**ABSTRACT**

A good game can therefore be thought of as a "family" of potentially interesting logic puzzles, and the play consists of each player posing such a puzzle to the other. Good players are the ones who find the most difficult puzzles to present to their opponents.

Every Web Developer must have a basic understanding of HTML, CSS, and JavaScript. Responsive Web Design is used in all types of modern web development. ECMAScript 5 (JavaScript 5) is supported in all modern browsers. Take a good look at it, especially the new array functions.Vanilla JavaScript This tutorial explains some basic HTML5, CSS3 and JS concepts. We will discuss data-attribute, positioning, perspective, transitions, flexbox, event handling, timeouts and ternaries. You are not expected to have much prior knowledge in programming.

Abstract Highly available and scalable. web hosting can be a complex and expensive proposition. Traditional scalable .web architectures have not only needed to implement complex solutions to ensure high levels of reliability, but they have also required an accurate forecast of traffic to provide a high level of customer service.

**INDEX**

**ABSTRACT**

1. **INTRODUCTION--------------------------------------------------------4**
   1. **About web development-------------------------------------------------4**
   2. **WORKING OF BROWSER-------------------------------------------6**

**1.2.1) Web Client-----------------------------------------------------7**

**1.2.2) Web Server----------------------------------------------------7**

**1.2.3) What is DNS Server-----------------------------------------9**

**1.2.4) How Does DNS Work---------------------------------------9**

**1.3) About Memory game---------------------------------------------------10**

**2-LITERATURE SURVEY-------------------------------------------------11**

**3- SYSTEM ANALYSIS-----------------------------------------------------12**

**3.1) Feasibility Study----------------------------------------------------------12**

**3.2) Modules --------------------------------------------------------------------13**

**3.3)Module Description------------------------------------------------------13**

**4-ARCHITECTURE----------------------------------------------------------14**

**4.1)SYSTEM REQUIREMENT:------------------------------------------15**

**4.2) INSTALLION SOFTWARE REQUIREMENT------------------16**

**4.3) Implementation of Memory Game----------------------------------18**

**5-PROJECT CODE-----------------------------------------------------------20**

**5.1)HTML file----------------------------------------------------------------19**

**5.2) Cascading Style Sheets-----------------------------------------------22**

**5.3)JavaScript---------------------------------------------------------------26**

**5.4) Images Used in Project------------------------------------------------31**

**6-SYSTEM TESTING----------------------------------------------------------------------34**

**6.1 TYPES OF TESTS---------------------------------------------------- 34**

**6.2 TEST STRATEGY AND APPROACH----------------------------36**

**7-RESULT-------------------------------------------------------------------------------------37**

**8-HOSTING-----------------------------------------------------------------------------------43**

**9-CONCLUSION-----------------------------------------------------------------------------44**

**10-REFERENCE-----------------------------------------------------------------------45**

**INTRODUCTION**

**About web development:-**

Web programming, also known as web development, is the creation of dynamic web applications. Examples of web applications are social networking sites like Facebook or e-commerce sites like Amazon.

The good news is that learning web development is not that hard!

In fact, many argue it’s the best form of coding for beginners to learn. It’s easy to set up, you get instant results and there’s plenty of online training available.

A lot of people learn web coding because they want to create the next Facebook or find a job in the industry. But it’s also a good choice if you just want a general introduction to coding, since it’s super easy to get started. No matter whether you’re looking for a career or just want to learn coding, learning how to develop for the web is for you. It’s one of the smartest decisions you will ever make!

There are two broad divisions of web development – front-end development (also called client-side development) and back-end development (also called server-side development).

Front-end development refers to constructing what a user sees when they load a web application – the content, design and how you interact with it. This is done with three codes – HTML, CSS and JavaScript.

HTML, short for Hyper Text Markup Language, is a special code for ‘marking up’ text in order to turn it into a web page. Every web page on the net is written in HTML, and it will form the backbone of any web application. CSS, short for Cascading Style Sheets, is a code for setting style rules for the appearance of web pages. CSS handles the cosmetic side of the web. Finally, JavaScript is a scripting language that’s widely used to add functionality and interactivity to web pages.

Back-end development controls what goes on behind the scenes of a web application. A back-end often uses a database to generate the front-end.

Here’s an example. Say you log into your Facebook account, and you are greeted with the latest updates in your News Feed. They’re not going to be the same updates that you saw yesterday.

How did the page change? Did a Facebook employee manually edit the page to update your news feed? Of course not. A script on the Facebook back-end would have received the updates and re-generated the front-end accordingly.

Back-end scripts are written in many different coding languages and frameworks, such as…

PHP

Ruby on Rails

ASP.NET

Perl

Java

Node.js (javascript)

Python

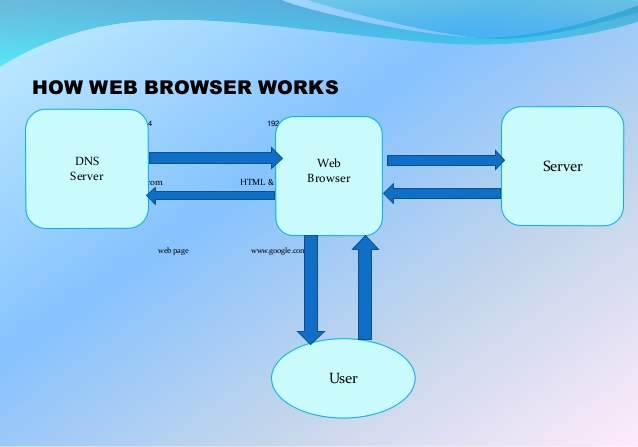
Without going into more detail, that’s really all there is to web development. So now, here are the best ways to start learning it.

