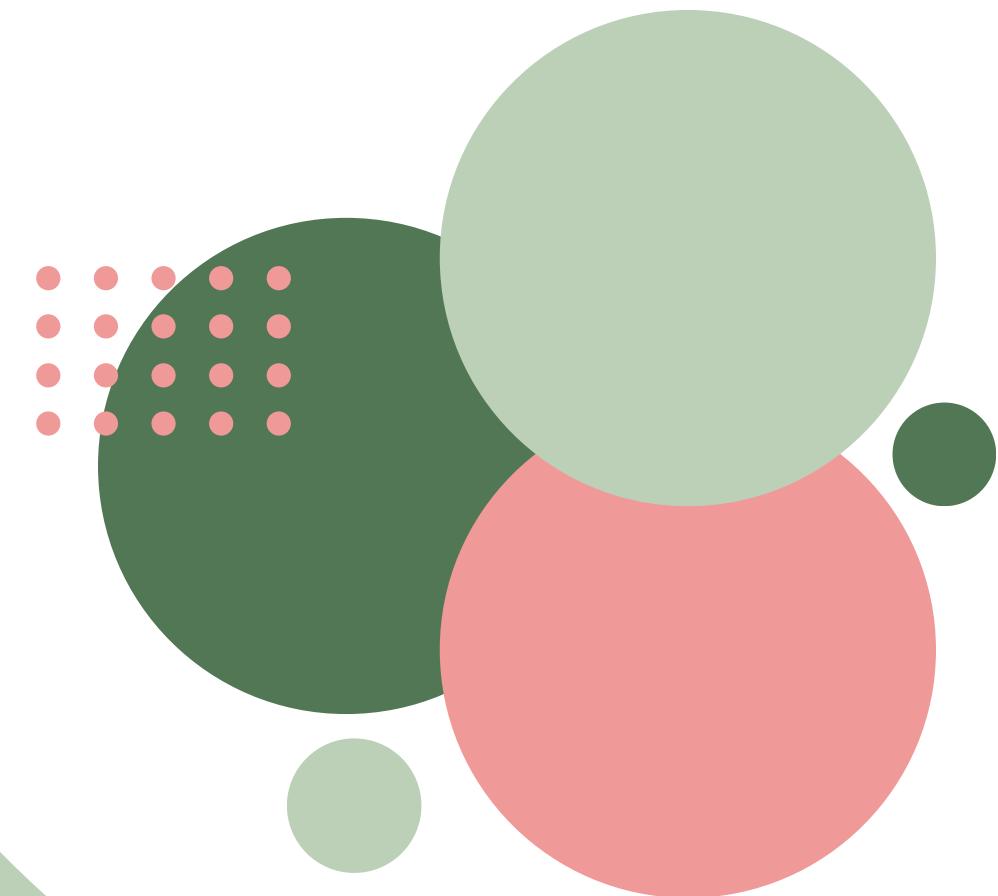


MATRIX CALCULATOR

JO A REUM



INDEX



PLAN

SKETCH

03

DESCRIPTION

04

DESIGN

MOCKUP

05

COLOR & FONT

06

FAVICON & BUTTON

07

PREVIEW

08

DEVELOPMENT

