

PROFESSIONAL TRAINING REPORT

at

**Sathyabama Institute of Science and Technology
(Deemed to be University)**

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering Degree in INFORMATION TECHNOLOGY
By

**Durga Prasad Kode
REG. NO. 39120027**



DEPARTMENT OF INFORMATION TECHNOLOGY

SCHOOL OF COMPUTING

**SATHYABAMA INSTITUTE OF SCIENCE AND
TECHNOLOGY**

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119, TAMILNADU**

APRIL 2022



SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A” by NAAC

(Established under Section 3 of UGC Act, 1956)

JEPPIAAR NAGAR, RAJIV GANDHI SALAI

CHENNAI- 600119

www.sathyabama.ac.in

DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of Durga Prasad Kode (Reg. No: 39120027) who carried out the project entitled “Web based game with 2 players (PPG-Ping Pong Game)” under my supervision from February 2022 to April 2022.

Internal Guide

Dr. R.M.Gomathi, M.Tech., Ph.D.

Head of the department

Dr. R.Subhashini M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I , **Durga Prasad Kode** hereby declare that the project report entitled “**Web based game with 2 players(PPG- Ping Pong Game)**” done by me under the guidance of

Dr. R.M.Gomathi, M.Tech., P.hD. is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Information Technology.

DATE:

PLACE: Chennai

A handwritten signature in blue ink, reading "K Durga Prasad", with a horizontal line underneath.

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D, Dean**, School of Computing, **Dr. S. Vigneshwari, M.E., Ph.D. and Dr. R.Subhashini, M.E., Ph.D., Heads of the Department of Information Technology** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. R.M.Gomathi,M.Tech.,P.hD.,** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project.

TRAINING CERTIFICATE

ABSTRACT

A **video game** or **computer game** is an electronic game that involves interaction with a user interface or input device such as a joystick, controller, keyboard, or motion sensing device to generate visual feedback. This feedback is shown on a video display device, such as a TV set, monitor, touchscreen, or virtual reality headset. Video games are often augmented with audio feedback delivered through speakers or headphones, and sometimes with other types of feedback, including haptic technology. Computer games are not all video games for example text adventure games, chess, and so on do not depend upon a graphics display.

One of the first digital games created was a two-dimensional electronic table tennis game, which was named Pong. This project creates an implementation of Pong which is controlled by two players using knobs, and displays the game on a 640x480 pixel VGA monitor. The knobs control potentiometers, which send an analog signal to the analog ports of the microcontroller. The microcontroller handles the game mechanics, including tracking the position of the ball and paddles, and keeping score. Using parallel ports, the microcontroller communicates the positions and score to the FPGA, which sends the necessary signals to a VGA monitor in order to display the game.

Ping Pong game is a two-player table tennis themed video game. The game involves two paddles and a moving ball. The players have to move paddles in upward or downward direction and save the ball from getting hit onto the wall. If the ball hits the wall then it's a score for another player.

Table tennis or Pong is the first commercially successful game. This game development project is for beginners and can clear many coding concepts in java. Many students searched it for their first year or second-year minor project. There are different milestones for the development of this game

1. Creation of moving paddles or object like tennis bat
2. Movement of the ball due to hits
3. Scoring System
4. Declaration of winner

CHAPTER	TITLE	PAGE NO.
1	INTRODUCTION	
1.1	OVER VIEW	1
1.2	PROCEDURE	3
2	SYSTEM ANALYSIS	
2.1	PURPOSE	4
2.1.1	Existing system	4
2.1.2	Proposed system	4
2.2	Problem definition	4
2.3	Feasibility study	4
2.3.1	Technical Feasibility	4
2.3.2	Economical feasibility	5
2.3.3	Operational feasibility	5
3	Software requirements and specifications	
3.1	Purpose	6
3.2	Scope	6
3.3	Over view	5
3.3.1	Product perspective	5
3.3.2	Product functionality	5
3.3.3	Users and characteristics	5
3.4	Specific Requirements	5
3.4.1	Hardware requirements	5
3.5	Functional requirements	6
3.6	Non-functional requirements	6
3.6.1	Performance	6
3.6.2	Reliability	6
3.6.3	Safety	6
3.6.4	Usability	7

3.7	Ping pong controls	7
4	Technologies Used	
4.1	HTML	8
4.2	CSS	9
4.3	JAVA SCRIPT	10
5	SOURCE CODE	
5.1	Welcome screen	11
5.2	Player's names	11
5.3	Game screen	12
6	CONCLUSION	13
7	SOURCE CODE	
7.1	Welcome screen (HTML)	14
7.2	Welcome screen (CSS)	14
7.3	Player's Name(HTML)	16
7.4	Player's Names (CSS)	17
7.5	Player's Names (JS)	19
7.6	Game Screen(HTML)	19
7.7	Game Screen(CSS)	20
7.8	Game Screen(JS)	24

CHAPTER 1

INTRODUCTION:

1.1 OVER VIEW

Ping Pong was the first commercially successful video game, and it helped to establish the video game industry along with the Magnavox Odyssey. Soon after its release, several companies began producing games that closely mimicked its gameplay. Eventually, Atari's competitors released new types of video games that deviated from Pong's original format to varying degrees, and this, in turn, led Atari to encourage its staff to move beyond Pong and produce more innovative games themselves.

