BCSE202L-DATA STRUCTURES AND ALGORITHMS

SUDOKU SOLVER

USING RECURSION AND BACKTRACKING

TABLE OF CONTEXT

Here's what we'll cover:

CHAPTER 1

Introduction

French newspapers featured variations of the Sudoku puzzles in the 19th century, and the puzzle has appeared since 1979 in puzzle books under the name Number Place. However, the modern Sudoku only began to gain widespread popularity in 1986 when it was published by the Japanese puzzle company Nikoli under the name Sudoku, meaning "single number”.

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

A typical Sudoku puzzle

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

The solution to the puzzle above

OBJECTIVE

Develop a sudoku validator and sudoku solver using backtracking algorithm

CHAPTER 2

Software Requirements:

1)HTML
2)CSS
3)JAVASCRIPT

Hardware Requirements:

Device with Internet access

CHAPTER 3

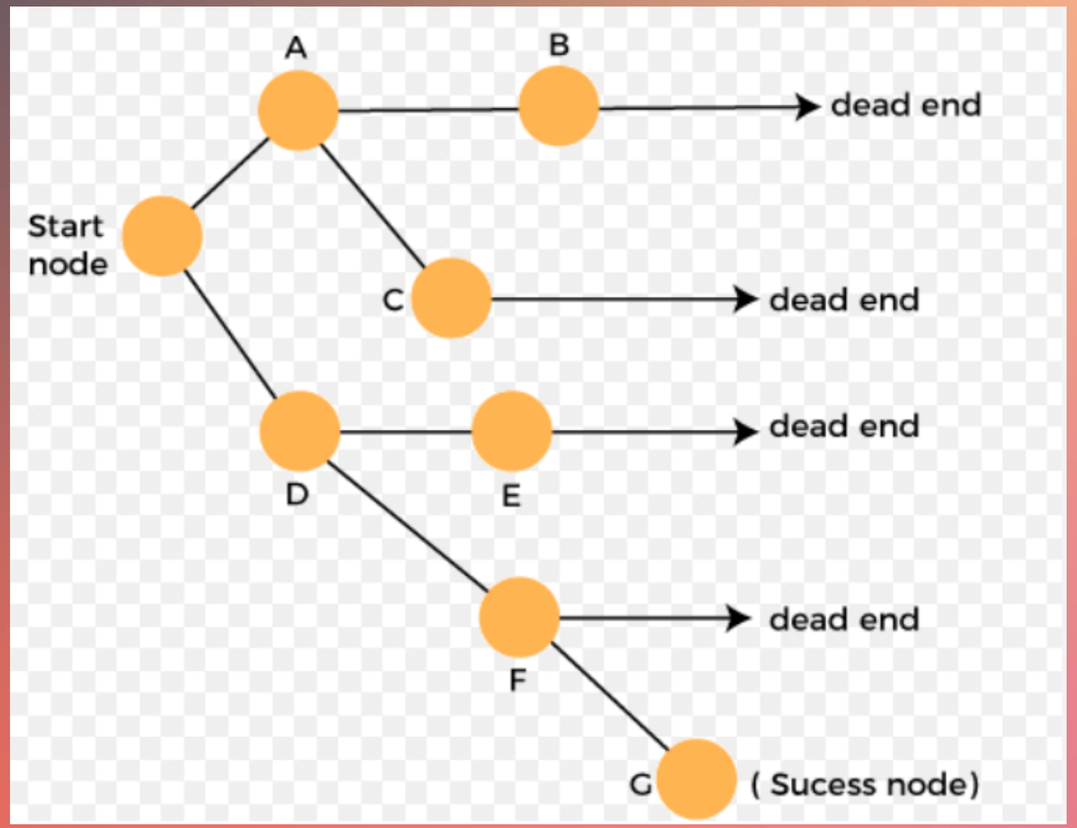
ALGORITHM USED :RECURSION AND BACKTRACKING

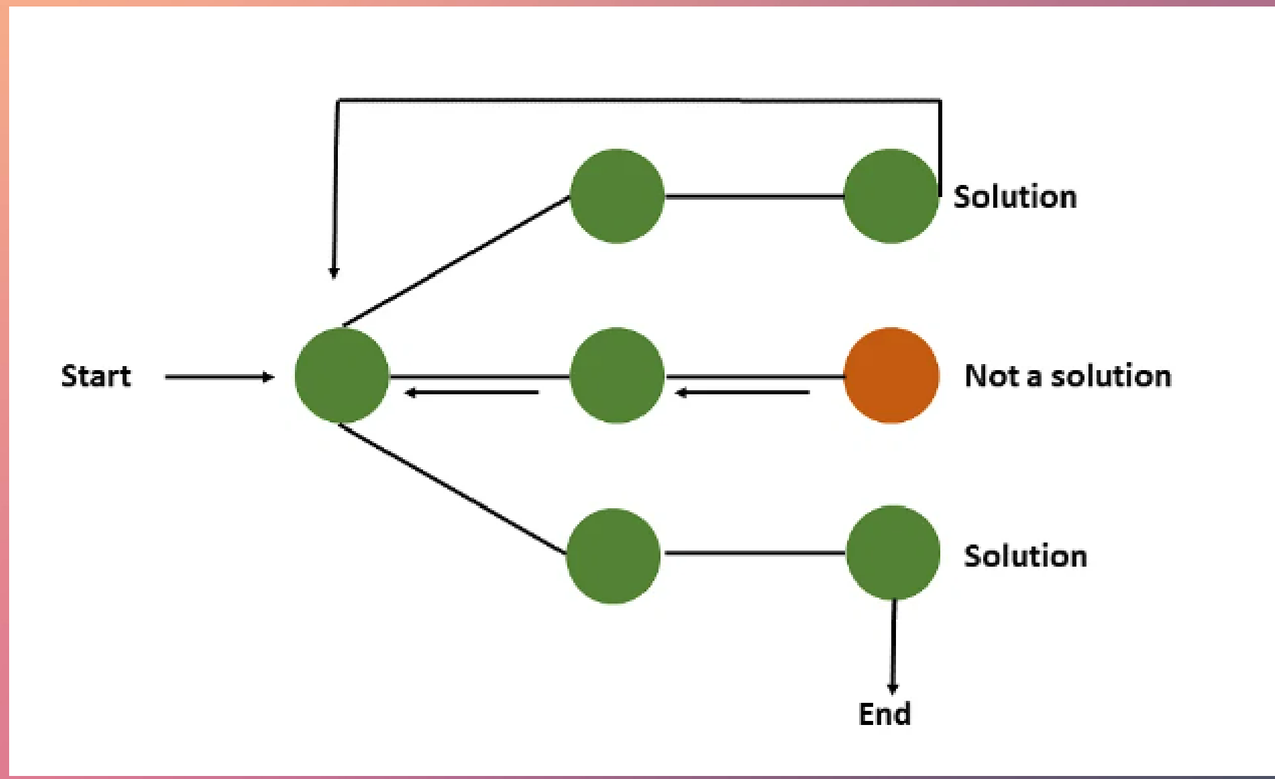
In backtracking algorithms you try to build a solution one step at a time. If at some step it becomes clear that the current path that you are on cannot lead to a solution you go back to the previous step (backtrack) and choose a different path. Briefly, once you exhaust all your options at a certain step you go back.

Approach For Solving Sudoku Using Recursive Backtracking Algorithm

1. Like all other Backtracking problems, we can solve Sudoku by one by one assigning numbers to empty cells.
2. Before assigning a number, we need to confirm that the same number is not present in current row, current column and current 3X3 subgrid.
3. If number is not present in respective row, column or subgrid, we can assign the number, and recursively check if this assignment leads to a solution or not. If the assignment doesnt lead to a solution, then we try next number for current empty cell. And if none of number (1 to 9) lead to solution, we return false.

