: steelblue

* ***JS:***

'use strict';  
  
*const* navBarItems = document.querySelectorAll('.*nav\_bar\_\_item*');  
  
*for* (*let* i = 0; i < navBarItems.length; i++) {  
 navBarItems[i].addEventListener('click', (*event*) => {  
 *const* subNavBarElement = *event*.target.firstElementChild  
 *if* (subNavBarElement){  
 console.log('SUB NAV BAR!');  
 subNavBarElement.classList.toggle('active');  
 }  
 });  
}

***Результат програми:***

******

******

***Виконання:***

* ***HTML:***

<!DOCTYPE *html*>  
<html *lang*="en">  
<head>  
 <meta *charset*="UTF-8">  
 <title>Task 10</title>  
 <link *rel*="stylesheet" *href*="./styles/css/task\_10.css">  
 <link *rel*="icon" *href*="../../../../img/icons/racoon\_animal\_icon\_123571.ico">  
</head>  
<body>  
<div *class*="wrapper">  
 <div *class*="books\_container">  
 <ol *class*="books\_container\_\_ol\_list">  
 <li *class*="ol\_list\_\_li\_book">Robert Cecil Martin - Clean Code</li>  
 <li *class*="ol\_list\_\_li\_book">Andy Hunt and Dave Thomas - The Pragmatic Programmer</li>  
 <li *class*="ol\_list\_\_li\_book">Erich Gamma, John Vlissides, Ralph Johnson, Richard Helm - Design Patterns: Elements of Reusable Object-Oriented Software</li>  
 <li *class*="ol\_list\_\_li\_book">Robert Cecil Martin - The Clean Coder: A Code of Conduct for Professional Programmers</li>  
 <li *class*="ol\_list\_\_li\_book">Kent Beck and Martin Fowler - Refactoring: Improving the Design of Existing Code</li>  
 <li *class*="ol\_list\_\_li\_book">Steve McConnell - Code Complete</li>  
 <li *class*="ol\_list\_\_li\_book">Eric Freeman, Elisabeth Robson, Bert Bates, Kathy Sierra - Head First Design Patterns</li>  
 </ol>  
 </div>  
</div>  
<script *src*="./js/task\_10.js"></script>  
</body>  
</html>

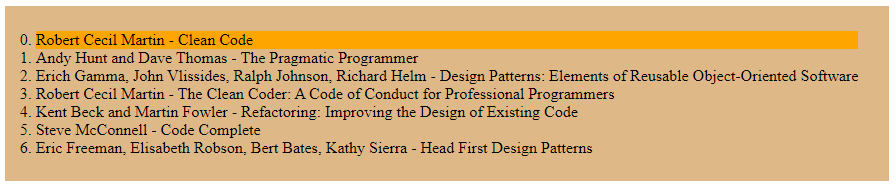
* ***SASS:***

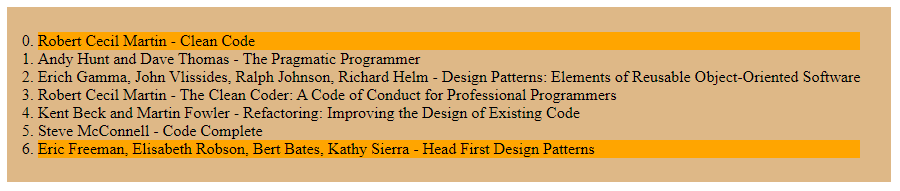
\*  
 margin: 0  
 padding: 0  
 box-sizing: border-box  
*// outline: 1px solid red*.*wrapper* width: 100%  
 height: 100vh  
 display: flex  
 justify-content: center  
 align-items: center  
  
.*books\_container* width: 46%  
 height: 18%  
 position: relative  
 display: flex  
 justify-content: center  
 align-items: center  
 background-color: burlywood  
  
