A

CREATIVITY AND INOVATION

PROJECT REPORT

on

"TIC TAC TOE GAME"

submitted in the partial fulfillment of the requirement for the award of degree of

BACHELOR OF TECHNOLOGY

on

COMPUTER SCIENCE & ENGINEERING
In Specialization with IBM Big Data

SUBMITTED BY
Anmol Prakash
(A40219016)
Batch: 2019-2023

Under the guidance of
Ms. Nishika
(Asst. Professor. Department of CSE)

Department of Computer Science & Engineering
Faculty of Engineering & Technology
P.D.M. University, Bahadurgarh

pdm UNIVERSITY
Established under the Haryana Private Universities (Amendment) Act, 2015

CERTIFICATE OF COMPLETION

This is to certify that the Project work entitled "**TIC TAC TOE GAME**" submitted by **Anmol Prakash** in fulfillment for the requirements of the award of "**Bachelor of Technology**" Degree in **"Computer Science & Engineering**" **(Big Data IBM)** at PDMU, Bahadurgarh, Haryana is an authentic work carried out by him under my supervision and guidance. To the best of my knowledge, the matter embodied in the project has not been submitted to any other University / Institute for the award of any Degree .

Guide

Ms. Nishika

A.P. CSE Deptt.

Faculty of Engineering & Technology

DECLARATION BY THE CANDIDATE

I hereby declare that the work presented in this report entitled **"TIC TAC TOE GAME"**, in fulfilment of the requirement for the award of the degree Bachelor of Technology in Computer Science & Engineering, submitted in CSE Department, PDMU, Bahadurgarh, Haryana is an authentic record of our own work carried out during our degree under the guidance of **Ms. Nishika**.

The work reported in this has not been submitted by me for award of any other degree or diploma.

Anmol Prakash
A40219016

ACKNOWLEDGEMENT

We are delighted to express our heartily and sincere gratitude and indebtedness to my project guide **Ms. Nishika, Department of Computer Science and Engineering, PDM University, Bahadurgarh, Haryan**a for her invaluable guidance, valuable advice and wholehearted cooperation. Her continuous inspiration, timely help and constructive criticism has made us complete this Project.

I would also like to convey my deep regards to all other faculty members who have bestowed their great effort and guidance at appropriate times without which it would have been very difficult on my part to finish this work. Finally I would also like to thank my friends for their advice and pointing out my mistakes.

Anmol Prakash
A40219016

ABSTRACT

Tic Tac Toe in Html game itself is very simple, but the programming concept is not. This system is a game application that is developed in HTML CSS platform. Moreover, this system is an easy and fundamental level tiny project for teaching purpose. Besides, the user can also modify this scheme according to their requirements. They can also create a perfect advance level project. Instead, there is no database as a back end of the system.

The game basically shows board which consists of nine squares. Each of these squares has an index number starting at the top left corner and ending at the bottom correct corner with the number 0. However, the game rule has 2 players. Moreover, this game is a simple tic tac toe game like that of the traditional which we typically play in 80's and 90's but in the modified way. Thus, user can reminisce the things of their childhood as well.

In this particular game each user have X and O to implement in 3X3 table. Also, your score will be reset and the score or number of wins by particular user is displayed over the screen.

TABLE OF CONTENT

TITLE **PAGE No.**

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviation	Description
HTML	Hyper Text Markup Language
SGML	Standard Generalised Markup Language
WWW	World Wide Web
XHTML	Extensible Hypertext Markup Language
IP	Internet Protocol
MIME	Multipurpose Internet Mail Extensions
W3C	World Wide Web Consortium
IIS	Internet Information Services
MySQL	My Structure Query Language
VSCode	Visual Studio Code
href	Hypertext Reference

CHAPTER 1

INTRODUCTION

Tic-tac-toe is not a very challenging game for human beings. If you're an enthusiast, you've probably moved from the basic game to some variant like three dimensional tic- tac-toe on a larger grid. If you sit down right now to play ordinary three-by-three tic-tac- toe with a friend, what will probably happen is that every game will come out a tie. Both you and your friend can probably play perfectly, never making a mistake that would allow your opponent to win. But can you describe how you know where to move each turn? Most of the time, you probably aren't even aware of alternative possibilities; you just look at the board and instantly know where you want to move. That kind of instant knowledge is great for human beings , because it makes you a fast player. But it isn't much help in writing a computer program. For that, you have to know very explicitly what your strategy is.

1.1 WEB DEVELOPMENT

Web development refers to the building, creating, and maintaining of websites. It includes aspects such as web design, web publishing, web programming, and database management. It is the creation of an application that works over the internet i.e. websites.

The word Web Development is made up of two words, that is:

- **Web:** It refers to websites, web pages or anything that works over the internet.
- **Development:** Building the application from scratch.

1.1.1 Web Development can be classified into two ways

Frontend Development.

Backend Development.

Frontend Development: The part of a website that the user interacts directly is termed as front end. It is also referred to as the 'client side' of the application.

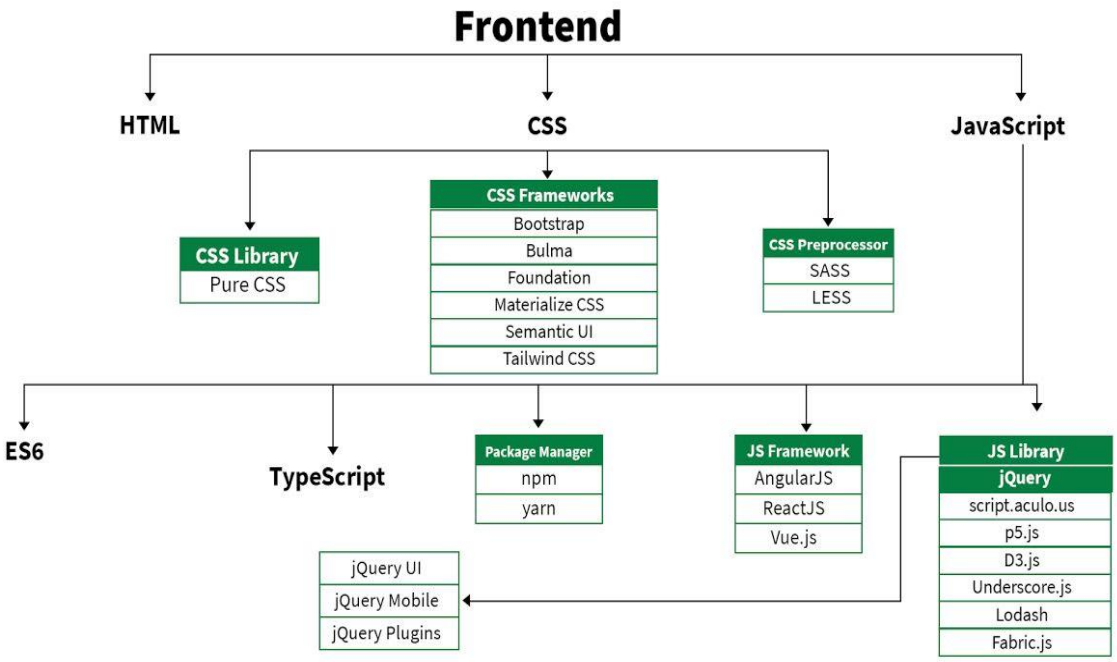

Figure 1.1: Frontend Flowchart

Backend Development: Backend is the server side of a website. It is the part of the website that users cannot see and interact. It is the portion of software that does not come in direct contact with the users. It is used to store and arrange data.

