**INDUSTRIAL TRAINING AT REPOZITORY-TECHNOLOGIES-PRIVATE-LIMITED,**

**HISAR**

**REPORT**

***Submitted by***

**SAMEER KATARIA**

**Roll Number: 190040820020**

***In partial fulfillment of the requirements for the award of the diploma in***

**COMPUTER ENGINEERING**

******

**GOVT. POLYTECHNIC**

**HISAR -125001**

**AUGUST, 2021**

**DECLARATION**

I affirm that the Industrial Training report titled "INDUSTRIAL TRAINING AT REPOZITORY-TECHNOLOGIES-PRIVATE-LIMITED, HISAR" being submitted in partial fulfilment of the requirements for the award of the Diploma in COMPUTER ENGINEERING is the original work carried out by me. It has not formed the part of any other project work submitted for award of any degree or diploma either in this or any other Institution.

**(Signature)**

**SAMEER KATARIA**

**Roll No: 190040820020**

**ACKNOWLEDGEMENT**

I would like to express my special thanks of gratitude to my teacher as who gave me the golden opportunity to do this wonderful project on the topic of Web application development, which also helped me in doing a lot of Research and I came to know about so many new things I am really thankful to them.

(Signature of the Student)

**Sameer Kataria**

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **SR NO** | **CONTENTS** | **PAGE NO.** |
| 1. | Acknowledgement | 3 |
| 2. | Table of Contents | 4 |
| 3. | Abstract | 5 |
| 4. | Introduction | 6 |
| 5. | Software Used | 7 |
| 6. | Work Assignment | 8 |
| 7. | Language Used | 9 - 13 |
| 8. | Programs | 14 - 17 |
| 9. | Output | 18 |
| 10. | Conclusion | 19 |
| 11. | Reference | 20 |

**ABSTRACT**

It is a Notes Keeper System. This diploma thesis describe a case study, where we focusing on developing notes maker mechanism, using JavaScript. Use of simple CSS. It is easy to use and easy to understandable. It can be available on multiple operating system. It can be used in offline mode which means no need of Internet. I learned how to use JavaScript through my industrial training and HTML to create a functioning, interactive/responsive webpage. In this version, only one person can add notes to it and is not sharable. Here, the user type a note in panel and then submit it. After submitting the notes will be saved in local host and gets displayed on the webpage.

**INTRODUCTION**

Repozitory Technologies is a Web Development service provider. It specializes in Block chain Technology, Ecommerce Development, Artificial Intelligence, Web Development, Mobile App Development, Software Development services. The company has its head office located at Hisar and was founded in the year 2015.

Over the past few months, deep learning has gone from Web Development, focusing on theory and practical works. Web developers often work for clients who are trying to get their product or service onto the web. The work is typically very project focused and involves collaborating with a team that helps to coordinate the client’s needs into the end product. The client could be a tech company, an organization, or a government. The work could involve front-end, back-end, or full-stack web development. Web development could be a good profession for you if you like solving logical problems, building useful things, and experimenting with new technologies. Web developers are in high demand, generally have a good work/life balance, and command comfortable salaries.

The front end is the stuff you see on the website in your browser, including the presentation of content and user interface elements like the navigation bar. Front-end developers use HTML, CSS, JavaScript, and their relevant frameworks to ensure that content is presented effectively and that users have an excellent experience.

The back end refers to the guts of the application, which live on the server. The back end stores and serves program data to ensure that the front end has what it needs. This process can become very complicated when a website has millions of users. Back-end developers use programming languages like Java, Python, and Ruby to work with data.

Full-stack developers are comfortable working with both the front and back ends. In The Odin Project, we focus on teaching you full-stack development, covering all aspects of web development.

Different types of web developers, we mentioned earlier about the different types of clients and employers you may work with.

Large tech companies, such as Google, Facebook, and Amazon, have very stringent hiring requirements. If you successfully meet these expectations, they offer excellent pay, benefits, and opportunities.

Start-ups are a bit like the Wild West. For a junior developer, it can feel like a trial by fire because of the pace of development. Start-ups often offer slightly lower salaries and require longer hours, but they may also offer equity in the company and highly unique environments.

**SOFTWARE USED**

1. Visual Studio Code
2. Google Chrome

**TECHNOLOGIES USED**

1. HTML
2. CSS
3. JAVASCRIPT
4. BOOTSTRAP
5. CLOUD FLARE CDN

**WORK ASSIGNMENT**

I made a notes making web-based project by honing my skills and Industrial training.

