Internship Report

on

Web Development

In partial fulfillment for the award of the degree of

Bachelor of technology

In

Computer science and engineering

By

D.VENKATA CHARAN

Regd.No:21785A0506

Under the guidance of

Name, Designation

PRODIGY INFOTECH, BANGALORE

15/July/2023 to 31/Aug/2023

SRI VENKATESWARA COLLEGE OF ENGINEERING AND TECHNOLOGY

(AUTONOMOUS)

R.V.S NAGAR, CHITTOOR-517 127(A.P)

(Approved by AICTE ,New Delhi , Approved by AICTE new Delhi , Accredited by NAAC Affiliated to JNTUA ,Ananthapur)

CERTIFICATE

This is to certify that Internship report entitled "**WEB DEVELOPMENT"** is the bonafide work done by the student in the sri venkateswara college of engineering and technology is the bonafide work of "**D.VENKATA CHARAN (Reg No: 21785A0506)"**in partial fulfillment of the requirement of the award of the **BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING** during the Academic Year 2023-2024.

Guide:	**Head Of Department**	**Principal**
Prodigy Infotech	Dr.P.Jyotheeswari M.tech.,Ph.D	Dr.M.Mohan Babu M.Tech.,Ph.D

SRI VENKATESWARA COLLEGE OF ENGINEERING AND TECHNOLOGY

(AUTONOMOUS)

R.V.S NAGAR, CHITTOOR-517 127(A.P)

(Approved by AICTE ,New Delhi , Approved by AICTE new Delhi , Accredited by NAAC Affiliated to JNTUA ,Ananthapur)

CIN: PIT/AUG23/0198

CERTIFICATE

OF COMPLETION

07/09/2023

THIS CERTIFICATE IS PROUDLY PRESENTED TO

D.CHARAN

for completing their **6 Weeks** Internship in **Web Development** with outstanding remarks at Prodigy InfoTech in **August 23.**

m

ACKNOWLEDGEMENT

This satisfaction that accompanies the successful completion of the task would be put incomplete without the mention of the people who made it possible, whose constant guidance and encouragement, crown the efforts with success.

We convey our gratitude to our esteemed management and honorable Principal, **Dr. M.MOHAN BABU** and head of the department, **DR.P.JYOTHEESWARI** for providing all the facilities and support.

We would like to thank **Mr. HARE RAM SINGH**, Department Coordinator, Computer Science and Engineering for providing us innovative ideas, motivating and inspiring us throughout the completion of this internship..

I sincerely express my gratitude towards the **Prodigy Infotech** forproviding this opportunity.

We also thank the teaching and non-teaching staff of our college for their valuable guidance and constant support during the completion of this internship

We express our deep sense of gratitude to our parents and well-wishers who helped us a lot in making this internship successful.

Date:
Place:

D.VENKATA CHARAN
21785A0506
CSE-B

TABLE OF CONTENTS

COMPANY PROFILE:

About Prodigy Infotech:

Prodigy Infotech is an internship and online training platform.It was founded by Sarvesh aggarwal ,an IIT Madras alumnus, in 2010 , thewebsite helps students find internships with organizations in India.

History:

The platform which was founded in 2010, started as a WordPress blog that aggregated internships across India and articles on education ,technology andskill gap. Prodigy Infotech launced its online internships in 2014. As of 2018, the platform had million students and 80,000 companies.

Awards And Recognition:

In 2011 , the website became a part of NAACOM 10k startups . In 2015 Prodigy Infotech was a finalist in people Matters TechHR 2025 Spotlight Awards under Futurism in Recruitment category.

ABOUT HTML

HTML:

- HTML stands for Hypertext Markup Language, and it is the most widely used to write web pages.
- Hypertext refers to the way in which Web pages (HTML documents) are linked together. Thus, the link available on a webpage is called Hypertext.
- As its name suggests, HTML is a Markup Language which means you use HTML to simply "mark-up" a text document with tags that tell a Web browser how to structure it to display.
- Originally, HTML was developed with the intent of defining the structure of documents like headings, paragraphs, lists, and so forth to facilitate the sharing of scientific information between researchers.

Basic HTML Document:

In its simplest form, following is an example of an HTML document:

```
<!DOCTYPE html>
<html>
<head>
<title>This is document title</title>
</head>
<body>
<h1>This is a heading</h1>
<p>Document content goes here ....</p>
</body>
```

- <html>

Output:

This is the heading

Document content goes here…

HTML Tags:

As told earlier, HTML is a markup language and makes use of various tags to format the content. These tags are enclosed within angle braces <Tag Name>. Except few tags, most of the tags have their corresponding closing tags. For example, <html> has its closing tag</html> and <body> tag has its closing tag </body> tag etc.