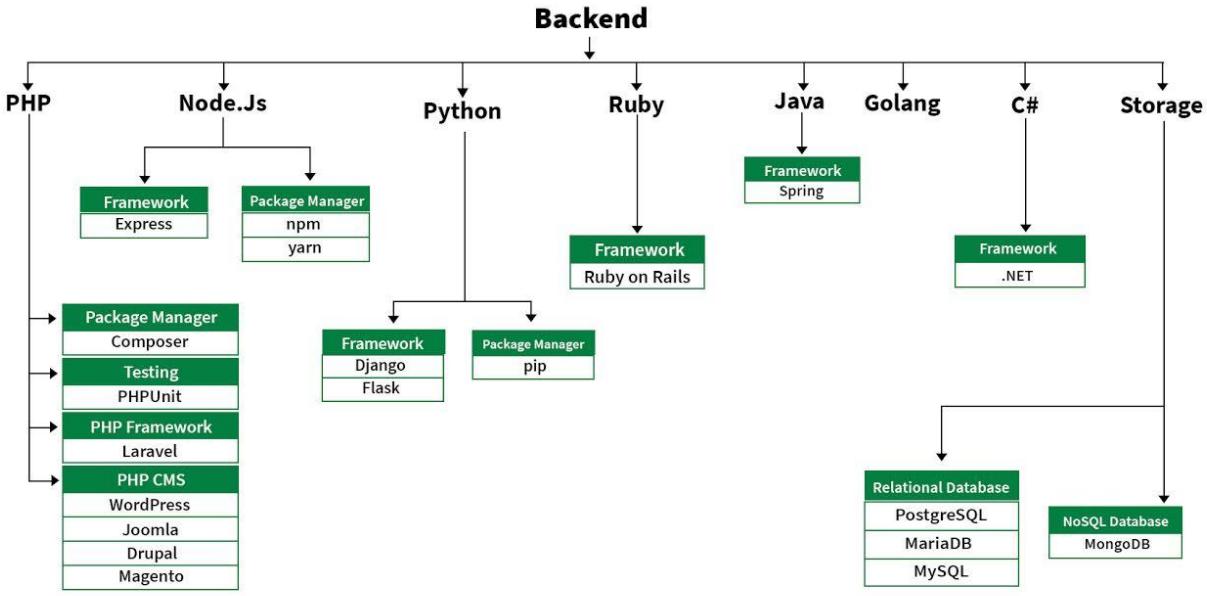

Figure 1.2 : Backend Flowchart

1.1.2 Working of website

Before you begin creating your own website and launch it to the Internet, it's important to know how websites work.

Here are some basic terms:

- A website is simply a collection of web pages of codes – codes that describes the layout, format and content on a page.
- The web server is a internet-connected computer that receives the request for a web page sent by your browser.
- The browser connects your computer to the server through an IP address. The IP address is obtained by translating the domain name. (**Don't worry, this part is all done automatically by your browser so you don't have to look up the IP addresses yourself.**)

In other words, in order to display your website on the Internet, you will need:

- A website
- A domain name
- A server

1.1.3 Different Types Of Website

- **BUSINESS WEBSITE**

  A business website is a website that represents a business or multiple businesses. If you own a business or planning to start one, you can go for a business website.

- **E-COMMERCE WEBSITE**

  An E-commerce website is a website where you can buy products and make transactions. For example, Amazon, Flipkart, etc.

- **PERSONAL WEBSITE**

  People create a personal website to demonstrate their art or share their thoughts or display their skills.

- **ENTERTAINMENT WEBSITE**

  An entertainment website is a website whose sole purpose is to provide entertaining content. Netflix, Tvf play, primevideo.com are a few examples of it.

- **EDUCATIONAL WEBSITE**

  An educational website majorly features educational content on its wall. They might sell online courses or provide you with information about educational careers or institutions.

- **NON-PROFIT WEBSITE**

  A non-profit website helps donors to check the authenticity of a non-profit organization and make the donation easily.

- **MEDIA WEBSITE**

  The media website contains news, reports and may or may not represent a media company. They also make a profit via advertisements generally and are aimed to provide genre-specific or general news/reports.

CHAPTER 2

FEASIBILITY STUDY

System feasibility is a test or evaluation of the complete system plan. Such an evaluation is necessary to define the application area along with its extension and complexity, to provide the scope of computerization together with suggested output and input format and potential benefits. The system study must examine whether a technically feasible solution is possible.

Depending on the result of the initial investigation, the survey is expanded to a more detailed study. Feasibility study is a best of the system proposal according to its Workability impact on the organization, ability to meet needs, and effective use of resources.

The feasibility study is an investigation that results in a written document that:

- Defines the scope of the problem.
- Identifies the elements of the problem.
- Identifies the evaluative criteria.
- Identifies possible alternative solutions.
- Evaluate each solution with the criteria.

The goal of the feasibility study is to discover possible solutions and to determine which of these appear to have promise and which are not feasible and why.

2.1 TECHNICAL FEASIBILITY

A study of functions, performance and constraints that may affect the ability to achieve an acceptable system. In an ever-changing software world, selecting one tool set and platform is a very difficult task. We should be extremely careful in the selection of the software platform and the tools for development. Always it should be possible to select a tool set and platform, which can seamlessly integrate into other software platforms and the support for the future, should be ensured.

This project is technically feasible as it is made using the JAVA technology which is platform independent and is very flexible to changes and is widely available. Also, as the ANDROID

STUDIO supports many languages, so it is very easy to upgrade the project to a better system

2.2 OPERATIONAL FEASIBILITY

Operational feasibility is the measure of how well a proposed system solves the problems and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development.

The "**Tic-tac-toe**" is operationally feasible as it has a very user-friendly environment which is very easy to use by the users. Also, the operations of every type of user can be easily studied by the design of the application. So, a new App developer can easily understand the project in a short time.

2.3 BEHAVIORAL FEASIBILITY

This aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to get used to the system which must be accepted as a necessity. The level of acceptance by the user solely depends on the developed system and has a modest technical requirement as only minimal or null changes are required for implementing the system.

This project is behaviorally feasible also as if we launch it; the users will accept it to save their time and money.

2.4 ECONOMICAL FEASIBILITY

Economic analysis is the most frequently used method for evaluating the effectiveness of a new system. More commonly known as cost/benefit analysis, the procedure is to determine the benefits and savings that are expected from a candidate system and compare them with costs. If benefits outweigh costs, then the decision is made to design and implement the system. An entrepreneur must accurately weigh the cost versus benefits before taking an action.

2.4.1 Cost Based Study

It is important to identify cost and benefit factors, which can be categorized as follows:

- Development costs
- Operating costs

This is an analysis of the costs to be incurred in the system and the benefits derivable out of the system.

2.4.2 Schedule Feasibility

A project will fail if it takes too long to be completed before it is useful. Typically, this means estimating how long the system will take to develop, and if it can be completed in a given time period using some methods like payback period. Schedule feasibility is a measure of how reasonable the project timetable is. Some projects are initiated with specific deadlines. It should be determined whether the deadlines are mandatory or desirable. This project is feasible by schedule as it completed in the desired time period with the reasonable timetables and deadlines.

CHAPTER 3

SOFTWARE REQUIREMENTS AND SYSTEM SPECIFICATION

Each and every application has some basic requirements be its software requirements of hardware requirements. Likely this tic tac toe game has some basic software requirements some of them are explained below:

3.1 HTML

HTML is an acronym which stands for **Hyper Text Markup Language** which is used for creating web pages and web applications. Let's see what is meant by Hypertext Markup Language, and Web page.

Hyper Text: HyperText simply means "Text within Text." A text has a link within it, is a hypertext. Whenever you click on a link which brings you to a new webpage, you have clicked on a hypertext. HyperText is a way to link two or more web pages (HTML documents) with each other.

Markup language: A markup language is a computer language that is used to apply layout and formatting conventions to a text document. Markup language makes text more interactive and dynamic. It can turn text into images, tables, links, etc.

Web Page: A web page is a document which is commonly written in HTML and translated by a web browser. A web page can be identified by entering an URL. A Web page can be of the static or dynamic type. **With the help of HTML only, we can create static web pages**.

3.1.1 Usages of HTML

HTML helps to structure our website well. The way a skeleton system gives a structure to the human body in a similar manner HTML act as a skeleton for a website, without it a website cannot be made. If you want to work as a Software Developer especially in the Web Development domain, then learning HTML is a must, because without knowledge of it you

cannot build a website.

Base for creating websites: HTML is the basic necessity a developer should know while building a website from scratch.

Learn web development: HTML is the first step towards learning Web Development. Once you learn HTML, you can build simple, static websites very easily.

Can become freelancer: Since web development has the best scope in freelancing, therefore learning HTML will surely help you to get the best deals of website development in the market.

3.1.2 Brief History of HTML

In the late 1980's , a physicist, Tim Berners-Lee who was a contractor at CERN, proposed a system for CERN researchers. In 1989, he wrote a memo proposing an internet based hypertext system.

Tim Berners-Lee is known as the father of HTML. The first available description of HTML was a document called "HTML Tags" proposed by Tim in late 1991. The latest version of HTML is HTML5.

3.1.3 Features of HTML

1) It is a very **easy and simple language**. It can be easily understood and modified.

2) It is very easy to make an **effective presentation** with HTML because it has a lot of formatting tags.

3) It is a **markup language**, so it provides a flexible way to design web pages along with the text.

4) It facilitates programmers to add a **link** on the web pages (by html anchor tag), so it enhances the interest of browsing of the user.

5) It is **platform-independent** because it can be displayed on any platform like Windows, Linux, and Macintosh, etc.