.*books\_container\_\_ol\_list* cursor: pointer  
  
.*ol\_list\_\_li\_book* &.*active* background-color: orange

* ***JS:***

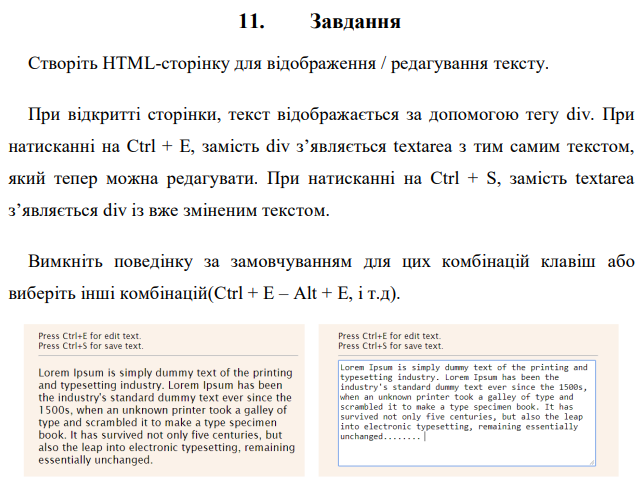
'use strict';  
  
*const booksToggle* = (*event*, *books*, *booksStatus*, *index*) => {  
 *event*.target.classList.toggle('active');  
  
 clickedBooksIndexes.push(*index*)  
  
 *if* (*booksStatus*[*index*] === 'active') {  
 *booksStatus*[*index*] = 'inactive';  
 } *else* {  
 *booksStatus*[*index*] = 'active'  
 }  
  
 console.log(*booksStatus*);  
};  
  
*const setBooks* = (*event*, *bookIndex*, *books*, *booksStatus*, *clickedBooksIndexes*) => {  
 *const* index = *parseInt*(*event*.target.getAttribute('value'));  
 *if* (!*booksStatus*.includes('active') || *event*.ctrlKey) {  
 *booksToggle*(*event*, *books*, *booksStatus*, index);  
 *return*;  
 }  
 *if* (*event*.shiftKey){  
 console.log('SHIFT');  
 *clickedBooksIndexes*.push(index)  
 *const* startIndex = *clickedBooksIndexes*[*clickedBooksIndexes*.length - 2];  
 *const* finishIndex = *clickedBooksIndexes*[*clickedBooksIndexes*.length - 1];  
 console.log(startIndex, finishIndex);  
 *if* (startIndex > finishIndex){  
 *for* (*let* i = startIndex; i >= finishIndex; i--) {  
 *if* (i !== startIndex)  
 books[i].classList.toggle('active');  
 booksStatus[i] = 'active';  
 }  
 }  
  
 *if* (startIndex < finishIndex){  
 *for* (*let* i = startIndex; i <= finishIndex; i++) {  
 *if* (i !== startIndex)  
 books[i].classList.toggle('active');  
 booksStatus[i] = 'active';  
 }  
 }  
 console.log(booksStatus);  
 *return*;  
 }  
 *if* (booksStatus.includes('active')) {  
 console.log(clickedBooksIndexes);  
 *const* activeIndex = clickedBooksIndexes[clickedBooksIndexes.length - 1];  
 books[activeIndex].classList.toggle('active');  
 booksStatus[activeIndex] = 'inactive';  
 booksToggle(event, books, booksStatus, index);  
 }  
}  
  
*const* books = document.querySelectorAll('.*ol\_list\_\_li\_book*');  
  
*let* booksStatus = [];  
*let* clickedBooksIndexes = []  
*for* (*let* i = 0; i < books.length; i++) {  
 books[i].classList.add('inactive');  
 books[i].setAttribute('value', i.toString());  
 booksStatus.push('inactive');  
  
 books[i].addEventListener('click', (event) => {  
 console.log('EVENT!');  
 setBooks(event, i, books, booksStatus, clickedBooksIndexes)  
 });  
}