HTML

10

CSS

11

JAVASCRIPT

12

END

REVIEW

19

PLAN

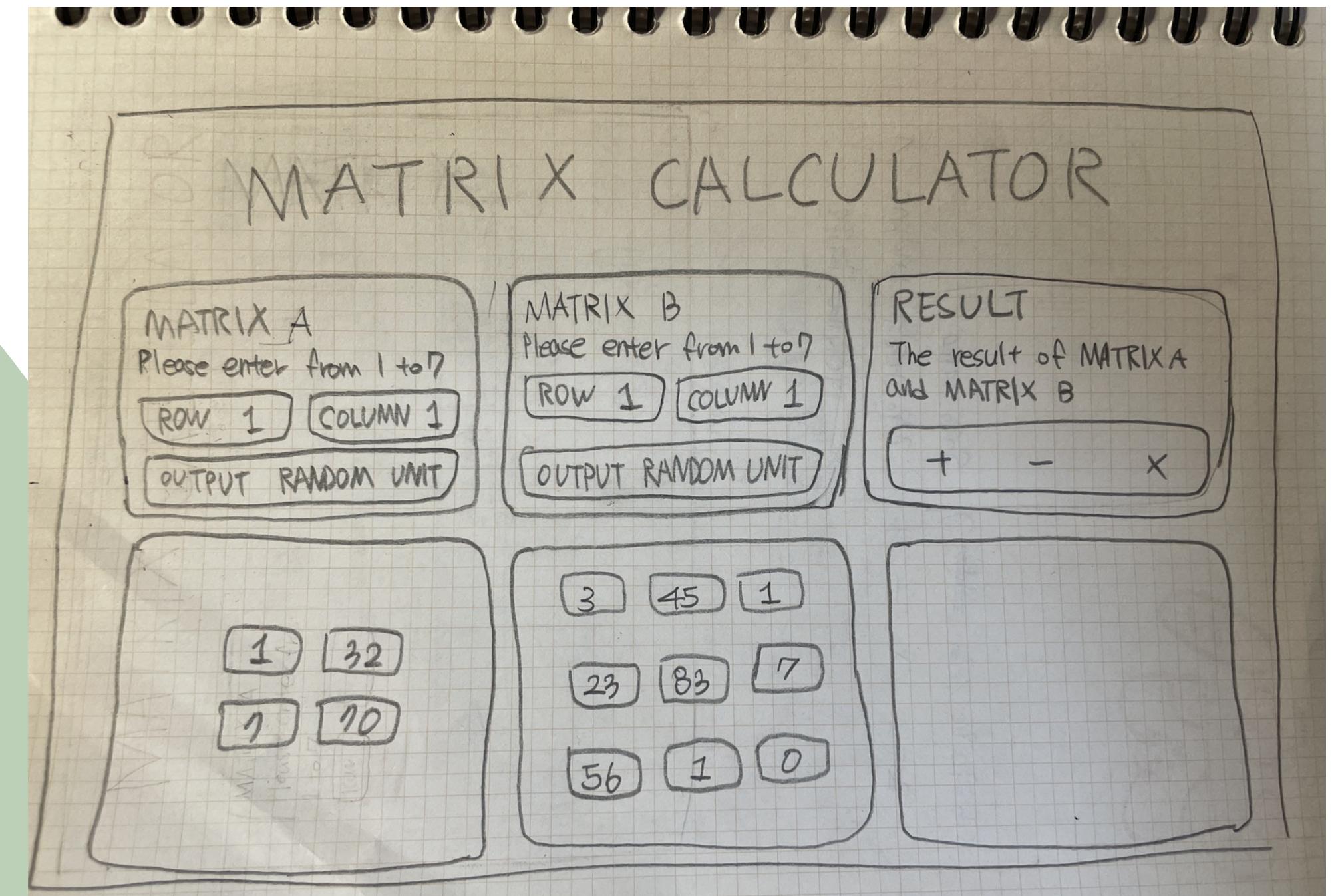
SKETCH

입력하는 공간

행과 열을 입력받고 버튼을 누를 수 있다.

출력하는 공간

입력하는 공간에서 행과 열을 입력받아
값의 개수에 맞게 출력할 수 있다.