It was one of the earliest arcade video games; it was created by Allan Alcorn as a training exercise assigned to him by Atari co-founder Nolan Bushnell, but Bushnell and Atari co-founder Ted Dabney were surprised by the quality of Alcorn's work and decided to manufacture the game. Bushnell based the game's concept on an electronic ping-pong game included in the Magnavox Odyssey, the first home video game console. In response, Magnavox later sued Atari for patent infringement.

Atari released several sequels to Pong that built upon the original's gameplay by adding new features. During the 1975 Christmas season, Atari released a home version of Pong exclusively through Sears retail stores. The home version was also a commercial success and led to numerous clones. The game was remade on numerous home and portable platforms following its release. Pong is part of the permanent collection of the Smithsonian Institution in Washington, D.C., due to its cultural impact.

Ping Pong, also known as Table tennis is a sport in which two or four players hit a lightweight ball, also known as the ping-pong ball, back and forth across a table using small paddles. The original PONG game is designed by Nolan Bushnell and released by Atari in 1972. PONG is an adaptation of table tennis to the video screen and is one of the first commercially successful game. It is also widely regarded to have usher in the video game era. Mini Pong, like the original PONG, is a basic simulation of the racket sport of table tennis. A sphere representing a ping pong ball travels across the screen in a linear trajectory. If the ball strikes the perimeter of the playing field, the obstacle in the middle of the room or one of the simulated paddles, the ball ricochets based on the angle of the impact. With every impact on the playing field or obstacle, the ball picks up speed. Speed is reduced upon hitting the paddle.

1.2 Procedure:

Create an HTML file with the name index.html.

Create a CSS file with the name styles.css and link it in the index.html file using the link tag.

Create a JS file with the name index.js and link it in the index.html file using a script tag.

Create a div for the game board, ball, and 2 divs for paddles, i.e. player-1 and player-2.

Give some style to your game in the CSS file.

Get a reference to paddles, balls, and game boards in JS.

Create a function in the index.js file with the name move ball.

Give the ball a random direction and a random speed by changing the x and y coordinates of the ball.

Apply collision with ball and game board's top/bottom side, i.e. if the ball touches the top/bottom of the board then multiply $-1 * y$ velocity of the ball.

Apply collision with ball and game left and right side of the game board. i.e. increase the score and change game state to serve the state.

Apply collision with ball and paddles. i.e. multiply $-1 * x$ velocity of the ball.

CHAPTER 2

2 System Analysis

2.1 Purpose

The primary goal of this project was to increase understanding of the video game industry's creation methods, communication behaviors, and attitudes for the purpose of building more accurate models of preservation and collection development.

2.1.1 Existing System

PING PONG Table tennis, also known as ping-pong is a sport in which two players hit a lightweight ball, also known as the ping-pong ball, back and forth across a table using small rackets. The game takes place on a hard table divided by a net. T-Rex Game or Dino Runner and initially code named Project Bolan, is a built-in browser game in the Google.

2.1.2 Proposed System

PING PONG We adapted Table Tennis game into digital platform with sound effects, score boards, player serves. Instead of rackets, rectangle shaped paddles are used and ball is replaced with square. Attractive background color makes the players more interested to focus with a maximum of 10 scores to become the winner.

2.2 Problem definition

This project is developed for indoor entertainment purposes due to COVID-19 pandemic.

2.3 FEASIBILITY STUDY

After the problem is clearly understood and solutions proposed, the next step is to conduct the feasibility study. Feasibility study is defined as evaluation or analysis of the potential impact of the proposed project or program. The objective is to determine whether the proposed system is feasible. There are three aspects of feasibility study to which the proposed system is subjected as discussed below.

2.3.1 Technical Feasibility

Technical feasibility assess whether the current technical resources are sufficient for the new system. We can upgrade the level of technology for supporting our website. We check whether the proposed system can be implemented in the present system without supporting the existing hardware.

2.3.2 Economical Feasibility

Economic feasibility determines whether the time and money are available to develop the system. There is no additional hardware used to develop the site, it is inexpensive to build.