1. **Displaying the Interface**

For displaying our main page or we can say structure of the website/local server, I have created an html document named Website.html. This will contain all of our code. Next, open the file in your preferred browser by using a VS-Code feature called open with live server. It will open our project on a local server on some random port if not set or we can change it manually if we are node.

1. **Making a New Note**

Now for taking a note from the user/client, I have worked in html and created a panel for the note and added a button to submit. And set the coordinates of these two and plus I have used bootstrap to make it responsive.

1. **Linking with JS**

Then I have linked my Website.html with app.js (JavaScript document) so that they can communicate and exchange data/information.

1. **Saving data input from user in Local Storage**

Here, I have used two methods getItem and setItem for storing the data input or we can say notes from the client in our local storage in the form of JSON (JavaScript Object Notation).

1. **Showing notes from Local Storage**

In this section I have used a method named getItem to collect data from the local storage and showing it on the webpage.

1. **Deleting a Specific note**

Here I have created a button named delete. If a user clicks on it, the specific note gets deleted.

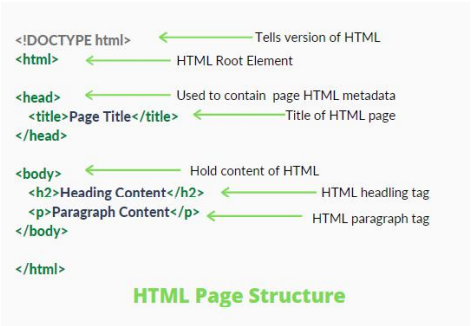
1. **Searching a Specific note**

We can search the specific note by typing the contents of it in search panel in navigation bar.

**LANGUAGES USED**

1. **HTML:-**

Its stands for HyperText Markup Language. It is used to design web pages using a markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages. A markup language is used to define the text document within tag which defines the structure of web pages. This language is used to annotate (make notes for the computer) text so that a machine can understand it and manipulate text accordingly. Most markup languages (e.g. HTML) are humanreadable. The language uses tags to define what manipulation has to be done on the text. HTML is a markup language used by the browser to manipulate text, images, and other content, in order to display it in the required format. HTML was created by Tim Berners-Lee in 1991. The first-ever version of HTML was HTML 1.0, but the first standard version was HTML 2.0, published in 1999. The basic structure of an HTML page is laid out below. It contains the essential building-block elements (i.e. doctype declaration, HTML, head, title, and body elements) upon which all web pages are created.



<DOCTYPE! html>: This is the document type declaration (not technically a tag). It declares a document as being an HTML document. The doctype declaration is not case- sensitive.

<html>: This is called the HTML root element. All other elements are contained within it.

<head>: The head tag contains the “behind the scenes” elements for a webpage. Elements within the head aren’t visible on the front-end of a webpage. HTML elements used inside the element include:

* <style>
* <title>
* <base>
* <noscript>
* <script>
* <meta>
* <meta>
* <link>
* <p>: t is used for defining a paragraph.
* <br>: It is used for a single-line break.
* <img>: It is used for defining an image with a given source.
* <sup>: It is used for defining superscripted data.
* <b>: It is used for defining bold text.
* <sub>: It is used for defining subscripted data, etc.

<body>: The body tag is used to enclose all the visible content of a webpage. In other words, the body content is what the browser will show on the front-end. An HTML document can be created using any text editor. Save the text file using .html or .htm. Once saved as an HTML document, the file can be opened as a webpage in the browser.

**II. CSS:-**

• Cascading Style Sheets, fondly referred to as CSS, is a simply designed language intended to simplify the process of making web pages presentable. CSS allows you to apply styles to web pages. More importantly, CSS enables you to do this independent of the HTML that makes up each web page.CSS is easy to learn and understand, but it provides powerful control over the presentation of an HTML document. CSS saves time: You can write CSS once and reuse the same sheet in multiple HTML pages.

• Easy Maintenance: To make a global change simply change the style, and all elements in all the webpages will be updated automatically.

• Search Engines: CSS is considered a clean coding technique, which means search engines won’t have to struggle to “read” its content.

• Superior styles to HTML: CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.

• Offline Browsing: CSS can store web applications locally with the help of an offline cache. Using this we can view offline websites.

