

**SIX WEEK INDUSTRIAL TRAINING REPORT ON
NODE CHAT APP**

AT

OOPS INFO SOLUTIONS

SUBMITTED IN PARTIAL FULFILLMENT FOR AWARD DEGREE OF

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

BY

SUBHAM DAS

(SG-17349)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

UIET, PANJAB UNIVERSITY SSG REGIONAL CENTRE,

HOSHIARPUR-146021, Punjab (INDIA)

Training Completion Certificate issued by industry

DECLARATION

The work embodied in the training report entitled, “**NODE CHAT APP**” submitted to the department of Computer Science and Engineering at UIET, Panjab University Swami Sarvanand Giri Regional Centre, Hoshiarpur for the six week industrial training, has been done by me. The training report is entirely based on my own work. All ideas and references have been duly acknowledged.

Name and Signature of student

Countersigned by:

(Supervisor)

ACKNOWLEDGEMENT

It gives me a great sense of pleasure to present the report of this computer science based project. This project would not have been possible without the kind support of my instructor at training centre. I am highly indebted to him for his guidance and constant supervision as well as for providing the necessary information regarding the project and also for their support in completing the project I would also not like to miss the opportunity to acknowledge the contribution of all the faculty members of the department for their kind assistance and cooperation during the development of our project. Last but not the least , my thanks and appreciation go to my parents and colleagues for being a constant support of help and encouragement throughout the project.

ABSTRACT

Node Chatting app is a tool used to chat with multiple users in real time. Also user can share his/her current location in the chat.

Technologies used in project are :

- HTML
- CSS
- JAVASCRIPT
- NODE.JS
- SOCKET.IO

CONTENTS

S.No.	Title	Page No.
Chapter 1	Introduction to project	1
1.1	Overview	1
1.2	Project composition	1
Chapter 3	Snapshots	2-3
Chapter 4	Technologies used	4-22
4.1	HTML	4
4.2	CSS	13
4.3	JAVASCRIPT	15
4.4	NODE.JS	19
4.5	SOCKET.IO	20
Chapter 5	Conclusion and future scope	23

CHAPTER 1 – INTRODUCTION TO PROJECT

Node Chatting app is a tool used to chat with multiple users in real time.

Technologies used in project are :

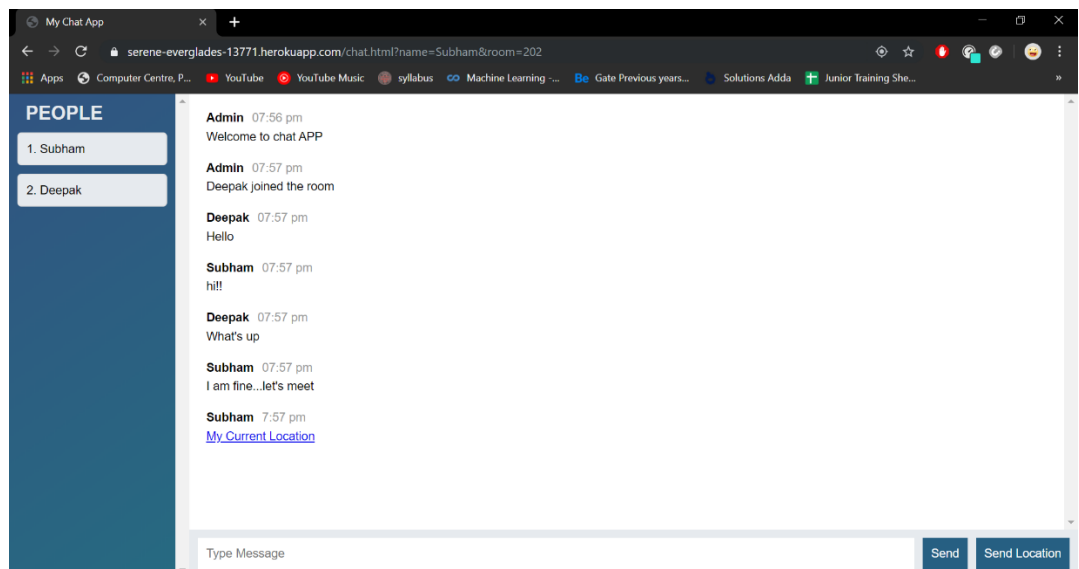
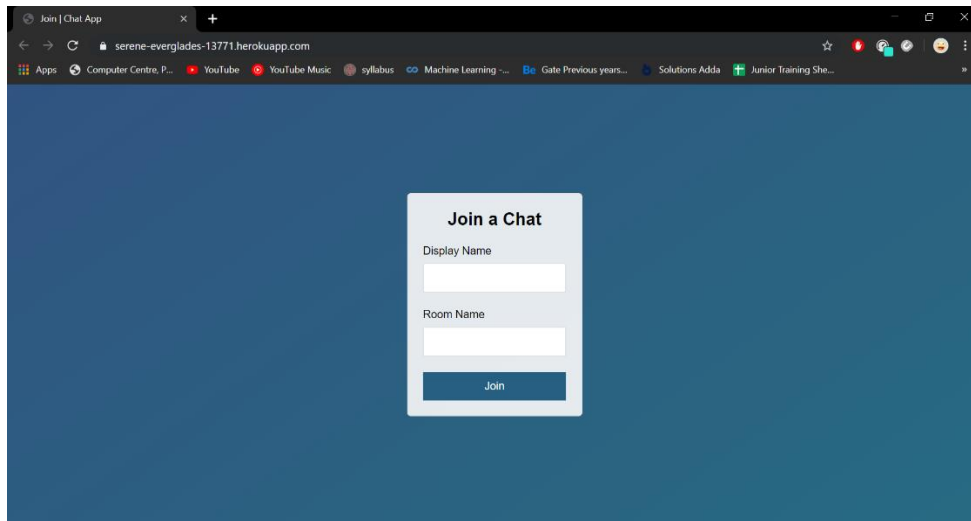
- HTML
- CSS
- JAVASCRIPT
- NODE.JS
- SOCKET.IO