FLOWCHART FOR BACKTRACKING ALGORITHM





CHAPTER 4

IMPLEMENTATION

```html
<html>
    <head>
        <title>sudoku solver</title>
        <link rel="stylesheet" href="styles.css">
    </head>

    <body>
        <h1 style="color:rgb(255, 170, 170)">SUDOKU SOLVER</h1>
        <table border="1" class="table" id="input_table">
            <tr>
              <td><input size="1" id="r1c1"></input></td>
              <td><input size="1" id="r1c2"></input></td>
              <td style="border-right:solid medium rgb(163, 154, 241)"><input size="1" id="r1c3"></input></td>
              <td><input size="1" id="r1c4"></input></td>
              <td><input size="1" id="r1c5"></input></td>
              <td style="border-right:solid medium rgb(163, 154, 241)"><input size="1" id="r1c6"></input></td>
              <td><input size="1" id="r1c7"></input></td>
              <td><input size="1" id="r1c8"></input></td>
              <td><input size="1" id="r1c9"></input></td>
            </tr>
            <tr>
              <td><input size="1" id="r2c1"></input></td>
              <td><input size="1" id="r2c2"></input></td>
              <td style="border-right:solid medium rgb(163, 154, 241)"><input size="1" id="r2c3"></input></td>
              <td><input size="1" id="r2c4"></input></td>
              <td><input size="1" id="r2c5"></input></td>
              <td style="border-right:solid medium rgb(163, 154, 241)"><input size="1" id="r2c6"></input></td>
              <td><input size="1" id="r2c7"></input></td>
              <td><input size="1" id="r2c8"></input></td>
              <td><input size="1" id="r2c9"></input></td>
            </tr>
```