**CSS Syntax:**

A CSS comprises style rules that are interpreted by the browser and then applied to the corresponding elements in your document.A style rule set consists of a selector and declaration block.

Selector -- h1

Declaration -- {color:blue;font size:12px;}

• The selector points to the HTML element you want to style.

• The declaration block contains one or more declarations separated by semicolons.

• Each declaration includes a CSS property name and a value, separated by a colon. For Example: –; color is property and blue is value. –; font-size is property and 12px is value.

• A CSS declaration always ends with a semicolon, and declaration blocks are surrounded by curly braces.

CSS Versions

1. CSS1

2. CSS2

3. CSS3

4. CSS4

Version 4 comes with:-

• CSS-Pro

• CSS-Mobile

Supported Browser:

• Google Chrome

• Microsoft Edge

• Firefox

• Opera

• Safari

Pure CSS Components: The Pure CSS has the collection of the following components:

• Base

• Grid

• Forms

• Buttons

• Tables

• Menus

**III. JAVASCRIPT:-**

JavaScript is a lightweight, cross-platform, and interpreted scripting language. It is well-known for the development of web pages, many non-browser environments also use it. JavaScript can be used for Client-side developments as well as Server-side developments. JavaScript contains a standard library of objects, like Array, Date, and Math, and a core set of language elements like operators, control structures, and statements.

• Client-side: It supplies objects to control a browser and it’s Document Object Model (DOM). Like if client-side extensions allow an application to place elements on an HTML form and respond to user events such as mouse clicks, form input, and page navigation. Useful libraries for the client-side are AngularJS, ReactJS, VueJS and so many others.

• Server-side: It supplies objects relevant to running JavaScript on a server. Like if the server-side extensions allow an application to communicate with a database, and provide continuity of information from one invocation to another of the application, or perform file manipulations on a server. The useful framework which is the most famous these days is node.js.

JavaScript can be added to your HTML file in two ways:

• Internal JS: We can add JavaScript directly to our HTML file by writing the code inside the <script> tag. The <script> tag can either be placed inside the <head> or

the <body> tag according to the requirement.

• External JS: We can write JavaScript code in other file having an extension .js and

then link this file inside the <head> tag of the HTML file in which we want to add

this code.

Syntax:

<script>

// JavaScript Code

</script>

History of JavaScript: It was created in 1995 by Brendan Eich while he was an engineer at Netscape. It was originally going to be named LiveScript but was renamed. Unlike most programming languages, the JavaScript language has no concept of input or output. It is designed to run as a scripting language in a host environment, and it is up to the host environment to provide mechanisms for communicating with the outside world. The most common host environment is the browser. Features of JavaScript: According to a recent survey conducted by Stack Overflow, JavaScript is the most popular language on earth. With advances in browser technology and JavaScript having moved into the server with Node.js and other frameworks, JavaScript is capable of so much more. Here are a few things that we can do with JavaScript:

• JavaScript was created in the first place for DOM manipulation. Earlier websites were mostly static, after JS was created dynamic Web sites were made.

• Functions in JS are objects. They may have properties and methods just like another object. They can be passed as arguments in other functions.

• Can handle date and time.

• Performs Form Validation although the forms are created using HTML.

• No compiler is needed.

Applications of JavaScript:

• Web Development: Adding interactivity and behavior to static sites JavaScript was invented to do this in 1995. By using AngularJS that can be achieved so easily.

• Web Applications: With technology, browsers have improved to the extent that a language was required to create robust web applications. When we explore a map in Google Maps then we only need to click and drag the mouse. All detailed view is just a click away, and this is possible only because of JavaScript. It uses Application Programming Interfaces(APIs) that provide extra power to the code. The Electron and React is helpful in this department.

• Server Applications: With the help of Node.js, JavaScript made its way from client to server and node.js is the most powerful on the server-side.

• Games: Not only in websites, but JavaScript also helps in creating games for leisure. The combination of JavaScript and HTML 5 makes JavaScript popular in game development as well. It provides the EaseJS library which provides solutions for working with rich graphics.

• Smartwatches: JavaScript is being used in all possible devices and applications. It provides a library PebbleJS which is used in smartwatch applications. This framework works for applications that require the internet for its functioning.

• Art: Artists and designers can create whatever they want using JavaScript to draw on HTML 5 canvas, make the sound more effective also can be used p5.js library.