**Types of browsers and there Engine :-**

The most famous world wide used web browser are

1. **Chrome:-** V8 is Google’s open source high-performance JavaScript and WebAssembly engine, written in C++. It is used in Chrome and in Node.js, among others. It implements ECMAScript and WebAssembly, and runs on Windows 7 or later, macOS 10.12+, and Linux systems that use x64, IA-32, ARM, or MIPS processors. V8 can run standalone, or can be embedded into any C++ application.
2. **Edge:-** Chakra is a JScript engine developed by Microsoft for its 32-bit version of the Internet Explorer 9 (IE9) web browser. The JScript engine is developed as closed source proprietary software. Microsoft has developed a different JavaScript engine based on the JScript, for the newer Microsoft Edge browser (also called Chakra).
3. **Firefox:-** SpiderMonkey is Mozilla's JavaScript engine written in C and C++. It is used in various Mozilla products, including Firefox, and is available under the MPL2. Standalone source code releases can be found on the Releases page. How to get SpiderMonkey source code, build it, and run the test suite.

**(1.2)WORKING OF BROWSER**



The World Wide Web is a system of Internet servers that support specially formatted documents. Web browsers are used to make it easy to access the World Wide Web. Browsers are able to display Web pages largely in part to an underlying Web protocol called HyperText Transfer Protocol (HTTP). HTTP defines how messages are formatted and transmitted, and what actions Web servers and browsers should take in response to various commands. It is what allows Web clients and Web servers to communicate with each other. When you enter a Web address (URL) in your browser, this actually sends an HTTP command to the Web server directing it to fetch and transmit the requested Web page and display the information in your browser. All Web servers serving Web sites and pages support the HTTP protocol.

**(1.2.1) Web CLIENT :-**

Web client can be said as an application or web browser (like Google Chrome, Edge, Opera, Firefox, Safari) which is installed in a computer and used to interact with Web servers upon user’s request. It is basically a consumer application which collects processed data from servers. A Client and a Server are two parts of a connection, these are two distinct machines, web client requests information, and the web server is basically a PC that is designed to accept requests from remote computers and send on the information requested.

**(1.2.2) Web Sever :-**

Web server is basically a system the deals with the client’s request and provide client with web pages by Hyper Text Transfer Protocol (HTTP) and files via File Transfer Protocol (FTP). URL (universal resource locator) is basically the address where you want to receive data from. When you type a URL in address bar of your web browser, web server sends request to the location where the domain name of this URL is saved.

Then the information you requested (i.e. the webpage) is accessed and provided you by server. This is how it works. So, we can conclude that the processing and providing webpage to the client is the main duty of a web server.

Image result for Difference between Web Browser, Web Server and Web Client

Developing a Web Server

Developing a webserver is not that rocket science. You can ever convert your PC into web server using a web server software. Many web server software are available such as:

Microsoft

NCSA

Apache

Netscape

Developing a web server is not difficult thing. But the processing of information, granting instant access to requested data and storing medium are key factors which measure the efficiency of a web server. HTTP is the protocol to deliver web contents. Web pages are then delivered to client in HTML format. HTML consists of the information like text based content, script shapes styles videos or images (if present) etc.

Features of Web Servers

Key features of web servers are

Virtual hosting

Large file support (multiple gigabytes)

Bandwidth regulating

Server-side scripting (for generating dynamic web pages)

A web server can handle up to 80000 connections at a time. However, a normal web server supports 500-1000 connections per IP.

**(1.2.3) what is DNS Server :-**

The Domain Name System (DNS) is the phonebook of the Internet. Humans access information online through domain names, like nytimes.com or espn.com. Web browsers interact through Internet Protocol (IP) addresses. DNS translates domain names to IP addresses so browsers can load Internet resources.

Each device connected to the Internet has a unique IP address which other machines use to find the device. DNS servers eliminate the need for humans to memorize IP addresses such as 192.168.1.1 (in IPv4), or more complex newer alphanumeric IP addresses such as 2400:cb 00:2048:1::c629:d7a2 (in IPv6).

**(1.2.4) How does DNS work:-**

The process of DNS resolution involves converting a hostname (such as www.example.com) into a computer-friendly IP address (such as 192.168.1.1). An IP address is given to each device on the Internet, and that address is necessary to find the appropriate Internet device - like a street address is used to find a particular home. When a user wants to load a webpage, a translation must occur between what a user types into their web browser (example.com) and the machine-friendly address necessary to locate the example.com webpage.

