AMRITSAR GROUP OF COLLAGE

Autonomous status Conferred by UGC|NAAC-A Grade

SIX WEEKS INDUSTRIAL TRAINING

On

Web Development

B.TECH (CSE) – V

SEMESTER BATCH (2020-2024)

Project Report On

"SNAKE CANDY"

Submitted to :

Dr. Sandeep Kad

Head of Department (CSE)

Submitted by:

Amitesh kumar

(2000061)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Amritsar College of Engineering and Technology Amritsar

ACKNOWLEDGEMENT

I am highly grateful to Dr. Sandeep Kad for providing this opportunity to carry out the six Weeks industrial training. It was purely on the basis of his experience and knowledge that I am able to clear all the theoretical and technical hurdles during the development phases of this project work.

I want to express gratitude to other faculty members of Department of Computer science and Technology for their intellectual support throughout the training Course.

I also thank all our friends who have more or less contributed to the preparation of this Project Report, we will be always indebted to them. This project completion has indeed helped us explore more knowledge avenues related to web development and I am sure it will help us in future too.

DECLARATION

I the undersigned solemnly declare that the synopsis on project SNAKE CANDY is based on my own work carried out during the course of our study.

I assert the statements made and conclusions drawn are an outcome of my research work. I further certify that

- The work contained in the report is original and has been done by me.
- The work has not been submitted to any other Institution for any other degree/diploma/certificate in this collage or any other of India and abroad.
- I have followed the guidelines provided by the collage in writing the report.
- Whenever I have used materials (data, theoretical analysis and text) from other sources, I have given due credit to them in the text of the report and giving their details in the references.

Student Name

Amitesh kumar

(2000061)

TABLE OF CONTENTS

❖ INTRODUCTION TO ORGANIZATION

ICE Technology Lab is a leading name in the technical training industry. We are known for churning out globally competent professionals who are miles ahead of others in skill set, knowledge and ability to deliver.

Getting a job is as difficult as beating the crowd because being in the corporate world demand a lot from the applicant because of which the applicants are putting their best, which results in the increment of difficulty level. You can see each and every thing is connected but the solution of this problem is either spending years to reach to a desired position or come to ICETL. ICETL provide the entire necessary computer training which helps the newbies and also the experienced workers so that they can achieve better recognition in this competitive world**.**

ICETL has specialization in 3 important domains namely:

• Training

• Development

• Placement

The company provides specialized training in 50+ leading technologies like, NET, Java, PHP, Ethical Hacking, Android, CCNA, AUTOCAD, MATLAB, EMBEDED SYSTEM and many more.

ICETL has a very committed team consisting of technical trainers who are continuously guiding, mentoring, admonish and coaching the students by providing them with exclusive personalized attention, which helps them to develop solid industry-oriented knowledge.

ICETL Services:

• Student Training

It is the best training company in whole northern India. It provides student training in various technologies like VLSI program, embedded system training, Microsoft certified training, java training etc. by professionals and experienced trainers who guides the students with best possible efforts. Various technologies are offered for student training with well-defined modules and different time period.

ICETL's training program is mainly constructed for students. The purpose of students training is to escalate career growth of students by educating them in the various fields of information technology.

• Fresher Training

If you are a fresher or last year pass out and wish to pursue long term industrial training/ summer training/ winter training or short term 6 weeks training, them ICETL would be the right choice for you. Because, ICETL is the best training company offering training for engineering students.

Fresher training is a very likely phase among scholars who have completed their degree.

• Professional Training

ICETL is mentoring the Engineering candidates and lending them a hand in flourishing and presenting themselves as professionals in specific valuable skill.

Professional Training has specific goals of bettering one's potential, efficiency and accomplishment. It works with broad sphere of professions and associations from all zones, whether they are self-employed or in abundant constitutions, to overture entrepreneur training and expansion contingency along with culture and work involvement placements.

• College Campus Training

This program is a manifesto for providing complete training for engineering students on latest technologies to college students from different areas like B. tech, M. tech, BCA etc. within college campus itself. College campus training is imperative because it has all the important attributes which are vital for students to know before they enter into the corporate life where practical training is given more emphasis than the theoretical learning.

• 6 Months Industrial Training

ICETL, the Good Industrial Training company for CS, CSE, IT, BCA, MCA, B. tech, M. tech, BE Engineering student provides 6 months Industrial training program in all the domains. The objective of 6 months Industrial Training is to enhance knowledge of the students on any one of the cutting-edge technologies according to the industry standards

without which the student degree is a mere degree. This is done by exposing students to real work life situations which equip them with the required skill needed for the corporate world.

- **College Seminar / Workshop**

Unlike lectures, where a speaker will address the audience, seminars and workshops are led by someone acting as a mediator. This allows for opening up the floor for discussion pertaining to the technology's world and hearing about other ideas on the topics in a setting that encourages conversation. Workshops and seminars can address common problems or issues that techy students may face on a regular basis. Participants can share their insights and thoughts on how to resolve the problem, which can offer a fresh perspective when dealing with the problem. You can find a seminars or workshops to help you develop new skills that can help you get ahead. ICETL provides paid and free seminar & workshops on different new technologies

❖ Web Development

Web development refers to building, creating, and maintaining websites. It includes aspects such as web design, web publishing, web programming, and database management.

While the terms "web developer" and "web designer" are often used synonymously, they do not mean the same thing. Technically, a web designer only designs website interfaces using HTML and CSS. A web developer may be involved in designing a website, but may also write web scripts in languages such as PHP and ASP. Additionally, a web developer may help maintain and update a database used by a dynamic website.

Web development includes many types of web content creation. Some examples include hand coding web pages in a text editor, building a website in a program like Dreamweaver, and updating a blog via a blogging website. In recent years, content management systems like WordPress, Drupal, and Joomla have also become popular means of web development. These tools make it easy for anyone to create and edit their own website using a web-based interface.

While there are several methods of creating websites, there is often a trade-off between simplicity and customization. Therefore, most large businesses do not use content management systems, but instead have a dedicated Web development team that designs and maintains the company's website(s). Small organizations and individuals are more likely to choose a solution like Wordpress that provides a basic website template and simplified editing tools.

NOTE: JavaScript programming is a type of web development that is generally not considered part of web design. However, a web designer may reference JavaScript libraries like jQuery to incorporate dynamic elements into a site's design.

This blog covers some of the crucial and most important information in the field of Web Development. Here, you will get a basic idea of what Web Development is and how it works. Once you understand the meaning of Web Development, you will come across various platforms, tools, technologies, and languages used in this field. Later, you will read about the types of Web Development**.**

Technology plays an integral part in our day-to-day lives. It surrounds us in the simplest of places to perform ordinary tasks and make room for great inventions. All the web pages and software that we use to make our lives easier are developed by Web Developers. However, the Web Development meaning is still not clear to many of us, so let's start there and learn more about what Web Development is.

In the process of Web Development, Developers build web pages and applications for either the Intranet, a private network, or the Internet. It does not necessarily focus on a website's design; rather, it is majorly concerned with the programming and coding part, which is the main reason for the functioning of the website.

It refers to developing and maintaining web pages, including bits and pieces of concepts like web design, web programming, web publishing, and database management. Further, it includes various types of web development tools and techniques such as text editors for manually coding the websites, Dreamweaver for developing a web page, using a blogging website to update blogs, and more.

From basic and simple websites to complex web applications and social media platforms and from numerous online shopping web pages to even content management systems (CMS), all the online tools and websites that we use regularly are part of Web Development. Besides, all these tools and websites are built by Web Developers.

- **This blog includes the topics mentioned below:**
    - Web Development Platforms

- Most Popular Web Development Tools
- Web Development Technologies
- Web Development Frameworks
- Web Development Languages
- 3 Layers of Web Development
- Future Scope of Web Development
- Conclusion

➢ Web Development Platforms

In the world we live in today, rather than being a trend, a website is a necessity. No matter how small or how big an organization is, to survive and succeed in this era of digitization, making a website and reaching out to the crowd using online methods are necessary. These days, people are also building their web pages to share information or their personal experience with others. Some of the most popular website development platforms include:

- WordPress
- Wix
- Weebly
- Jimdo

➢ Most Popular Web Development Tools

To build functioning websites efficiently and easily, there are various tools that Web Developers can use. These tools allow them to build well-functioning websites with friendly user interfaces. Some of the widely used Web Development tools are as follows:
GitHub

- Sass
- CodePen
- AngularJS
- TypeScript
- Sketch
- JQuery

- Sublime Text
- Bootstrap
- Grunt
- Chrome DevTools
- NPM
- Visual Studio Code

➢ Web Development Technologies

To learn ‘What is Web Development?’ in-depth, you need to understand and have knowledge of the technologies used in Web Development. Below are some of the most common Web Development technologies that every Web Developer should learn to work with:

- HTML
- Browsers
- Programming languages
- CSS
- Frameworks
- Databases
- Libraries

- Servers
- Clients
- Frontend Development
- Backend Development
- Protocols
- Data formats
- API

➢ **Web Development Frameworks**

Web Development frameworks are created by Developers to make it easier to develop and work with various programming languages. These frameworks remove all the redundant and difficult tasks that are involved in the process of setting up a web page. The frameworks either do the tasks themselves or make it easier for Developers to complete those tasks.