***Результат програми:***

******



******

******

***Виконання:***

* ***HTML:***

<!DOCTYPE *html*>  
<html *lang*="en">  
<head>  
 <meta *charset*="UTF-8">  
 <title>Task 11</title>  
 <link *rel*="stylesheet" *href*="./styles/css/task\_11.css">  
 <link *rel*="icon" *href*="../../../../img/icons/racoon\_animal\_icon\_123571.ico">  
</head>  
<body>  
<div *class*="wrapper">  
 <div *class*="container">  
 <div *class*="container\_\_hotkeys">  
 <p *class*="hotkeys\_\_p\_ctrl\_e" *id*="ctrl\_e">Press <strong>CTRL+E</strong> for edit text</p>  
 <p *class*="hotkeys\_\_p\_ctrl\_s" *id*="ctrl\_s">Press <strong>CTRL+S</strong> for save text</p>  
 </div>  
 <div *class*="container\_\_div\_text">  
 <div *class*="div\_text\_\_text">  
 <p>  
 Lorem ipsum dolor sit amet, consectetur adipisicing elit. Aliquid assumenda atque cum eaque enim eveniet incidunt laboriosam maiores nemo numquam odit,  
 praesentium quidem ratione totam unde velit veniam vero, voluptatibus! At maiores minus nulla quis voluptatibus! Adipisci atque, aut consequatur culpa ea esse  
 et eum eveniet ex facilis, impedit ipsum maxime modi molestias pariatur porro quo quod voluptatum? Enim, et. Ab excepturi libero nam non! Aut minima quis quod  
 rem soluta. Aperiam dignissimos exercitationem explicabo hic, ipsam nemo possimus. A aspernatur enim natus obcaecati quibusdam, sed veritatis. Alias, illum,  
 laudantium. Consequatur delectus dignissimos expedita illum in laborum laudantium mollitia nisi obcaecati omnis, porro quas reiciendis similique voluptatibus  
 voluptatum? Aliquam aspernatur commodi enim esse exercitationem illo itaque laboriosam natus odit quos!  
 </p>  
 </div>  
 </div>  
</div>  
<script *src*="./js/task\_11.js"></script>  
</body>  
</html>

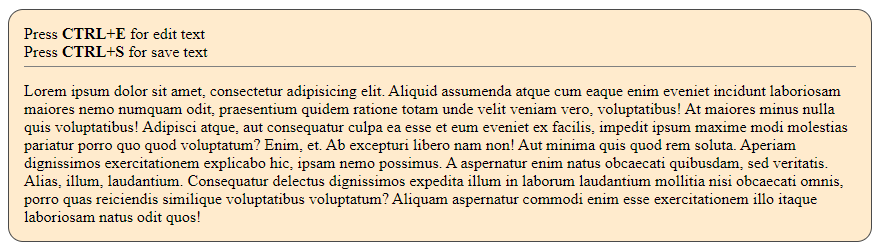
* ***SASS:***

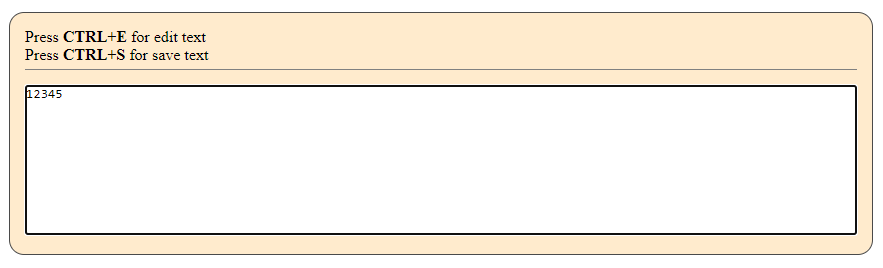
\*  
 margin: 0  
 padding: 0  
 box-sizing: border-box  
*// outline: 1px solid red*.*wrapper* width: 100%  
 height: 100vh  
 display: flex  
 justify-content: center  
 align-items: center  
  
.*container* width: 45%  
 display: flex  
 flex-direction: column  
 justify-content: center  
 border: 1px solid #4a4a4a  
 border-radius: 15px  
 background-color: blanchedalmond  
 padding: 15px  
  
.*container\_\_hotkeys* padding-bottom: 5px  
 margin-bottom: 15px  
 border-bottom: 1px solid grey  
  
