Sudoku Game

A
Mini Project Report

Submitted in partial fulfilment of the
Requirements for the award of the Degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE & ENGINEERING

By

P Lalith **1602-19-733-021**

D Midhun **1602-19-733-025**

Department of Computer Science & Engineering

Vasavi College of Engineering (Autonomous)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad-31

2021

Vasavi College of Engineering (Autonomous)

(Affiliated to Osmania University)

Hyderabad-500 031

Department of Computer Science & Engineering

BONAFIDE CERTIFICATE

This is to certify that the project entitled **"Sudoku Game"** being submitted by **Midhun Danda and Lalith Srinivas,** bearing **1602-19-733-025 and 1602-19-733-021** in partial fulfilment of the requirements for the award of the degree of Bachelor of Engineering in Computer Science & Engineering is a record of bonafide work carried out by him/her under my guidance.

Dr. T. Adilakshmi,

Professor & HOD,

Dept. of CSE,

ACKNOWLEDGEMENT

We would like to express our heartfelt gratitude to Mr. Sriperambuduri Vinay Kumar, our project guide, for her valuable guidance and constant support, along with her capable instructions and persistent encouragement.

We are grateful to our Head of Department, Dr. T. Adilakshmi, for her steady support and the provision of every resource required for the completion of this project.

We would like to take this opportunity to thank our Principal, Dr. S. V. Ramana, as well as the management of the institute, for having designed an excellent learning atmosphere.

We are thankful to and fortunate enough to get constant encouragement, support and guidance which helped us in successfully completing our project work.

ABSTRACT

Sudoku game consists of a 9×9 board with numbers and blanks on it. The goal is to fill the blank spaces with suitable numbers. These numbers can be filled keeping in mind some rules. The rule for filling these empty spaces is that the number should not appear in the same row, same column or in the same 3×3 grid. In this project, we will build an automatic sudoku solver having sum of rows, columns, local square to be same. This model can be used on harder puzzles as well as sufficient trainings. Ordinary methods of brute force would become completely inefficient for solving. So, it is essential to optimize methods for solving this problem inorder to save on time and space.

TABLE OF CONTENTS

1.INTRODUCTION

Basically, Sudoku contains varied sizes of grid such as 4x4 grids, 8x8 grids up to 100x100 grids. but 9x9 grids are the foremost a la mode among others. There are many types of Sudoku that have been invented such as the normal Sudoku, Geometry Sudoku, Diagonal Sudoku and others.

The end result of the video games is associated with the set of rules of locating the answers to the puzzle. This paper recognition to 9x9 grids Sudoku and Java programming language could be used to solve the 9x9 grids of Sudoku. It can help many people to solve Sudoku by using Java language. Therefore, the objective of this paper is to construct a Sudoku Using backtracking program using HTML, CSS, JavaScript, PHP.

Rules:

Solving a Sudoku puzzle may be as a substitute tricky; however, the policies of the game are pretty easy. Solving a sudoku puzzle does now no longer require knowledge of mathematics; easy good judgment suffices. The goal of sudoku is to input a digit from 1 thru nine in every cell, in one of these manners that:

1 - Each horizontal row includes every digit precisely once.

2 - Each vertical column includes every digit exactly once.

3 - Each sub grid or place includes every digit precisely once.

2.METHODOLOGY

The brute force approach or algorithm which traverses through all the vacant cells in random order to fill the numbers in a sequential manner or backtracking when the number is found to be not valid. This is the process that is repeated until the value is allocated in the last cell(81st) is discovered.

In our project the user is first greeted with the sign-up page. The users can sign up by filling all the required details. The signup page is connected to the database using php script. After successful registration, the user is redirected to the home page where they can see the three labels containing name of the player, level of the game and the button to start the game. The user must choose the level of the game we want to play and click the new game button for starting the game.