Below are a few frameworks that Web Developers generally use:

- **Node.js**: A JavaScript framework for server-side
- **Ruby on Rails**: A full-stack framework developed with the help of Ruby
- **Ionic**: A mobile framework
- **Django**: A full-stack framework developed in Python
- **Bootstrap**: A user interface (UI) framework to develop using CSS/HTML/JavaScript
- **Cordova/PhoneGap**: A mobile framework used to expose native Android and iOS APIs to use while writing JavaScript
- **WordPress**: A CMS developed on PHP
- **.NET**: A full-stack framework developed by Microsoft
- **Foundation**: A UI framework used to build with JavaScript/CSS/HTML
- **Angular.js**, **Backbone.js, and Ember.js**: Front-end JavaScript frameworks
- **Drupal**: A CMS framework developed using PHP

Web Development Languages

Programming languages are used to communicate with other systems and computers and give them necessary instructions. Developers have a plethora of languages to choose from to do the same. Some of these languages are given below:

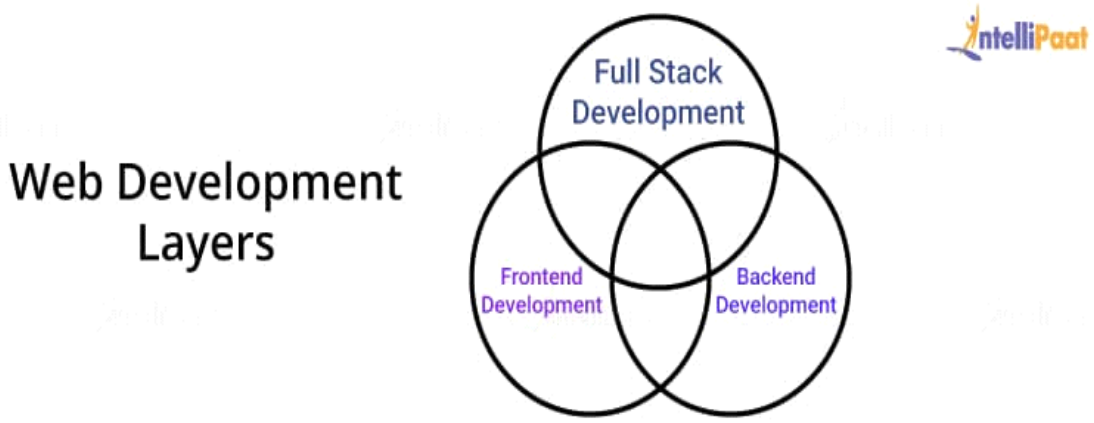
- Python
- Java
- Ruby
- JavaScript
- Go
- Coffee Script
- PHP
- Swift
- Objective-C

Among these, Java Web Development and PHP Web Development are the most frequently used when compared to the others.

- **3 Layers of Web Development**

The Web Development process can broadly be defined in the following three layers:

- Frontend Web Development
- Backend Web Development
- Full Stack Web Development



- **Frontend Web Development**

This type of Web Development helps convert information and available data into a graphical user interface using CSS, HTML, and JavaScript such that users can easily interact with the provided interface. It deals with the part of the website that users can see and easily comprehend.

For instance, on your browser, you as a user see and interact with the website's frontend. It consists of all the things that you can view and experience directly, including colors, text, images, buttons, etc.

- **Backend Web Development**

The backend part of any web page is the portion that users don't or can't see. It is the backbone of the website. Users cannot see it, but it is the part responsible for the functioning of the website. It allows developers to store and organize data, as well as ensures the proper working and functioning of everything available on the frontend or the client-side.

The backend portion interacts with the frontend part of the web page by sending and receiving data that is shown live on the website. When users enter some data, fill out forms, or purchase something, the browser sends that request to the backend (the server-side) which, in turn, sends back the data as frontend code for the web page to comprehend, interpret, and make it visible for you.

- **Full Stack Web Development**

Full Stack development is the combined development of the frontend and the backend of a web page. It includes the development of the graphics and design of the web page, along with managing the database for data organization and storage.

- **Future Scope of Web Development**

With technologies like Machine Learning, Artificial Intelligence (AI), Data Science, and more rising day by day in terms of need and use, they can immensely help in the field of Web Development, thereby increasing the opportunities for Web Developers. No matter what requirements change the future holds, **Web Developer Skills** will always be in demand. Web Development offers some of the most secure and promising careers that you can opt for with numerous job opportunities available.

- **Conclusion**

In this blog, you learned about 'What is Web Development?' and all the requirements needed to build a web page. You also came across various types and layers of Web Development. To learn the necessary skills and become a professional in this field, you can also register for a Web Development course with benefits like online support and job assistance.

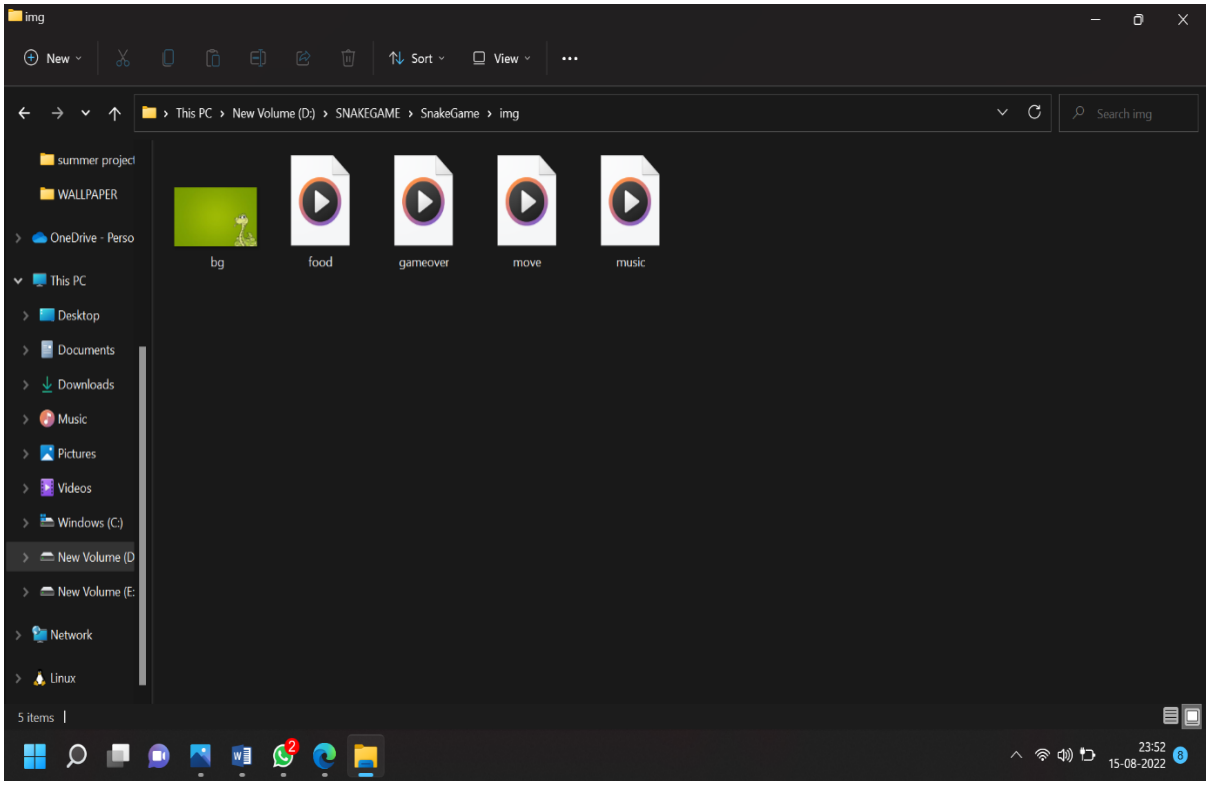
❖ Introduction to Project

➢ Coding Snake Game In JavaScript

We will produce a snake game with the ease of HTML, CSS, and Javascript. Building this game will push you to boost your web development skills. It's very easy to make this game just by using the right tools at the right time. We will use sounds, animations, user input like upward, downward keys , etc. So let's begin with some starter files that I have downloaded in the folder from the internet such as :