HTML FOR SKELETON OF THE WEBSITE

CREATING A FORM FIELD TO GET INPUT FROM THE USER

```
32              <tr>
33                <td><input size="1" id="r3c1"></input></td>
34                <td><input size="1" id="r3c2"></input></td>
35                <td style="border-right:solid medium rgb(163, 154, 241)"><input size="1" id="r3c3"></input></td>
36                <td><input size="1" id="r3c4"></input></td>
37                <td><input size="1" id="r3c5"></input></td>
38                <td style="border-right:solid medium rgb(163, 154, 241)"><input size="1" id="r3c6"></input></td>
39                <td><input size="1" id="r3c7"></input></td>
40                <td><input size="1" id="r3c8"></input></td>
41                <td><input size="1" id="r3c9"></input></td>
42              </tr>
43              <tr>
44                <td style="border-top:solid medium rgb(163, 154, 241)"><input size="1" id="r4c1"></input></td>
45                <td style="border-top:solid medium rgb(163, 154, 241)"><input size="1" id="r4c2"></input></td>
46                <td style="border-right:solid medium rgb(163, 154, 241)"><input size="1" id="r4c3"></input></td>
47                <td style="border-top:solid medium rgb(163, 154, 241)"><input size="1" id="r4c4"></input></td>
48                <td style="border-top:solid medium rgb(163, 154, 241)"><input size="1" id="r4c5"></input></td>
49                <td style="border-right:solid medium rgb(163, 154, 241)"><input size="1" id="r4c6"></input></td>
50                <td style="border-top:solid medium rgb(163, 154, 241)"><input size="1" id="r4c7"></input></td>
51                <td style="border-top:solid medium rgb(163, 154, 241)"><input size="1" id="r4c8"></input></td>
52                <td><input size="1" id="r4c9"></input></td>
53              </tr>
54              <tr>
55                <td><input size="1" id="r5c1"></input></td>
56                <td><input size="1" id="r5c2"></input></td>
57                <td style="border-right:solid medium rgb(163, 154, 241)"><input size="1" id="r5c3"></input></td>
58                <td><input size="1" id="r5c4"></input></td>
59                <td><input size="1" id="r5c5"></input></td>
60                <td style="border-right:solid medium rgb(163, 154, 241)"><input size="1" id="r5c6"></input></td>
61                <td><input size="1" id="r5c7"></input></td>
```

```
      <td><input size="1" id="r5c8"></input></td>
      <td><input size="1" id="r5c9"></input></td>
    </tr>
    <tr>
      <td><input size="1" id="r6c1"></input></td>
      <td><input size="1" id="r6c2"></input></td>
      <td style="border-right:solid medium rgb(163, 154, 241)"><input size="1" id="r6c3"></input></td>
      <td><input size="1" id="r6c4"></input></td>
      <td><input size="1" id="r6c5"></input></td>
      <td style="border-right:solid medium rgb(163, 154, 241)"><input size="1" id="r6c6"></input></td>
      <td><input size="1" id="r6c7"></input></td>
      <td><input size="1" id="r6c8"></input></td>
      <td><input size="1" id="r6c9"></input></td>
    </tr>
    <tr>
      <td style="border-top:solid medium rgb(163, 154, 241)"><input size="1" id="r7c1"></input></td>
      <td style="border-top:solid medium rgb(163, 154, 241)"><input size="1" id="r7c2"></input></td>
      <td style="border-right:solid medium rgb(163, 154, 241)"><input size="1" id="r7c3"></input></td>
      <td style="border-top:solid medium rgb(163, 154, 241)"><input size="1" id="r7c4"></input></td>
      <td style="border-top:solid medium rgb(163, 154, 241)"><input size="1" id="r7c5"></input></td>
      <td style="border-right:solid medium rgb(163, 154, 241)"><input size="1" id="r7c6"></input></td>
      <td style="border-top:solid medium rgb(163, 154, 241)"><input size="1" id="r7c7"></input></td>
      <td style="border-top:solid medium rgb(163, 154, 241)"><input size="1" id="r7c8"></input></td>
      <td><input size="1" id="r7c9"></input></td>
    </tr>
    <tr>
      <td><input size="1" id="r8c1"></input></td>
      <td><input size="1" id="r8c2"></input></td>
      <td style="border-right:solid medium rgb(163, 154, 241)"><input size="1" id="r8c3"></input></td>
      <td><input size="1" id="r8c4"></input></td>
```

```
92                <td><input size="1" id="r8c5"></input></td>
93                <td style="border-right:solid medium rgb(163, 154, 241)"><input size="1" id="r8c6"></input></td>
94                <td><input size="1" id="r8c7"></input></td>
95                <td><input size="1" id="r8c8"></input></td>
96                <td><input size="1" id="r8c9"></input></td>
97              </tr>
98              <tr>
99                <td><input size="1" id="r9c1"></input></td>
100               <td><input size="1" id="r9c2"></input></td>
101               <td style="border-right:solid medium rgb(163, 154, 241)"><input size="1" id="r9c3"></input></td>
102               <td><input size="1" id="r9c4"></input></td>
103               <td><input size="1" id="r9c5"></input></td>
104               <td style="border-right:solid medium rgb(163, 154, 241)"><input size="1" id="r9c6"></input></td>
105               <td><input size="1" id="r9c7"></input></td>
106               <td><input size="1" id="r9c8"></input></td>
107               <td><input size="1" id="r9c9"></input></td>
108             </tr>
109           </table>
110           <br>
111           <button type="button" onclick="myFunction()" >SUBMIT</button>
112           <p id="message1" class="message1">   </p>
113
114           <table border="1" class="table" id="output_table">
115             <tr>
116               <td id="ar1c1"></td>
117               <td id="ar1c2"></td>
118               <td id="ar1c3"></td>
119               <td id="ar1c4"></td>
120               <td id="ar1c5"></td>
121               <td id="ar1c6"></td>
```

```
122                 <td id="ar1c7"></td>
123                 <td id="ar1c8"></td>
124                 <td id="ar1c9"></td>
125               </tr>
126               <tr>
127                 <td id="ar2c1"></td>
128                 <td id="ar2c2"></td>
129                 <td id="ar2c3"></td>
130                 <td id="ar2c4"></td>
131                 <td id="ar2c5"></td>
132                 <td id="ar2c6"></td>
133                 <td id="ar2c7"></td>
134                 <td id="ar2c8"></td>
135                 <td id="ar2c9"></td>
136               </tr>
137               <tr>
138                 <td id="ar3c1"></td>
139                 <td id="ar3c2"></td>
140                 <td id="ar3c3"></td>
141                 <td id="ar3c4"></td>
142                 <td id="ar3c5"></td>
143                 <td id="ar3c6"></td>
144                 <td id="ar3c7"></td>
145                 <td id="ar3c8"></td>
146                 <td id="ar3c9"></td>
147               </tr>
148               <tr>
149                 <td id="ar4c1"></td>
150                 <td id="ar4c2"></td>
151                 <td id="ar4c3"></td>
```