Upon clicking the button for the new game, the user will face the game page where a 9X9 grid with some of the cells filled up. The user must fill the remaining empty cells to finish the game. We will have timer start off. We will also have few buttons to pause and stop the game. Upon finishing the last cell of the grid, the user will be redirected to the result page where he will be congratulated, and the total time taken for finishing the game is shown. Along with two other buttons which are for starting a new game and submitting his record for leader board evaluation. If the user has clicked the new game button, he will be redirected to the home page where he can start the new game after selecting level of the game else if the user has selected to submit his information, he will come across a form for filling up the information required. Upon submitting the form, the information is collected into the database. All the actions are done using JS.

3.IMPLEMENTATION

INDEX.HTML

```
<!DOCTYPE html>
<html lang="en">

<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <!-- mobile status bar color -->
   <meta name="theme-color" content="#fff">
   <title>Sudoku</title>
   <link rel="shortcut icon" href="./static/images/icon.png" type="image/x-icon">
   <!-- google font -->
   <link rel="preconnect" href="https://fonts.googleapis.com">
   <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
   <link href="https://fonts.googleapis.com/css2?family=Potta+One&display=swap"
rel="stylesheet">
   <!-- box icons -->
   <link href='https://unpkg.com/boxicons@2.0.9/css/boxicons.min.css'
rel='stylesheet'>
   <!-- app css -->
   <link rel="stylesheet" href="./app.css">

</head>

<body>

   <!-- top nav -->
   <nav>
      <div class="nav-container">
         <a href="#" class="nav-logo">
            Sudoku
         </a>
         <div class="dark-mode-toggle" id="dark-mode-toggle">
            <i class="bx bxs-sun"></i>
            <i class="bx bxs-moon"></i>
         </div>
      </div>
   </nav>
   <!-- end top nav -->
```

```
    <!-- main -->
    <div class="main">
        <div class="screen">
            <!-- start screen -->
            <div class="start-screen active" id="start-screen">
                <input type="text" placeholder="Your name" maxlength="11"
class="input-name" id="input-name">
                <div class="btn" id="btn-level">
                    Easy
                </div>
                <div class="btn" id="btn-continue">Continue</div>
                <div class="btn btn-blue" id="btn-play">New game</div>
            </div>
            <!-- end start screen -->



            <!-- game screen -->
            <div class="main-game" id="game-screen">
                <div class="main-sudoku-grid">
                    <!-- 81 cells -->
                    <div class="main-grid-cell"></div>
                    <div class="main-grid-cell"></div>
                    <div class="main-grid-cell"></div>
                    <div class="main-grid-cell"></div>
                    <div class="main-grid-cell"></div>
                    <div class="main-grid-cell"></div>
                    <div class="main-grid-cell"></div>
                    <div class="main-grid-cell"></div>
                    <div class="main-grid-cell"></div>
                    <div class="main-grid-cell"></div>
                    <div class="main-grid-cell"></div>
                    <div class="main-grid-cell"></div>
                    <div class="main-grid-cell"></div>
                    <div class="main-grid-cell"></div>
                    <div class="main-grid-cell"></div>
                    <div class="main-grid-cell"></div>
                    <div class="main-grid-cell"></div>
                    <div class="main-grid-cell"></div>
                    <div class="main-grid-cell"></div>
                    <div class="main-grid-cell"></div>
                    <div class="main-grid-cell"></div>