6) It facilitates the programmer to add **Graphics, Videos, and Sound** to the web pages which makes it more attractive and interactive.

7) HTML is a case-insensitive language, which means we can use tags either in lower-case or upper-case.

3.1.4 HTML EXAMPLE

HTML is a markup language which is used for creating attractive web pages with the help of styling, and which looks in a nice format on a web browser. An HTML document is made of many HTML tags and each HTML tag contains different content.

Let's see a simple example of HTML.

```
<!DOCTYPE>
<html>
<head>
<title>Web page title</title>
</head>
<body>
<h1>Write Your First Heading</h1>
<p>Write Your First Paragraph.</p>
</body>
</html>
```

3.1.5 Description of HTML Example

- **<!DOCTYPE>:** It defines the document type or it instruct the browser about the version of HTML.
- **<html >** :This tag informs the browser that it is an HTML document. Text between html tag describes the web document. It is a container for all other elements of HTML except <!DOCTYPE>

- **<head>:** It should be the first element inside the <html> element, which contains the metadata(information about the document). It must be closed before the body tag opens.
- **<title>:** As its name suggested, it is used to add title of that HTML page which appears at the top of the browser window. It must be placed inside the head tag and should close immediately. (Optional)
- **<body>** : Text between body tag describes the body content of the page that is visible to the end user. This tag contains the main content of the HTML document.
- **<h1>** : Text between <h1> tag describes the first level heading of the webpage.
- **<p>** : Text between <p> tag describes the paragraph of the webpage.

3.1.6 HTML Web Browser

The purpose of a web browser (Chrome, Edge, Firefox, Safari) is to read HTML documents and display them correctly.

A browser does not display the HTML tags, but uses them to determine how to display the document:

Figure 3.1 : Web Browser

3.2 CSS

It stands for Cascading Style Sheet, it is a style sheet language used to shape the HTML elements that will be displayed in the browsers as a web-page. Without using CSS, the website

which has been created by using HTML, will look dull. Basically CSS gives the outer cover on any HTML elements. If you consider HTML as a skeleton of the web- page then the CSS will be the skin of the skeleton. The Internet media type (MIME type) of CSS is text/CSS. The CSS was developed by the World Wide Web Consortium (W3C) in the year of 1996. The CSS can be applied to HTML documents in different ways.

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.

3.2.1 Importance to Learn CSS

- Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.
- CSS is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain. I will list down some of the key advantages of learning CSS:
- **Create Stunning Web site** - 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.
- **Become a web designer** - If you want to start a career as a professional web designer, HTML and CSS designing is a must skill.
- **Control web** - 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.
- **Learn other languages** - Once you understand the basic of HTML and CSS then other related technologies like JavaScript, php, or angular are become easier to understand.

3.2.2 Brief History of CSS

CSS is another thing you may learn just after understanding HTML. CSS stands for the

cascading style sheets, which Hakon Wium Lie created in 1994. Hakon Wium Lie is considered the father of CSS as he created this amazing thing. And he used to work with the father of HTML, Mr. Tim Berners-Lee, when he was working in CERN.

3.2.3 Features of CSS

- **Opportunity in Web designing:** If anyone wants to begin a career in web designing professionally, it is essential to have knowledge of CSS and HTML.
- **Website Design:** With the use of CSS, we can control various styles, such as the text color, the font style, the spacing among paragraphs, column size and layout, background color and images, design of the layout, display variations for distinct screens and device sizes, and many other effects as well.
- **Web Control:** CSS has controlling power on the documents of HTML, so it is easy to learn. It is integrated with the HTML and the XHTML markup languages.
- **Other Languages:** Once we have knowledge of some basics of CSS and HTML, other associated technologies like Angular, PHP, and JavaScript are become clearer to understand.

3.2.4 CSS Syntax

A CSS rule set contains a selector and a declaration block.

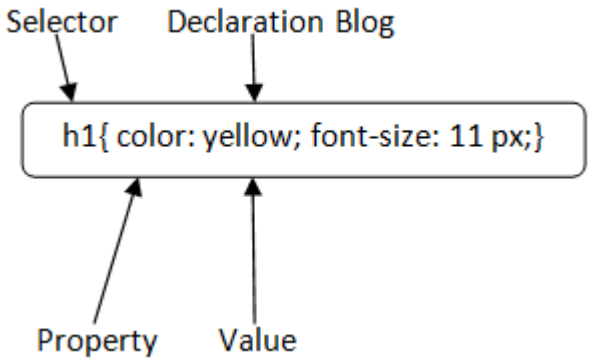

Figure 3.2: CSS Syntax

Selector: Selector indicates the HTML element you want to style. It could be any tag like <h1>, <title> etc.

Declaration Block: The declaration block can contain one or more declarations separated by a semicolon. For the above example, there are two declarations:

- color: yellow;
- font-size: 11 px;

Each declaration contains a property name and value, separated by a colon.

Property: A Property is a type of attribute of HTML element. It could be color, border etc.

Value: Values are assigned to CSS properties. In the above example, value "yellow" is assigned to color property.

- Selector{Property1: value1; Property2: value2; .....;}

3.2.5 Ways to add CSS

CSS is added to HTML pages to format the document according to information in the style sheet. There are three ways to insert CSS in HTML documents.

3.2.5.1 Inline CSS

3.2.5.2 Internal CSS

3.2.5.3 External CSS

3.2.5.1 Inline CSS

We can apply CSS in a single element by inline CSS technique.

The inline CSS is also a method to insert style sheets in HTML document. This method mitigates some advantages of style sheets so it is advised to use this method sparingly.

If you want to use inline CSS, you should use the style attribute to the relevant tag.

Syntax:

<htmltag style="cssproperty1:value; cssproperty2:value;"> **</htmltag>**

3.2.5.2 Internal CSS

Internal CSS is used to apply CSS on a single document or page. It can affect all the elements of the page. It is written inside the style tag within head section of html.

For example:

<style>
p{color:blue}
</style>

3.2.5.3 External CSS

The external style sheet is generally used when you want to make changes on multiple pages. It is ideal for this condition because it facilitates you to change the look of the entire web site by changing just one file.

It uses the <link> tag on every pages and the <link> tag should be put inside the head section.

Example:

<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>

The external style sheet may be written in any text editor but must be saved with a .css extension. This file should not contain HTML elements.

3.3 JavaScript

JavaScript (js) is a light-weight object-oriented programming language which is used by several websites for scripting the webpages. It is an interpreted, full-fledged programming language that enables dynamic interactivity on websites when applied to an HTML document. It was introduced in the year 1995 for adding programs to the webpages in the Netscape Navigator browser. Since then, it has been adopted by all other graphical web browsers. With JavaScript, users can build modern web applications to interact directly without reloading the page every time. The traditional website uses js to provide several forms of interactivity and simplicity.

Although, JavaScript has no connectivity with Java programming language. The name was suggested and provided in the times when Java was gaining popularity in the market. In addition to web browsers, databases such as CouchDB and MongoDB uses JavaScript as their scripting and query language.

3.3.1 Features of JavaScript

There are following features of JavaScript:

1. All popular web browsers support JavaScript as they provide built-in execution environments.
2. JavaScript follows the syntax and structure of the C programming language. Thus, it is a structured programming language.
3. JavaScript is a weakly typed language, where certain types are implicitly cast (depending on the operation).
4. JavaScript is an object-oriented programming language that uses prototypes rather than using classes for inheritance.
5. It is a light-weighted and interpreted language.
6. It is a case-sensitive language.
7. JavaScript is supportable in several operating systems including, Windows, macOS, etc.
8. It provides good control to the users over the web browsers.

3.3.2 History of JavaScript

In 1993, **Mosaic**, the first popular web browser, came into existence. In the **year 1994**, **Netscape** was founded by **Marc Andreessen**. He realized that the web needed to become more dynamic. Thus, a 'glue language' was believed to be provided to HTML to make web designing easy for designers and part-time programmers. Consequently, in 1995, the company recruited **Brendan Eich** intending to implement and embed Scheme programming language to the browser. But, before Brendan could start, the company merged with **Sun Microsystems** for adding Java into its Navigator so that it could compete with Microsoft over the web technologies and platforms. Now, two languages were there: Java and the scripting language. Further, Netscape decided to give a similar name to the scripting language as Java's. It led to 'Javascript'. Finally, in May 1995, Marc Andreessen coined the first code of Javascript named '**Mocha**'. Later, the marketing team replaced the name with '**LiveScript**'. But, due to trademark reasons and certain other reasons, in December 1995, the language was finally renamed to 'JavaScript'. From then, JavaScript came into existence.