LOGIC TO CREATE A TABLE WHICH GETS INPUT AND A TABLE THAT SHOWS OUTPUT

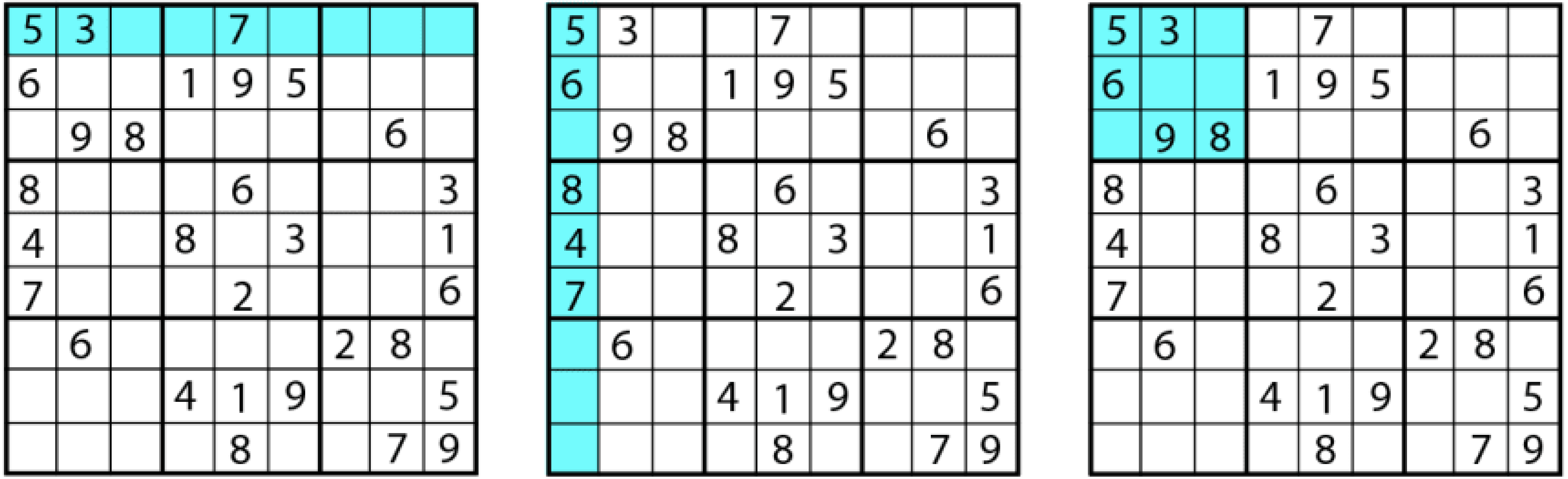
```
152                 <td id="ar4c4"></td>
153                 <td id="ar4c5"></td>
154                 <td id="ar4c6"></td>
155                 <td id="ar4c7"></td>
156                 <td id="ar4c8"></td>
157                 <td id="ar4c9"></td>
158               </tr>
159               <tr>
160                 <td id="ar5c1"></td>
161                 <td id="ar5c2"></td>
162                 <td id="ar5c3"></td>
163                 <td id="ar5c4"></td>
164                 <td id="ar5c5"></td>
165                 <td id="ar5c6"></td>
166                 <td id="ar5c7"></td>
167                 <td id="ar5c8"></td>
168                 <td id="ar5c9"></td>
169               </tr>
170               <tr>
171                 <td id="ar6c1"></td>
172                 <td id="ar6c2"></td>
173                 <td id="ar6c3"></td>
174                 <td id="ar6c4"></td>
175                 <td id="ar6c5"></td>
176                 <td id="ar6c6"></td>
177                 <td id="ar6c7"></td>
178                 <td id="ar6c8"></td>
179                 <td id="ar6c9"></td>
180               </tr>
181               <tr>
```

```
                <td id="ar7c1"></td>
                <td id="ar7c2"></td>
                <td id="ar7c3"></td>
                <td id="ar7c4"></td>
                <td id="ar7c5"></td>
                <td id="ar7c6"></td>
                <td id="ar7c7"></td>
                <td id="ar7c8"></td>
                <td id="ar7c9"></td>
            </tr>
            <tr>
                <td id="ar8c1"></td>
                <td id="ar8c2"></td>
                <td id="ar8c3"></td>
                <td id="ar8c4"></td>
                <td id="ar8c5"></td>
                <td id="ar8c6"></td>
                <td id="ar8c7"></td>
                <td id="ar8c8"></td>
                <td id="ar8c9"></td>
            </tr>
            <tr>
                <td id="ar9c1"></td>
                <td id="ar9c2"></td>
                <td id="ar9c3"></td>
                <td id="ar9c4"></td>
                <td id="ar9c5"></td>
                <td id="ar9c6"></td>
                <td id="ar9c7"></td>
                <td id="ar9c8"></td>
                <td id="ar9c9"></td>
            </tr>
        </table>

        <style>
            body {background-image: url('background2.avif');
            background-repeat: no-repeat;background-size: 100% 100%; text-align:center ;color: floralwhite;}
        </style>
```