DESCRIPTION

입력하는 공간

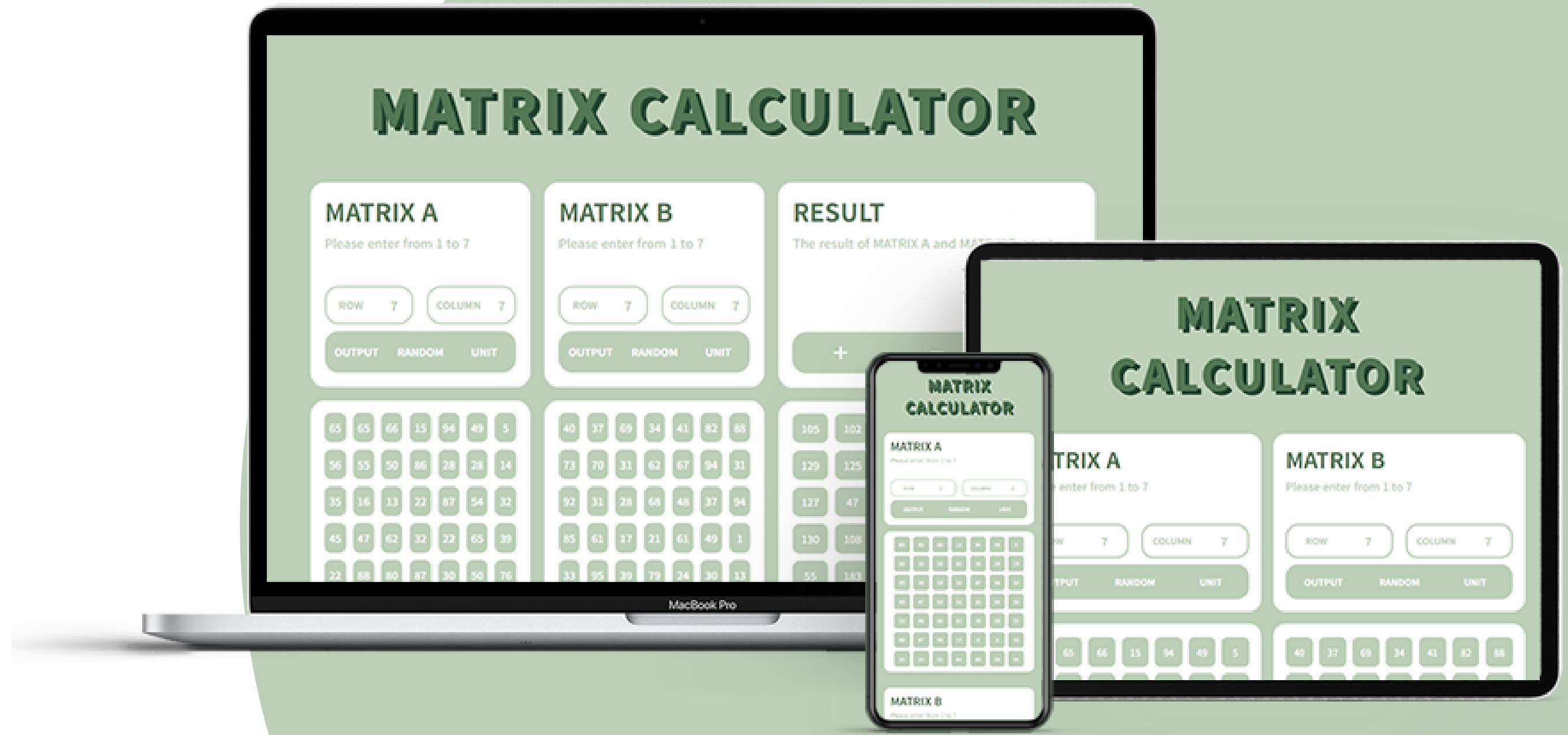
- 행과 열에 각각 1부터 7의 숫자를 입력해야 한다.
- 1부터 7이 아닌 숫자를 입력 시 오류 메시지와 함께 에러가 발생한다.
- UNIT 버튼은 행과 열의 개수가 같아야 하며 틀릴 시 오류 메시지와 에러가 난다.

출력하는 공간

- 입력하는 공간에서 받아온 행과 열의 개수에 맞게 출력한다.
- 각각 계산하고 싶은 값을 0에서 99까지 입력 해야한다.
- 0부터 99까지의 숫자가 아닌 값을 입력하면 오류 메시지와 함께 에러가 발생한다.