3.3.3 Application of JavaScript

JavaScript is used to create interactive websites. It is mainly used for:

- Client-side validation,
- Dynamic drop-down menus,
- Displaying date and time,
- Displaying pop-up windows and dialog boxes (like an alert dialog box, confirm dialog box and prompt dialog box),
- Displaying clocks etc.

3.3.4 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.

3.3.5 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.

JavaScript Example

```
<script>
document.write("Hello JavaScript by JavaScript");
</script>
```

3.4 VS Code

Visual Studio Code is a lightweight but powerful source code editor which runs on your

desktop and is available for Windows, macOS and Linux. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages (such as C++, C#, Java, Python, PHP, Go) and runtimes (such as .NET and Unity)**.**

Visual Studio Code supports macOS, Linux, and Windows - so you can hit the ground running, no matter the platform.

At its heart, Visual Studio Code features a lightning fast source code editor, perfect for day-to-day use. With support for hundreds of languages, VS Code helps you be instantly productive with syntax highlighting, bracket-matching, auto-indentation, box-selection, snippets, and more. Intuitive keyboard shortcuts, easy customization and community-contributed keyboard shortcut mappings let you navigate your code with ease.

For serious coding, you'll often benefit from tools with more code understanding than just blocks of text. Visual Studio Code includes built-in support for IntelliSense code completion, rich semantic code understanding and navigation, and code refactoring.

And when the coding gets tough, the tough get debugging. Debugging is often the one feature that developers miss most in a leaner coding experience, so we made it happen. Visual Studio Code includes an interactive debugger, so you can step through source code, inspect variables, view call stacks, and execute commands in the console.

3.4.1 Features of Visual Studio Code

Visual Studio Code combines the simplicity of a source code editor with powerful developer tooling, like IntelliSense code completion and debugging. First and foremost, it is an editor that gets out of your way. The delightfully frictionless edit-build-debug cycle means less time fiddling with your environment, and more time executing on your ideas.

- **Available for macOS, Linux, and Windows**

Visual Studio Code supports macOS, Linux, and Windows - so you can hit the ground running, no matter the platform.

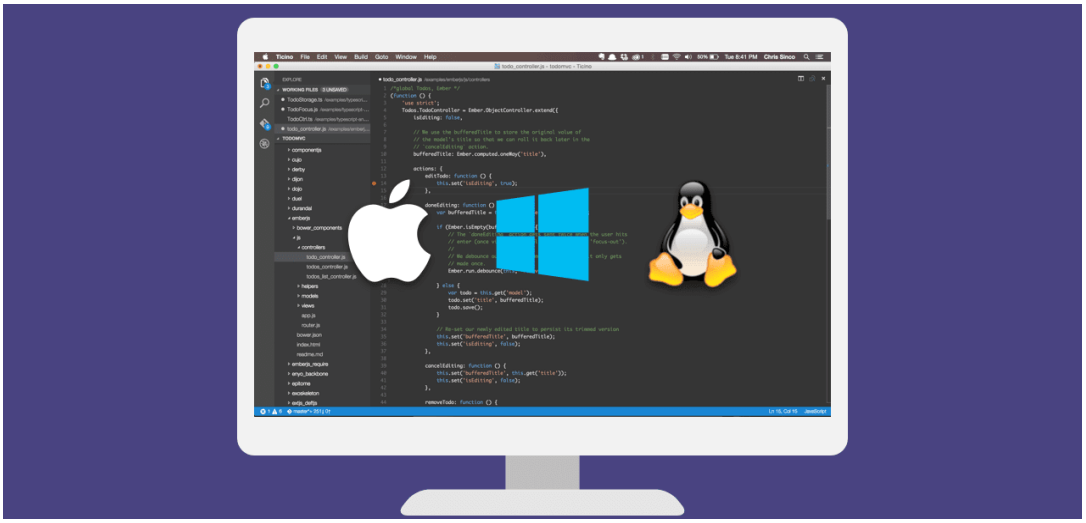

Figure 3.3: VS platform

- **Edit, build, and debug with ease**

At its heart, Visual Studio Code features a lightning fast source code editor, perfect for day-to-day use. With support for hundreds of languages, VS Code helps you be instantly productive with syntax highlighting, bracket-matching, auto-indentation, box-selection, snippets, and more. Intuitive keyboard shortcuts, easy customization and community-contributed keyboard shortcut mappings let you navigate your code with ease.

For serious coding, you'll often benefit from tools with more code understanding than just blocks of text. Visual Studio Code includes built-in support for IntelliSense code completion, rich semantic code understanding and navigation, and code refactoring.

And when the coding gets tough, the tough get debugging. Debugging is often the one feature that developers miss most in a leaner coding experience, so we made it happen. Visual Studio Code includes an interactive debugger, so you can step through source code, inspect variables, view call stacks, and execute commands in the console.

VS Code also integrates with build and scripting tools to perform common tasks making everyday workflows faster. VS Code has support for Git so you can work with source control without leaving the editor including viewing pending changes diffs.

- **Make it your own**

Customize every feature to your liking and install any number of third-party extensions. While most scenarios work "out of the box" with no configuration, VS Code also grows with you,

and we encourage you to optimize your experience to suit your unique needs. VS Code is an open-source project so you can also contribute to the growing and vibrant community on GitHub.

- **Built with love for the Web**

VS Code includes enriched built-in support for Node.js development with JavaScript and TypeScript, powered by the same underlying technologies that drive Visual Studio. VS Code also includes great tooling for web technologies such as JSX/React, HTML, CSS, SCSS, Less, and JSON.

- **Robust and extensible architecture**

Architecturally, Visual Studio Code combines the best of web, native, and language-specific technologies. Using Electron, VS Code combines web technologies such as JavaScript and Node.js with the speed and flexibility of native apps. VS Code uses a newer, faster version of the same industrial-strength HTML-based editor that has powered the "Monaco" cloud editor, Internet Explorer's F12 Tools, and other projects. Additionally, VS Code uses a tools service architecture that enables it to integrate with many of the same technologies that power Visual Studio, including Roslyn for .NET, TypeScript, the Visual Studio debugging engine, and more.

Visual Studio Code includes a public extensibility model that lets developers build and use extensions, and richly customize their edit-build-debug experience.

- **Ready, set, code**

If you prefer a code editor-centric development tool or are building cross-platform web and cloud applications, we invite you to try out Visual Studio Code and let us know what you think!.

- **Easy to customize**

Customize every feature to your liking and install any number of third-party extensions. While most scenarios work "out of the box" with no configuration, VS Code also grows with you, and we encourage you to optimize your experience to suit your unique needs. VS Code is an open-source project so you can also contribute to the growing and vibrant community on GitHub.

- **User Interface of VS code**

The user interface of VS code editor is shown in the screenshot given below. Observe that the editor includes various features to create a new project or import from an existing project.