Above example of HTML document uses the following tags:

Tags:

<!DOCTYPE...>

This tag defines the document type and HTML version.

<html>

This tag encloses the complete HTML document and mainly comprises of document header which is represented by <head>...</head> and document body which is represented by <body>...</body> tags.

<head>

This tag represents the document's header which can keep other HTML tags like <title>, <link> etc.

<title>

The <title> tag is used inside the <head> tag to mention the document title.

<**body>**

This tag represents the document's body which keeps other HTML tags like <h1>, , <p> etc.

<h1>

This tag represents the heading.

<p>

This represents a paragraph.

HTML Document Structure:

A typical HTML document will have the following structure:

Document declaration tag

<html>

<head>

Document header related tags

</head>

<body>

Document body related tags

</body>

</html>

The <!DOCTYPE> Declaration:

The <!DOCTYPE> declaration tag is used by the web browser to understand the version of the HTML used in the document. Current version of HTML is 5 and it makes use of the following declaration:

<!DOCTYPE html>

There are many other declaration types which can be used in HTML document depending on what version of HTML is being used.

Heading Tags:

- Any document starts with a heading. You can use different sizes for your headings. HTML also has six levels of headings, which use the elements <h1>, <h2>, <h3>, <h4>, <h5>, and <h6>. While displaying any heading, browser adds one line before and one line after that heading.

Example:

```
<!DOCTYPE html>nmn
<html>
<head>
<title>Heading Example</title>
</head>
<body>
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
<h4>This is heading 4</h4>
<h5>This is heading 5</h5>
<h6>This is heading 6</h6>
</body>
</html>
```

Output:

This is heading 1

This is heading 2

This is heading 3

This is heading 4

This is heading5

This is heading6

Paragraph Tag:

- The tag offers a way to structure your text into different paragraphs. Each paragraph of text should go in between an opening and a closing tag as shown below in the example:

Example:

```
<!DOCTYPE html>
<html>
<head>
<title>Paragraph Example</title>
</head>
<body>
<p>Here is a first paragraph of text.</p>
<p>Here is a second paragraph of text.</p>
<p>Here is a third paragraph of text.</p>
</body>
</html>
```

Output:

Here is a first paragraph of text.

Here is a second paragraph of text.

Here is a third paragraph of text.

Line Break Tag:

- Whenever you use the
 element, anything following it starts from the next line. This tag is an example of an empty element, where you do not need opening and closing there is nothing to go in between them.
- The
 tag has a space between the characters br and the forward slash. If you omit this space, older browsers will have trouble rendering the line break, while if you miss the forward slash character and just use
 it is not valid in XHTML.

Example:

```
<!DOCTYPE html>
<html>
<head>
<title>Line Break Example</title>
</head>
<body>
<p>Hello<br />
You delivered your assignment on time.<br />
Thanks<br />
Mahnaz</p>
</body>
</html>
```

Output:

Hello

You delivered your assignment on time

HTML FORMATTING

Bold Text:

Anything that appears within ... element, is displayed in bold as shown below:

Example:

- <!DOCTYPE html>
- <html>
- <head>
- </head>
- The following word uses a bold typeface.
- </body>
- </html>

Output:

The following word uses a bold typeface.

Italic Text:

- <body>

➢ Anything that appears within ... element is displayed in italicized as shown below:

Example:

- <!DOCTYPE html>
- <html>
- <head>
- <title>Italic Text Example</title>
- </head>

- <body>
- The following word uses a italicized typeface.
- </body>
- </html>

Output:

➢ The following word uses an *italicized* typeface.

Underlined Text:

➢ Anything that appears within <u>...</u> element, is displayed with underline as shown below:

Example:

- <!DOCTYPE html>
- <html>
- <head>
- <title>Underlined Text Example</title>
- </head>
- <body>
- The following word uses a <u>underlined</u> typeface.
- </body>
- </html>

Output:

➢ The following word uses an <u>underlined</u> typeface.

Strike Text:

- Anything that appears within <strike>...</strike> element is displayed with strikethrough,which is a thin line through the text as shown below:

Example:

```
<!DOCTYPE html>
<html>
<head>
<title>Strike Text Example</title>
</head>
<body>
<p>The following word uses a <strike>strikethrough</strike> typeface.</p>
</body>
</html>
```

Output:

The following word uses a ~~strikethrough~~ typeface.

Monospaced Font:

The content of a <tt>...</tt> element is written in monospaced font. Most of the fonts are known as variable-width fonts because different letters are of different widths (for example,the letter 'm' is wider than the letter 'i'). In a monospaced font, however, each letter has the
same width.