.*div\_text\_\_text* width: 100%  
  
 &.*active* display: inherit  
  
.*div\_text\_\_textarea* width: 100%  
  
 &.*active* display: inherit  
  
 .*textarea\_\_change\_text* width: 100%  
 height: 150px  
 resize: none

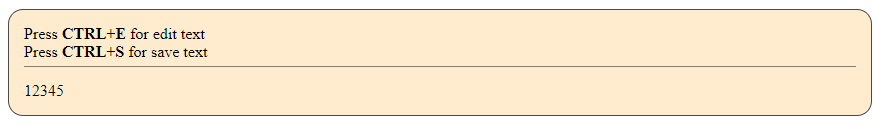
* ***JS:***

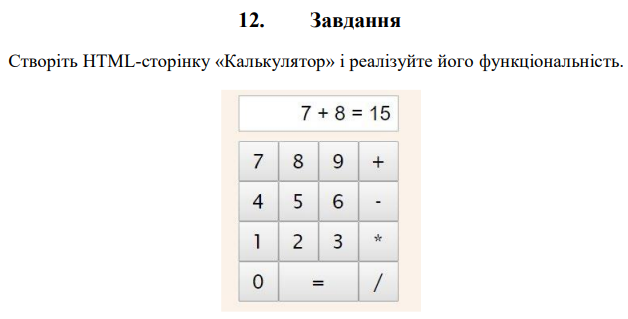
'use strict';  
  
*const deleteElemFromArr* = (*array*, *deleteElem*) => {  
 *const* index = *array*.indexOf(*deleteElem*);  
  
 *if* (index > -1) { *// only splice array when item is found  
 array*.splice(index, 1); *// 2nd parameter means remove one item only* }  
  
 *return array*}  
  
*const removeSpacesAndBreakLines* = (*splitText*) => {  
 *for* (*let* i = 0; i < *splitText*.length; i++) {  
 *if* (*splitText*[i] === '' || /^[ \n\r]$/.test(*splitText*[i]))  
 *splitText* = *deleteElemFromArr*(*splitText*, *splitText*[i]);  
 }  
  
 *return splitText*}  
  
document.addEventListener('keydown', (*keyDownEvent*) => {  
 *// if ((keyDownEvent.ctrlKey || keyDownEvent.metaKey) && (keyDownEvent.key === 'e' || keyDownEvent.key === 'E') ||  
 // (keyDownEvent.ctrlKey || keyDownEvent.metaKey) && (keyDownEvent.key === 's' || keyDownEvent.key === 'S')) {  
 // keyDownEvent.preventDefault();  
 // }  
 if* ((*keyDownEvent*.key === 'e' || *keyDownEvent*.key === 'E')) {  
 console.log('ctrl+e')  
  
 *const* divText = document.querySelector('.*div\_text\_\_text*');  
  
 *let* splitText = divText.firstElementChild.innerHTML.replace(/\n/g, '').split(' ');  
 *for* (*let* i = 0; i < 3; i++) {  
 splitText = *removeSpacesAndBreakLines*(splitText);  
 }  
 *const* text = splitText.join(' ');  
  
 *const* divTextarea = document.createElement('div');  
 *const* textarea = document.createElement('textarea');  
  
 divTextarea.setAttribute('class', 'div\_text\_\_textarea');  
 textarea.setAttribute('name', 'textareaTextChange');  
 textarea.setAttribute('id', 'textareaTextChange');  
 textarea.setAttribute('class', 'textarea\_\_change\_text');  
  
 textarea.value = text;  
  
 divTextarea.appendChild(textarea);  
 divText.parentElement.appendChild(divTextarea);  
  
 divText.remove();  
 }  
  
 *if* ((*keyDownEvent*.key === 's' || *keyDownEvent*.key === 'S')) {  
 console.log('ctrl+s');  
  
 *const* divTextarea = document.querySelector('.*div\_text\_\_textarea*');  
 *const* textarea = document.querySelector('#textareaTextChange');  
 *const* textareaText = textarea.value;  
  
 *const* divText = document.createElement('div');  
 *const* p = document.createElement('p');  
  
 divText.setAttribute('class', 'div\_text\_\_text');  
  
 p.textContent = textareaText;  
  
 divText.appendChild(p);  
 divTextarea.parentElement.appendChild(divText);  
  
 divTextarea.remove();  
 }  
}, *false*);