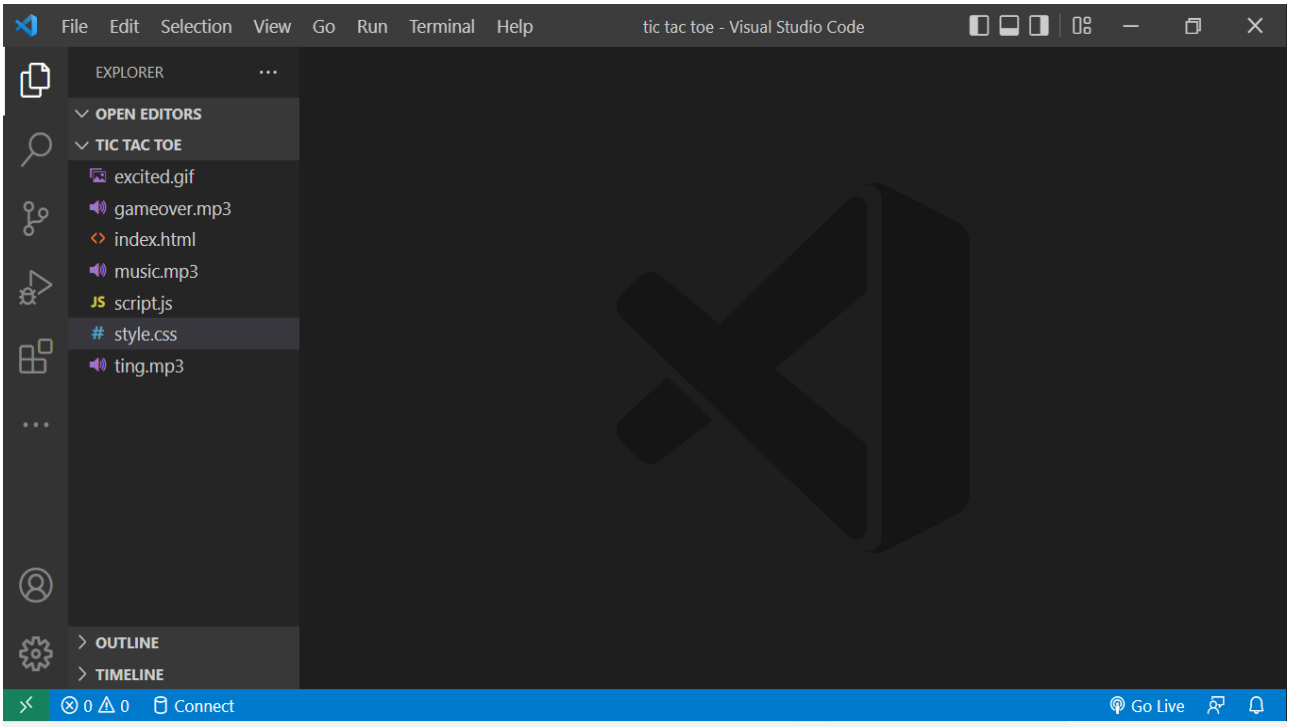

Figure 3.4: UI of VS Code IDE

CHAPTER 4
DESIGN MODULE

4.1 ACTIVITY OF GAME

we will start designing the code for the game logic and identify what objects we will have in our design.

We start with an Activity Diagram to illustrate the game flow.

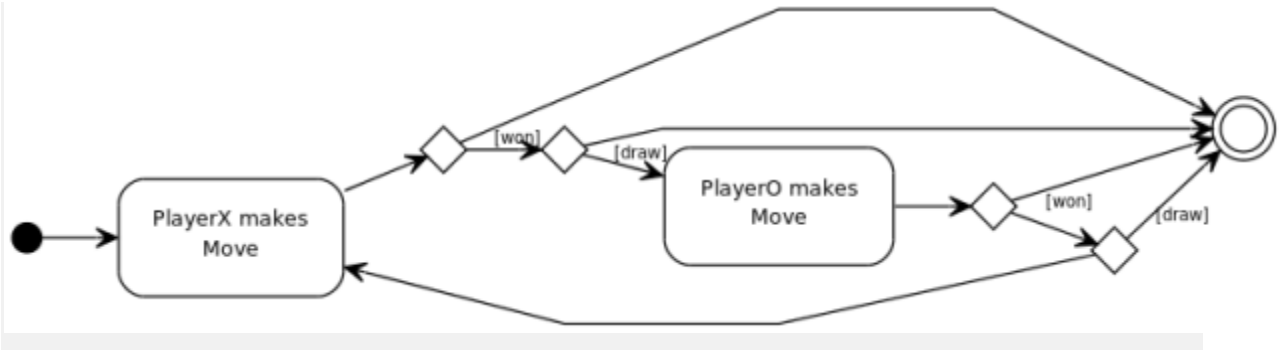

Figure 4.1: Activity of Game

The game starts with the player that marks with the X symbol. If the move results in a winning combination of cells, the game ends. If player's move does not win, but there are no more cells left to mark, the game ends in a draw. if neither these condition are true then the next player may take his/her turn.

Unfortunately, while this diagram provides a cognitive flow of the game that works, it does not satisfy our requirements set out for the Player or the View. We need to decide where in this diagram we are going to notify the View and Player of board changes and win conditions.

Views need to be refreshed when the game starts and after each player has made a move. We need to notify the view of how the game ends when it does.

Here is a revised version of the diagram for the model with the View notifications added.

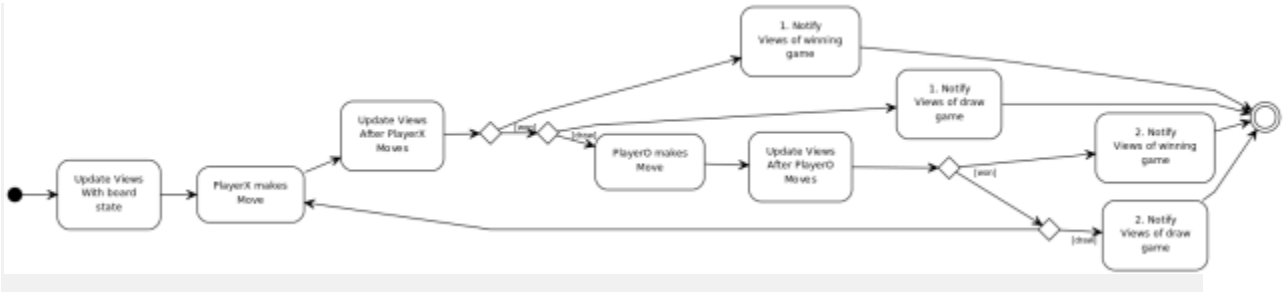

Figure 4.2: Flow for view Notifications

Player notification are the same as the Views, but instead of getting notified of general win conditions the player is notified it his/she has won or lost. The player is also notified when it is his/her turn to make a move.

4.2 DATA FLOW DIAGRAM

A DFD shows the flow of data through a system. It is used to describe and analyse the movement of data through a system-manual or automated-including the processes, stores of data, and delays in the system. Data flow diagrams are the central tool and the basis from which other components are developed.

A data flow diagram (DFD) depicts how data interacts with a system. They are also known as data flow graphs, bubble charts or pert networks. Data flow diagrams are extremely useful in modelling many aspects of a business function because they systematically subdivide a task into its basic parts; help in the analyst understand the system that they are trying to model. Data flow diagrams use a variety of symbols to represent a provider of data such as a customer or management

Here is the basic diagram modified to include the player notifications.

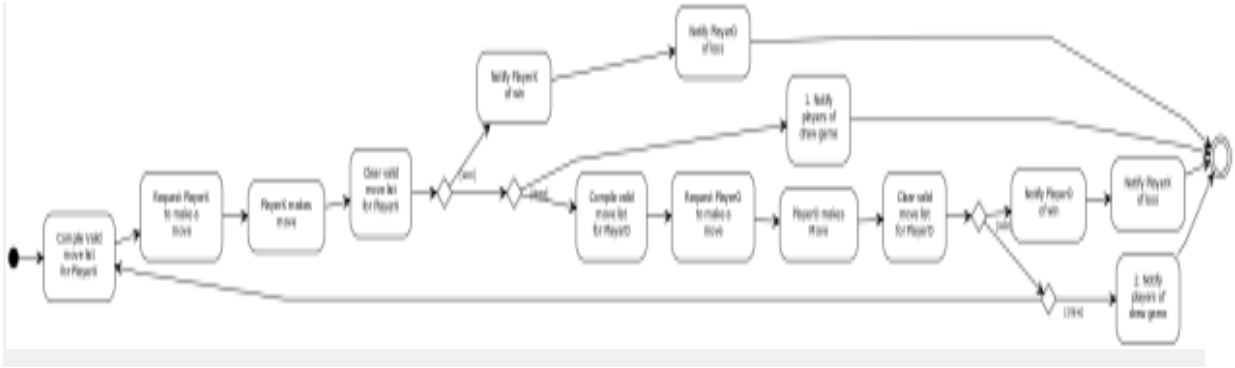

Figure 4.3: Flow Diagram

The diagram shows that the valid moves for a player is cleared after the player has made a move. This must be done in turn based games to prevent a player from make a move out of turn.

It would be an easy exercise to combine the Player and View activities into one diagram and I would actually like to do so, but the diagram would just be too big. I will say that the View should be notified before the Players are notified.

In the next post we will start extracting Objects from the diagram based on responsibilities and behaviour.