In order to understand the process behind the DNS resolution, it’s important to learn about the different hardware components a DNS query must pass between. For the web browser, the DNS lookup occurs “behind the scenes” and requires no interaction from the user’s computer apart from the initial require

**(1.3) About Memory Game :-**

A large range of Free Online memory Matching games for young and old: games for kids, for adults or seniors. It's the famous Memory® game, know as Concentration card game or Matching Game, where you need to match pairs by turn over 2 cards at a time. You can find the Quick game rules here, or you can also Download the Printable game rules in PDF format.

Matching Games have several levels of difficulty, you can change the number of cards on the board thanks to the buttons located below the game.

On Memorize, all Matching games have a 2 players mode, you can play with a friend or against the computer. There are also Big and Difficult matching games with many cards, or games grouped by specific themes like Animals, Cartoons, Sport, Learning games, Arts & Culture, Christmas, and many others... There is something for everyone!

The games on Memorize, are the online version of the famous Memory® matching cards game published by Ravens-burger in the 1960s which later became a classic of the board games in many countries all over there world.The matching games are compatible with all browsers and all devices

There are many great benefits when one plays memory-enhancing games, some of which are: Memory games exercise the brain, making it more sharp and alert. If you play memory games at least thirty minutes every day, your concentration and focusing ability will improve

**(1.4) EXISTING SYSTEM**

The content and the game adjust automatically to your device, so don't hesitate to play on a tablet or a smartphone. The purpose of this memory game: The purpose of this memory game is to memorize the locations of the cards in the game and to make pairs of cards by turning them over 2 by 2. When the 2 cards match, it's a pair!

Test and train your memory skills with our selection of great memory games. Brain Games; Puzzle Games; Daily Puzzles; Word Games; Crosswords; Sudoku; Memory; Math Games; Simon Says. Test your memory and remember the pattern of lights and sounds. Play. Casino Cards Memory. Memory game with cards. Play. Look Look. Fun Memory game: match the same cats. Play. Emoji Pairs

The purpose of this memory game is to memorize the locations of the cards in the game and to make pairs of cards by turning them over 2 by 2. When the 2 cards match, it's a pair! You win the pair and have the right to play again, otherwise the cards are automatically turned face down and you have to make a new try.

**(1.5) PROPSED SYSTEM**

This game is about testing memory

We have created memory games for adults-on levels of difficulty and you can increase the challenge step by step. There are games in each level, and they challenge the brain by changing every time. There is no time limit to beat the games, that depends on each player.

Implemented New and better Animation and used frontend view compatible view to the user and completely optimized which consume very low internet fast enough to the users to access by the users

1. **LITERATURE SURVEY**

The Memory Game. Search this site. Abstract; Literature Review; Problem & Hypothesis; Materials & Procedure; Data & Results; Conclusion; Bibliography; Display Poster; Abstract. Bibliography. Conclusion. Data & Results. Display Poster. Literature Review. Materials & Procedure. Problem & Hypothesis. Research Paper. Sitemap. etc. Her results were that it does affect your body.

One of the best web based games for memory exercises, Balls and Boxes will train your brain to keep track of the objects as they change their positions. All Online Games, fun brain teasers, memory techniques, Relaxing Games, Thinking Games Count the Sheep II Count the Sheep II is part memory game, part focus game.

Improve your kids memory and concentration with brain and memory games. It is a great way to improve your child’s cognitive skills that is what Kids World Fun brings through brain games. These memory games, when played regularly, can make your kid smarter and vigilant. Why blow all your money over tutorials that can bore your child to wits?

**3. SYSTEM ANALYSIS**

3.1 FEASIBILITY STUDY:-

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out.

Three key considerations involved in the feasibility analysis are

ECONOMICAL FEASIBILITY

TECHNICAL FEASIBILITY

SOCIAL FEASIBILITY

**ECONOMICAL FEASIBILITY:**

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

### TECHNICAL FEASIBILITY:

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

**SOCIAL FEASIBILITY:**

The 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 use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity.

**(3.2) Modules**

* Domain Address
* Web Hosting
* Front End vs Back End

**(3.3) Modules Description**

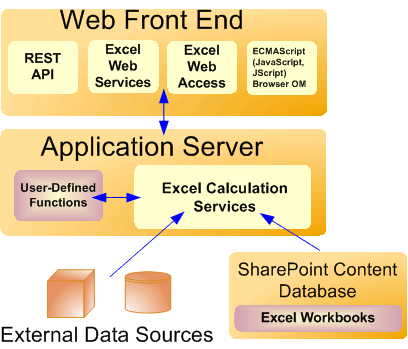
**Domain Address:-**

An Internet Protocol, or IP, address is different than a domain name. The IP address is an actual set of numerical instructions. It communicates exact information about the address in a way that is useful to the computer but makes no sense to humans. The domain name functions as a link to the IP address. Links do not contain actual information, but they do point to the place where the IP address information resides. It is convenient to think of IP addresses as the actual code and the domain name as a nickname for that code. A typical IP address looks like a string of numbers. It could be 232.17.43.22, for example. However, humans cannot understand or use that code. To summarize, the domain name is a part of the URL, which points to the IP address.