Example:

```
<!DOCTYPE html>
<html>
<head>
<title>Monospaced Font Example</title>
</head>
<body>
<p>The following word uses a <tt>monospaced</tt> typeface.</p>
</body>
</html>
```

Output:

The following word uses a monospaced typeface.

Superscript Text:

- The content of a ^{...} element is written in superscript; the font size used is the same size as the characters surrounding it but is displayed half a character's height above the other characters.

Example:

```
<!DOCTYPE html>
<html>
<head>
<title>Superscript Text Example</title>
</head>
<body>
```

- The following word uses a ^{superscript} typeface.
- </body>
- </html>

Output:

The following word uses a superscript typeface.

Subscript Text:

The content of a _{...} element is written in subscript; the font size used is the same as the characters surrounding it, but is displayed half a character's height beneath the other characters.

Example:

- <!DOCTYPE html>
- <html>
- <head>
- <title>Subscript Text Example</title>
- </head>
- <body>
- The following word uses a _{subscript} typeface.
- </body>
- </html>

Output:

The following word uses a subscript typeface.

CSS

What is 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 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, as well as a variety of other effects.
- CSS is easy to learn and understand but it provides a powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

Advantages of CSS:

- **CSS saves time** - You can write CSS once and then reuse the same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many web pages as you want.
- **Pages load faster** - If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So, less code means faster download times.
- **Easy maintenance** - To make a global change, simply change the style, and all the elements in all the web pages will be updated automatically.
- **Superior styles to HTML** - CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.zz

more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cellphones or for printing.

- **Global web standards** – Now HTML attributes are being deprecated and it is being recommended to use CSS. So it's a good idea to start using CSS in all the HTML pages to make them compatible with future browsers.

CSS-SYNTAX:

- A CSS comprises of style rules that are interpreted by the browser and then applied to thecorresponding elements in your document. A style rule is made of three parts:

Selector:

A selector is an HTML tag at which a style will be applied. This could be any tag like <h1> or <table> etc.

Property:

A property is a type of attribute of HTML tag. Put simply, all the HTML attributes are converted into CSS properties. They could be color, border, etc.

Value:

Values are assigned to properties. For example, color property can have the value either red or #F1F1F1 etc.

You can put CSS Style Rule Syntax as follows:

selector { property: value }

Example: You can define a table border as follows:

table{ border :1px solid #C00; }

- Here table is a selector and border is a property and the given value 1px solid

#C00 is the value of that property.

- You can define selectors in various simple ways based on your comfort. Let me put these selectors one by one.

The Type Selectors:

- This is the same selector we have seen above. Again, one more example to give a color to all level 1 headings:

```
h1 {
color: #36CFFF;
}
```

The Universal Selectors:

- Rather than selecting elements of a specific type, the universal selector quite simply matches the name of any element type:

SYNTAX:

```
13
{
color: #000000;
}
```

- This rule renders the content of every element in our document in black.

The Child Selectors:

- You have seen the descendant selectors. There is one more type of selector, which is very similar to descendants but have different functionality. Consider the following example:

```
body > p {
color: #000000;
```

- }

➢ This rule will render all the paragraphs in black if they are a direct child of the <body> element. Other paragraphs put inside other elements like or <td> would not have any effect of this rule.

The Attribute Selectors:

➢ You can also apply styles to HTML elements with particular attributes. The style rule below will match all the input elements having a type attribute with a value of text:

- input[type="text"]{
- color: #000000;
- }

➢ The advantage to this method is that the <input type="submit" /> element is unaffected, and the color applied only to the desired text fields.

➢ There are following rules applied to attribute selector.

➢ **p[lang]** - Selects all paragraph elements with a lang attribute.

➢ **p[lang="fr"]** - Selects all paragraph elements whose lang attribute has a value of exactly "fr".

➢ **p[lang~="fr"]** - Selects all paragraph elements whose lang attribute contains the word "fr".

➢ **p[lang|="en"]** - Selects all paragraph elements whose lang attribute contains values that are exactly "en", or begin with "en-".

➢ **Multiple Style Rules:**

➢ You may need to define multiple style rules for a single element. You can define these rules to combine multiple properties and corresponding values into a single

block as defined in the following example:

```
h1 {
color: #36C;
font-weight: normal;
letter-spacing: .4em;
margin-bottom: 1em;
text-transform: lowercase;
```

CSS Comments:

- Many times, you may need to put additional comments in your style sheet blocks. So, it is very easy to comment any part in the style sheet. You can simply put your comments inside /*.....this is a comment in style sheet.... */.
- You can use /* ....*/ to comment multi-line blocks in similar way you do in C and C++ programming languages.