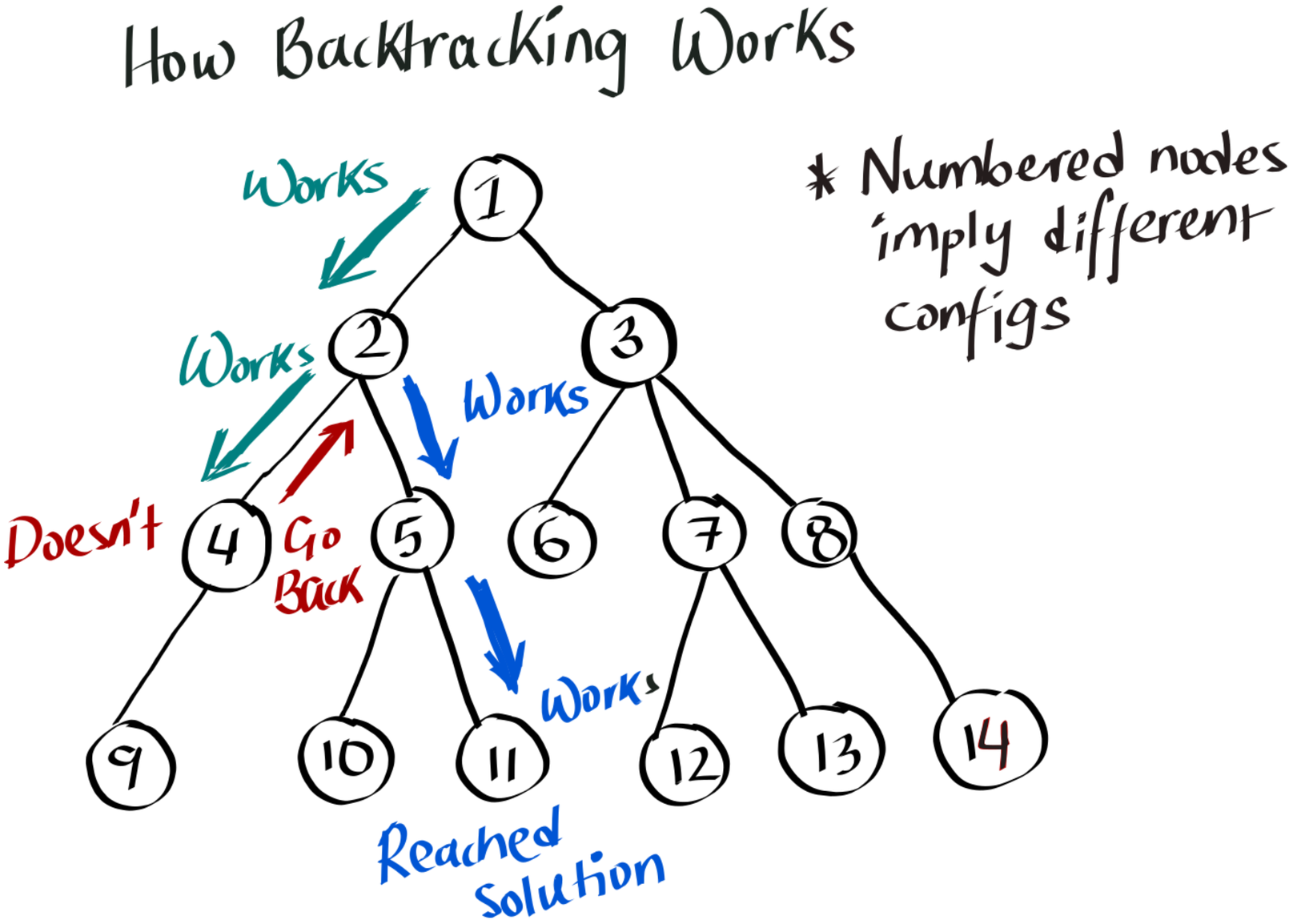
LINKING CSS STYLE TO THE HTML FILE

VALIDATING A SUDOKU



Every number in a row, column and a 9 cell block must be unique from each other respectively.

SUDOKU SOLVING ALGORITHM



```
<script>
  let grid=[[0,0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0,0]];
  function validator(v,row,column,node) {
      for (let i=0;i<9;i++) {
          if (v[row][i]==node) return false;
          if (v[i][column]==node) return false;
          let a=3*(parseInt(row/3))+parseInt(i/3);
          let b=3*(parseInt(column/3))+parseInt(i%3);
          if (v[a][b]==node) return false;
      }
      return true;
  }
  function SolveSudoku() {
      for (let i=0;i<9;i++) {
          for (let j=0;j<9;j++) {
              if (grid[i][j]>=1 && grid[i][j]<=9) {}
              else {
                  for (let c=1;c<10;c++) {
                      if (validator(grid,i,j,c)==true) {
                          grid[i][j]=c;
                          if (SolveSudoku(grid)==true) return true;
                          else grid[i][j]=0;
                      }
                  }
                  return false;
              }
          }
      }
      return true;
  }
```

JAVASCRIPT TO IMPLEMENT THE BACKTRACKING LOGIC

```
250         function myFunction() {
251
252
253             grid[0][0]=document.querySelector("#r1c1").value;
254             grid[0][1]=document.querySelector("#r1c2").value;
255             grid[0][2]=document.querySelector("#r1c3").value;
256             grid[0][3]=document.querySelector("#r1c4").value;
257             grid[0][4]=document.querySelector("#r1c5").value;
258             grid[0][5]=document.querySelector("#r1c6").value;
259             grid[0][6]=document.querySelector("#r1c7").value;
260             grid[0][7]=document.querySelector("#r1c8").value;
261             grid[0][8]=document.querySelector("#r1c9").value;
262
263             grid[1][0]=document.querySelector("#r2c1").value;
264             grid[1][1]=document.querySelector("#r2c2").value;
265             grid[1][2]=document.querySelector("#r2c3").value;
266             grid[1][3]=document.querySelector("#r2c4").value;
267             grid[1][4]=document.querySelector("#r2c5").value;
268             grid[1][5]=document.querySelector("#r2c6").value;
269             grid[1][6]=document.querySelector("#r2c7").value;
270             grid[1][7]=document.querySelector("#r2c8").value;
271             grid[1][8]=document.querySelector("#r2c9").value;
272
273             grid[2][0]=document.querySelector("#r3c1").value;
274             grid[2][1]=document.querySelector("#r3c2").value;
275             grid[2][2]=document.querySelector("#r3c3").value;
276             grid[2][3]=document.querySelector("#r3c4").value;
277             grid[2][4]=document.querySelector("#r3c5").value;
278             grid[2][5]=document.querySelector("#r3c6").value;
279             grid[2][6]=document.querySelector("#r3c7").value;
280             grid[2][7]=document.querySelector("#r3c8").value;
281             grid[2][8]=document.querySelector("#r3c9").value;
282
283             grid[3][0]=document.querySelector("#r4c1").value;
284             grid[3][1]=document.querySelector("#r4c2").value;
285             grid[3][2]=document.querySelector("#r4c3").value;
286             grid[3][3]=document.querySelector("#r4c4").value;
287             grid[3][4]=document.querySelector("#r4c5").value;
288             grid[3][5]=document.querySelector("#r4c6").value;