CHAPTER 5

IMPLEMENTATION & RESULTS

5.1 Main Page

All code implementation is performed on VS code. Before starting any coding I mentioned all the names to be displayed over the page:

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" />
    <title>Tic Tac Toe</title>
    <link rel="stylesheet" href="style.css" />
  </head>
  <body>
    <nav>
      <ul>
        <li>Welcome to Tic-Tac-Toe Game</li>
      </ul>
    </nav>

    <div class="gameContainer">
      <div class="container">
        <div class="line"></div>
      <div class="box bt-0 bl-0"><span class="boxtext"></span></div>
      <div class="box bt-0"><span class="boxtext"></span></div>
      <div class="box bt-0 br-0"><span class="boxtext"></span></div>
```

```
  <div class="box bl-0"><span class="boxtext"></span></div>
  <div class="box"><span class="boxtext"></span></div>
  <div class="box br-0"><span class="boxtext"></span></div>
  <div class="box bl-0 bb-0"><span class="boxtext"></span></div>
  <div class="box bb-0"><span class="boxtext"></span></div>
  <div class="box bb-0 br-0"><span class="boxtext"></span></div>
</div>
<div class="gameInfo">
  <div>
    <span class="info">Turn for X</span>
    <button id="reset">Reset</button>
  </div>
  <div class="imgbox">
    <img src="excited.gif" alt="" />
  </div>
</div>
<div class="rulesinfo">
<h1>Rules For Tic-Tac-Toe :</h1>
<p>
  <b>1.</b> Play occurs on a 3 by 3 grid of <br>9 empty squares.<br>
  <b>2.</b> Two players alternate marking<br>
   empty squares, the first player <br>
   marking Xs and the second player<br>
   marking Os.<br>
 <b>3.</b> If one player places three of <br>
  the same marks in a row, that player<br>
  wins.<br>
 <b>4.</b> If the spaces are all filled and<br>
   there is no winner,the game ends in<br>
   a draw.
</p>
</div>
```

```
    </div>

    <script src="script.js"></script>
  </body>
</html>
```

style.css :

```
@import
url('https://fonts.googleapis.com/css2?family=Baloo+Bhaina+2&family=Roboto&display=s
wap');
*{
    margin: 0;
    padding: 0;
}

nav{
    background-color: rgb(37, 9, 37);
    color: white;
    height: 65px;
    font-size: 27px;
    align-items: center;
    padding: 0 12px;
    font-family: 'Roboto', sans-serif;
    font-size: 47px;
}

nav ul li{
    text-align: center;
}

.gameContainer{
```

```
    display: flex;
    margin-top: 50px;
    margin-left: 500px;
}

.container{
    margin-left: -331px;
    display: grid;
    grid-template-rows: repeat(3, 10vw);
    grid-template-columns: repeat(3, 10vw);
    font-family: 'Roboto', sans-serif;
    position: relative;
}

.box{
    border: 2px solid black;
    font-size: 8vw;
    cursor: pointer;
    display: flex;
    justify-content: center;
    align-items: center;
    padding: 2px;
}

.box:hover{
    background-color: rgb(242, 234, 250);
}

.info {
    font-size: 22px;
}
```

```
.gameInfo{
    padding: 0 34px;
    font-family: 'Baloo Bhaina 2', cursive;
}

.gameInfo h1{
    font-size: 2rem;
}

.imgbox img{
    width: 0;
    transition: width 1s ease-in-out;
}

.rulesinfo{
    justify-content: center ;
    background-color:#f3e7f9 ;
    padding: 0 78px;
    padding-left: 78px;
    font-family: 'Baloo Bhaina 2', cursive;
}
.rulesinfo h1{
    font-size: 2.5rem;
}
.rulesinfo p{
    font-size: 24px;
    font-family: 'Baloo Bhaina 2', cursive;
}

.br-0{
    border-right: 0;
}
```

```
.bl-0{
   border-left: 0;
}

.bt-0{
   border-top: 0;
}

.bb-0{
   border-bottom: 0;
}

#reset {
   margin: 0 23px;
   padding: 1px 18px;
   background: #f3e7f9;
   border-radius: 6px;
   cursor: pointer;
   font-family: 'Baloo Bhaina 2';
   font-size: 25px;
   font-weight: bolder;
}

.line{
   background-color: black;
   height: 3px;
   width: 0;
   position: absolute;
   background-color: #911d91;
   transition: width 1s ease-in-out;
}
```

```
@media screen and (max-width: 950px)
{
   .gameContainer{
      flex-wrap: wrap;
   }
   .gameInfo{
      margin-top: 34px;
   }
   .gameInfo h1{
      font-size: 1.5rem;
   }
   .container {
      grid-template-rows: repeat(3, 20vw);
      grid-template-columns: repeat(3, 20vw);
   }
```

script.js :

```
console.log("Welcome to Tic Tac Toe")
let music = new Audio("music.mp3")
let audioTurn = new Audio("ting.mp3")
let gameover = new Audio("gameover.mp3")
let turn = "X"
let isgameover = false;

// Function to change the turn
const changeTurn = ()=>{
   return turn === "X"? "0": "X"
}

// Function to check for a win
```

```
const checkWin = ()=>{
    let boxtext = document.getElementsByClassName('boxtext');
    let wins = [
        [0, 1, 2, 5, 5, 0],
        [3, 4, 5, 5, 15, 0],
        [6, 7, 8, 5, 25, 0],
        [0, 3, 6, -5, 15, 90],
        [1, 4, 7, 5, 15, 90],
        [2, 5, 8, 15, 15, 90],
        [0, 4, 8, 5, 15, 45],
        [2, 4, 6, 5, 15, 135],
    ]
    wins.forEach(e =>{
        if((boxtext[e[0]].innerText === boxtext[e[1]].innerText) && (boxtext[e[2]].innerText === boxtext[e[1]].innerText) && (boxtext[e[0]].innerText !== "") ){
            document.querySelector('.info').innerText = boxtext[e[0]].innerText + " Won"
            isgameover = true
             document.querySelector('.imgbox').getElementsByTagName('img')[0].style.width = "200px";
             document.querySelector(".line").style.transform = `translate(${e[3]}vw, ${e[4]}vw) rotate(${e[5]}deg)`
            document.querySelector(".line").style.width = "20vw";
        }
    })
}

// Game Logic
// music.play()
let boxes = document.getElementsByClassName("box");
Array.from(boxes).forEach(element =>{
    let boxtext = element.querySelector('.boxtext');
    element.addEventListener('click', ()=>{
```

```
        if(boxtext.innerText === ''){
            boxtext.innerText = turn;
            turn = changeTurn();
            audioTurn.play();
            checkWin();
            if (!isgameover){
                document.getElementsByClassName("info")[0].innerText  = "Turn for " + turn;
            }
        }
    })
})

// Add onclick listener to reset button
reset.addEventListener('click', ()=>{
    let boxtexts = document.querySelectorAll('.boxtext');
    Array.from(boxtexts).forEach(element => {
        element.innerText = ""
    });
    turn = "X";
    isgameover = false
    document.querySelector(".line").style.width = "0vw";
    document.getElementsByClassName("info")[0].innerText  = "Turn for " + turn;
    document.querySelector('.imgbox').getElementsByTagName('img')[0].style.width = "0px"
})