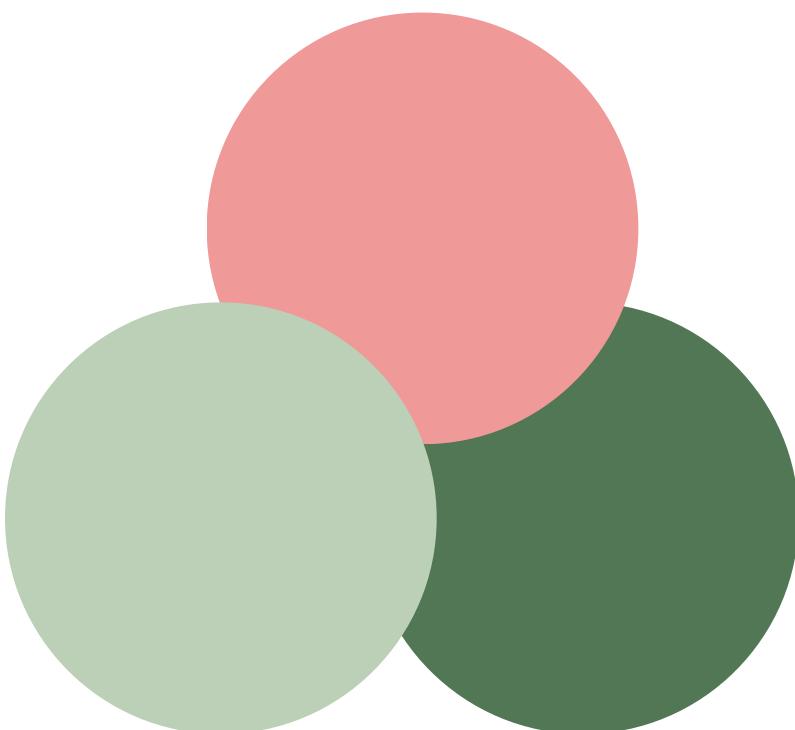
DESIGN

MOCKUP



COLOR & FONT

#F09999



#BCD0B7 #527754

초록색 계열은 가장 안정적인
색으로, 눈을 자극할 우려가
적어 눈의 피로를 풀어주기도 합니다.
눈이 아프지 않도록 초록색 계열로 선택
하였으며, 초록색 계열을 대비해 분홍색 계열로
에러 메시지 색상을 표현할 수 있도록 하였다.