Example:

```
/* This is an external style sheet file */
h1, h2, h3 {
color: #36C;
font-weight: normal;
letter-spacing: .4em;
margin-bottom: 1em;
text-transform: lowercase;
}
/* end of style rules. */
```

ABOUT JAVASCRIPT

What is JavaScript?

- JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages.
- It is an interpreted programming language with object-oriented capabilities.
- JavaScript was first known as LiveScript, but Netscape changed its name to JavaScript, possibly because of the excitement being generated by Java.
- JavaScript made its first appearance in Netscape 2.0 in 1995 with the name LiveScript.
- The general-purpose core of the language has been embedded in Netscape, Internet Explorer, and other web browsers.
- The ECMA-262 Specification defined a standard version of the core JavaScript
- language.

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

Advantages of JavaScript:

The merits of using JavaScript are:

- **Less server interaction**: You can validate user input before sending the page

off to the server. This saves server traffic, which means less load on your server.

- **Immediate feedback to the visitors:** They don't have to wait for a page reload to see if they have forgotten to enter something.
- **Increased interactivity**: You can create interfaces that react when the user hovers over them with a mouse or activates them via the keyboard.
- **Richer interfaces**: You can use JavaScript to include such items as drag-and-drop components and sliders to give a Rich Interface to your site visitors.

Limitations of JavaScript:

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

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

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

JavaScript Development Tools:

- One of major strengths of JavaScript is that it does not require expensive development tools. You can start with a simple text editor such as Notepad. Since an interpreted language inside the context of a web browser, you don't even need to make our life simpler, various vendors have come up with very nice JavaScript Editing tools. Some of them are listed here:
- **Microsoft FrontPage:** Microsoft has developed a popular HTML editor called FrontPage. FrontPage also provides web developers with a number of JavaScript tools to assist in the creation of interactive websites.
- **Macromedia Dreamweaver MX**: Macromedia Dreamweaver MX is a very popular HTML and JavaScript editor in the professional web development crowd. It provides several handy prebuilt JavaScript components, integrates

JAVA SCRIPT SYNTAX

Syntax:

- JavaScript can be implemented using JavaScript statements that are placed within the <script>... </script> HTML tags in a web page.
- You can place the <script> tags, containing your JavaScript, anywhere within you web page, but it is normally recommended that you should keep it within the <head> tags.
- The <script> tag alerts the browser program to start interpreting all the text between these tags as a script. A simple syntax of your JavaScript will appear as follows.

- <script ...>
- JavaScript code
- </script>

The script tag takes two important attributes:

- **Language**: This attribute specifies what scripting language you are using. Typically, its value will be javaScript. Although recent versions of HTML (and XHTML, its successor) have phased out the use of this attribute.
- **Type**: This attribute is what is now recommended to indicate the scripting language in use and its value should be set to "text/javascript".

- <script language="javascript" type="text/javascript">

Your First JavaScript Code:

Let us take a sample example to print out "Hello World". We added an optional HTML comment that surrounds our JavaScript code. This is to save our code from a browser that does not support JavaScript. The comment ends with a "//-->". Here "//" signifies a comment in JavaScript, so we add that to prevent a browser from reading the end of the HTML comment as a piece of JavaScript code. Next, we call a function document.write which writes a string into our HTML document.

This function can be used to write text, HTML, or both. Take a look at the following

Example:

```
<html>
<body>
<script language="javascript" type="text/javascript">
<!--
document.write ("Hello World!")
//-->
</script>
</body>
</html>
```

Result:

Hello World!

Comments in JavaScript:

JavaScript supports both C-style and C++-style comments. Thus:

- Any text between a // and the end of a line is treated as a comment and is ignored by JavaScript.
- Any text between the characters /* and */ is treated as a comment. This may span multiple lines.
- JavaScript also recognizes the HTML comment opening sequence <!--
- JavaScript treats this as a single-line comment, just as it does the // comment.
- The HTML comment closing sequence --> is not recognized by JavaScript so it should be written as //-->.

Example:

The following example shows how to use comments in JavaScript.