289             grid[3][6]=document.querySelector("#r4c7").value;
290             grid[3][7]=document.querySelector("#r4c8").value;
291             grid[3][8]=document.querySelector("#r4c9").value;
292
293             grid[4][0]=document.querySelector("#r5c1").value;
294             grid[4][1]=document.querySelector("#r5c2").value;
295             grid[4][2]=document.querySelector("#r5c3").value;
296             grid[4][3]=document.querySelector("#r5c4").value;
297             grid[4][4]=document.querySelector("#r5c5").value;
298             grid[4][5]=document.querySelector("#r5c6").value;
299             grid[4][6]=document.querySelector("#r5c7").value;
300             grid[4][7]=document.querySelector("#r5c8").value;
301             grid[4][8]=document.querySelector("#r5c9").value;
302
303             grid[5][0]=document.querySelector("#r6c1").value;
304             grid[5][1]=document.querySelector("#r6c2").value;
305             grid[5][2]=document.querySelector("#r6c3").value;
306             grid[5][3]=document.querySelector("#r6c4").value;
307             grid[5][4]=document.querySelector("#r6c5").value;
308             grid[5][5]=document.querySelector("#r6c6").value;
309             grid[5][6]=document.querySelector("#r6c7").value;
```

```
                grid[5][7]=document.querySelector("#r6c8").value;
                grid[5][8]=document.querySelector("#r6c9").value;

                grid[6][0]=document.querySelector("#r7c1").value;
                grid[6][1]=document.querySelector("#r7c2").value;
                grid[6][2]=document.querySelector("#r7c3").value;
                grid[6][3]=document.querySelector("#r7c4").value;
                grid[6][4]=document.querySelector("#r7c5").value;
                grid[6][5]=document.querySelector("#r7c6").value;
                grid[6][6]=document.querySelector("#r7c7").value;
                grid[6][7]=document.querySelector("#r7c8").value;
                grid[6][8]=document.querySelector("#r7c9").value;

                grid[7][0]=document.querySelector("#r8c1").value;
                grid[7][1]=document.querySelector("#r8c2").value;
                grid[7][2]=document.querySelector("#r8c3").value;
                grid[7][3]=document.querySelector("#r8c4").value;
                grid[7][4]=document.querySelector("#r8c5").value;
                grid[7][5]=document.querySelector("#r8c6").value;
                grid[7][6]=document.querySelector("#r8c7").value;
                grid[7][7]=document.querySelector("#r8c8").value;
                grid[7][8]=document.querySelector("#r8c9").value;

                grid[8][0]=document.querySelector("#r9c1").value;
                grid[8][1]=document.querySelector("#r9c2").value;
                grid[8][2]=document.querySelector("#r9c3").value;
                grid[8][3]=document.querySelector("#r9c4").value;
                grid[8][4]=document.querySelector("#r9c5").value;
                grid[8][5]=document.querySelector("#r9c6").value;
                grid[8][6]=document.querySelector("#r9c7").value;
                grid[8][7]=document.querySelector("#r9c8").value;
                grid[8][8]=document.querySelector("#r9c9").value;

                SolveSudoku();

                let output_table = document.getElementById('output_table');
                for (let i=0;i<9;i++) {
                    for (let j=0;j<9;j++) {
                        output_table.rows[i].cells[j].innerHTML = grid[i][j];
                    }
                }
            }
        </script>
    </body>
</html>
```