Noto Sans Korean

Noto Sans Korean

Noto Sans Korean

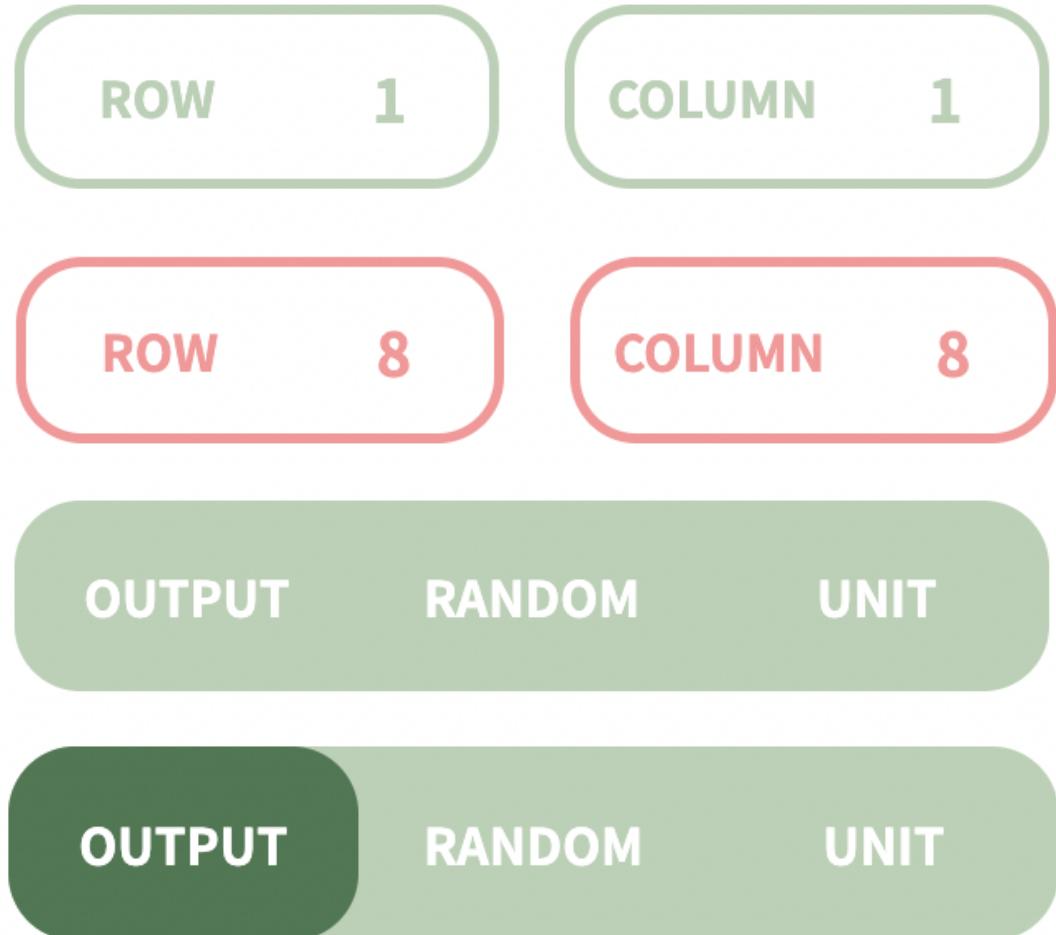
Noto Sans Korean

FAVICON & BUTTON



- 계산기에 맞게 디자인하여 제작

40
100



- 에러 발생 시 분홍색 계열로 변경
- 마우스 및 버튼을 눌렀을 시 진한 초록색으로 변경하여 선택되었음을 표시

PREVIEW

MATRIX CALCULATOR

MATRIX A

Please enter from 1 to 7

ROW 5 COLUMN 5

OUTPUT RANDOM UNIT

74	9	12	43	18
29	20	7	57	49
69	65	55	36	10
60	73	33	97	72
77	32	22	47	84

MATRIX B

Please enter from 1 to 7

ROW 5 COLUMN 5

OUTPUT RANDOM UNIT

70	89	14	0	26
54	9	99	49	15
37	95	24	4	59
57	4	18	50	34
18	44	85	96	5

RESULT

The result of MATRIX A and MATRIX B

+ - ×

144	98	26	43	44
83	29	106	106	64
106	160	79	40	69
117	77	51	147	106
95	76	107	143	89

DESIGN PREVIEW

MATRIX CALCULATOR

MATRIX A
Please enter from 1 to 7

ROW 5 COLUMN 5
OUTPUT RANDOM UNIT

74	9	12	43	18
29	20	7	57	49
69	65	55	36	10
60	73	33	97	72
77	32	22	47	84

MATRIX B
Please enter from 1 to 7

ROW 5 COLUMN 5
OUTPUT RANDOM UNIT

70	89	14	0	26
54	9	99	49	15
37	95	24	4	59
57	4	18	50	34
18	44	85	96	5

RESULT
The result of MATRIX A and MATRIX B

+ - ×

144	98	26	43	44
83	29	106	106	64
106	160	79	40	69
117	77	51	147	106
95	76	107	143	89

MATRIX CALCULATOR

MATRIX A
Please enter from 1 to 7

ROW 5 COLUMN 5
OUTPUT RANDOM UNIT

74	9	12	43	18
29	20	7	57	49
69	65	55	36	10
60	73	33	97	72
77	32	22	47	84

MATRIX B
Please enter from 1 to 7

ROW 5 COLUMN 5
OUTPUT RANDOM UNIT

70	89	14	0	26
54	9	99	49	15
37	95	24	4	59
57	4	18	50	34
18	44	85	96	5

RESULT
The result of MATRIX A and MATRIX B

+ - ×

144	98	26	43	44
83	29	106	106	64
106	160	79	40	69
117	77	51	147	106
95	76	107	143	89

SITE

DEVELOPMENT

HTML

```

<div class="input_list">
  <div id="secondRow" class="input_box">
    <label for="secondRowSize"><span id="secondRowLabel">ROW</span></label>
    <input id="secondRowSize" type="number" name="secondRowSize" min="1" max="7" value="1" step="1" maxlength="1">
  </div>
  <div id="secondColumn" class="input_box">
    <label for="secondRowSize"><span id="secondColumnLabel">COLUMN</span></label>
    <input id="secondColumnSize" type="number" name="secondColumnSize" min="1" max="7" value="1" step="1" maxlength="1">
  </div>
</div>
<div class="btn_box">
  <button id="secondOutputBtn" class="output_btn" type="button"><span>OUTPUT</span></button>
  <button id="secondRandomBtn" class="random_btn" type="button"><span>RANDOM</span></button>
  <button id="secondUnitBtn" class="unit_btn" type="button"><span>UNIT</span></button>
</div>
<div id="secondInputMatrix" class="matrix_box"></div>
</section>