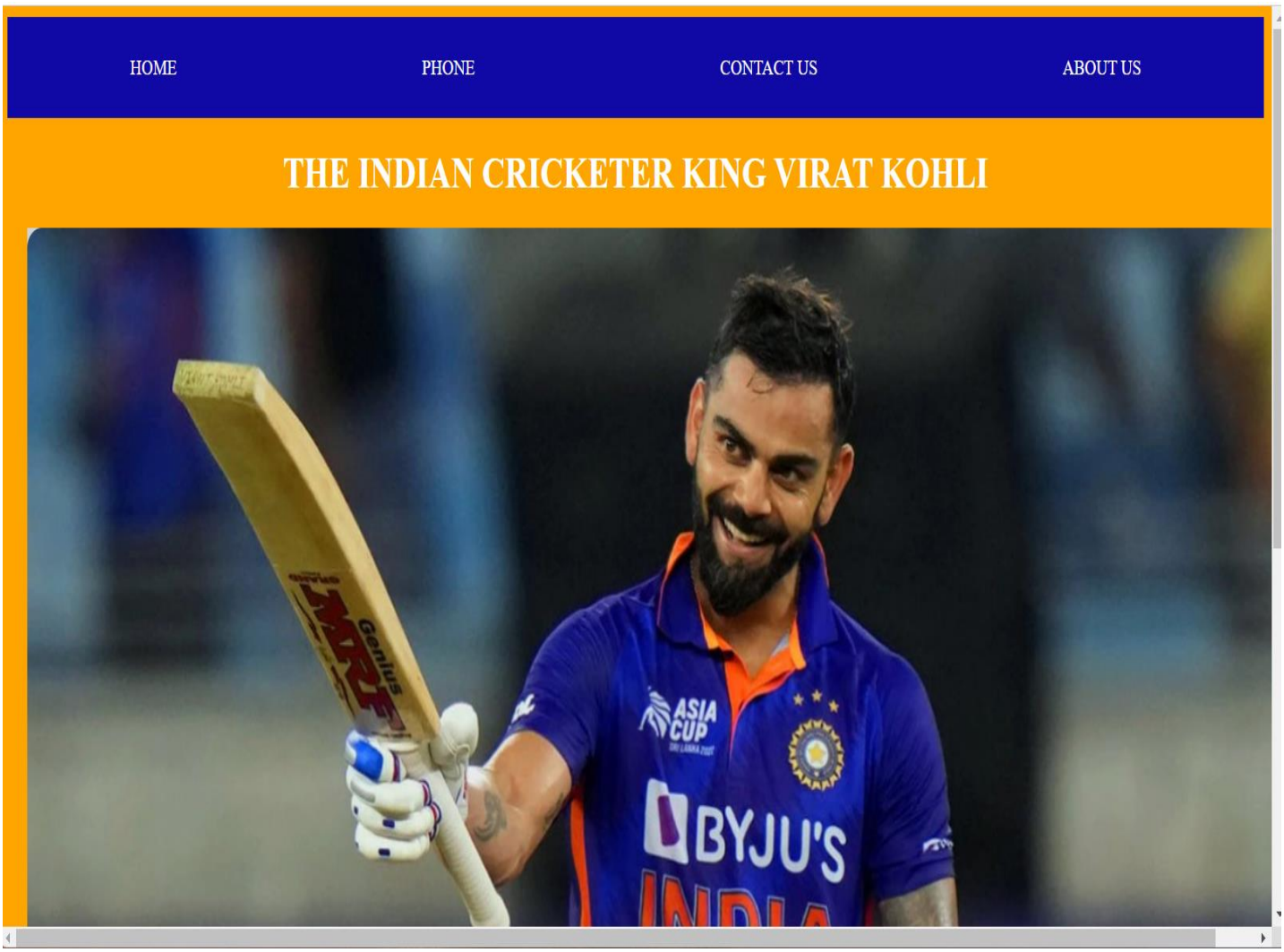
```
<script language="javascript" type="text/javascript">
<!--
// This is a comment. It is similar to comments in C++
/*
  This is a multiline comment in JavaScript
  It is very similar to comments in C Programming
*/
//-->
</script>
```

PROJECT

RESPONSIVE LANDING PAGE:

Creating a responsive landing page involves using HTML and CSS and possibly some java script code to ensure that the page layout and design adapt well to different sizes and devices

Here how a Responsive landing page looks like:



3.0 CONCLUSION

1. I believe that trail has shown conclusively that it is both possible and desirable to use python as the principle teaching language.

2. It is free (as in both cost and source code).

3. It is trivial to install on windows PC allowing students to take their interest further. For many the hurdle of installing a pascal or C Compiler on a windows machine is either too expensive or too complicated.

4. Most importantly, it's clean syntax offers increased understanding and enjoyment for students.

7.REFERENCES

- **https://in.linkedin.com/company/prodigyinfotech**
- **https://www.w3schools.com/**
- **https://internship.aicteindia.org/login_new.php**
- **https://learn.verzeo.in/users/sign_in**
- **https://prodigyinfotech.dev/**