                    <div class="main-grid-cell"></div>
                    <div class="main-grid-cell"></div>
```

```
<div class="main-grid-cell"></div>
<div class="main-grid-cell"></div>
<div class="main-grid-cell"></div>
<div class="main-grid-cell"></div>
<div class="main-grid-cell"></div>
<div class="main-grid-cell"></div>
<div class="main-grid-cell"></div>
<div class="main-grid-cell"></div>
<div class="main-grid-cell"></div>
<div class="main-grid-cell"></div>
<div class="main-grid-cell"></div>
<div class="main-grid-cell"></div>
<div class="main-grid-cell"></div>
<div class="main-grid-cell"></div>
<div class="main-grid-cell"></div>
<div class="main-grid-cell"></div>
<div class="main-grid-cell"></div>
<div class="main-grid-cell"></div>
<div class="main-grid-cell"></div>
<div class="main-grid-cell"></div>
<div class="main-grid-cell"></div>
<div class="main-grid-cell"></div>
<div class="main-grid-cell"></div>
<div class="main-grid-cell"></div>
<div class="main-grid-cell"></div>
<div class="main-grid-cell"></div>
<div class="main-grid-cell"></div>
<div class="main-grid-cell"></div>
<div class="main-grid-cell"></div>
<div class="main-grid-cell"></div>
<div class="main-grid-cell"></div>
<div class="main-grid-cell"></div>
<div class="main-grid-cell"></div>
<div class="main-grid-cell"></div>
<div class="main-grid-cell"></div>
<div class="main-grid-cell"></div>
<div class="main-grid-cell"></div>
<div class="main-grid-cell"></div>
<div class="main-grid-cell"></div>
<div class="main-grid-cell"></div>
<div class="main-grid-cell"></div>
<div class="main-grid-cell"></div>
<div class="main-grid-cell"></div>
<div class="main-grid-cell"></div>
```

```
    <div class="main-grid-cell"></div>
    <div class="main-grid-cell"></div>
    <div class="main-grid-cell"></div>
    <div class="main-grid-cell"></div>
    <div class="main-grid-cell"></div>
    <div class="main-grid-cell"></div>
    <div class="main-grid-cell"></div>
    <div class="main-grid-cell"></div>
    <div class="main-grid-cell"></div>
    <div class="main-grid-cell"></div>
    <div class="main-grid-cell"></div>
    <div class="main-grid-cell"></div>
    <div class="main-grid-cell"></div>
    <div class="main-grid-cell"></div>
</div>

<div class="main-game-info">
    <div class="main-game-info-box main-game-info-name">
        <span id="player-name">tuat</span>
    </div>
    <div class="main-game-info-box main-game-info-level">
        <span id="game-level">Easy</span>
    </div>
</div>

<div class="main-game-info-box main-game-info-time">
    <span id="game-time">10:20</span>
    <div class="pause-btn" id="btn-pause">
        <i class="bx bx-pause"></i>
    </div>
</div>

<div class="numbers">
    <div class="number">1</div>
    <div class="number">2</div>
    <div class="number">3</div>
    <div class="number">4</div>
    <div class="number">5</div>
    <div class="number">6</div>
    <div class="number">7</div>
    <div class="number">8</div>
    <div class="number">9</div>
    <div class="delete" id="btn-delete">X</div>
</div>
```

```
            </div>
            <!-- end game screen -->

            <!-- pause screen -->
            <div class="pause-screen" id="pause-screen">
                <div class="btn btn-blue" id="btn-resume">Resume</div>
                <div class="btn" id="btn-new-game">New game</div>
            </div>
            <!-- end pause screen -->

            <!-- result screen -->
            <div class="result-screen" id="result-screen">
                <div class="congrate">Completed</div>
                <div class="info">Time</div>
                <div id="result-time"></div>
                <div class="btn" id="btn-new-game-2">New game</div>
            </div>
            <!-- end result screen -->
        </div>
    </div>
    <!-- end main -->

    <script src="./static/js/constant.js"></script>
    <script src="./static/js/sudoku.js"></script>
    <script src="./static/js/app.js"></script>
</body>

</html>
```