```



- input을 통해 값을 입력할 수 있도록 태그 작성
- button 태그를 작성하여 각 버튼을 눌러 연산할 수 있도록 생성

DEVELOPMENT

CSS

```
/* 해당영역 흔들기 애니메이션 */
.vibration {
    animation: vibration 0.1s infinite;
}

@keyframes vibration {
    from {
        transform: rotate(2deg);
    }
    to {
        transform: rotate(-2deg);
    }
}
```

Please enter from 1 to 7

- 에러가 발생했을 경우 글씨가 흔들리는 현상을 주기 위해 작성

DEVELOPMENT

CSS

- 반응형으로 제작

```
/* Tablet */
@media all and (min-width:768px){
    .title{
        font-size:0.7rem;
        letter-spacing: 0.06rem;
        text-shadow:0.07rem 0.06rem #21432e;
    }

    .matrix_flex{
        display:flex;
        flex-wrap: wrap;
    }

    .matrix_flex .item{
        flex: 1 1 40%;
    }

    .matrix_cal:nth-child(2){
        margin-left:0.2rem;
    }
}
```

```
/* PC */
@media all and (min-width:1132px){

    .title{
        font-size:0.8rem;
    }

    .matrix_flex{
        display:flex;
    }

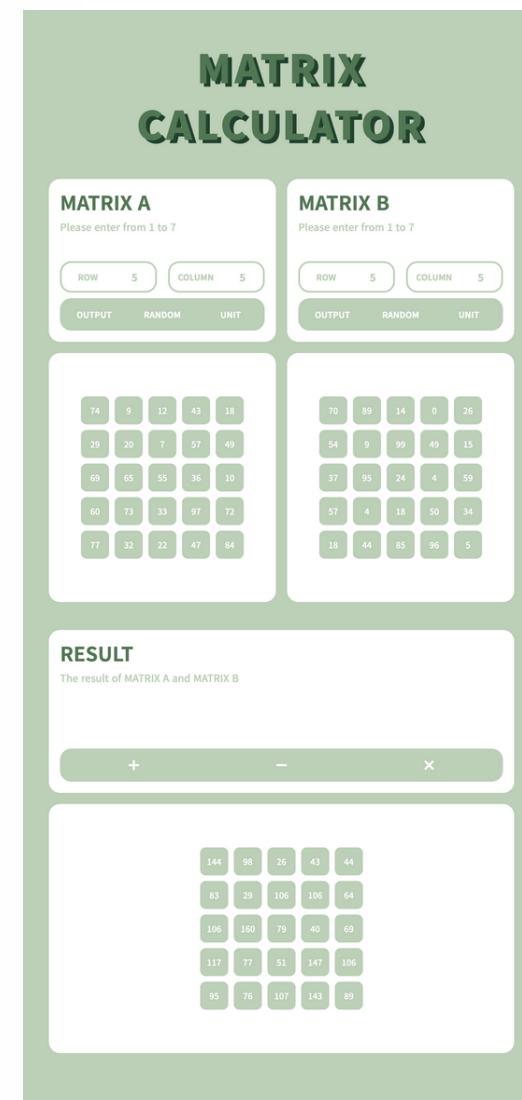
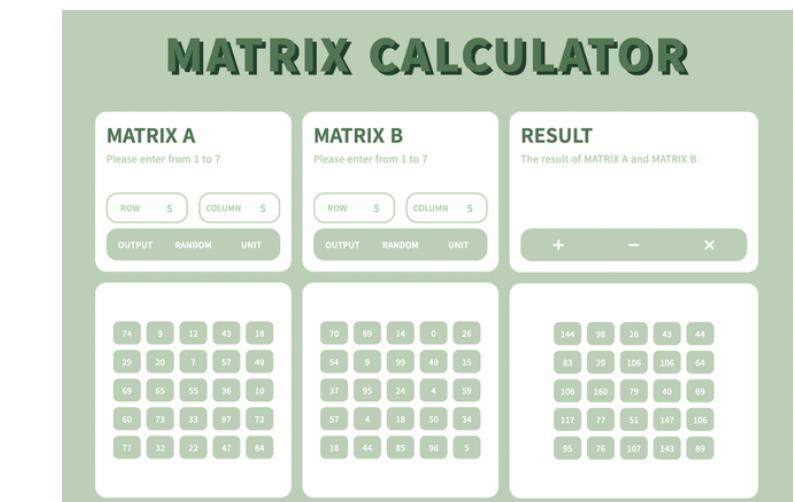
    .matrix_flex .item{
        flex:1;
    }

    .matrix_cal:nth-child(3){
        margin-left:0.2rem;
    }

    .matrix_cal .matrix_box{
        margin-bottom: 0;
        height: 3.9rem;
    }

    .matrix_cal .matrix_box .matrix_center .matrix_input .matrix_result,
    .matrix_cal .matrix_box .matrix_center .matrix_input .div_matrix{
        padding:0.1rem 0;
    }

    .matrix_cal .result_box{
        min-width:4.5rem;
    }
}
```