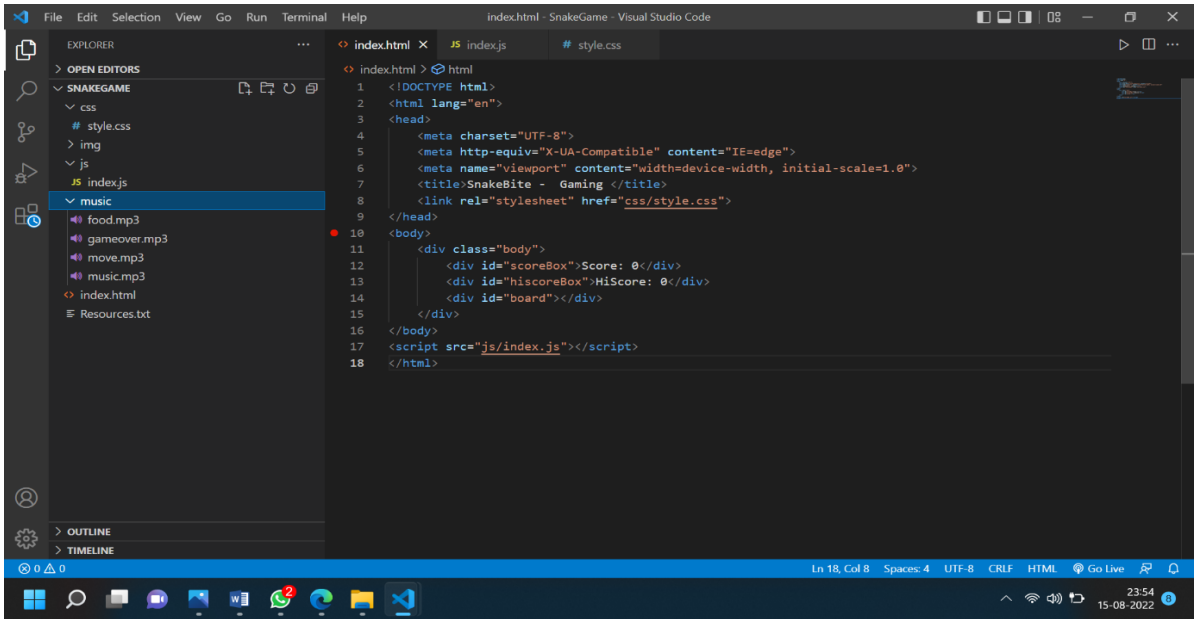
- **bg.jpg** (will use for the background of the game)
- **food.mp3** (will play when the snake eats the food)
- **gameover.mp3** (This sound will be played when the game is over)
- **move.mp3** (will be played when the snake changes the direction)
- **music.mp3** (We will use this file for the background music)

You can also use sounds of your choice or if you are willing to use the same files, it is available in the source code at the end of the blog.



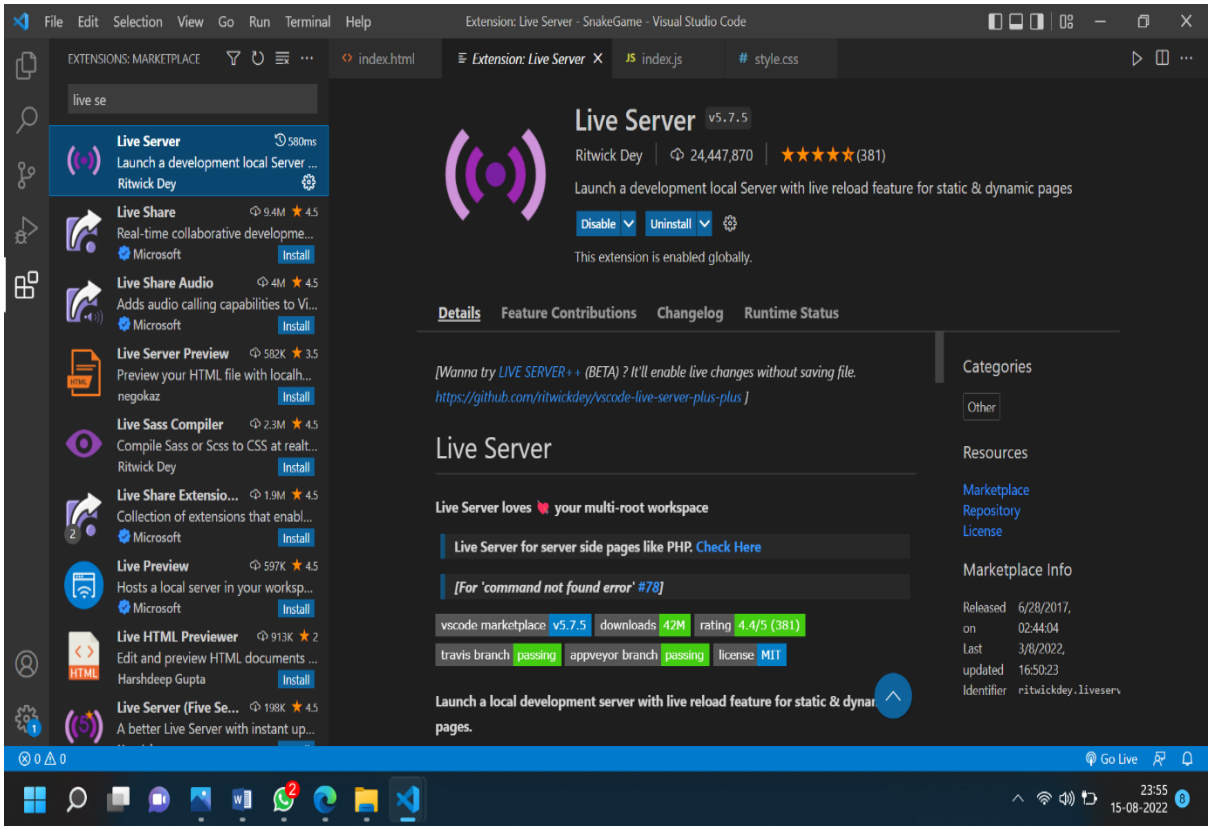
Let's open VS code in the folder then we will create one HTML file named **index.html** and four folders named CSS, **js, img,** and **music**. Now inside the folder named **css,** we will create a CSS file named **style.css**(for styling) and in the folder named **js,** we will create the JavaScript file named **script.js**.

Afterward, we will move the image named **bg.jpg** into the folder named **img** and all the sounds into the folder named **music** as shown below :



Now before moving ahead, let me tell you that I will be using the live server extension to live reload the features of our game. I recommend you to use the same.

To know the process of installation and to run the Live Server look at the video given below:



In the index.html file, we will create a boilerplate. If you are using a text editor and if there is no emmet abbreviation, then you can type it there, and for your time convenience there is one

shortcut in VS code, that is '!+Enter'. This will help you to create a boilerplate automatically. Then we will give the title as “Snake Candy”.

- **CREATING STRUCTURE OF GAME**

Inside the <body> tag of the **index.html** file,

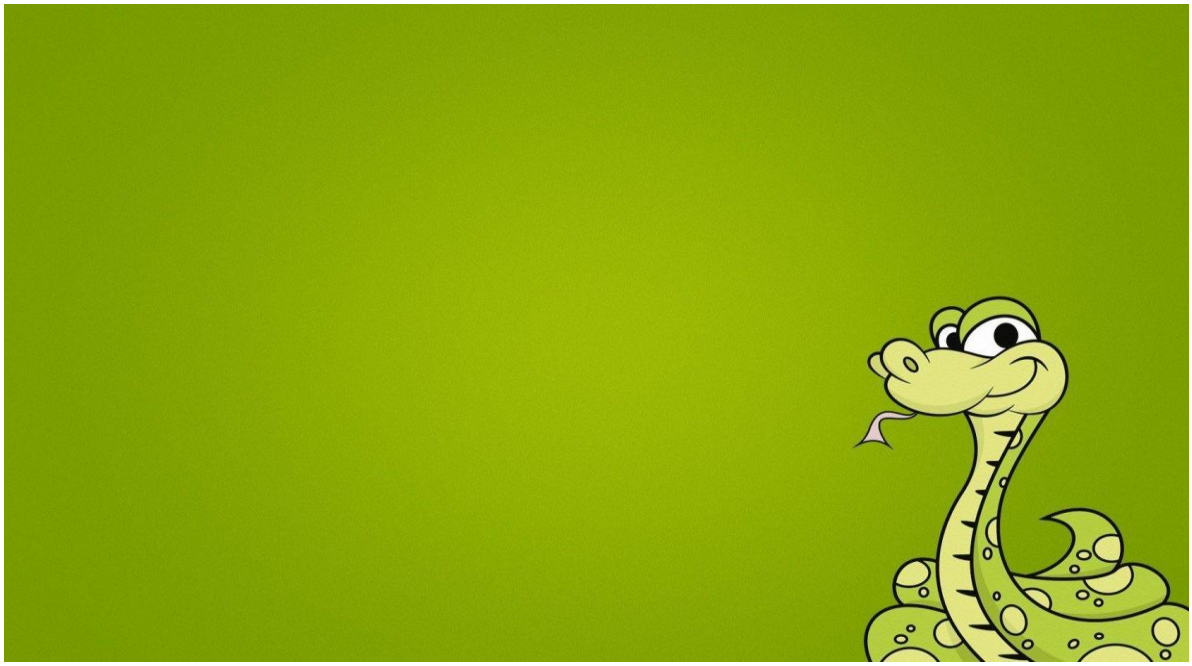
- We will create a ** block** with the class named **body** and inside it, we will create one child ** block** with the id named **board**. This div block will be our board (playing area of the game).
- Now we will create two more child blocks with id named **scorebox** and **highcore** inside block with the class named **body** and above the block with id named **board**. These two blocks will display the current score and high score of the game.
- We will write the text “Score: 0” inside the block with id named **scorebox** and inside block with id named **highscore**, we will write the text “Hiscore: 0”.