**WEB Hosting:-**

Web hosting is a service that allows organizations and individuals to post a website or web page onto the Internet. A web host, or web hosting service provider, is a business that provides the technologies and services needed for the website or webpage to be viewed in the Internet. Websites are hosted, or stored, on special computers called servers. When Internet users want to view your website, all they need to do is type your website address or domain into their browser. Their computer will then connect to your server and your webpages will be delivered to them through the browser.

Most hosting companies require that you own your domain in order to host with them. If you do not have a domain, the hosting companies will help you purchase one.



**Front End vs Back End:-**

The front end of a website or application refers to the part that users see and interact with. That said, front-end development refers to the creation and management of the front end of a website or application. This includes making sure the various coding is working and being actively presented to those interacting with a webpage. For example, a front-end developer would work to ensure all of a website's fonts, menus and buttons are in proper working order for users to interact with. This means that while web design is focused on the look of a website, front-end development is the means by which these designs and "looks" are shown on the website itself.

* Implement a website or application's visual elements
* Use responsive design in the creation of a website or application's user interface
* Test the website or application for usability
* Troubleshoot any code that isn't working
* Improve a website or application's visual architecture
* Modify website and application interfaces
* Collaborate with back-end developers

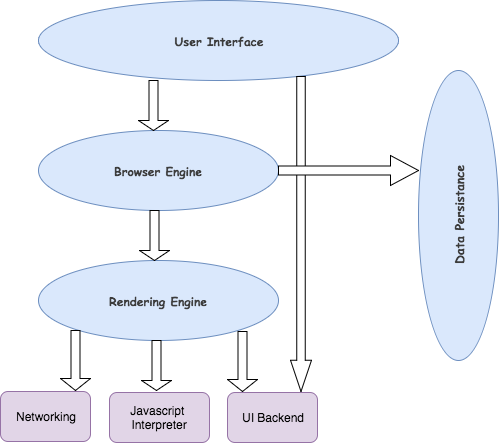
Back-end development refers to the behind-the-scenes of a website or application that isn't visible to a user. In other words, it's the storage and communication between a website or application's database and browser. The three components of a back end include the server, application and the database.

* Analyze data, processes and codes
* Participate in training to stay on top of current practices
* Collaborate with other team members as well as front-end developers
* Report data to necessary parties
* Create functional APIs and site core
* Monitor server status
* Design user interface

**SYSTEM ARCHITECTURE**

High-level architecture of browser

The below image shows the main components of a web browser:



**1)The User Interface**: The user interface is the space where User interacts with the browser. It includes the address bar, back and next buttons, home button, refresh and stop, bookmark option, etc. Every other part, except the window where requested web page is displayed, comes under it.

**2)The Browser Engine**: The browser engine works as a bridge between the User interface and the rendering engine. According to the inputs from various user interfaces, it queries and manipulates the rendering engine.

**3)The Rendering Engine:** The rendering engine, as the name suggests is responsible for rendering the requested web page on the browser screen. The rendering engine interprets the HTML, XML documents and images that are formatted using CSS and generates the layout that is displayed in the User Interface. However, using plugins or extensions, it can display other types data also. Different browsers user different rendering engines:

\* Internet Explorer: Trident

\* Firefox & other Mozilla browsers: Gecko

\* Chrome & Opera 15+: Blink

\* Chrome (iPhone) & Safari: Webkit

**4)Networkin**g: Component of the browser which retrieves the URLs using the common internet protocols of HTTP or FTP. The networking component handles all aspects of Internet communication and security. The network component may implement a cache of retrieved documents in order to reduce network traffic.

**5)JavaScript Interpreter**: It is the component of the browser which interprets and executes the javascript code embedded in a website. The interpreted results are sent to the rendering engine for display. If the script is external then first the resource is fetched from the network. Parser keeps on hold until the script is executed.

**6)UI Backend**: UI backend is used for drawing basic widgets like combo boxes and windows. This backend exposes a generic interface that is not platform specific. It underneath uses operating system user interface methods.

**7)Data Persistence/Storag**e: This is a persistence layer. Browsers support storage mechanisms such as localStorage, IndexedDB, WebSQL and FileSystem. It is a small database created on the local drive of the computer where the browser is installed. It manages user data such as cache, cookies, bookmarks and preferences.

**(4.1) SYSTEM REQUIREMENT**

**HARDWARE REQUIREMENTS:**

System : I 3 Core processor.

Hard Disk : 50 GB.

Monitor : 15’’ LED

Input Devices : Keyboard, Mouse

Ram : 4 GB

**SOFTWARE REQUIREMENTS:**

Operating system : Windows 7.

Coding Language : HTML,CSS,JAVASCRIPT

Editor : VS Code

Hosting : Heroku

**(4.2) Implementation Memory Game:-**

Memory puzzle game using python and pygame modules. Python is one of the easiest globally used programming languages, and using the pygame modules with it will give you a chance to easily implement many game applications. Pygame is an open free source that you can download later easily. You will learn about most of the objects and functions in the pygame that will help you to build your own game project. Also you will learn how to insert drawings, images and texts to your game.