***Результат програми:***

******

******

******

******

***Виконання:***

* ***HTML:***

<!DOCTYPE *html*>  
<html *lang*="en">  
<head>  
 <meta *charset*="UTF-8">  
 <title>Task 12</title>  
 <link *rel*="stylesheet" *href*="./styles/css/task\_12.css">  
 <link *rel*="icon" *href*="../../../../img/icons/racoon\_animal\_icon\_123571.ico">  
</head>  
<body>  
<div *class*="wrapper">  
 <div *class*="container">  
 <div *class*="container\_\_calculator">  
 <input *type*="text" *name*="calculatorField" *id*="calculatorField" *class*="calculator\_\_field" *readonly*>  
 <button *type*="button" *id*="calculatorTotalClear" *class*="calculator\_\_total\_clear">C</button>  
 <button *type*="button" *id*="calculator7" *class*="calculator\_\_number" *data-num*="7">7</button>  
 <button *type*="button" *id*="calculator8" *class*="calculator\_\_number" *data-num*="8">8</button>  
 <button *type*="button" *id*="calculator9" *class*="calculator\_\_number" *data-num*="9">9</button>  
 <button *type*="button" *id*="calculatorAdd" *class*="calculator\_\_action" *data-action*="Add">+</button>  
 <button *type*="button" *id*="calculator4" *class*="calculator\_\_number" *data-num*="4">4</button>  
 <button *type*="button" *id*="calculator5" *class*="calculator\_\_number" *data-num*="5">5</button>  
 <button *type*="button" *id*="calculator6" *class*="calculator\_\_number" *data-num*="6">6</button>  
 <button *type*="button" *id*="calculatorSubtract" *class*="calculator\_\_action" *data-action*="Subtract">-</button>  
 <button *type*="button" *id*="calculator1" *class*="calculator\_\_number" *data-num*="1">1</button>  
 <button *type*="button" *id*="calculator2" *class*="calculator\_\_number" *data-num*="2">2</button>  
 <button *type*="button" *id*="calculator3" *class*="calculator\_\_number" *data-num*="3">3</button>  
 <button *type*="button" *id*="calculatorMultiply" *class*="calculator\_\_action" *data-action*="Multiply">\*</button>  
 <button *type*="button" *id*="calculator0" *class*="calculator\_\_number" *data-num*="0">0</button>  
 <button *type*="button" *id*="calculatorEqual" *class*="calculator\_\_action" *data-equals*="">=</button>  
 <button *type*="button" *id*="calculatorDivide" *class*="calculator\_\_action" *data-action*="Divide">/</button>  
 </div>  
 </div>  
 <script *src*="./js/task\_12.js"></script>  
</body>  
</html>

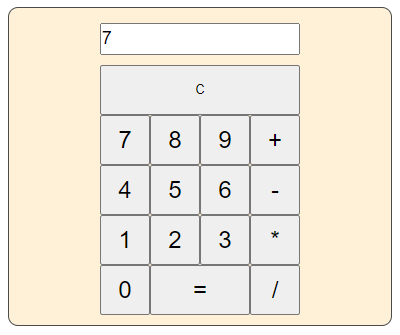
* ***SASS:***

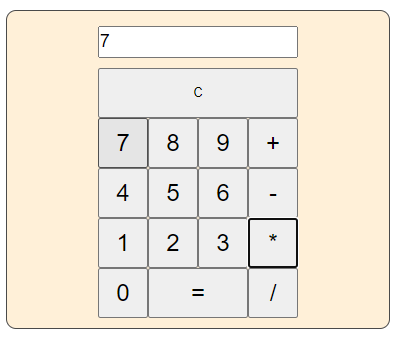
\*  
 margin: 0  
 padding: 0  
 box-sizing: border-box  
*// outline: 1px solid red*.*wrapper* width: 100%  
 height: 100vh  
 display: flex  
 justify-content: center  
 align-items: center  
  
.*container* width: 20%  
 height: 33%  
 display: flex  
 flex-direction: column  
 justify-content: center  
 align-items: center  
 border: 1px solid #4a4a4a  
 border-radius: 10px  
 background-color: #fff0d8  
 padding: 15px  
  
.*container\_\_calculator* width: 57%  
 height: 100%  
 display: flex  
 flex-wrap: wrap  
 justify-content: center  
 align-items: center  
  
 .*calculator\_\_field* width: 100%  
 height: 32px  
 font-size: 18px  
 margin-bottom: 10px  
  
 .*calculator\_\_number*, .*calculator\_\_action* width: 50px  
 height: 50px  
 font-size: 24px  
  