SIGNUP.HTML

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <style>
            .top-popup1{
                position: absolute;
                top: 50%;
                left: 50%;
                transform: translate(-50%,-50%);
```

```
        width: 400px;
        border-radius: 10px;
}
    .top-popup1 h1{
        text-align: center;
        padding: 0 0 20px 0;
        border-bottom: 1px solid silver;
        color: #00aeef;
    }
    .top-popup1 form{
        padding: 0 40px;
        box-sizing: border-box;
    }
    .top-popup1.txt-field{
         position: relative;
         border-bottom: 2px solid #adadad;
         margin: 30px 0;
    }
    .txt-field input{
        width: 100%;
        padding: 0 5px;
        height: 40px;
        font-size: 16px;
        border: none;
        background: none;
        outline: none;
    }
    body{
        background-color: white;
        margin:0;
        padding:0;
        font-family: "Potta One", cursive;
        height: 100vh;
```

```
    overflow: hidden;
}
.txt-field span::before{
    content: '';
    position:absolute;
    top: 40px;
    left: 0;
    width: 0%;
    height: 2px;
    background: #2691d9;
    transition: .5s;
}
.txt-field input:focus ~ label,
.txt-field input:valid ~ label{
    top: -5px;
    color: #2691d9;
}
.txt-field input:focus ~ span::before,
.txt-field input:valid ~ span::before{
    width: 100%;
}
.btn{
    width: 100%;
    height: 50px;
    border: 1px solid;
    background: white;
    border-radius: 25px;
    font-size: 18px;
    color: #00aeef;
    font-weight: 700;
    cursor: pointer;
    outline: none;
}
```

```
        .btn.hover{
            border-color: #00aeef;
            transition: .5s;
        }
    </style>
  </head>
  <body>
    <div class="top-popup1">
      <form method="post" action="connect.php">
        <h1>SignUp</h1>
        <div class="txt-field">
          <span></span>
          <input type="text" placeholder="Enter First Name" name="name"
class="btn"><br><br>
        </div>
        <div class="txt-field">
          <span></span>
          <input type="text" placeholder="Enter Email" name="email" class="btn"
><br><br>
        </div>
        <div class="txt-field">
          <span></span>
          <input type="password" placeholder="Enter Password" name="password"
class="btn"><br><br>
        </div>
          <button type="submit" class="btn"><a href="" style="text-decoration:none;
color:#00aeef;">Sign Up</a></button><br><br>
          <button type="button" class="btn"
onclick="closeForm()">Close</button><br>
        </form>
    </div>
  </body>
</html>
```

End.html

```
<!DOCTYPE html>
<html lang="en">
   <head>
      <style>
         .top-popup1{
            position: absolute;
            top: 50%;
            left: 50%;
            transform: translate(-50%,-50%);
            width: 400px;
            border-radius: 10px;
       }
         .top-popup1 h1{
            text-align: center;
            padding: 0 0 20px 0;
            border-bottom: 1px solid silver;
            color: #00aeef;
         }
         .top-popup1 form{
            padding: 0 40px;
            box-sizing: border-box;
         }
         .top-popup1.txt-field{
             position: relative;
             border-bottom: 2px solid #adadad;
             margin: 30px 0;
         }
         .txt-field input{
            width: 100%;
            padding: 0 5px;
            height: 40px;
            font-size: 16px;
```

```
    border: none;
    background: none;
    outline: none;
}
body{
    background-color: white;
    margin:0;
    padding:0;
    font-family: "Potta One", cursive;
    height: 100vh;
    overflow: hidden;
}
.txt-field span::before{
    content: '';
    position:absolute;
    top: 40px;
    left: 0;
    width: 0%;
    height: 2px;
    background: #2691d9;
    transition: .5s;
}
.txt-field input:focus ~ label,
.txt-field input:valid ~ label{
    top: -5px;
    color: #2691d9;
}
.txt-field input:focus ~ span::before,
.txt-field input:valid ~ span::before{
    width: 100%;
}
.btn{
    width: 100%;
```