**Learn step-by-step**

In a video that plays in a split-screen with your work area, your instructor will walk you through these steps:

Introduction:- Implementing language

Events:- working with Event functionality

Rectangles:- designing rectangle grid

Shapes:- learn designing shapes and design

Game Logic:- Implementing Logic is very difficult step and time consuming process a which make project completely completely working

Game Functions:- As defining game is a major part and accessing game according to functions required

Texts and Images:- Implementing text and images at real time working helps work much more visualization with color full content

**(5) PROJECT CODE**

**5.1) HTML FILE**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>COSMOS</title>

<!--GOOGLE FONTS-->

<link

href="https://fonts.googleapis.com/css2?family=VT323&display=swap"

rel="stylesheet"

/>

<!--BOOTSTRAP-->

<link

rel="stylesheet"

href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"

integrity="sha384-JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZOnIxN0hoP+VmmDGMN5t9UJ0Z"

crossorigin="anonymous"

/>

<script

src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"

integrity="sha384-B4gt1jrGC7Jh4AgTPSdUtOBvfO8shuf57BaghqFfPlYxofvL8/KUEfYiJOMMV+rV"

crossorigin="anonymous"

></script>

<link rel="stylesheet" href="styles2.css" />

</head>

<body id="game">

<header>

<h1 class="text-center mt-4 mb-2 headTag">COSMOS CONSCIOUS</h1>

</header>

<section class="container mx-auto">

<section class="mx-auto">

<div class="row">

<div class="col-12 text-center points">

<p>Battles won: <span id="point"> 0</span></p>

</div>

</div>

<div id="won" class="text-center">

<h1>You have conquered the galaxy!</h1>

<p>Battles won: <span id="finalPoints">0</span></p>

<button class="btn" id="playAgain">Conquer another galaxy</button>

</div>

<section class="memory-game">

<div class="memory-card" data-cards="card1">

<img class="front-face" src="Images/alien.svg" alt="avacado" />

<img class="back-face" src="Images/vortex.svg" alt="sky" />

</div>

<div class="memory-card" data-cards="card1">

<img class="front-face" src="Images/alien.svg" alt="avacado" />

<img class="back-face" src="Images/vortex.svg" alt="sky" />

</div>

<div class="memory-card" data-cards="card2">

<img class="front-face" src="Images/space-gun.svg" alt="burgers" />

<img class="back-face" src="Images/vortex.svg" alt="sky" />

</div>

<div class="memory-card" data-cards="card2">

<img class="front-face" src="Images/space-gun.svg" alt="burgers" />

<img class="back-face" src="Images/vortex.svg" alt="sky" />

</div>

<div class="memory-card" data-cards="card3">

<img class="front-face" src="Images/spaceship.svg" alt="carrot" />

<img class="back-face" src="Images/vortex.svg" alt="sky" />

</div>

<div class="memory-card" data-cards="card3">

<img class="front-face" src="Images/spaceship.svg" alt="carrot" />

<img class="back-face" src="Images/vortex.svg" alt="sky" />

</div>

<div class="memory-card" data-cards="card4">

<img class="front-face" src="Images/plant.svg" alt="cheese" />

<img class="back-face" src="Images/vortex.svg" alt="sky" />

</div>

<div class="memory-card" data-cards="card4">

<img class="front-face" src="Images/plant.svg" alt="cheese" />

<img class="back-face" src="Images/vortex.svg" alt="sky" />

</div>

<div class="memory-card" data-cards="card5">

<img class="front-face" src="Images/alien (1).svg" alt="pancakes" />

<img class="back-face" src="Images/vortex.svg" alt="sky" />

</div>

<div class="memory-card" data-cards="card5">

<img class="front-face" src="Images/alien (1).svg" alt="pancakes" />

<img class="back-face" src="Images/vortex.svg" alt="sky" />

</div>

<div class="memory-card" data-cards="card6">

<img class="front-face" src="Images/universe.svg" alt="salad" />

<img class="back-face" src="Images/vortex.svg" alt="sky" />

</div>

<div class="memory-card" data-cards="card6">

<img class="front-face" src="Images/universe.svg" alt="salad" />

<img class="back-face" src="Images/vortex.svg" alt="sky" />

</div>

</section>

</section>

</section>

<script src="script2.js"></script>

</body>

</html>

**5.2) CASCADING STYLE SHEET:-**

\* {

padding: 0;

margin: 0;

box-sizing: border-box;

font-family: "VT323", monospace;

}

body {

height: 100vh;

background-image: url("./Images/final.jpg");

background-size: cover;

background-position: center;

position: relative;

}

.open {

width: (100%);

display: block;

justify-content: center;

align-items: center;

margin-top: 100px;

}

.mainTag {

color: #c5c5c5;

font-weight: bolder;

}

.welcome {

color: #ffffff;

font-weight: 800;

width: 70%;

margin: auto;

margin-bottom: 20px;

}

.btn-handle {

border: 2px solid #2580b3;

color: #fff;

}

.btn-handle:hover {

background: linear-gradient(115.87deg, #14f1d9 0%, #3672f8 100%);

}

.headTag,

.points > p {

color: #c5c5c5;

font-weight: bolder;

}

#point {

font-weight: 1300;

}

.memory-game {

width: calc(25% - 10px);

height: calc(33.333% - 10px);

margin: auto;

display: flex;

flex-wrap: wrap;

perspective: 1000px;

z-index: -1;