2.3.3 Operational Feasibility

Operational feasibility determines if the human resources are available to operate the system once it has been installed. No extra training is needed to use this system. Anyone who has knowledge in internet in english language can easily use the system. The resources that are required to implement or install are already available with the organization.

CHAPTER 3

Software Requirement Specification

3.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the Game Development. It illustrate the purpose and complete description for the development of the system. It explain system constraints, interface. This document is primarily intended to be proposed for digital games and a reference for the upcoming students of the college.

3.2 Scope

Our project has made it easier for everyone to create their own digital games. Playing video games fulfills a purpose in their lives. This could include gaming for: relaxation, opportunities to exert control, enjoyment, creativity, socialization, prevent boredom, challenge, and achievement. It could also be used as a coping method or stress management.

3.3 Overview

The purpose of this document is to help the reader to visualize the solution to the project presented.

3.3.1 Product Perspective

Our project is mainly focused on entertainment purposes. This project can be implemented in any computer.

3.3.2 Product Functionality

In PING PONG two players can play at a time, where each player can handle one of the two paddles at a time.

3.3.3 Users and Characterstics

In PING PONG there are two players each can control one of the two paddles to direct the ball to the opponent's side.

3.4 Specific Requirements

3.4.1 Hardware Requirements

- ^ System: IBM-Compatible PC
- ^ Processor: Intel® Pentium® IV 2.4 GHz or AMD 3500+
- ^ Speed: Above 1GHz
- ^ RAM capacity:2 GB
- ^ Graphics: OpenGL 2.1 or OpenGL ES 2
- ^ Keyboard: Standard

- ^ Mouse: Standard
- ^ Monitor: SVGA Color

3.5 Functional Requirements

It contains three main modules.

Two players in PING PONG

Two players in PING PONG Each player will control the paddle in their respective side to hit the ball to prevent it from going out of boundary. Each player can interact with the ball only when the ball reaches near their paddle.

3.6 Non Functional Requirements

Non-functional requirements define the overall qualities or attributes of the resulting system. Non-functional system place restrictions on the product being developed, the developed process, and specify external constraints that the product must meet. Examples of non-functional requirements include safety, security, usability, reliability and performance requirements. Project management issues (costs, time and schedule) are often considered as non-functional requirements. The principal non - functional constraints which are relevant to critical systems :

- ^ performance
- ^ security
- ^ safety
- ^ usability

3.6.1 Performance

Performance requirements concern the speed of operation of a system.

Types of performance requirements :

Response requirements (how quickly the system reacts to a user input).

Throughput requirements (how much the system can accomplish within a specified amount of time).

Availability requirements (is the system available for service when requested by end users).

The speed of operation of this system is adequate for the requirements.

3.6.2 Reliability

Reliability is the ability of a system to perform its required function under stated conditions for a specified period of time.

constraints on the runtime behavior of the system. This system is reliable because its functionalities can be done on the required conditions.

3.6.3 Safety

Safety requirements are not required which exclude unsafe situation from the possible solution of the system.

3.6.4 Usability

Usability is the ease with which a user can learn to operate, prepare inputs for, and interpret outputs of system or components. Usability requirements include :
well-formed user interfaces.

3.7 PING PONG CONTROLS

Up - To move the Paddle up for Player1

Down - To move the Paddle down for Player1

W - To move the Paddle up for Player2

S - To move the Paddle down for Player2

Enter - To start the game

CHAPTER 4

4.1 HTML: Hyper Text Markup Language

HTML (Hyper Text Markup Language) is the most basic building block of the Web. It defines the meaning and structure of web content.

Other technologies besides HTML are generally used to describe a web page's appearance/presentation (CSS) or functionality/behavior (JavaScript).

"Hypertext" refers to links that connect web pages to one another, either within a single website or between websites. Links are a fundamental aspect of the Web.

By uploading content to the Internet and linking it to pages created by other people, you become an active participant in the World Wide Web.

HTML uses "markup" to annotate text, images, and other content for display in a Web browser.

HTML markup includes special "elements" such as <head>, <title>, <body>, <header>, <footer>, <article>, <section>, <p>, <div>, , , <aside>, <audio>, <canvas>, <datalist>, <details>, <embed>, <nav>, <output>, <progress>, <video>, , , and many others.

An HTML element is set off from other text in a document by "tags", which consist of the element name surrounded by " " and " ".

The name of an element inside a tag is case insensitive. That is, it can be written in uppercase, lowercase, or a mixture.

HTML is a markup language that web browsers use to interpret and compose text, images, and other material into visual or audible web pages. Default characteristics for every item of HTML markup are defined in the browser, and these characteristics can be altered or enhanced by the web page designer's additional use of CSS. Many of the text elements are found in the 1988 ISO technical report TR 9537 Techniques for using SGML, which in turn covers the features of early text formatting languages such as that used by the RUNOFF command developed in the early 1960s for the CTSS (Compatible Time-Sharing System) operating system: these formatting commands were derived from the commands used by typesetters to manually format documents.