• Machine Learning: This JavaScript ml5.js library can be used in web development by using machine learning.

Limitations of JavaScript:

• Performance: JavaScript does not provide the same level of performance as offered by many traditional languages as a complex program written in JavaScript would be comparatively slow. But as JavaScript is used to perform simple tasks in a browser, so performance is not considered a big restriction in its use.

• Complexity: To master a scripting language, programmers must have a thorough knowledge of all the programming concepts, core language objects, client and server-side objects otherwise it would be difficult for them to write advanced scripts using JavaScript.

• Weak error handling and type checking facilities: It is weakly typed language as there is no need to specify the data type of the variable. So wrong type checking is not performed by compile.

**PROGRAMS**

**• HTML Code**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <meta http-equiv="X-UA-Compatible" content="ie=edge">

    <title>Notes App</title>

    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"

        integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">

<!-- Adding CSS -->

<style>

        body {

            background: url('https://images.unsplash.com/photo-1507842217343-583bb7270b66?ixlib=rb-1.2.1&ixid=MnwxMjA3fDB8MHxzZWFyY2h8MXx8bGlicmFyeXxlbnwwfHwwfHw%3D&w=1000&q=80');

            height: 100vh;

            background-size: cover;

            background-repeat: no-repeat;

            background-position: center;

            background-color: antiquewhite;

        }

    </style>

</head>

<body>

    <nav class="navbar navbar-expand-lg navbar-dark bg-dark">

        <a class="navbar-brand" href="#">Magic Notes</a>

        <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent"

            aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

            <span class="navbar-toggler-icon"></span>

        </button>

        <div class="collapse navbar-collapse" id="navbarSupportedContent">

            <ul class="navbar-nav mr-auto">

            </ul>

            <form class="form-inline my-2 my-lg-0">

                <input class="form-control mr-sm-2" id="searchTxt" type="search" placeholder="Search"

                    aria-label="Search">

                <button class="btn btn-outline-success my-2 my-sm-0" type="submit">Search</button>

            </form>

        </div>

    </nav>

    <div class="container my-3">

        <h1 style="    color: #f8e1c3;

        background-color: #6a6666;

        text-align: center;

        font-size: 3.5rem">Welcome To Magic Notes</h1>

        <div class="card">

            <div class="card-body">

                <h5 class="card-title">Add a note</h5>

                <div class="form-group">

                    <textarea class="form-control" id="addTxt" rows="3"></textarea>

                </div>

                <button class="btn btn-primary" id="addBtn">Add Note</button>

            </div>

        </div>

        <hr>

        <h1 style="color:#f8e1c3;background-color: #6a6666;">Your Notes</h1>

        <hr>

        <div id="notes" class="row container-fluid"> </div>

    </div>

<!-- Adding jquery -->

    <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"

        integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"

        crossorigin="anonymous"></script>

<!-- Adding cdn file by Cloud flare -->

    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"

        integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"

        crossorigin="anonymous"></script>

<!-- Adding bootstrap -->

    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"

        integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"

        crossorigin="anonymous"></script>

<!-- Linking JS Externally -->

    <script src="/app.js"></script>

</body>

</html>

* **JAVASCRIPT**

console.log("Welcome to notes app. This is app.js");

showNotes();

// If user adds a note, add it to the localStorage

let addBtn = document.getElementById("addBtn");

addBtn.addEventListener("click", function (e) {

  let addTxt = document.getElementById("addTxt");

  let notes = localStorage.getItem("notes");

  if (notes == null) {

    notesObj = [];

  } else {

    notesObj = JSON.parse(notes);

  }

  notesObj.push(addTxt.value);

  localStorage.setItem("notes", JSON.stringify(notesObj));

  addTxt.value = "";

  showNotes();

});

// Function to show elements from localStorage

function showNotes() {

  let notes = localStorage.getItem("notes");

  if (notes == null) {

    notesObj = [];

  } else {

    notesObj = JSON.parse(notes);

  }

  let html = "";

  notesObj.forEach(function (element, index) {

    html += `

            <div class="noteCard my-2 mx-2 card" style="width: 18rem;">

                    <div class="card-body">

                    <input type="checkbox">

                        <h5 style="display:contents" class="card-title" >Note ${index + 1}</h5>

                        <p class="card-text"> ${element}</p>

                        <button id="${index}"onclick="deleteNote(this.id)" class="btn btn-primary">Delete Note</button>