 *#calculatorEqual*, *#calculatorClear* width: 100px  
 height: 50px  
  
 *#calculatorTotalClear* width: 100%  
 height: 50px

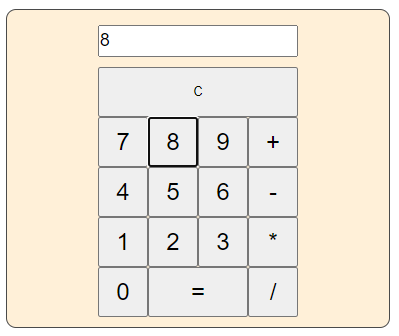
* ***JS:***

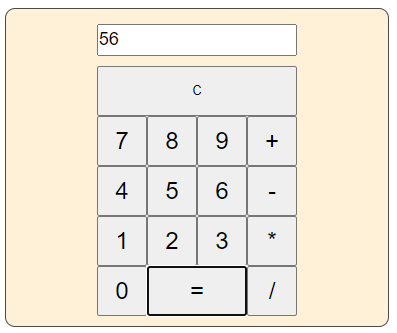
'use strict';  
  
*const getElement* = (*element*) => {  
 *if* (*element*.charAt(0) === "#") {  
 *return* document.querySelector(*element*);  
 }  
  
 *return* document.querySelectorAll(*element*);  
};  
  
*const* calculatorField = *getElement*("#calculatorField");  
*const* calculatorEqual = *getElement*("#calculatorEqual");  
*const* calculatorNumbers = *getElement*(".calculator\_\_number");  
*const* calculatorActions = *getElement*(".calculator\_\_action");  
*let* currentNumber = "", firstNumber = "", resultNumber, operator;  
  
*const setNumber* = *function* () {  
 *if* (resultNumber) {  
 currentNumber = *this*.getAttribute("data-num");  
 resultNumber = "";  
 } *else* {  
 currentNumber += *this*.getAttribute("data-num");  
 }  
  
 calculatorField.value = currentNumber;  
};  
  
*const moveNumber* = *function* () {  
 firstNumber = currentNumber;  
 currentNumber = "";  
 operator = *this*.getAttribute("data-action");  
  
 calculatorEqual.setAttribute("data-result", "");  
};  
  
*const displayNumber* = () => {  
 firstNumber = *parseFloat*(firstNumber);  
 currentNumber = *parseFloat*(currentNumber);  
  
 *switch* (operator) {  
 *case* "Add":  
 resultNumber = firstNumber + currentNumber;  
 *break*;  
  
 *case* "Subtract":  
 resultNumber = firstNumber - currentNumber;  
 *break*;  
  
 *case* "Multiply":  
 resultNumber = firstNumber \* currentNumber;  
 *break*;  
  
 *case* "Divide":  
 resultNumber = firstNumber / currentNumber;  
 *break*;  
  
 *default*:  
 resultNumber = currentNumber;  
 }  
  
 *if* (!*isFinite*(resultNumber)) {  
 *if* (*isNaN*(resultNumber)) {  
 resultNumber = "NaN!";  
 } *else* {  
 resultNumber = "Dividing by zero!";  
 }  
 }  
  
 calculatorField.value = resultNumber;  
 calculatorEqual.setAttribute("data-result", resultNumber);  
  
 firstNumber = 0;  
 currentNumber = resultNumber;  
};  
  
*const clearAll* = () => {  
 firstNumber = "";  
 currentNumber = "";  
 calculatorField.value = "0";  
 calculatorEqual.setAttribute("data-result", resultNumber);  
};  
  
*for* (*let* i = 0; i < calculatorNumbers.length; i++) {  
 calculatorNumbers[i].onclick = *setNumber*;  
}  
  
*for* (*let* i = 0; i < calculatorActions.length; i++) {  
 calculatorActions[i].onclick = *moveNumber*;  
}  
  
calculatorEqual.onclick = *displayNumber*;  
  
*getElement*("#calculatorTotalClear").onclick = *clearAll*;

***Результат програми:***









***Репозиторій лабораторної роботи:***

* https://github.com/AndriiBabushko/Frontend

***Висновок:*** під час виконання лабораторної роботи було отримано навички обробки подій в мові JavaScript.