The first publicly available description of HTML was a document called "HTML Tags", first mentioned on the Internet by Tim Berners-Lee in late 1991. It describes 18 elements comprising the initial, relatively simple design of HTML. Except for the hyperlink tag, these were strongly influenced by SGML guid, an in-house Standard Generalized Markup Language

(SGML)-based documentation format at CERN. Eleven of these elements still exist in HTML 4.

4.2 CSS: Cascading Style Sheets

CSS is a stylesheet language used to describe the presentation of a document written in HTML or XML (including XML dialects such as SVG, MathML or XHTML).

CSS describes how elements should be rendered on screen, on paper, in speech, or on other media.

CSS is among the core languages of the and is standardized across Web browsers according to W3C specifications.

Previously, development of various parts of CSS specification was done synchronously, which allowed versioning of the latest recommendations.

You might have heard about CSS1, CSS2.1, CSS3. However, CSS4 has never become an official version.

From CSS3, the scope of the specification increased significantly and the progress on different CSS modules started to differ so much, that it became more effective to develop and release recommendations separately per module.

Instead of versioning the CSS specification, W3C now periodically takes a snapshot of the latest stable state of the CSS specification.

The name cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.

The CSS specifications are maintained by the World Wide Web Consortium (W3C). Internet media type (MIME type) text/css is registered for use with CSS by RFC 2318 (March 1998). The W3C operates a free CSS validation service for CSS documents.

Inheritance in CSS is not the same as inheritance in class-based programming languages, where it is possible to define class B as "like class A, but with modifications". With CSS, it is possible to style an element with "class A, but with modifications". However, it is not possible to define a CSS class B like that, which could then be used to style multiple elements without having to repeat the modifications.

4.3 JS: JAVA SCRIPT

JS is a lightweight, interpreted, or just-in-time compiled programming language with first-class functions.

While it is most well-known as the scripting language for Web pages, many non-browser environments also use it, such as Node.js, Apache CouchDB and Adobe Acrobat.

JavaScript is a prototype-based, multi-paradigm, single-threaded, dynamic language, supporting object-oriented, imperative, and declarative (e.g. functional programming) styles.

Read more about JavaScript.

This section is dedicated to the JavaScript language itself, and not the parts that are specific to Web pages or other host environments.

For information about API specifics to Web pages, please see Web APIs and DOM.

The standards for JavaScript are the ECMAScript Language Specification (ECMA-262) and the ECMAScript Internationalization API specification (ECMA-402).

The JavaScript documentation throughout MDN is based on the latest draft versions of ECMA-262 and ECMA-402.

And in cases where some proposals for new ECMAScript features have already been implemented in browsers, documentation and examples in MDN articles may use some of those new features.

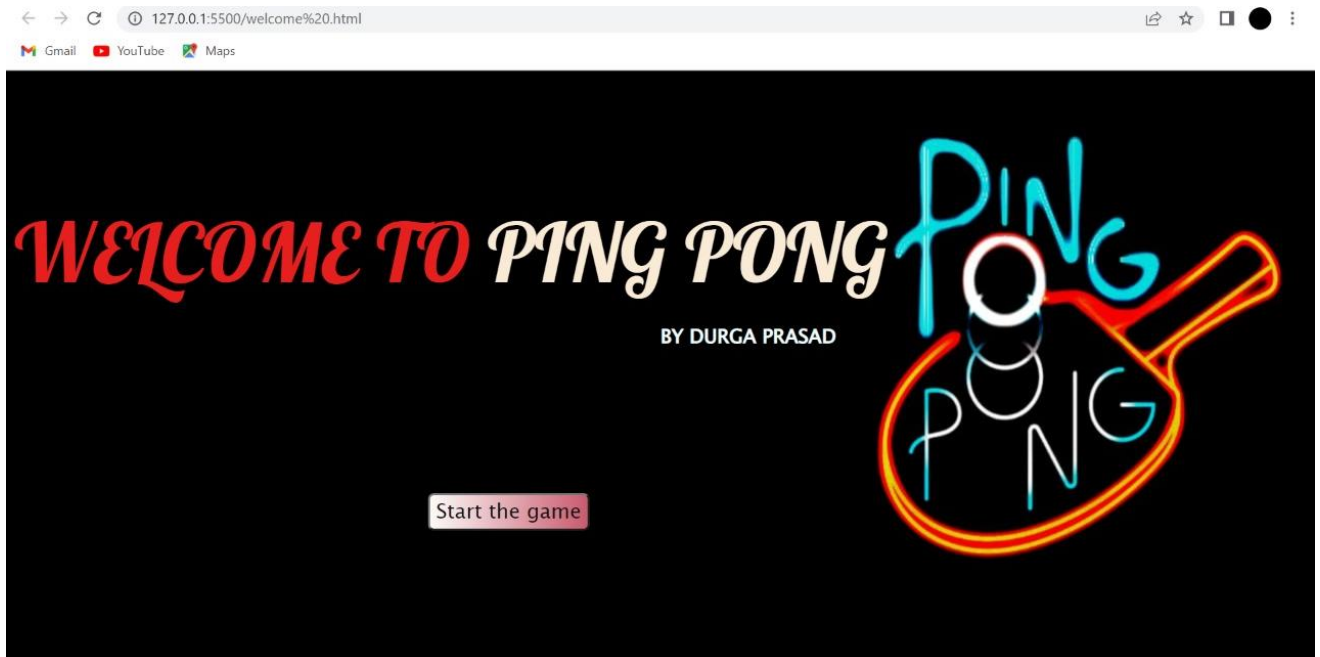
Do not confuse JavaScript with the Java programming language.