```
            height: 50px;
            border: 1px solid;
            background: white;
            border-radius: 25px;
            font-size: 18px;
            color: #00aeef;
            font-weight: 700;
            cursor: pointer;
            outline: none;
        }
        .btn.hover{
            border-color: #00aeef;
            transition: .5s;
        }
    </style>
  </head>
 <body>
    <div class="top-popup1">
        <form method="post" action="connect2.php">
          <h1>User_Form</h1>
          <div class="txt-field">
             <span></span>
          <input type="text" placeholder="Enter Name" name="User_name"
class="btn"><br><br>
        </div>
        <div class="txt-field">
          <span></span>
          <input type="text" placeholder="Enter Email" name="Email" class="btn"
><br><br>
        </div>
        <div class="txt-field">
          <span></span>
```

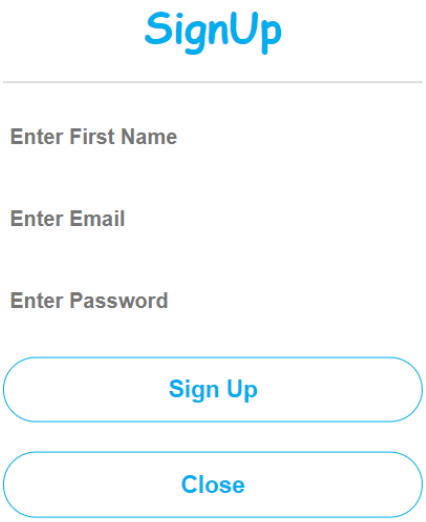
```
            <input type="time" placeholder="Enter Time" name="Time"
class="btn"><br><br>
        </div>
            <button type="submit" class="btn"><a href="index.html" style="text-
decoration:none; color:#00aeef;">Submit</a></button><br><br>
          </form>
      </div>
   </body>
</html>
```

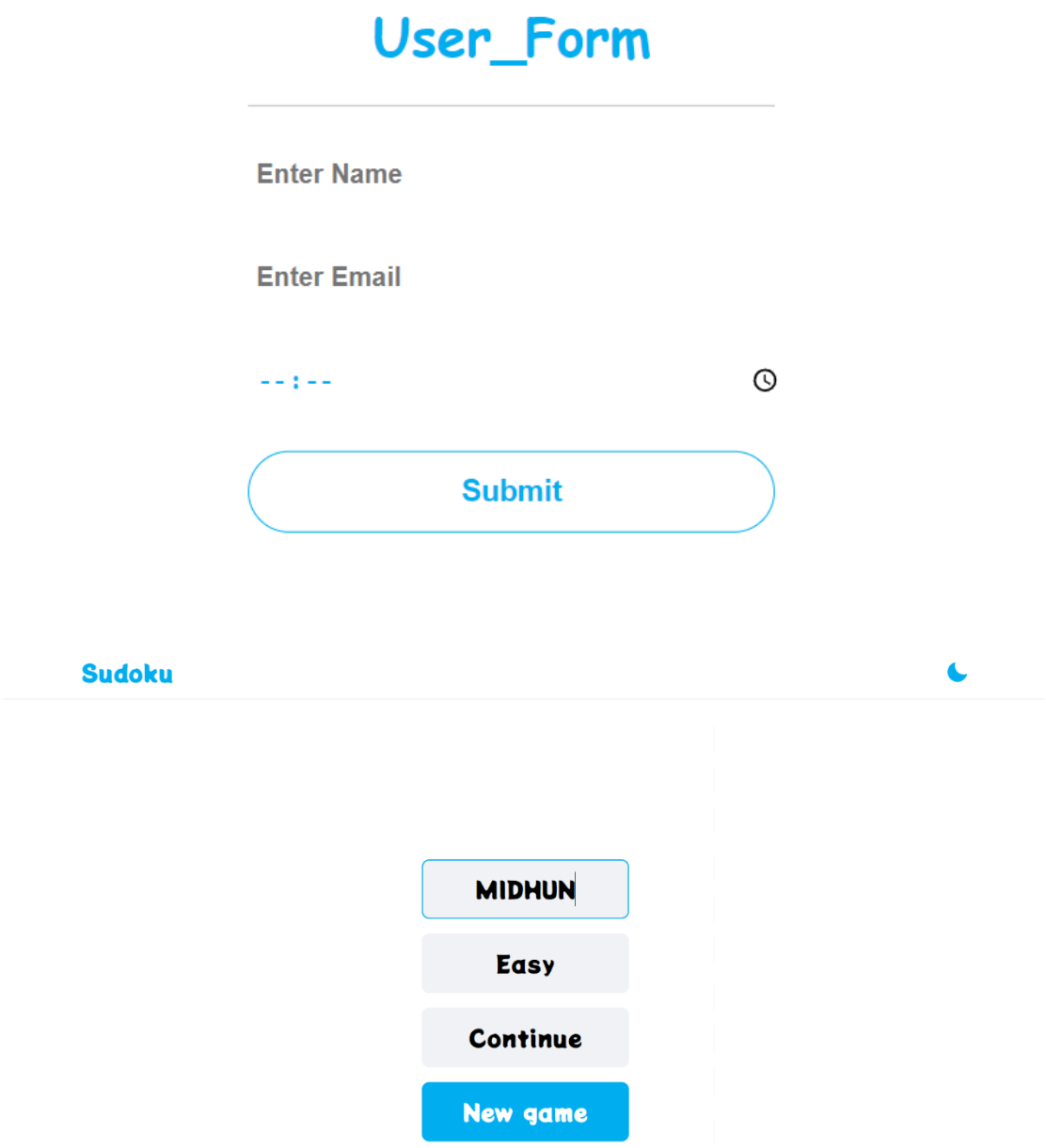
connect PHP file

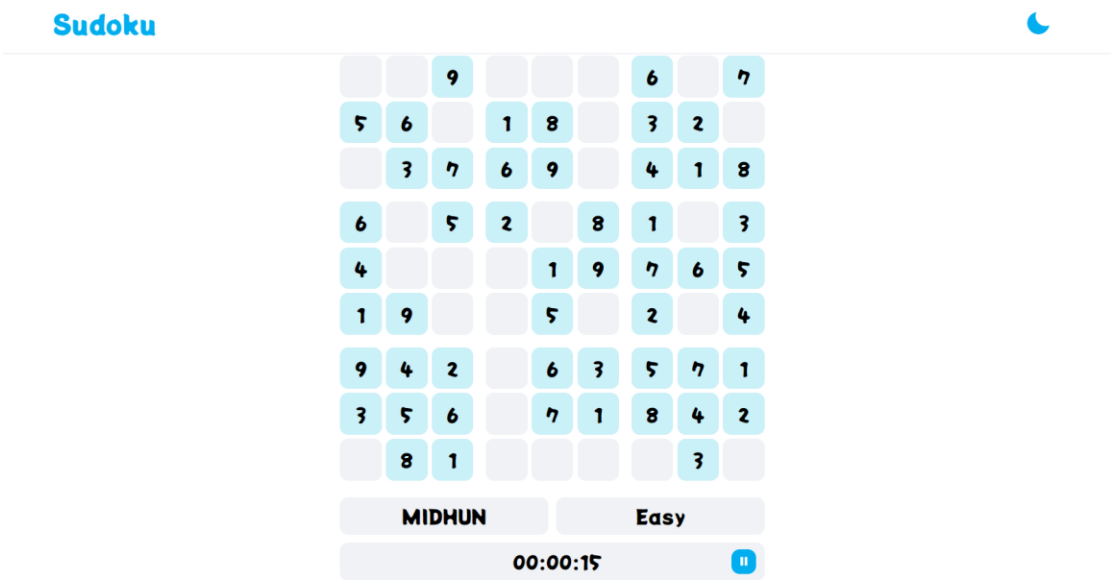
```
<html>
<body>
<?php
        $name = $_POST['name'];
        $email = $_POST['email'];
        $password = $_POST['password'];