                    </div>

                </div>`;

  });

  let notesElm = document.getElementById("notes");

  if (notesObj.length != 0) {

    notesElm.innerHTML = html;

  } if (notesObj.length === 0) {

    notesElm.innerHTML = `Nothing to show! Use "Add a Note" section above to add notes.`;

    notesElm.style.color = "#f8e1c3";

  }

}

// Function to delete a note

function deleteNote(index) {

  let notes = localStorage.getItem("notes");

  if (notes == null) {

    notesObj = [];

  } else {

    notesObj = JSON.parse(notes);

  }

  notesObj.splice(index, 1);

  localStorage.setItem("notes", JSON.stringify(notesObj));

  showNotes();

}

let search = document.getElementById('searchTxt');

search.addEventListener("input", function () {

  let inputVal = search.value.toLowerCase();

  let noteCards = document.getElementsByClassName('noteCard');

  Array.from(noteCards).forEach(function (element) {

    let cardTxt = element.getElementsByTagName("p")[0].innerText;

    if (cardTxt.includes(inputVal)) {

      element.style.display = "block";

    }

    else {

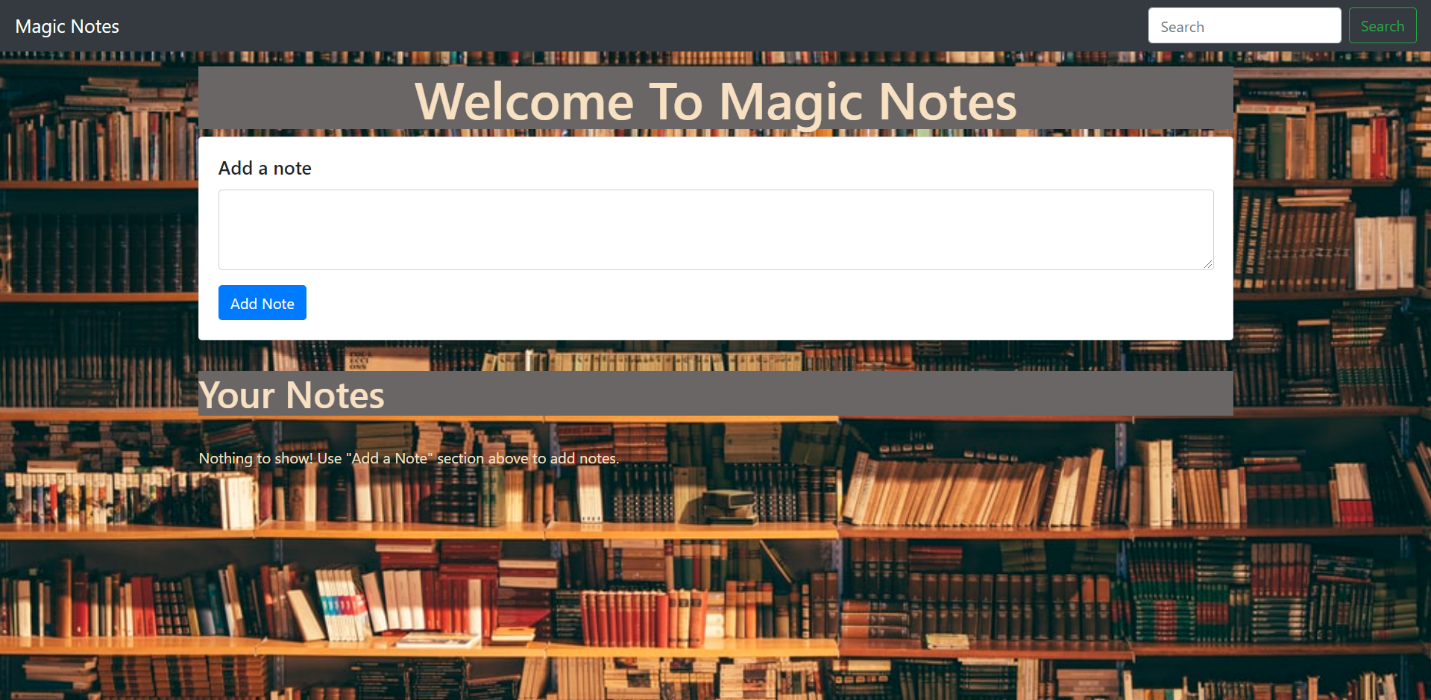
      element.style.display = "none";