Both "Java" and "JavaScript" are trademarks or registered trademarks of Oracle in the U.S. and other countries.

However, the two programming languages have very different syntax, semantics, and use.

CHAPTER 5

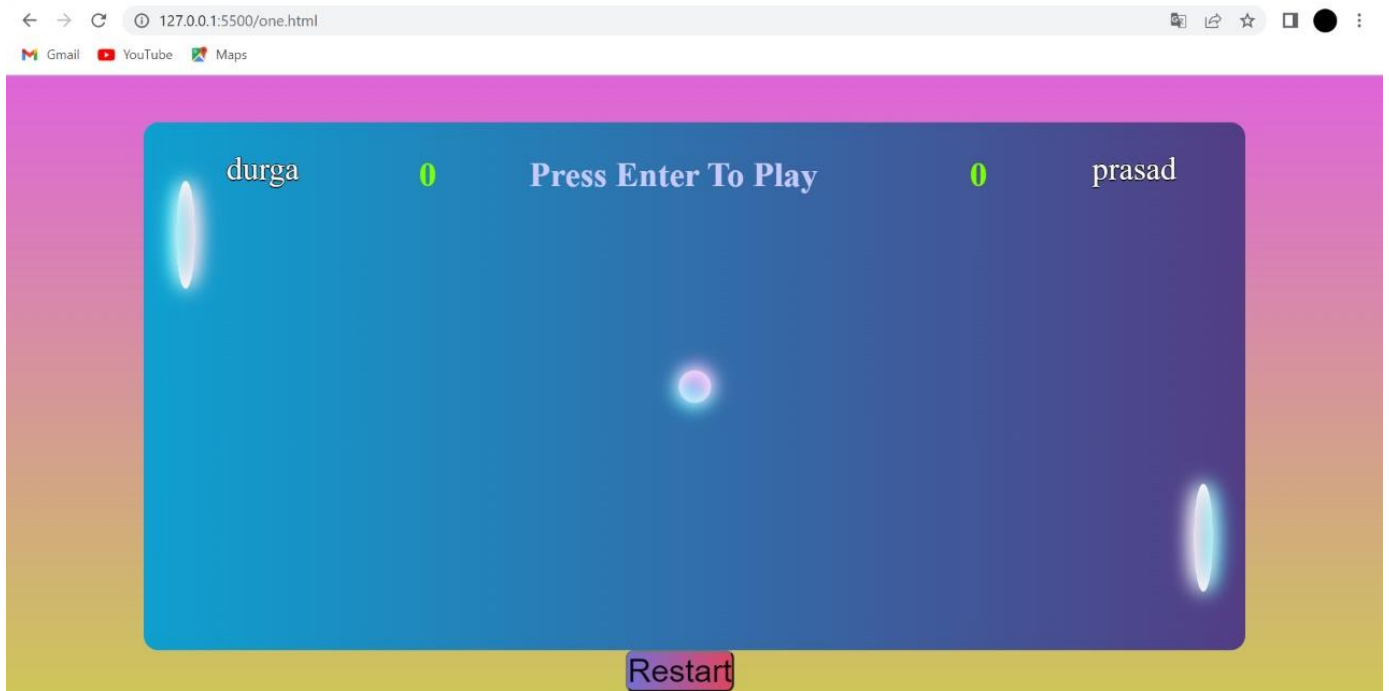
5.1 Welcome Screen



5.2 Player's Names:

A screenshot of a web browser displaying a form for entering player names. The browser's address bar shows the URL "127.0.0.1:5500/names.html". The page has a gradient background transitioning from orange on the left to blue on the right. In the center, there are two input fields. The first is labeled "Username 1" and contains the text "durga". The second is labeled "Username 2" and contains the text "prasad". Below the input fields, there are two buttons: "Submit" and "Continue".