      $conn = new mysqli('127.0.0.1','root',  ','test11');
        if($conn->connect_error){
                echo "$conn->connect_error";
                die("Connection Failed : ". $conn->connect_error);
        } else {
                $stmt = $conn->prepare("insert into signin(name, email, password)
values(?, ?, ?)");
                $stmt->bind_param("ssi", $name, $email, $password);
                $execval = $stmt->execute();
                echo $execval;
                echo "Registration successfully...";
                echo '<a href="index.html">Link</a>';
                $stmt->close();
                $conn->close();
        }
?>
```

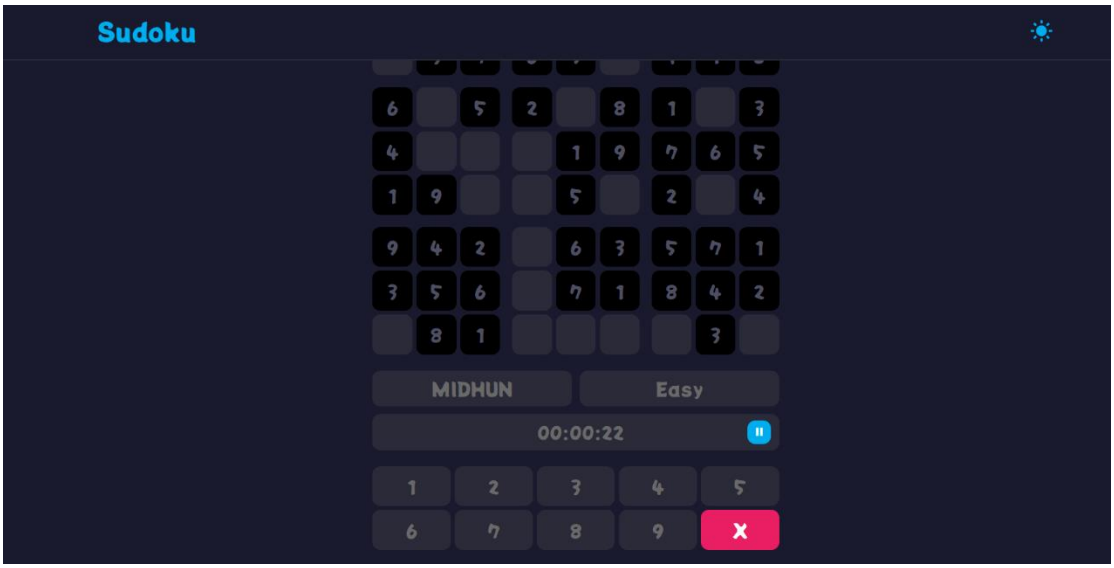
```
<body>
</html>