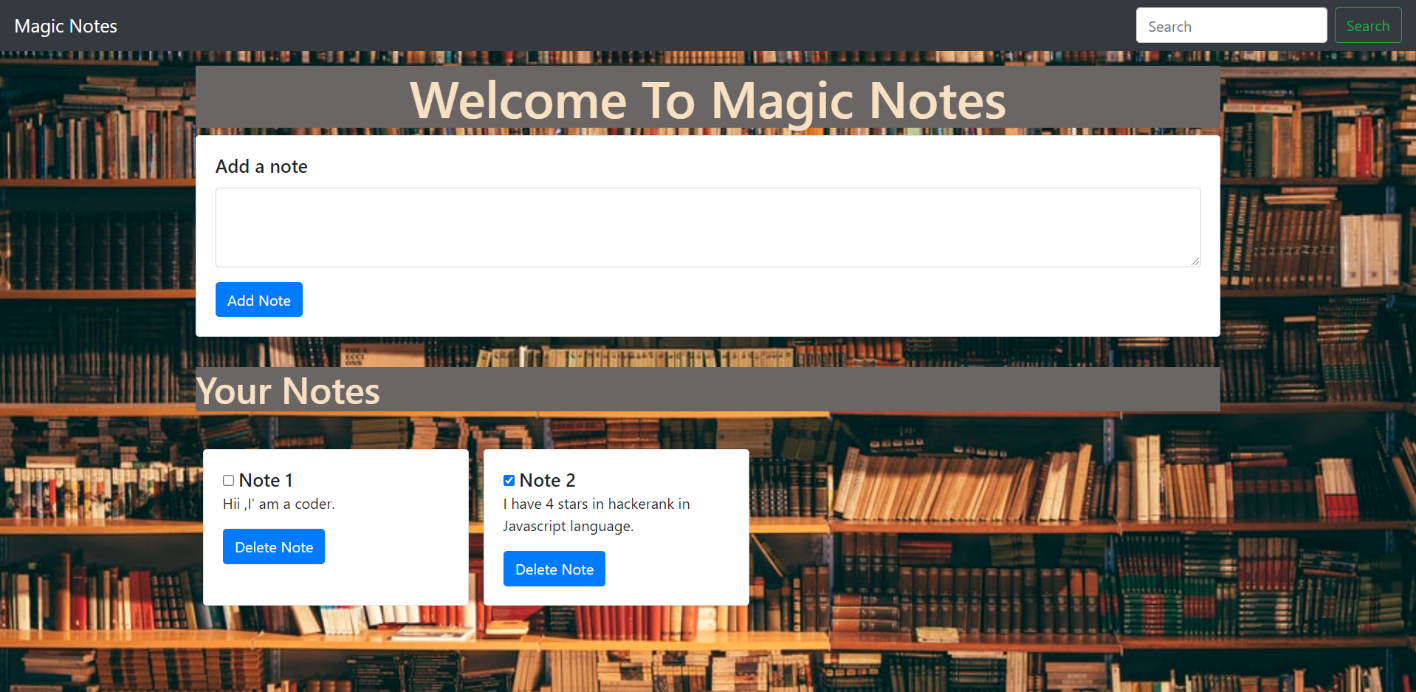
    }

  })

})

**OUTPUT**





**CONCLUSION**

This project has really been for entertainment and enjoyable. It has made us learn and understand the many trivial concepts of HTML, CSS, and JavaScript Language. As I have used lots of JavaScript method an in-built function it provides various controls, add, delete, buttons, search panel, navigation bar and so many things build a user friendly application. Finally it has taught us a valuable lifelong lesson about the improvements and working and interacting in a group.

**Further Features:**

1. Add Title

2. Mark a note as Important

3. Separate notes by user

4. Sync and host to web server

5. Setting reminder

6. Sharing notes on different platforms

**REFERENCES**

1. [www.youtube.com](http://www.youtube.com)
2. [www.w3schools.com](http://www.w3schools.com)
3. [www.geeksforgeeks.com](http://www.geeksforgeeks.com)
4. [www.google.com](http://www.google.com)
5. cdnjs.cloudflare.com