5.3 GAME SCREEN:



CHAPTER 6

CONCLUSION

PING PONG: There are two players that can play the game at a time. Each player can handle one of the paddles using the control keys (up and down keys, w and s keys).

Single player will control the jumping of Mr.Toad using the space bar and try to pass between the pipes.

In the project, two clients comes together to connect and plays the game.

The project can be taken a step forward by adding more than two players or by setting a difficulty level (My View).

CHAPTER 7

SOURCE CODE:

7.1 Welcome screen (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">

  <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=Lobster&display=swap"
rel="stylesheet">

  <link rel="stylesheet" href="style.css">

  <div><span  class="wlc"  >WELCOME  TO</span>  <span  class="ping"  >PING
PONG</span> </div>

  <h3>BY DURGA PRASAD</h3>

  <a href="/names.html"><button>Start the game</button> </a>

</head>

<body>

</body>

</html>
```

7.2 Welcome Screen CSS

```
div{
  font-family: 'Lobster', cursive;
  text-align: left;
  font-size: 80px;
  margin-top: 10%;
  border-color: black 15px;
}
```

```

.wlc{
    color: rgb(231, 31, 31);

}

.ping{

    color: antiquewhite;
}

h3{

    margin-left: 50%;
    color: azure;
    font-family: 'Lucida Sans', 'Lucida Sans Regular', 'Lucida Grande', 'Lucida Sans Unicode',
Geneva, Verdana, sans-serif;

}

button{

    margin-left: 32%;
    margin-top: 120px;
    color: rgb(16, 17, 18);
    font-family: 'Lucida Sans', 'Lucida Sans Regular', 'Lucida Grande', 'Lucida Sans Unicode',
Geneva, Verdana, sans-serif;
    font-size: 20px;
    background-image: linear-gradient(to right, #fdfbf8, #ca5b6f);
    border-radius: 6px;

}

body{
    background-image: url(/bg1.jpeg);
    background-repeat: no-repeat;
    background-size: 1280px;

```

```
background-position: center;

}
```

7.3 Players Names (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">
  <link      href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet"                                integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">
  <link rel="stylesheet" href="style1.css">
  <title>Names</title>
</head>

<body>

  <div class="center" >
    <form name="form1" >
      <label for="fname" id="onee" >Username 1</label><br>
      <input type="text" id="txtName" name="fname" ><br>
      <label for="lname" id="twoo" >Username 2</label><br>
      <input type="text" id="txtEmail" name="lname" ><br><br>
    </form>
  </div>

  <div class="zoo" >
    <p id="p1"></p>
    <p id="p2"></p>
```



```
</div>
<button class="btn1" onclick="testVariable()" >Submit</button>
<a href="/one.html"><button class="btn2" >Continue</button> </a>
```

```
<script src="/namescript.js"></script>
<script src="/three.js"></script>
```

```
</body>
</html>
```

7.4 Players Names (CSS)

```
div{
  font-size: 20px;
  border-color: rgb(169, 33, 33) 15px;
}
.btn1{
  position: absolute;
  top: 70%;
  left: 46%;
  color: rgb(15, 0, 0);
  font-family: 'Lucida Sans', 'Lucida Sans Regular', 'Lucida Grande', 'Lucida Sans Unicode',
Geneva, Verdana, sans-serif;
  font-size: 20px;
  background-image: linear-gradient(to right, #073a87, #c67373);
  border-radius: 6px;
  padding-right: 10px;
  text-align: center;
}

.btn2{

  position: absolute;
  top: 80%;
```

```
    left: 45%;
    color: rgb(15, 0, 0);
    font-family: 'Lucida Sans', 'Lucida Sans Regular', 'Lucida Grande', 'Lucida Sans Unicode',
Geneva, Verdana, sans-serif;
    font-size: 20px;
    background-image: linear-gradient(to right, #6290d6, #a11e1e);
    border-radius: 6px;
    padding-right: 10px;
    padding-left: 10px;
}
```

```
body{

    background-image: linear-gradient(to right, #e39d58, #5f92de);

}
```

```
input{
    width: 100%;
    margin-bottom: 50px;
}
```

```
.center{

    position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);
}
```

```
#onee{

    padding-left: 10px;
```

```

}

#twoo{
    padding-left: 10px;
}

.zoo{
    color: #060606;
    font-size: 0px;
}

```

7.5 Players names (JAVA SCRIPT)

```

function testVariable() {
    debugger
    var strText = document.getElementById("txtName").value;
    var strText1 = document.getElementById("txtEmail").value;
    console.log(strText);
    document.getElementById("p1").innerHTML = strText;
    document.getElementById("p2").innerHTML = strText1;

    localStorage.setItem("playerone",strText);
    localStorage.setItem("playertwo",strText1);
}