connect2 PHP file
<html>
<?php
        $User_name = $_POST['User_name'];
        $Email = $_POST['Email'];
        $Time = $_POST['Time'];

      $conn = new mysqli('127.0.0.1','root',  '','test11');
        if($conn->connect_error){
                echo "$conn->connect_error";
                die("Connection Failed : ". $conn->connect_error);
        } else {
                $stmt = $conn->prepare("insert into user_data1(User_name, Email,
Time) values(?, ?, ?)");
                $stmt->bind_param("ssi", $User_name, $Email, $Time);
                $execval = $stmt->execute();
                echo $execval;
                echo "Want to play another game";

           echo '<a href="index.html">Link</a>';
                $stmt->close();
                $conn->close();
        }
?>
```

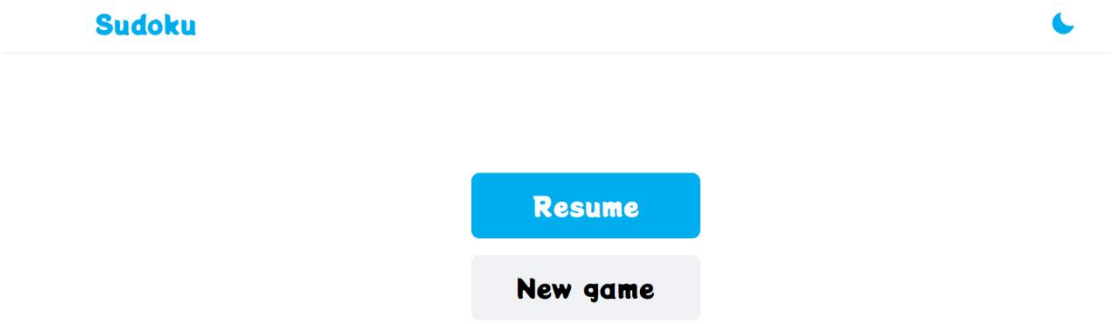
4.OUTPUT

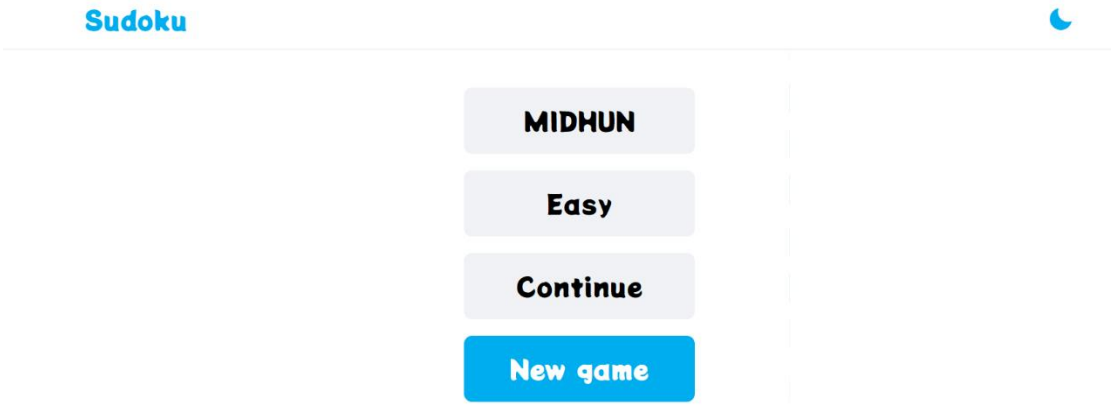

SignUp
Enter First Name
Enter Email
Enter Password
Sign Up
Close


User_Form
Enter Name
Enter Email
--:--
Submit
Sudoku
MIDHUN
Easy
Continue
New game


Sudoku
MIDHUN
Easy
00:00:15


Sudoku
MIDHUN
Easy
00:00:22


Sudoku
Resume
New game


Sudoku
MIDHUN
Easy
Continue
New game

5.KNOWLEDGE LEARNT FROM PROJECT

As part of this project, we have used several tools like Html, CSS, JavaScript, Php for making the website. We have learnt about the various approaches of solving Sudoku. One such method is backtracking as part of the course; we have learnt about how artificial intelligence is useful in solving real life complex problems and its various methods of implementation. We also learnt about various intelligent systems and approaches.

6. FUTURE IMPROVEMENTS THAT CAN BE DONE IN SUDOKU GAME DESIGN (9X9):

1) It is always better if we give the Instant Help by giving hints to the users while playing the game.
2) One can also keep the score system based on the time, accuracy, and database to keep track of the top ten records.
3) The code written is not of the highest quality, and it can be furtherly improved by optimizing.
4) In terms of time and the number of lines of code written we see that it is the longest code, and it can be minimized furtherly.
5) Also, a constraint Programming language like Prolog would be better suited to this kind of task because those languages include the ability to do some of the solving work automatically.
6) The other improvement that can happen is that the game that we design should also run in the Mobile or the Android device, but this will run only in the Desktop.
7) The next improvement that can be done is instead of displaying the numbers in the grid form, we could make use of apple's ARKit framework and using the images of the numbers instead of labels could be of better. Those pictures can be skewed and resized so that they can be mapped into scene for a more natural presentation of solution.
8) We can also add the option of saving an overlaid image into the user's picture gallery.

7.CONCLUSION

In sudoku game design project we have designed an online sudoku game website where the users can play. We have used html, CSS, php and js as our technologies. The sudoku game can usually be solved using different approaches but we have chosen brute force approach as we are making it a website. Sudoku game can not only be considered as game for relaxation, but it is also helpful to sharpen our brain logical and reasoning skills. further improvements can also be done in this project such as leaderboard, personalized layout for game panel, etc.…