                color: dodgerblue; font-size: 17px;
                cursor: pointer;
}

/* Player turn space */ #print {
font-family: Verdana,
Geneva, Tahoma, sans-serif;
```

color: dodgerblue; font-size: 20px;

}

/* Main Container */ #main {

text-align: center;

}

/* Game Instruction Text */ #ins {

font-family: Verdana,

Geneva, Tahoma, sans-serif

}

</style>

5.2 Project Details

Figure below represents all files present in the project:

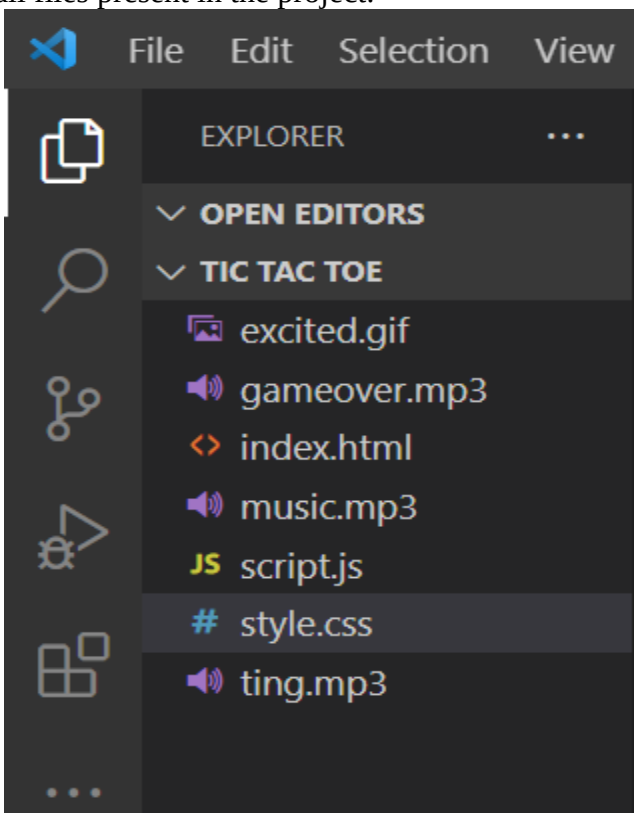

Figure 5.1: All files in project

Folder Snake HTML contains all the files related to the project, folder named tic.css contains all the rules designing part of the game , and tic.js contain all the logic part of the game.

5.3 Game Interface:

This webpage represents the implementation of index.html, style.css, script.js which is the user interface visible to a user.

On the Right side we implement the rules to play this Tic Tac Toe game.

There is button called as “Reset” , after press this button it replays the game again.

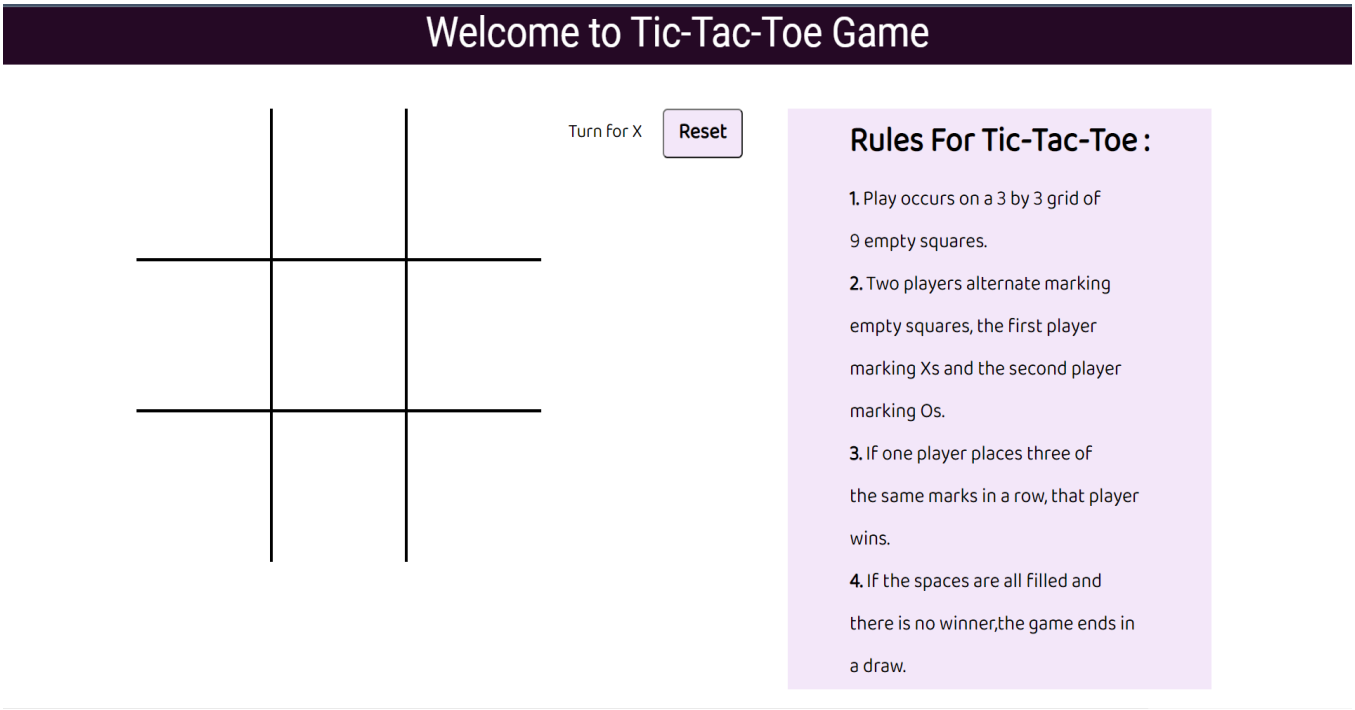

Figure 5.2 : Game Interface

Winning message

This image below represents which player win the current game.

1. When Player ‘X’ won the match , then a cat cheers for the wining of “X” player.

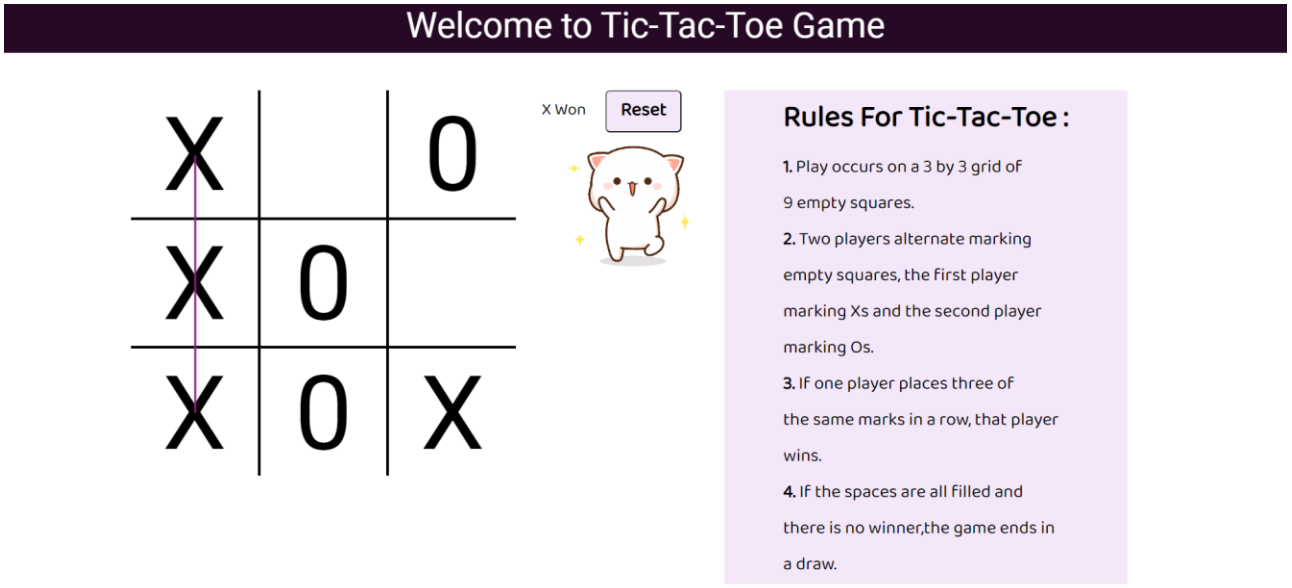

Figure 5.3: Player X won

2. When Player ‘O’ won the match , then a cat cheers for the wining of “O” player.

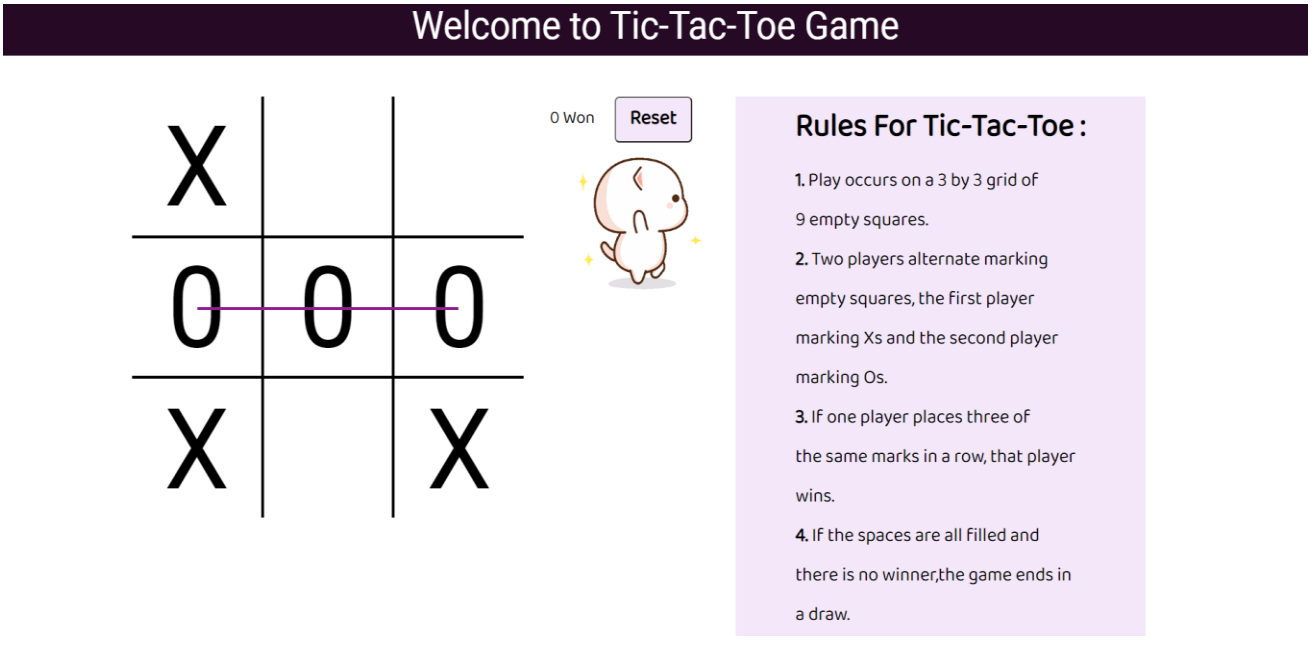

Figure 5.4: Player O won

3. When Game Ties . then it shows nothing , for replay the game again you have to simply press Reset button.

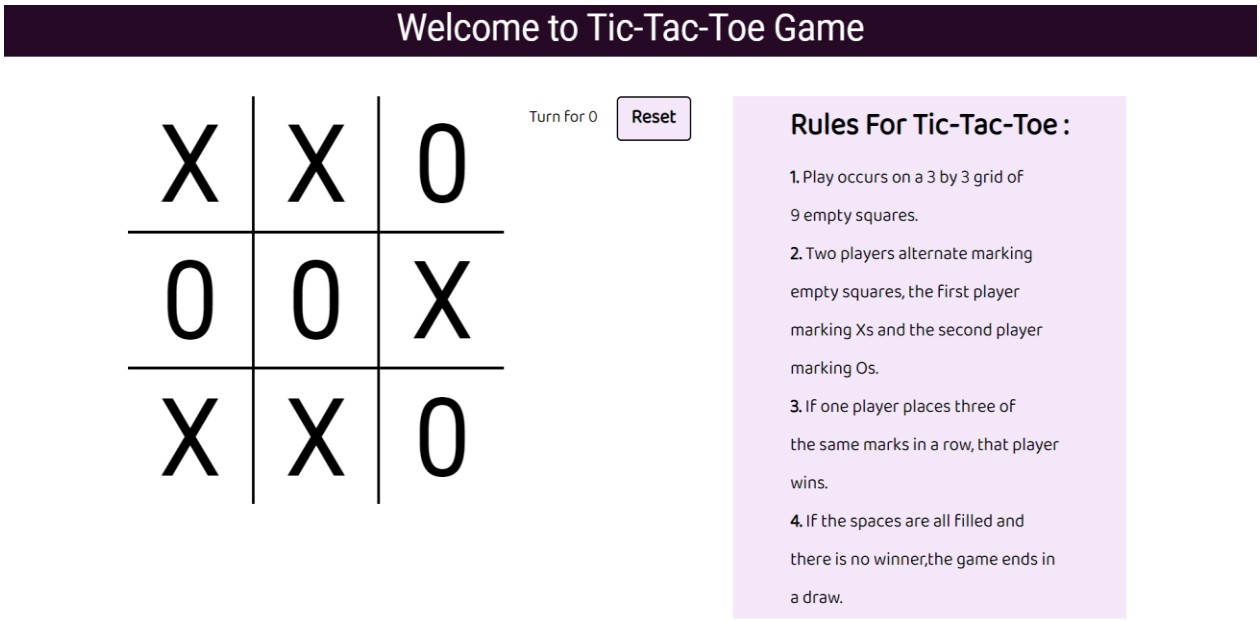

Figure 5.5: Game Ties

CHAPTER 6

TESTING AND DEBUGGING

Software Testing is the process of executing a program or system with the intent of finding errors or it involves any activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results.

Software testing is a process used to identify the correctness, completeness, and quality of developed computer software.

6.1. TESTING FRAMEWORK

To begin with, let us look at the traditional Software Development life cycle. The figure below depicts the same.

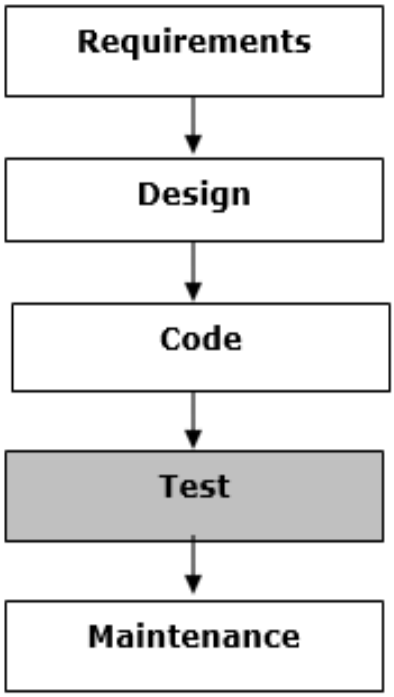

Figure 6.1: Traditional SRS

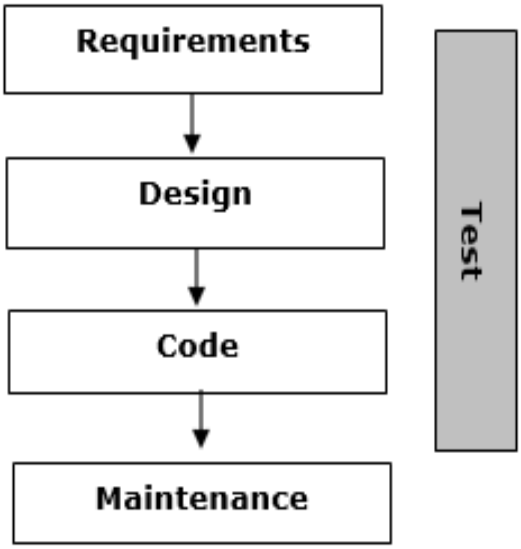

Figure 6.2: Advanced SRS

In the above diagram (Figure 6.1), the Testing phase comes after the Coding is complete and before the product is launched and goes into maintenance. But the recommended test process involves testing in every phase of the life cycle (Figure 6.2).

During the requirements phase, the emphasis is upon validation to determine that the defined requirements meet the needs of the organization. During the design and program phases, the emphasis is on verification to ensure that the design and programs accomplish the defined requirements.

6.2 TESTING TECHNIQUES

In this project the following testing techniques are used.

6.2.1 Unit Testing