1. Styling body

We will use an image named **bg.jpg** for the background. You can also use any image of your choice or if you are willing to use the same image, it is available in the source code at the end of the description. So let’s start styling in the **style.css** file by adding a universal selector to reset CSS, then we will target the block by a class named **body** and will add some CSS properties in it such as :

- **background: url("../img/bg.jpg");**(sets the background image).
- **min-height: 100vh;**(defines the minimum height of an element).
- **background-size: 100vw 100vh;**(specifies the size of the background images).
- **background-repeat: no-repeat;**(sets if/how a background image will be repeated).
- **display: flex;**(specifies the display behavior (the type of rendering box) of an element).
- **justify-content: center;**(aligns the flexible container's items when the items do not use all available space on the main-axis horizontally).
- **align-items: center;**(specifies the default alignment for items inside the flexible container).

The WebPage will look like this:



2. Styling board (playing area of the game)

Let's style the block with the class named **board** which is the child of the block with class named **body** in the **index.html** file, so we will target that block in **style.css** file by class named **board** and in it we will add some CSS properties such as :

- **background: linear-gradient(rgb(170, 236, 170), rgb(236, 236, 167));**(sets the background using linear-gradient() function).
- **width: 90vmin;**(sets the width of an element).
- **height: 92vmin;**(sets the height of an element).
- **border: 2px solid black;**(sets the border on element).
- **display: grid;**(specifies the display behavior (the type of rendering box) of an element).
- **grid-template-rows: repeat(18, 1fr);**(specifies the number and the heights of the rows in a grid layout).
- **grid-template-columns: repeat(18, 1fr);**(specifies the number and the widths of columns in a grid layout).

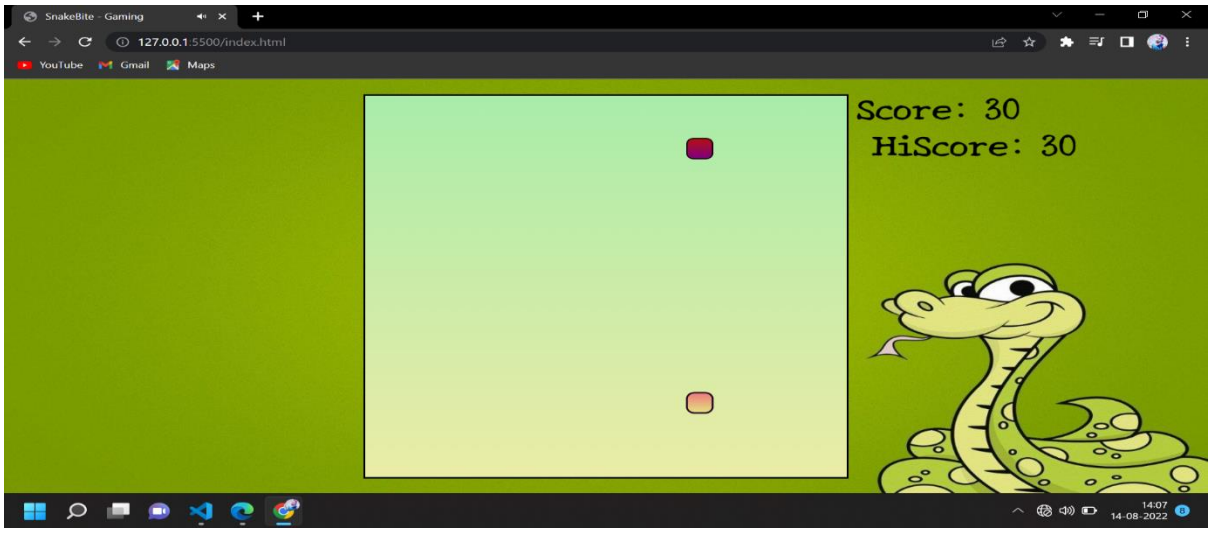
➢ **Now we will target the block of a score box by id named scorebox and we will add some CSS properties such as :**

- **position: absolute;**(specifies the type of positioning method used for an element).
- **top: 9px;**(affects the vertical position of a positioned element).
- **right: 200px;**(affects the horizontal position of a positioned element).
- **font-size: 39px;**(sets the size of a font).
- **font-weight: bold;**(sets how thick or thin characters in text should be displayed).
- **font-family: 'New Tegomin', serif;**(specifies the font for an element).

Let's target the block of a high score by id named **highscore** and will add some CSS properties such as :

- **position: absolute;**(specifies the type of positioning method used for an element).
- **top: 59px;**(affects the vertical position of a positioned element).
- **right: 140px;**(affects the horizontal position of a positioned element).
- **font-size: 39px;**(sets the size of a font).
- **font-weight: bold;**(sets how thick or thin characters in text should be displayed).
- **font-family: 'New Tegomin', serif;**(specifies the font for an element).

The WebPage will look like this:



4. Styling of the head, snake, and food

We will style the **head** which will be used in our JavaScript file to display the head of the snake so we will add some CSS properties in **head** such as :

- **background: linear-gradient(rgb(240, 124, 124), rgb(228, 228, 129));**(sets the background using linear-gradient() function).
- **border: 2px solid rgb(34, 4, 34);**(sets the border on element).
- **transform: scale(1.02);**(applies a 2D or 3D transformation to an element).
- **border-radius: 9px;**(defines the radius of the element's corners).

You can refer to the code from the snippet given below.

Let's style the **snake** which will be used in our JavaScript file to display the snake so we will add some CSS properties in the **snake** such as :

- **background-color: purple;**(sets the background color of an element).
- **border: .25vmin solid white;**(sets the border on element).
- **border-radius: 12px;**(defines the radius of the element's corners).

And the last one is **food** which will be used in our JavaScript file to display the food so we will add some CSS properties in **food** such as :

- **background: linear-gradient(red, purple);**(sets the background using linear-gradient() function).
- **border: .25vmin solid black;**(sets the border on element).
- **border-radius: 8px;**(defines the radius of the element's corners).

JAVASCRIPT FOR GAME

- **Game Constants & Variables**

We will create all constants and variables in a JavaScript file named **java.script** that will be used in our game as shown below.

- **Game Functions**

1. Main Function

We will create the main function for rendering and here instead of the **setInterval()** method we will use the **window.requestAnimationFrame()** function for the reason . The **ctime** is our current time and we will create an if statement for time such that it will render the game when

the condition is false and after that, this function will call **gameEngine()** function which will be created later. You can refer to the code from the snippet given below.

2. Function for collision

We will create the function named **isCollide** and give parameter **snake**. First, we will write the logic for if a snake bumps into **itself,** for that we will create a loop in **snakeArr** and then create the if statement in it with the condition for collision into itself and if the condition is true it will return true. Now we will write the logic for if the snake bumps into the **wall**, for that we will create one more if statement with the condition for collision into the wall and if the condition is true it will return true. **3. gameEngine() Function**

Inside this function, we will write the logic for,

- updating the snake array and food.
- if the snake has eaten the food, increment the score and regenerate the food.
- moving the snake.
- displaying the snake.
- displaying the food.

All these points will see one by one as follows :

- **Logic for updating the snake array and food**

  This function will update the snake array and food. At first, we will create an if statement with condition that if the function **isCollide()** with parameter **snakeArr** returns true means the snake collides with itself or with the wall, then it will play **gameOverSound** and the background music that is **musicSound** will be paused and will set the **inputDir** to its initial value then we will create an alert with text message "Game Over. Press any key to play again!". After that we will reset the **snakearr** then it will play **musicSound** again and it will set the score equal to zero.

- **Logic for if the snake has eaten the food, increment the score and regenerate the food**

  Now we will create an if statement with the condition that if the snake has eaten the food then we will play the **foodSound** and increment the score by 1. We will store the highscore in

localstorage and Whenever the snake eats the food we will add one head to it using the **unshift()** function. After the snake eats the food we will regenerate the food by generating random values between a and b which are the variables created with the values such as a=2 and b=16.

- **Logic for moving the snake**

  To move the snake we will iterate the whole snake body for that we will create a decrementing **for loop** in **snakArr** and will write some logic to shift the element one by one from the end.