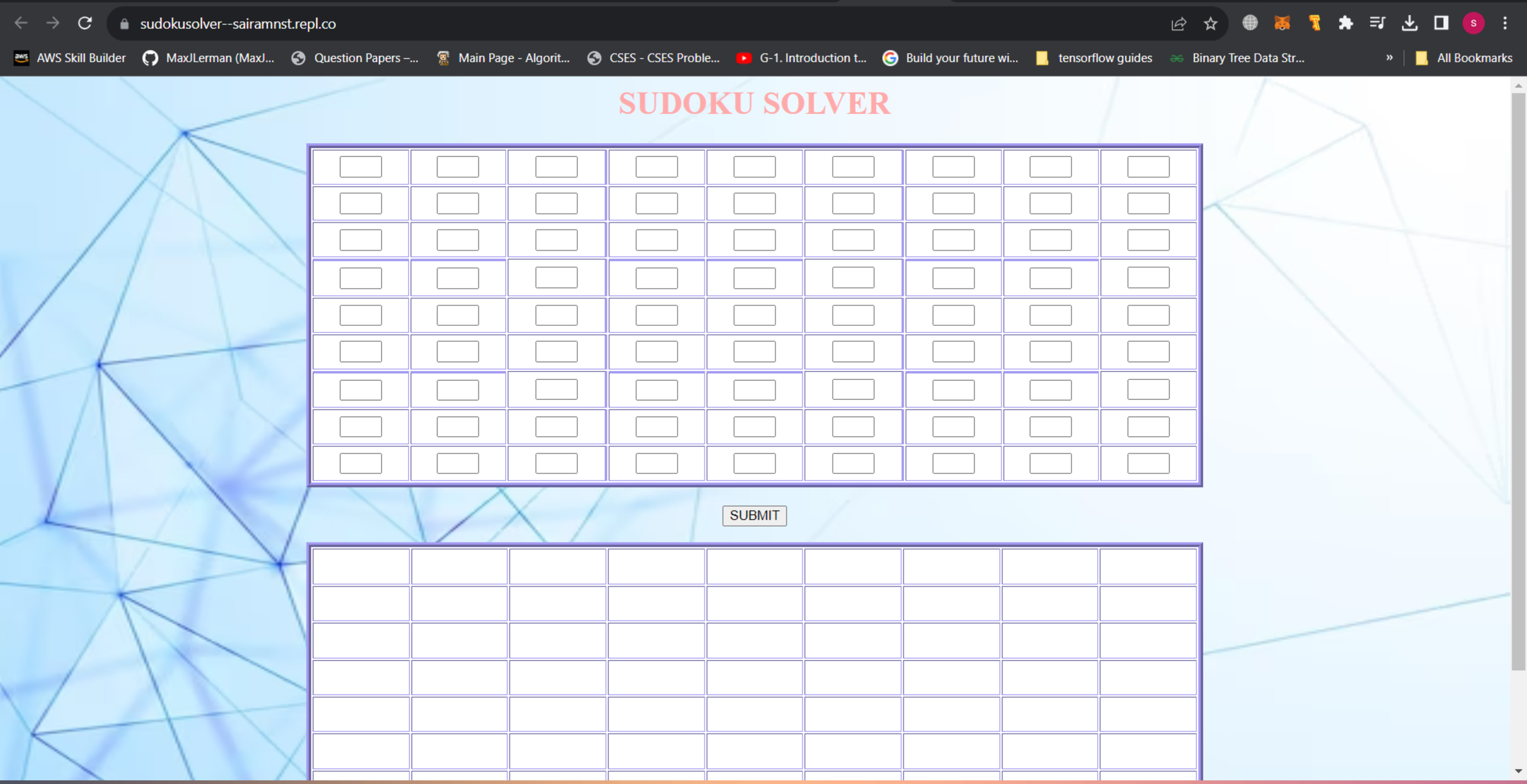
LOGIC TO GET INPUT VALUE FROM WEBSITE TO JAVASCRIPT AND TO PRINT OUTPUT

CSS FILE FOR WEBSITE STYLING

```css
.table{
    margin-left: auto;
    margin-right: auto;
    border: 1;
    border-color: rgb(163, 154, 241);
    border-style:ridge;
    border-width: thick;
    color: black;
    width: 60%;
    height: 50%;
    background-color: white;
    text-align: center;

}


.message1 {
    color:rgb(0, 0, 0);
    font-weight: bold;
}
```



INITIAL WEBPAGE

Hard Sudoku 8

4						3		7
		9		8			2	
	5		3		2			6
6	9		7				5	
		3				7		8
					5			
					6		3	
		5		7	9			2
2	4					8		

4	6	2	5	9	1	3	8	7
1	3	9	6	8	7	4	2	5
7	5	8	3	4	2	1	9	6
6	9	1	7	3	8	2	5	4
5	2	3	9	6	4	7	1	8
8	7	4	2	1	5	9	6	3
9	8	7	4	2	6	5	3	1
3	1	5	8	7	9	6	4	2
2	4	6	1	5	3	8	7	9

SAMPLE SUDOKU GAME

4						3		7
		9		8			2	
	5		3		2			6
6	9		7				5	
		3				7		8
					5			
					6		3	
		5		7	9			2
2	4					8		

SUBMIT

GIVE IN THE INPUT FROM THE SAMPLE PROBLEM

4	2	8	5	6	1	3	9	7
3	6	9	4	8	7	1	2	5
7	5	1	3	9	2	4	8	6
6	9	4	7	1	8	2	5	3
5	1	3	9	2	4	7	6	8
8	7	2	6	3	5	9	1	4
9	8	7	2	4	6	5	3	1
1	3	5	8	7	9	6	4	2
2	4	6	1	5	3	8	7	9

AFTER CLICKING THE SUBMIT BUTTON, WE CAN SEE THE ANSWER TO THE QUESTION ENTERED

CHAPTER 5

RESULT:

The development of the Sudoku solver was a comprehensive endeavor that involved a combination of backtracking algorithm using Javascript and HTML and CSS for the UI. The primary objective was to create an sudoku solver which is capable of solving within few seconds. Thus a sudoku solver can mimic human-like strategic thinking

CHAPTER 6

CONCLUSION

The AI sudoku solver, driven by the backtracking algorithm is a remarkable demonstration of computational intelligence. It excels in finding optimal moves while navigating the intricate complexities of sudoku. The efficiency and strategic depth of the bot are a testament to the power of algorithmic thinking and optimization in the world of competitive gaming showcasing the continuous evolution of AI-driven game-playing systems.

FUTURE ENHANCEMENTS

Explore the integration of machine learning techniques, such as deep reinforcement learning, to enable the bot to learn and adapt dynamically during gameplay.

Liked our project?

Check out the code in our GitHUB repository: https://github.com/sairamnst/sudokusolver

Check out our website hosted on repl: https://sudokusolver--sairamnst.repl.co/

BIBLIOGRAPHY

STRIVER’S DSA SERIES FOR RECURSION AND BACKTRACKING: https://www.youtube.com/watch?v=FWAIf_EVUKE&pp=ygUVc3RyaXZlciBzdWRva3Ugc29sdmVy

Thank you!

M.SAIRAM 22BPS1081 | P.ARVIND 22BPS1086