DEVELOPMENT

JAVASCRIPT

```
function countMatrix(inputMatrix, countRowNumber, countColumnNumber){  
    for(let i=0; i<countRowNumber;i++){  
        inputMatrix[i] = [];  
        for(let j=0;j<countColumnNumber;j++){  
            inputMatrix[i][j] = 0;  
        }  
    }  
}
```

- 행렬을 초기화 시켜주기 위한 코드

DEVELOPMENT

JAVASCRIPT

```

function visualizeMatrix(selectId, matrix){
    let matrixString = '<div class="matrix_center">';
    for(let i=0; i<matrix.length; i++){
        let tempString = '<div class="matrix_input">';
        for(let j=0; j<matrix[i].length; j++){
            tempString += '<input id=""';
            tempString += selectId + "_" + String(i) + "_" + String(j);
            tempString += '" class="matrix_result" type="number" min="0" max="99" value=""';
            tempString += matrix[i][j];
            tempString += '" step="1" maxlength="1">';
        }
        tempString += '</div>';
        matrixString += tempString;
    }
    matrixString += '</div>';
    document.getElementById(selectId).innerHTML = matrixString;
    //console.log(matrix);
}

```

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

- 행렬 수 입력 시 입력한 값을 받아 그 개수에 맞게 칸 생성

DEVELOPMENT

JAVASCRIPT

```
function checkValueAttribute(){
    this.onchange = function(event){
        const inputId = (event.target.id);
        const parseInfo = (event.target.id).split("_");
        const matrixName = parseInfo[0];
        const matrixRowSize = parseInfo[1];
        const matrixColumnSize = parseInfo[2];
        const changedValue = Number(event.target.value);

        switch(matrixName){
            case 'firstInputMatrix':
                firstInputMatrix[matrixRowSize][matrixColumnSize] = changedValue;
                visualizeMatrix("firstInputMatrix", firstInputMatrix);
                if(Number(firstInputMatrix[matrixRowSize][matrixColumnSize]) > 99){
                    firstInputMatrix[matrixRowSize][matrixColumnSize] = 99;
                    visualizeMatrix("firstInputMatrix", firstInputMatrix);
                    errorOutputBox(inputId);
                    setTimeout(function(){
                        resetOutput(inputId);
                    }, 400);
                }
        }
    }
}
```

- 생성된 행렬 값을 추적하고 변경한다면 값을 바꿔주는 행동
- 99 이상의 숫자를 입력하면 99로 강제 입력

DEVELOPMENT

JAVASCRIPT

Please enter from 1 to 7

ROW

8

COLUMN

8

- 1부터 7인 숫자를 입력해야 하며
7 이상의 숫자를 입력할 경우 에러 발생
- 1부터 7의 숫자를 입력했을 경우 행렬
수에 맞게 칸 생성

```

document.getElementById("firstOutputBtn").onclick = function(){
    const firstRowSize = document.getElementById("firstRowSize").value;
    const firstColumnSize = document.getElementById("firstColumnSize").value;

    if(firstRowSize > 7 && firstColumnSize > 7){
        errorInputBox("firstText","firstRow","firstRowLabel", "firstRowSize");
        errorInputBox("firstText","firstColumn","firstColumnLabel", "firstColumnSize");
    }else if(firstRowSize > 7){
        resetColor("firstText","firstColumn","firstColumnLabel", "firstColumnSize");
        errorInputBox("firstText","firstRow","firstRowLabel", "firstRowSize");
    }else if(firstColumnSize > 7){
        resetColor("firstText","firstRow","firstRowLabel", "firstRowSize");
        errorInputBox("firstText","firstColumn","firstColumnLabel", "firstColumnSize");
    }else{
        resetColor("firstText","firstColumn","firstColumnLabel", "firstColumnSize");
        resetColor("firstText","firstRow","firstRowLabel", "firstRowSize");

        firstInputMatrix = [];
        countMatrix(firstInputMatrix, firstRowSize, firstColumnSize);
        visualizeMatrix("firstInputMatrix", firstInputMatrix );

        checkValueAttribute();
    }
}