- **Logic for displaying the snake**

  For displaying the snake first we will clear the innerHTML of the **board** so when the game loads the snake will render only once. Then we will create a for Each loop in **snakeArr** and inside it, we will create a new element **div** inside a variable named **snakeElement**, then giving style to it such as **gridRowStart** is **e.y** and **grid Cloumn Start** is **e.x**. Now we will create an if-else statement such as if the index is equal to 0 then the class named **head** will be added to **snakeElement** else the class named **snake** will be added to **snakeElement**. After that we will append the **snakeElement** as a child inside the **board**

- **Logic for displaying the food**

  Let's create an element **div** in a variable named **foodElement** and will give the style such as **gridRowStart** is **food.y** and **gridColumnStart** is **food.x** then inside **foodElement** we will add the class named **food**. After that, we will append the **foodElemnet** as a child inside the **board**

2. High Score of the game

We will create a variable named **high score** to store the high score from local storage. Now we will create an if-else statement for the high score such as if the high score is null then set the **his coreval** to 0 and update it in local storage else fetch the high score and set it into the inner HTML of **high score Box** .

❖ HARDWARE AND SOFTWARE REQUIREMENTS

➢ HARDWARE REQUIREMENT

Hardware requirements include that hardware which is required for its working. It includes:

- Pentium 1 Computer
- 100 MB RAM

➢ SOFTWARE REQUIREMENTS

The technical specifications of requirements for the software are as follows:

- Any Operating System (Windows, Linux, MAC)
- Visual studio
- Any web browser(Chrome , Firefox , etc)

❖ Objective of Project

This game aims to change the way people think of traditional snake game. It will offer the experience of commercial multilayer games to the player retaining the simplicity of traditionl snake game.The major objectives of this project are:

● Create a snake game that will have all the functionality of traditional snake games.

● Introduce multilayer functionality in the game that will allow several players to play a game simultaneously. It should be able to give the experience of a real time multiplayer game to the players.

● Introduce computer controlled intelligent opponent (unique feature of this game) to make the game more challenging and interesting. The movement and action of these intelligent opponents will be controlled by computer whose aim will be to eat the food before human players capture it.

❖ CODING

➢ HTML CODES

```
<!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">
    <title>SnakeMania - Ek Gaming Katha</title>
    <link rel="stylesheet" href="css/style.css">
</head>
<body>
    <div class="body">
        <div id="scoreBox">Score: 0</div>
        <div id="hiscoreBox">HiScore: 0</div>
        <div id="board"></div>
    </div>
</body>
<script src="js/index.js"></script>
</html>
```

➢ **CSS**

```
@import
url('https://fonts.googleapis.com/css2?family=New+Tegomin&display=sw
ap');
*{
    padding: 0;
    margin: 0;
}

.body{
    background: url("../img/bg.jpg");
    min-height: 100vh;
    background-size: 100vw 100vh;
    background-repeat: no-repeat;
    display: flex;
    justify-content: center;
    align-items: center;
}

#scoreBox{
    position: absolute;
    top: 9px;
    right: 200px;
    font-size: 39px;
    font-weight: bold;
    font-family: 'New Tegomin', serif;
}
```

```css
#hiscoreBox{
    position: absolute;
    top: 59px;
    right: 140px;
    font-size: 39px;
    font-weight: bold;
    font-family: 'New Tegomin', serif;
}

#board{
     background: linear-gradient(rgb(170, 236, 170), rgb(236, 236, 167));
    width: 90vmin;
    height: 92vmin;
    border: 2px solid black;
    display: grid;
    grid-template-rows: repeat(18, 1fr);
    grid-template-columns: repeat(18, 1fr);
}

.head{
     background: linear-gradient(rgb(240, 124, 124), rgb(228, 228, 129));
    border: 2px solid rgb(34, 4, 34);
    transform: scale(1.02);
    border-radius: 9px;
}

.snake{
    background-color: purple;
    border: .25vmin solid white;
    border-radius: 12px;
}
```

```
.food{
    background: linear-gradient(red, purple);
    border: .25vmin solid black;
    border-radius: 8px;
}
```

- **JS**

```
// Game Constants & Variables
let inputDir = {x: 0, y: 0};
const foodSound = new Audio('music/food.mp3');
const gameOverSound = new Audio('music/gameover.mp3');
const moveSound = new Audio('music/move.mp3');
const musicSound = new Audio('music/music.mp3');
let speed = 5;
let score = 0;
let lastPaintTime = 0;
let snakeArr = [
    {x: 13, y: 15}
];

food = {x: 6, y: 7};

// Game Functions
function main(ctime) {
```

```
    window.requestAnimationFrame(main);
    // console.log(ctime)
    if((ctime - lastPaintTime)/1000 < 1/speed){
        return;
    }
    lastPaintTime = ctime;
    gameEngine();
}

function isCollide(snake) {
    // If you bump into yourself
    for (let i = 1; i < snakeArr.length; i++) {
        if(snake[i].x === snake[0].x && snake[i].y === snake[0].y){
            return true;
        }
    }

    // If you bump into the wall

     if(snake[0].x >= 18 || snake[0].x <=0 || snake[0].y >= 18 ||
snake[0].y <=0){
        return true;
    }

    return false;
}

function gameEngine(){

    // Part 1: Updating the snake array & Food

    if(isCollide(snakeArr)){
        gameOverSound.play();
```

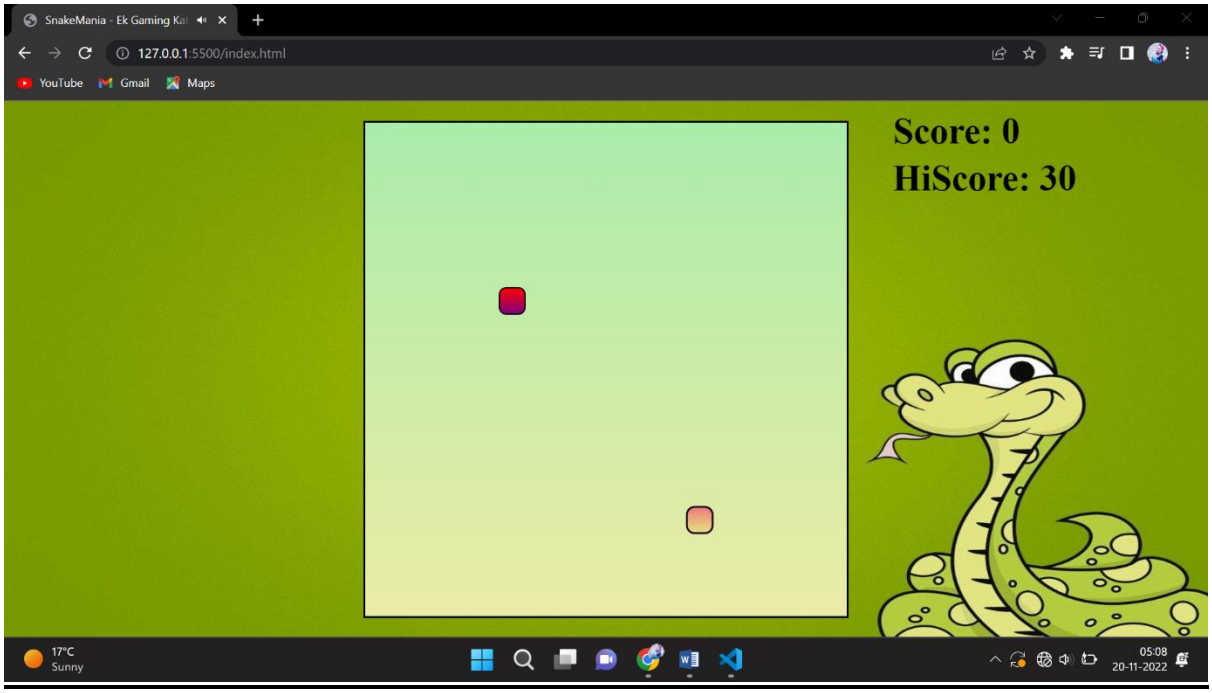
```
        musicSound.pause();
        inputDir =  {x: 0, y: 0};
        alert("Game Over. Press any key to play again!");
        snakeArr = [{x: 13, y: 15}];
        musicSound.play();
        score = 0;
    }

    // If you have eaten the food, increment the score and regenerate
the food

    if(snakeArr[0].y === food.y && snakeArr[0].x ===food.x){
        foodSound.play();
        score += 1;
        if(score>hiscoreval){
            hiscoreval = score;
                                    localStorage.setItem("hiscore",
JSON.stringify(hiscoreval));
            hiscoreBox.innerHTML = "HiScore: " + hiscoreval;
        }
        scoreBox.innerHTML = "Score: " + score;
             snakeArr.unshift({x:  snakeArr[0].x  +  inputDir.x,  y:
snakeArr[0].y + inputDir.y});
        let a = 2;
        let b = 16;
           food  =  {x:  Math.round(a  +  (b-a)*  Math.random()),  y:
Math.round(a + (b-a)* Math.random())}
    }

    // Moving the snake

    for (let i = snakeArr.length - 2; i>=0; i--) {
        snakeArr[i+1] = {...snakeArr[i]};
```