}

.memory-card {

width: calc(25% - 10px);

height: calc(33.333% - 10px);

margin: 5px;

position: relative;

transform: scale(1);

transform-style: preserve-3d;

transition: transform 0.5s;

border-radius: 5px;

background-image: linear-gradient(-225deg, #76c3f0 0%, #2580b3 100%);

}

.memory-card:active {

transform: scale(0.97);

transition: transform 0.2s;

}

.memory-card.flip {

transform: rotateY(180deg);

}

.front-face,

.back-face {

width: 100%;

height: 100%;

padding: 10px;

position: absolute;

border-radius: 5px;

backface-visibility: hidden;

}

.front-face {

transform: rotateY(180deg);

}

#won {

visibility: hidden;

border-radius: 10px;

background: #ffbdf1;

background-image: linear-gradient(to bottom, #76c3f0 0%, #2580b3 100%);

margin: auto;

position: absolute;

top: 50%;

left: 50%;

margin-right: -50%;

transform: translate(-50%, -50%);

z-index: 1;

border: 5px solid #064d75;

}

#won p {

font-weight: bolder;

}

#playAgain {

border: 2px solid #2580b3;

border-radius: 5px;

background: #2580b3;

}

#playAgain:hover {

border: 2px solid #064d75;

background: linear-gradient(115.87deg, #14f1d9 0%, #3672f8 100%);

}

.mainTag {

font-size: 80px;

margin-top: 60px;

}

.welcome {

font-size: 35px;

}

btn-handle {

font-size: 40px;

padding: 10px 20px;

border-radius: 10px;

}

.memory-game {

width: 640px;

height: 530px;

}

.front-face,

.back-face {

padding: 10px;

}

#won {

padding: 50px;

}

.headTag {

font-size: 70px;

}

.points > p,

#won > h1,

#won p,

#playAgain {

font-size: 35px;

}

**5.3) JAVA SCRIPT:-**

const cards = document.querySelectorAll(".memory-card");

const score = document.getElementById("point");

const finalScore = document.getElementById("finalPoints");

const won = document.getElementById("won");

const play = document.getElementById("playAgain");

const button = document.getElementsByClassName("btn-handle");

const ship = document.getElementById("ship");

const body = document.getElementsByTagName("body")[0];

var points = 0;

var finalPoint = 0;

var win = 0;

let hasFlippedCard = false;

let lockBoard = false;

let firstCard, secondCard;

function flipCard() {

if (lockBoard) return;

if (this === firstCard) return;

this.classList.add("flip");

if (!hasFlippedCard) {

hasFlippedCard = true;

firstCard = this;

return;

}

secondCard = this;

checkCards();

}

function checkCards() {

let isMatch = firstCard.dataset.cards === secondCard.dataset.cards;

isMatch ? cardsMatch() : cardsDontMatch();

}

function cardsMatch() {

firstCard.removeEventListener("click", flipCard);

secondCard.removeEventListener("click", flipCard);

points += 4;

finalPoint = points;

win += 2;

finalScore.innerHTML = finalPoint;

score.innerHTML = points;

if (win === 12) {

won.style.visibility = "visible";

}

resetBoard();

}

function cardsDontMatch() {

lockBoard = true;

setTimeout(() => {

firstCard.classList.remove("flip");

secondCard.classList.remove("flip");

resetBoard();

}, 1000);

points -= 1;

finalPoint = points;

score.innerHTML = points;

}

function resetBoard() {

[hasFlippedCard, lockBoard] = [false, false];

[firstCard, secondCard] = [null, null];

}

function playAgain() {

location.reload();

}

play.addEventListener("click", playAgain);

(function shuffle() {

cards.forEach((card) => {

let randomPos = Math.floor(Math.random() \* 12);

card.style.order = randomPos;

});

})(); //IIFE

cards.forEach((card) => card.addEventListener("click", flipCard));