Goal of Unit testing is to uncover defects using formal techniques like Boundary Value Analysis (BVA), Equivalence Partitioning, and Error Guessing. Defects and deviations in Date formats, Special requirements in input conditions (for example Text box where only numeric or alphabets should be entered), selection based on Combo Box's, List Box's, Option buttons, Check Box's would be identified during the Unit Testing phase. Following validation checks are performed during the unit testing

6.2.1.1 **Validation for Numeric Entries: -** This validation checks whether an entry in the text box is numeric or not. If not, a suitable message is flashed to user.

6.2.1.2 **Validation for Mandatory Entries: -** This validation checks whether mandatory entries are filled or not. If not, a suitable message is flashed to user.

6.2.1.3 **Validation for Unauthorized Entry: -** This validation checks for validity of user for making entries or modification. If user is not authorized than such a user is prevented from executing the software.

6.2.2 Integration Testing

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design.

In these testing forms means for relevant schemes that have already been tested during the unit testing are integrated into one module namely online monitoring of land reforms schemes, this system is then tested thoroughly for the errors.

CHAPTER 7
CONCLUSION & FUTURE SCOPE

7.1 CONCLUSION

The project tic tac toe game is built for 2 player game which is authentic and inspired from earlier version of tic tac toe which we use to play with friends. This particular game is developed successfully over the website. Code used in this website is kind of basic but require human interaction to be completed.

Tic tac toe is originally a simple game but occupies a lot of mathematical concepts. The probability to enter at any point in the 3X3 matrix is kind of exception to every player so this game is authentic to its earlier version , and main objective of developing game is successfully accomplished.

7.2 LIMITAIONS

- This website is restricted to windows devices only.
- Both the user had to be present at the same location to play this game.
- Login System is not included yet.
- 2 persons are compulsorily required to play , bot is not introduced yet.
- It is not available on the network yet.

7.3 FUTURE SCOPE

- Purchasing domain name to make it available over the world.
- Giving source codes for better experience and reducing searches.
- More optimized code in backend.
- Improving graphical user interface.
- Making personal and secured database of every player.
- Providing options to the users through which they can suggest us more inspiring ideas like animation of sound effects.

- Creating a particular IDs for frequent users to make user connected to our website.
- Making all functionality able to work online by establishing a secure connection to the server.

REFRENCES

[1] JavaTPoint: https://www.javatpoint.com/

[2] W3Schools: https://www.w3schools.com/

[3] Geeksforgeeks: https://www.geeksforgeeks.org/

[4] YouTube : CodeWithHarry

[5] W3 School : https://www.w3schools.com/