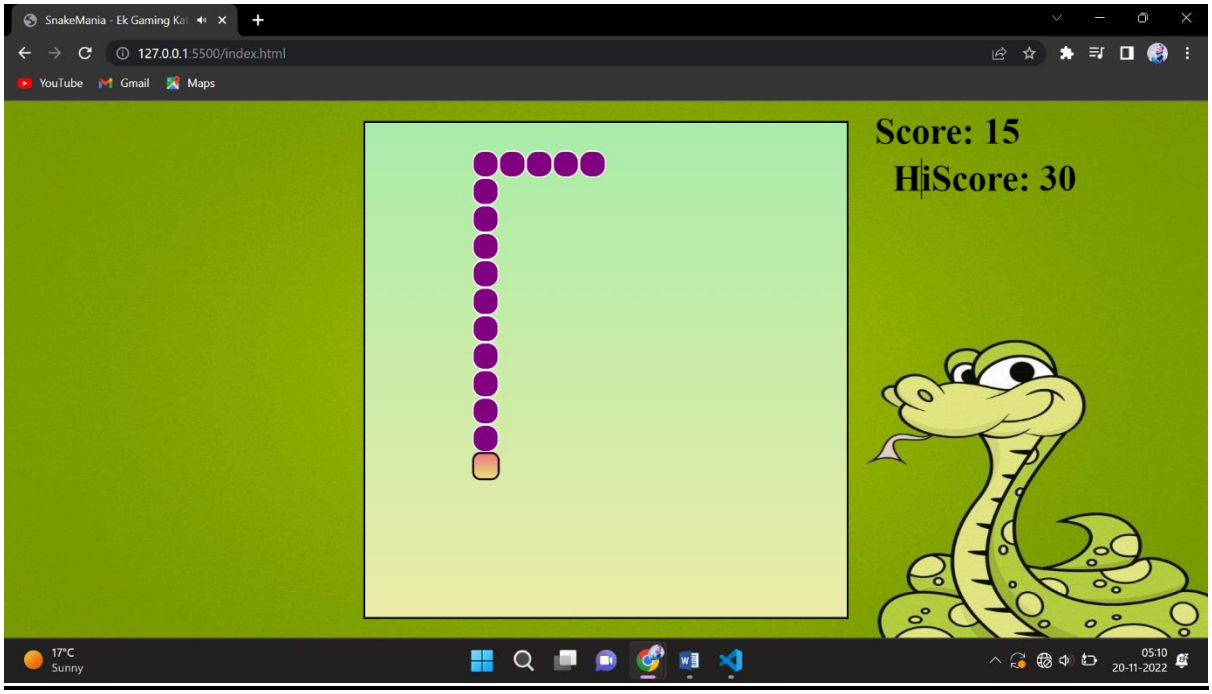
```
    }

    snakeArr[0].x += inputDir.x;
    snakeArr[0].y += inputDir.y;




    // Part 2: Display the snake and Food
    // Display the snake


    board.innerHTML = "";
    snakeArr.forEach((e, index)=>{
        snakeElement = document.createElement('div');
        snakeElement.style.gridRowStart = e.y;
        snakeElement.style.gridColumnStart = e.x;

        if(index === 0){
            snakeElement.classList.add('head');
        }
        else{
            snakeElement.classList.add('snake');
        }
        board.appendChild(snakeElement);
    });

    // Display the food

    foodElement = document.createElement('div');
    foodElement.style.gridRowStart = food.y;
    foodElement.style.gridColumnStart = food.x;
```

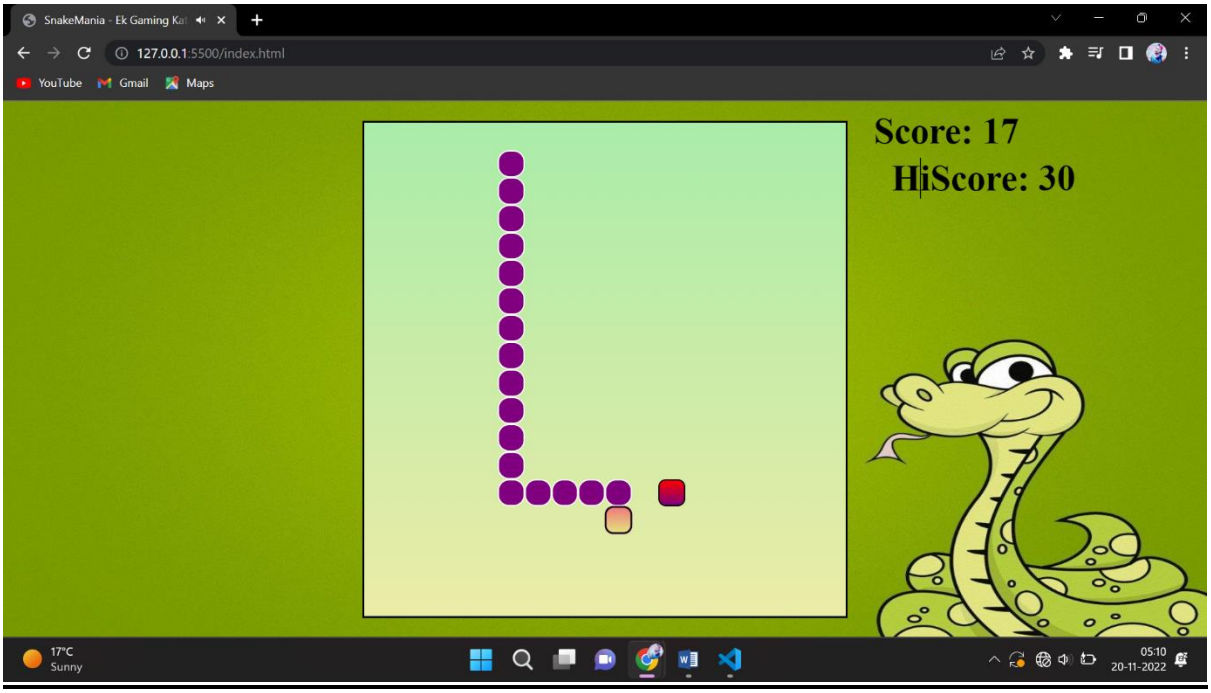
```
    foodElement.classList.add('food')
    board.appendChild(foodElement);


}


// Main logic starts here

musicSound.play();
let hiscore = localStorage.getItem("hiscore");
if(hiscore === null){
    hiscoreval = 0;
    localStorage.setItem("hiscore", JSON.stringify(hiscoreval))
}
else{
    hiscoreval = JSON.parse(hiscore);
    hiscoreBox.innerHTML = "HiScore: " + hiscore;
}

window.requestAnimationFrame(main);
window.addEventListener('keydown', e =>{
    inputDir = {x: 0, y: 1} // Start the game
    moveSound.play();
    switch (e.key) {
        case "ArrowUp":
            console.log("ArrowUp");
            inputDir.x = 0;
            inputDir.y = -1;
            break;

        case "ArrowDown":
            console.log("ArrowDown");
            inputDir.x = 0;
```