```

7.6 GAME SCREEN (HTML):

```

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

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>PING PONG GAME</title>
    <link rel="stylesheet" href="./two.css">
</head>

```

```

<body>

<div class="board">
  <div class="ball">
    <div class="ball_effect"></div>
  </div>
  <div class="paddle_1 paddle"></div>
  <div class="paddle_2 paddle"></div>
  <h1 class="player_1_score">0</h1>
  <h1 class="player_2_score">0</h1>
  <h1 class="message">
    Press Enter To Play
  </h1>
</div>
<div>
  <p id="p1"></p>
  <p id="p2"></p>
</div>
<div> <a href="/names.html"> <button >Restart</button></a></div>

<script src="/three.js"></script>
<script src="/namescript.js"></script>
</body>

</html>

```

7.7 GAME SCREEN (CSS) :

```

* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

```

```
body {  
  height: 100vh;  
  width: 100vw;  
  background-image: linear-gradient(  
    to top, #cdc759, #de64da);  
  display: flex;  
  justify-content: center;  
  align-items: center;  
}
```

```
.board {  
  height: 85vh;  
  width: 80vw;  
  background-image: linear-gradient(  
    to right, #0ea0d0, #533d85);  
  border-radius: 14px;  
}
```

```
.ball {  
  height: 30px;  
  width: 30px;  
  border-radius: 50%;  
  position: fixed;  
  top: calc(50% - 15px);  
  left: calc(50% - 15px);  
}
```

```
.ball_effect {  
  height: 100%;  
  width: 100%;  
  border-radius: 100px;  
  animation: spinBall 0.1s linear infinite;  
  box-shadow: inset 0 0 18px #fff,
```

```

    inset 6px 0 18px violet,
    inset -6px 0 18px #0ff,
    inset 6px 0 30px violet,
    inset -6px 0 30px #0ff,
    0 0 18px #fff, -4px 0 18px
    violet, 4px 0 18px #0ff;
}

```

```

@keyframes spinBall {
    100% {
        -webkit-transform: rotate(360deg);
        transform: rotate(360deg);
    }
}

```

```

.paddle {
    height: 100px;
    width: 18px;
    border-radius: 50%;
    position: fixed;
}

```

```

.paddle_1 {
    top: calc(7.5vh + 55px);
    left: calc(10vw + 30px);
    box-shadow: inset 0 0 18px #fff,
        inset -6px 0 18px #f3bad6,
        inset 6px 0 18px #0ff,
        inset -6px 0 30px #f3bad6,
        inset 6px 0 30px #0ff,
        0 0 18px #fff, 4px 0 18px
        #f3bad6, -4px 0 18px #0ff;
}

```

```
.paddle_2 {
  top: calc(85vh + 7.5vh - 100px - 55px);
  right: calc(10vw + 30px);
  box-shadow: inset 0 0 18px #fff,
    inset 6px 0 18px #f3bad6,
    inset -6px 0 18px #0ff,
    inset 6px 0 30px #f3bad6,
    inset -6px 0 30px #0ff,
    0 0 18px #fff, -4px 0 18px
    #f3bad6, 4px 0 18px #0ff;
}
```

```
.player_1_score {
  height: 50px;
  width: 50px;
  color: chartreuse;
  position: fixed;
  left: 30vw;
  margin-top: 30px;
}
```

```
.player_2_score {
  height: 50px;
  width: 50px;
  color: chartreuse;
  position: fixed;
  left: 70vw;
  margin-top: 30px;
}
```

```
.message {
  position: fixed;
  /* color: #48426d; */
  height: 10vh;
```

```
width: 30vw;
color: #c9cbff;
left: 38vw;
margin: 30px auto auto auto;
}
```

```
button{

background-image: linear-gradient(
    to right, #766fd8, #e24561);
border-radius: 7px;
position: absolute;
bottom: 5px;
left: 45%;
text-align: center;
color: rgb(10, 8, 9);
font-size: 30px
}
```

```
#p1{

position: absolute;
left: 16%;
top: 12%;
font-size: 30px;
color:rgb(249, 249, 249);
text-shadow: -1px 0 black, 0 1px black, 1px 0 black, 0 -1px black;

}
```

```
#p2{

position: absolute;
right: 15%;
```



```

top: 12%;
font-size: 30px;
color:rgb(255, 255, 255);
text-shadow: -1px 0 black, 0 1px black, 1px 0 black, 0 -1px black;
}