```

DEVELOPMENT

JAVASCRIPT

```
document.getElementById("plusBtn").onclick = function(){
    const firstRowSize = document.getElementById("firstRowSize").value;
    const firstColumnSize = document.getElementById("firstColumnSize").value;
    const secondRowSize = document.getElementById("secondRowSize").value;
    const secondColumnSize = document.getElementById("secondColumnSize").value;

    resultMatrix = [];
    if(firstInputMatrix.length == 0 || secondInputMatrix.length == 0){
        errorResultBox("resultText", "Please enter MATRIX");
    }else if(firstRowSize != firstInputMatrix.length) || (firstColumnSize != firstInputMatrix[0].length){
        errorResultBox("resultText", "Please enter MATRIX A");
    }else if(secondRowSize != secondInputMatrix.length) || (secondColumnSize != secondInputMatrix[0].length){
        errorResultBox("resultText", "Please enter MATRIX B");
    }else if(firstRowSize == secondRowSize) && (secondColumnSize == firstColumnSize){
        resetResult("resultText", "The result of MATRIX A and MATRIX B");
        countMatrix(resultMatrix, firstRowSize, firstColumnSize);
        console.log(firstInputMatrix.length);
        for(let i=0; i<firstInputMatrix.length;i++){
            for(let j=0; j<firstInputMatrix[i].length;j++){
                resultMatrix[i][j] = firstInputMatrix[i][j] + secondInputMatrix[i][j];
            }
        }
        visualizeResultMatrix("resultMatrix", resultMatrix );
    }
}
```

- 두 행렬을 연산하여 결과에 노출
- 값이 없을 경우 에러
- 두 행렬의 행과 열의 수가 맞아야 연산 가능
- 곱하기 버튼을 누를 경우에는 행렬 A의 행과 행렬 B의 열의 수가 맞아야 연산 가능

DEVELOPMENT

JAVASCRIPT

```
else if(firstRowSize == firstColumnSize){  
    resetColor("firstText", "firstColumn", "firstColumnLabel", "firstColumnSize");  
    resetColor("firstText", "firstRow", "firstRowLabel", "firstRowSize");  
  
    firstInputMatrix = [];  
    countMatrix(firstInputMatrix, firstRowSize, firstColumnSize);  
    for(let i=0;i<firstRowSize;i++){  
        firstInputMatrix[i][i] = 1;  
        visualizeMatrix("firstInputMatrix", firstInputMatrix );  
    }  
}
```

1	0	0
0	1	0
0	0	1

- 단위 행렬은 행과 열의 수가 맞아야 하며 대각선이 1인 행렬
- 행과 열의 수가 맞을 경우 실행되도록 작성

END

REVIEW

행렬 계산기 프로젝트를 진행하면서 자바스크립트 부분에서 연산이 잘 안되거나 오류가 발생하는 일이 생겨 많은 고비가 있었습니다. 곱하기 부분에서 A의 행과 B의 열의 개수도 맞아야 하며 행과 열을 하나씩 곱하여 더하는 방식이다 보니 복잡하게만 생각했습니다. 하지만 차근차근 계산식을 풀어 적어보면서 규칙을 알게 되었고 for 문으로 해결할 수 있었습니다. 이러한 경험을 통해 자바스크립트를 배우는 과정에서 기초적인 부분을 활용하여 계속해서 되짚어보고 복습하며 프로젝트를 진행할 수 있었습니다.

이 프로젝트를 통해 기초적인 부분을 다질 수 있는 기회를 얻었고, 혼자 문제를 해결해 나갈 수 있는 용기와 열정을 더 얻게 되었습니다. 더 나아가서는 조금 더 심화된 자바스크립트를 공부하여 다음 프로젝트는 더욱 발전된 코드로 성장하고 싶습니다.



THANK YOU