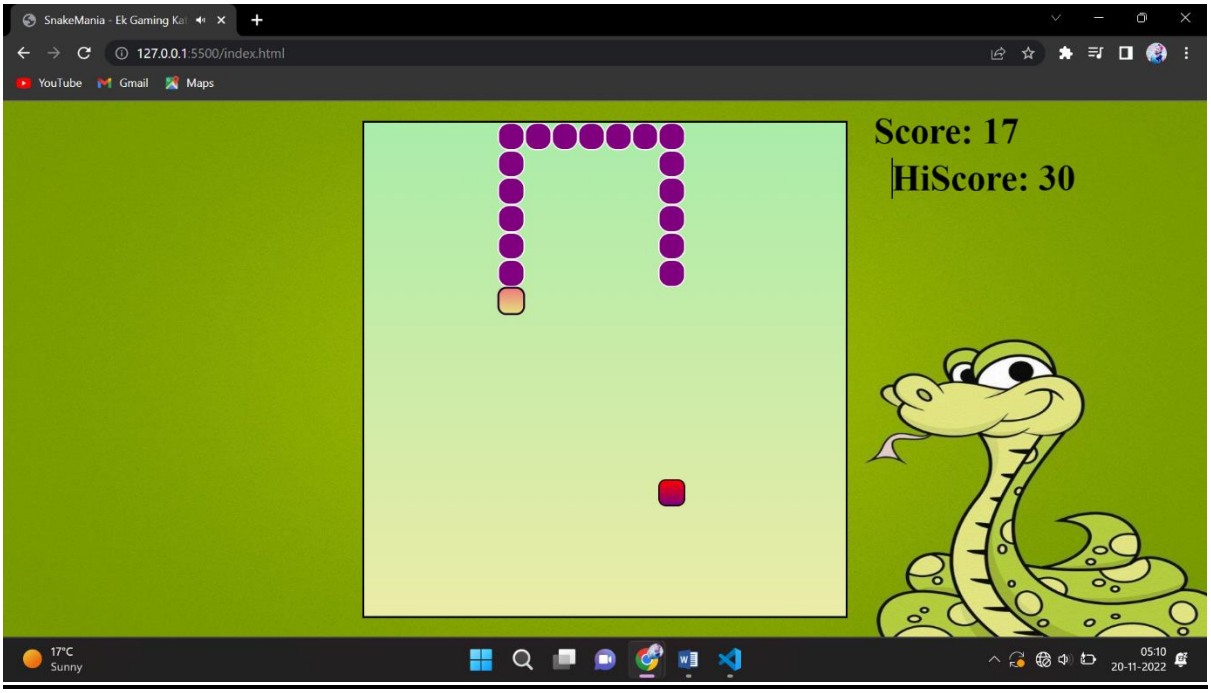
```
            inputDir.y = 1;
            break;

        case "ArrowLeft":
            console.log("ArrowLeft");
            inputDir.x = -1;
            inputDir.y = 0;
            break;

        case "ArrowRight":
            console.log("ArrowRight");
            inputDir.x = 1;
            inputDir.y = 0;
            break;
        default:
            break;
    }

});
```

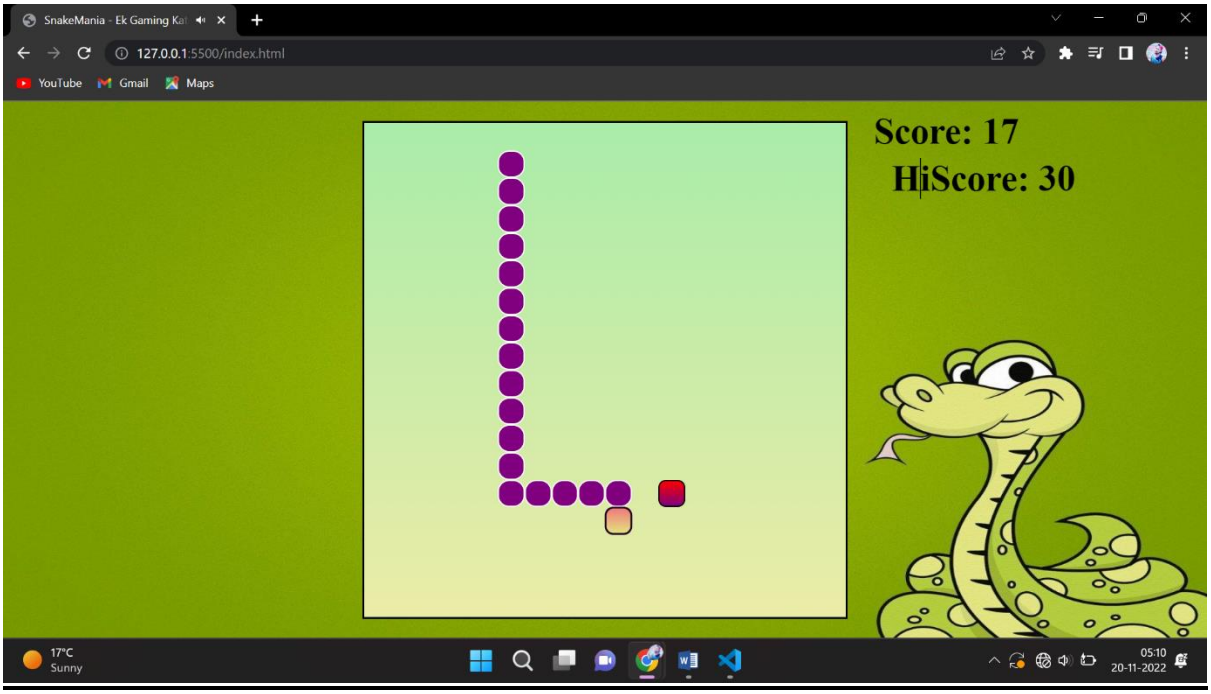
❖ SNAPSHOTS


SnakeMania - Ek Gaming Ka
127.0.0.1:5500/index.html
YouTube
Gmail
Maps
Score: 0
HiScore: 30
17°C
Sunny
05:08
20-11-2022


SnakeMania - Ek Gaming Ka
127.0.0.1:5500/index.html
YouTube
Gmail
Maps
Score: 15
HiScore: 30
17°C
Sunny
05:10
20-11-2022


SnakeMania - Ek Gaming Ka
127.0.0.1:5500/index.html
YouTube
Gmail
Maps
Score: 17
HiScore: 30
17°C
Sunny
05:10
20-11-2022


SnakeMania - Ek Gaming Ka
127.0.0.1:5500/index.html
YouTube
Gmail
Maps
Score: 17
HiScore: 30
17°C
Sunny
05:10
20-11-2022


SnakeMania - Ek Gaming Ka
127.0.0.1:5500/index.html
YouTube
Gmail
Maps
Score: 17
HiScore: 30
17°C
Sunny
05:10
20-11-2022

❖ Features of Project

- **The Six Main Features of Good Game Design**

1) Game Controls. Controls are the rudiments of a game. ...
2) Payoff. Games are largely enjoyable because they're rewarding. ...
3) Graphics. Humans just can't help it. ...
4) Storyline. ...
5) Strategy and Surprise. ...
6) Rising Challenges.

❖ Front End

- HTML
- CSS
- JAVASCRIPT

➢ **HTML**

HTML stands for Hypertext Markup Language, and it is the most widely used language to write Web Pages.

• Hypertext refers to the way in which Web pages (HTML documents) are linked together. Thus, the link available on a webpage is called Hypertext.

• As its name suggests, HTML is a Markup Language which means you use HTML to simply "mark-up" a text document with tags that tell a Web browser how to structure it to display.

Originally, HTML was developed with the intent of defining the structure of documents like headings, paragraphs, lists, and so forth to facilitate the sharing of scientific information between researchers.

Now, HTML is being widely used to format web pages with the help of different tags available in HTML language.

- **Advantages And Disadvantages Of HTML:**

HTML(Hypertext Markup Language) is the language used to foster website pages. It's anything but a real programming language yet a markup language. Hypertext is the text that has a connection installed into it that focuses on an alternate page or site. Predominantly HTML is utilized for organizing a website page and making an establishment. Fundamentally, Html is the foundation of website pages. Every one of the sites you see on the web is utilizing HTML dependent to some degree.

Each web engineer or website specialist needs to learn HTML, in the first place. The most recent form of HTML is HTML5 which is truly present-day and strong. It works great alongside CSS3. In the event that you are considering realizing this language, you should know these focuses. There is a lot of weaknesses and drawbacks of HTML.

- **CSS**

Cascading **S**tyle **S**heets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.

CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs,variations in display for different devices and screen sizes as well as a variety of other effects.

CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

Advantages of CSS

- **CSS saves time** − You can write CSS once and then reuse same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.
- **Pages load faster** − If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So less code means faster download times.
- **Easy maintenance** − To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.
- **Superior styles to HTML** − CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.
- **Multiple Device Compatibility** − Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cell phones or for printing.
- **Global web standards** − Now HTML attributes are being deprecated and it is being recommended to use CSS. So its a good idea to start using CSS in all the HTML pages to make them compatible to future browsers.

Who Creates and Maintains CSS?

CSS is created and maintained through a group of people within the W3C called the CSS Working Group. The CSS Working Group creates documents called specifications. When a specification has been discussed and officially ratified by the W3C members, it becomes a recommendation.

These ratified specifications are called recommendations because the W3C has no control over the actual implementation of the language. Independent companies and organizations create that software.

NOTE − The World Wide Web Consortium, or W3C is a group that makes recommendations about how the Internet works and how it should evolve.

CSS Versions

Cascading Style Sheets level 1 (CSS1) came out of W3C as a recommendation in December 1996. This version describes the CSS language as well as a simple visual formatting model for all the HTML tags.

CSS2 became a W3C recommendation in May 1998 and builds on CSS1. This version adds support for media-specific style sheets e.g. printers and aural devices, downloadable fonts, element positioning and tables.

➢ JAVA SCRIPT

JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

JavaScript was first known as **LiveScript,** but Netscape changed its name to JavaScript, possibly because of the excitement being generated by Java. JavaScript made its first appearance in Netscape 2.0 in 1995 with the name **LiveScript**. The general-purpose core of the language has been embedded in Netscape, Internet Explorer, and other web browsers.