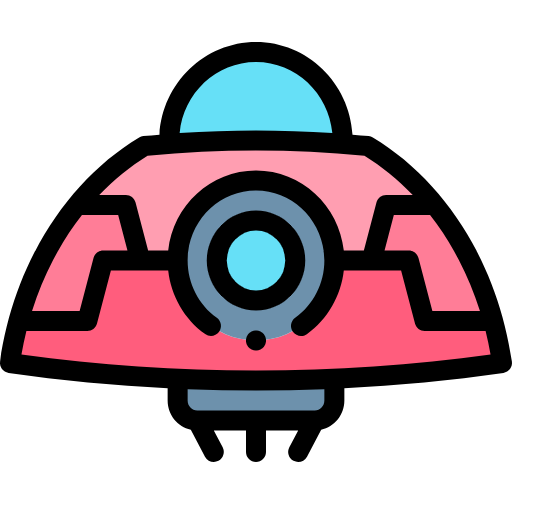
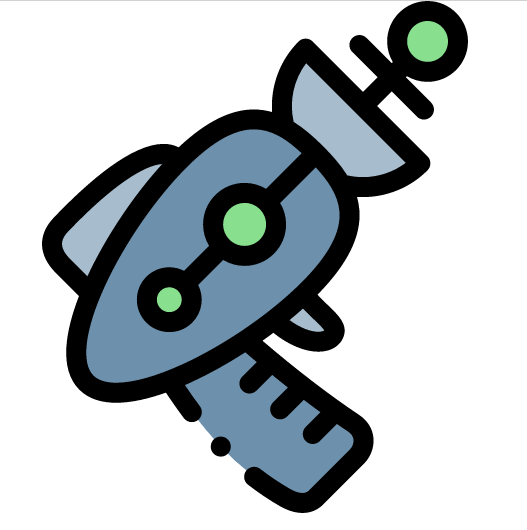
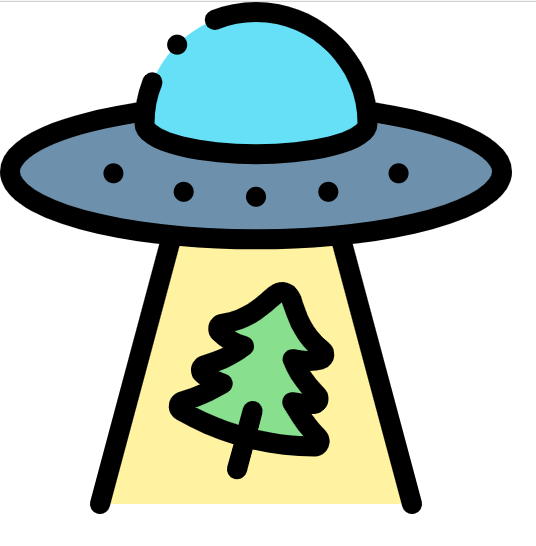
function shipMove() {

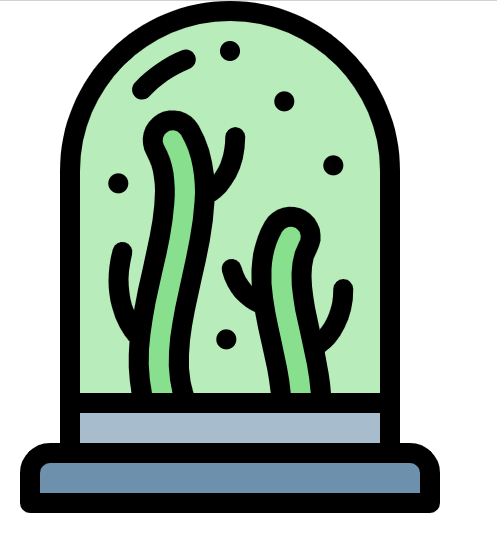
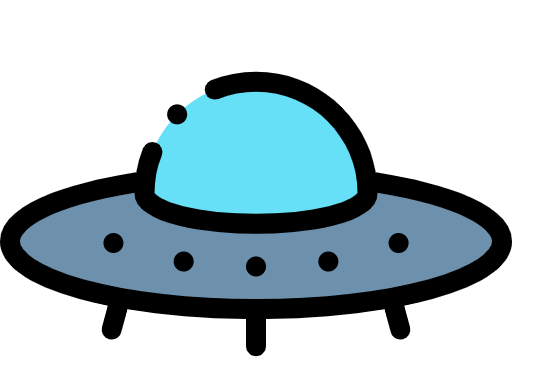
ship.classList.add = "animate\_\_slideOutRight";