```

7.8 GAME SCREEN (JAVA SCRIPT):

```

let gameState = 'start';
let paddle_1 = document.querySelector('.paddle_1');
let paddle_2 = document.querySelector('.paddle_2');
let board = document.querySelector('.board');
let initial_ball = document.querySelector('.ball');
let ball = document.querySelector('.ball');
let score_1 = document.querySelector('.player_1_score');
let score_2 = document.querySelector('.player_2_score');
let message = document.querySelector('.message');
let paddle_1_coord = paddle_1.getBoundingClientRect();
let paddle_2_coord = paddle_2.getBoundingClientRect();
let initial_ball_coord = ball.getBoundingClientRect();
let ball_coord = initial_ball_coord;
let board_coord = board.getBoundingClientRect();
let paddle_common =
    document.querySelector('.paddle').getBoundingClientRect();

let dx = Math.floor(Math.random() * 4) + 3;
let dy = Math.floor(Math.random() * 4) + 3;
let dxd = Math.floor(Math.random() * 2);
let dyd = Math.floor(Math.random() * 2);

var namee=localStorage.getItem("playerone");
var name1=localStorage.getItem("playertwo");
document.getElementById("p1").innerHTML = namee;
document.getElementById("p2").innerHTML = name1;

```

```

document.addEventListener('keydown', (e) => {
  if (e.key === 'Enter') {
    gameState = gameState === 'start' ? 'play' : 'start';
    if (gameState === 'play') {
      message.innerHTML = 'Game Started';
      message.style.left = 42 + 'vw';
      requestAnimationFrame(() => {
        dx = Math.floor(Math.random() * 4) + 3;
        dy = Math.floor(Math.random() * 4) + 3;
        dxd = Math.floor(Math.random() * 2);
        dyd = Math.floor(Math.random() * 2);
        moveBall(dx, dy, dxd, dyd);
      });
    }
  }
  if (gameState === 'play') {
    if (e.key === 'w') {
      paddle_1.style.top =
        Math.max(
          board_coord.top,
          paddle_1_coord.top - window.innerHeight * 0.06
        ) + 'px';
      paddle_1_coord = paddle_1.getBoundingClientRect();
    }
    if (e.key === 's') {
      paddle_1.style.top =
        Math.min(
          board_coord.bottom - paddle_common.height,
          paddle_1_coord.top + window.innerHeight * 0.06
        ) + 'px';
      paddle_1_coord = paddle_1.getBoundingClientRect();
    }

    if (e.key === 'ArrowUp') {

```

```

paddle_2.style.top =
    Math.max(
        board_coord.top,
        paddle_2_coord.top - window.innerHeight * 0.1
    ) + 'px';
paddle_2_coord = paddle_2.getBoundingClientRect();
}
if (e.key == 'ArrowDown') {
paddle_2.style.top =
    Math.min(
        board_coord.bottom - paddle_common.height,
        paddle_2_coord.top + window.innerHeight * 0.1
    ) + 'px';
paddle_2_coord = paddle_2.getBoundingClientRect();
}
}
});

```

```

function moveBall(dx, dy, dxd, dyd) {
if (ball_coord.top <= board_coord.top) {
    dyd = 1;
}
if (ball_coord.bottom >= board_coord.bottom) {
    dyd = 0;
}
if (
    ball_coord.left <= paddle_1_coord.right &&
    ball_coord.top >= paddle_1_coord.top &&
    ball_coord.bottom <= paddle_1_coord.bottom
) {
    dxd = 1;
    dx = Math.floor(Math.random() * 4) + 3;
    dy = Math.floor(Math.random() * 4) + 3;
}
}

```

```

if (
  ball_coord.right >= paddle_2_coord.left &&
  ball_coord.top >= paddle_2_coord.top &&
  ball_coord.bottom <= paddle_2_coord.bottom
) {
  dxd = 0;
  dx = Math.floor(Math.random() * 4) + 3;
  dy = Math.floor(Math.random() * 4) + 3;
}
if (
  ball_coord.left <= board_coord.left ||
  ball_coord.right >= board_coord.right
) {
  if (ball_coord.left <= board_coord.left) {
    score_2.innerHTML = +score_2.innerHTML + 1;
  } else {
    score_1.innerHTML = +score_1.innerHTML + 1;
  }
  gameState = 'start';

  ball_coord = initial_ball_coord;
  ball.style = initial_ball.style;
  message.innerHTML = 'Press Enter to Play Pong';
  message.style.left = 38 + 'vw';
  return;
}
ball.style.top = ball_coord.top + dy * (dyd == 0 ? -1 : 1) + 'px';
ball.style.left = ball_coord.left + dx * (dxd == 0 ? -1 : 1) + 'px';
ball_coord = ball.getBoundingClientRect();
requestAnimationFrame(() => {
  moveBall(dx, dy, dxd, dyd);
});
}

```