The ECMA-262 Specification defined a standard version of the core JavaScript language.

- JavaScript is a lightweight, interpreted programming language.
- Designed for creating network-centric applications.
- Complementary to and integrated with Java.
- Complementary to and integrated with HTML.
- Open and cross-platform

Client-Side JavaScript

Client-side JavaScript is the most common form of the language. The script should be included in or referenced by an HTML document for the code to be interpreted by the browser.

It means that a web page need not be a static HTML, but can include programs that interact with the user, control the browser, and dynamically create HTML content.

The JavaScript client-side mechanism provides many advantages over traditional CGI server-side scripts. For example, you might use JavaScript to check if the user has entered a valid e-mail address in a form field.

The JavaScript code is executed when the user submits the form, and only if all the entries are valid, they would be submitted to the Web Server.

JavaScript can be used to trap user-initiated events such as button clicks, link navigation, and other actions that the user initiates explicitly or implicitly.

- **Advantages of JavaScript**

The merits of using JavaScript are −

- **Less server interaction** − You can validate user input before sending the page off to the server. This saves server traffic, which means less load on your server.
- **Immediate feedback to the visitors** − They don't have to wait for a page reload to see if they have forgotten to enter something.
- **Increased interactivity** − You can create interfaces that react when the user hovers over them with a mouse or activates them via the keyboard.
- **Richer interfaces** − You can use JavaScript to include such items as drag-and-drop components and sliders to give a Rich Interface to your site visitors.

- **Limitations of JavaScript**

We cannot treat JavaScript as a full-fledged programming language. It lacks the following important features −

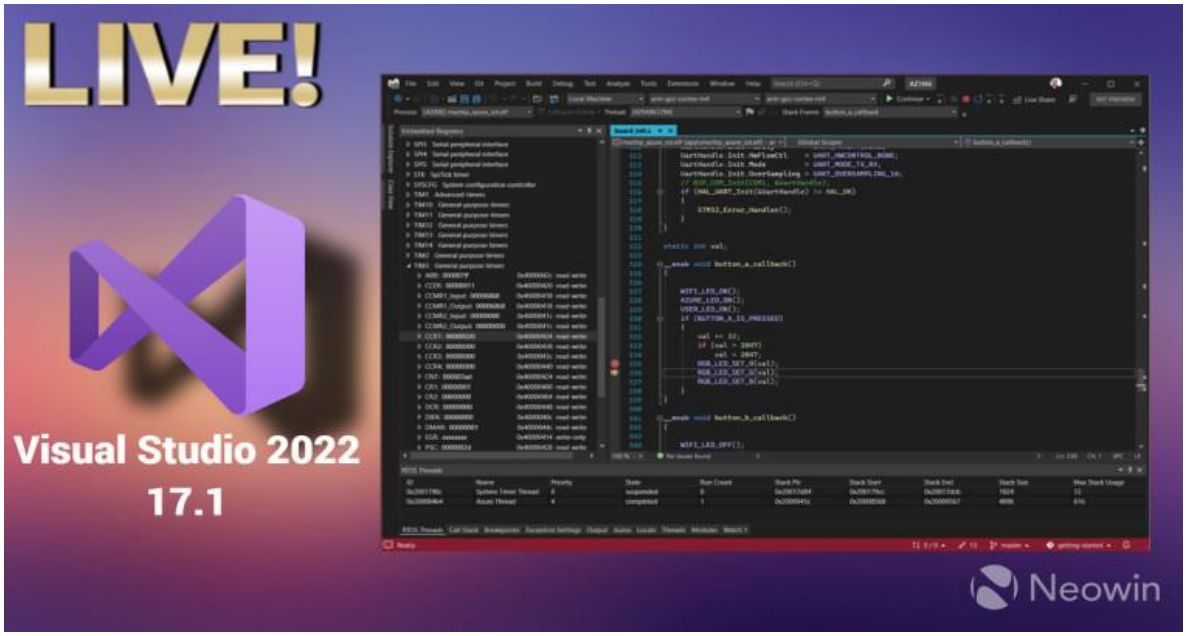
- Client-side JavaScript does not allow the reading or writing of files. This has been kept for security reason.
- JavaScript cannot be used for networking applications because there is no such support available.
- JavaScript doesn't have any multi-threading or multiprocessor capabilities.

Once again, JavaScript is a lightweight, interpreted programming language that allows you to build interactivity into otherwise static HTML pages.

❖ TOOL USE

- **Visual Studio**

  Microsoft Visual Studio is an IDE made by Microsoft and used for different types of software development such as computer programs, websites, web apps, web services, and mobile apps. It contains completion tools, compilers, and other features to facilitate the software development process.



History of Visual Studio

Visual Studio has been around for over 20 years. Its first version was Visual Studio 97. Since then there were a lot of different versions, the current one is Microsoft Visual Studio 2019.

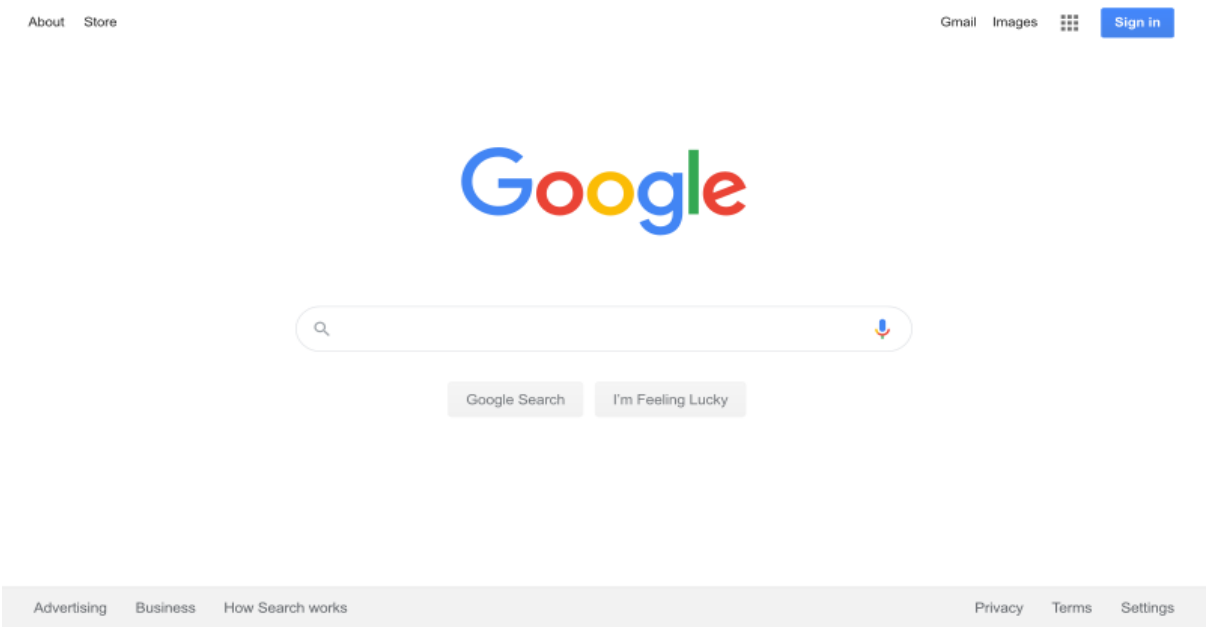
How does it work?

The Visual Studio IDE (integrated development environment) is a software program for developers to write and edit their code. Its user interface is used for software development to edit, debug and build code. Visual Studio includes a code editor supporting IntelliSense (the code completion component) as well as code refactoring. The integrated debugger works both as a source-level debugger and a machine-level debugger. Other built-in tools include a code

profiler, designer for building GUI applications, web designer, class designer, and database schema design

- **GOOGLE CHROME**

Google Chrome is a cross-platform web browser developed by Google. It was first released in 2008 for Microsoft Windows, built with free software components from Apple WebKit and Mozilla Firefox. It was later ported to Linux, macOS, iOS, and Android, where it is the default browser. The browser is also the main component of ChromeOS, where it serves as the platform for web applications.



Most of Chrome's source code comes from Google's free and open-source software project Chromium, but Chrome is licensed as proprietary freeware. WebKit was the original rendering engine, but Google eventually forked it to create the Blink engine; all Chrome variants except iOS now use Blink. 122

As of October 2021, StatCounter estimates that Chrome has a 68% worldwide browser market share (after peaking at 72.38% in November 2018) on personal computers (PC), is most used on tablets (having surpassed Safari), and is also dominant on smartphones, and at 65% across all platforms combined. Because of this success, Google has expanded the "Chrome" brand name to other products: ChromeOS, Chromecast, Chromebook, Chromebit, Chromebox, and Chromebase.

❖ Reference

- ICE Technology Lab
- www.stackoverflow.com
- www.gfg.com
- www.w3school.com
- www.tutorialspoint.com
- www.github.com
- www.android.com