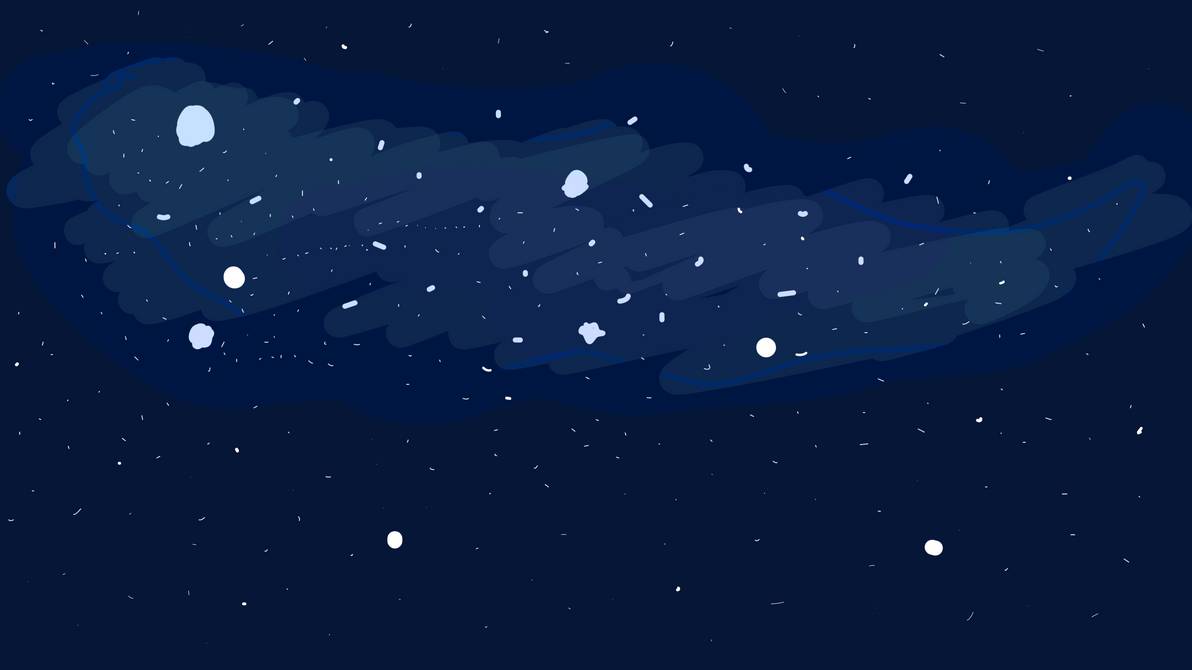
}

button.addEventListener("click", shipMove);

**(5.4) IMAGES USED IN PROJECT**



6. **SYSTEM TESTING**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies , assemblies and/or a finished product.

**6.1 TYPES OF TESTS**

**Unit testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration.

**Integration testing**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

**Functional test:** Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked. Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

**System Test**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration orient ed system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

**White Box Testing**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

**Black Box Testing:-**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

**Unit Testing:**-

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.Field testing will be performed manually and functional tests will be written in detail.

**Test objectives**

* All field entries must work properly.
* Pages must be activated from the identified link.
* The entry screen, messages and responses must not be delayed.

**Features to be tested**

* Verify that the entries are of the correct format
* No duplicate entries should be allowed
* All links should take the user to the correct page.

# Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**Acceptance Testing**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**6.2 TEST STRATEGY AND APPROACH**

**TEST STRATEGY** clarifies the major tasks and challenges of a test project. It is created to inform project managers, testers, and developers about key issues of the testing process. This includes the testing objective, methods of testing new functions, total time and resources required for the project, and the testing environment. The test approach is typically covered in the test strategy. It is a process to find effective way to proceed with testing of particular application and feature.

TEST STRATEGY SPECIFIES:

* Role of every team member
* Environments setup required
* Testing tools needed
* Risks involved

**TEST APPROACH** explains the type of testing that will be performed on the project. A test approach considers the following:

1) Criteria for approach selection

2) Approach for organizing tests

3) Approach for executing tests.

**Types:**  
 1. Proactive: In this approach, test designing is started ASAP to discover bugs before build's availability.  
 2. Reactive: In this approach, QA is not started till development team has completed their work on the feature.  
 Usually, Proactive test approach is followed. Proactive test approach and effective test strategy makes life easier of the tester.

**OUTPUT:-**







**8)HOSTING PROJECT**

Hosting project can solve portable issues and the project is about E-learning purpose

The project can be portable by hosting it online to the world . The hosting Require Domain name (which contain IP address in text format) which is easier to searching and remembering the name

As Github helps the project to store online and live to the world just by used name we can see our project and this issue haven’t solved completely

As so far HEROKU helps for solving the secure hosting problem by providing it free to the users with there subdomain the code can be upload through Github or using there own CLI commands through Terminal .

The easier way is upload the code in Github using Gitbash or VS CODE Terminal the code can be uploaded and linking the github account to heroku helps hosting project simple way

**(9) CONCLUSION**

Web Development On Web Conclusion Conclusion.The purpose and objective of Web.Web project is achieved. By providing extremely rich graphical user interface, web page designing is easy and in an aesthetic form. Flexibility in designing makes user explore their imagination and thus, even a novice user can dream and accomplish their wish of web page designing

Web Services are based on Loosely Coupled Architecture. This means that the interfaces of Web Services are dynamic (changes during a given timeline) in nature. But the client logic does not necessarily have to change while interacting with the service. This facilitates the integration of multiple software in a more efficient way.

js is an open source JavaScript framework through which you can build and render 2D games within a browser, with the help of HTML5 and Web Audio. While drawing a simple 2D object in raw requires you to deal with a huge amount of JavaScript code, Babylon.js lets you do the same thing with a minimal amount of code and very low level of complexity.

Web Hosting Resources. Categories. By Web Hosting Staff . Choosing a web host requires that you take an appraisal of your website, the technology used to create it, your web page elements and your future growth. You will also have to make room for emerging technologies and web trends

**10)REFERENCE**

[1]W3 school is the basic reference for the web development

[2]Sir Timothy John Berners-Lee OM KBE FRS FREng FRSA FBCS, is the founder of html. Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser.

[3] M. K. Aguilera, R. Janakiraman, and L. Xu, “Using Erasure Codes Efficiently for Storage in a Distributed System,” in International Conference on Dependable Systems and Networks (DSN), 2005, pp. 336–345.

[4] Heroku was founded in 2007 by Orion Henry, James Lindenbaum, and Adam Wiggins. The company was acquired by Salesforce in 2011, and the Heroku platform is now part of Salesforce Platform. 2007 July 2007

[5] Vellayappa Ayyadurai Shiva, December 2, 1963 Internet came in the late 1960s with the creation of ARPANET, or the Advanced Research Projects Agency Network. Originally funded by the U.S. Department of Defense, ARPANET