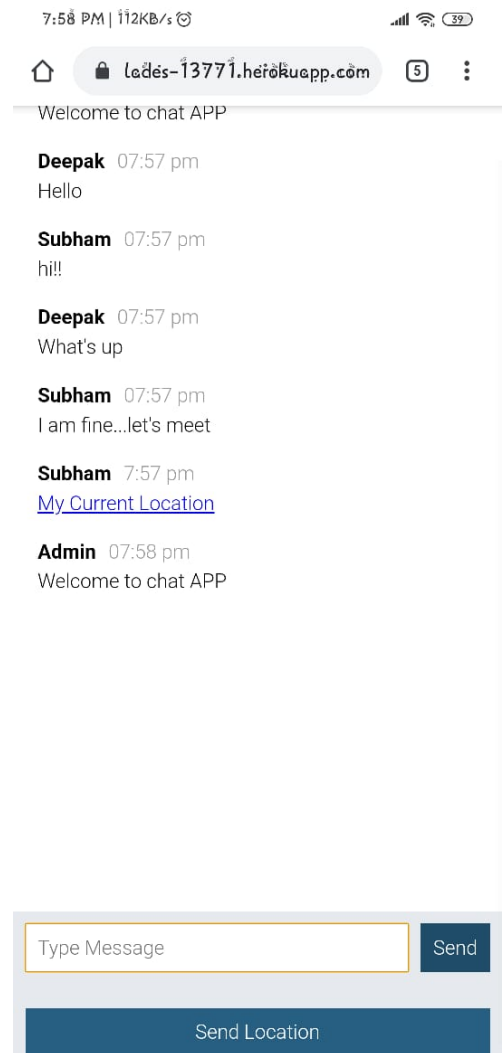
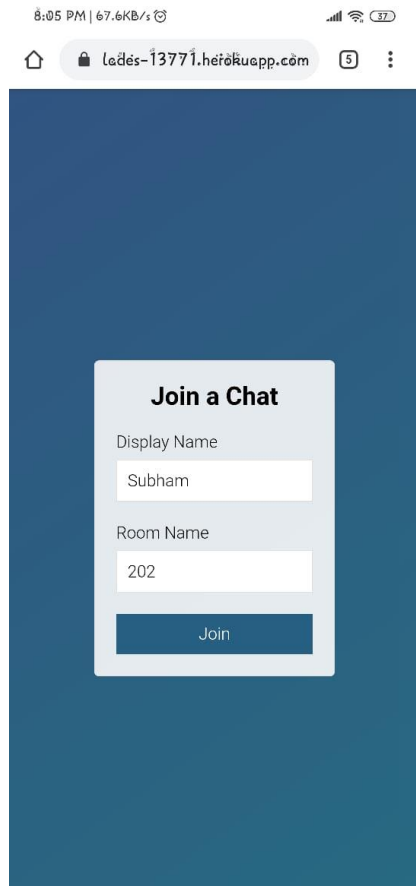
Messaging apps now have more global users than traditional social networks—which means they will play an increasingly important role in the distribution of digital journalism in the future. Drawing upon our interviews and case studies, we identify a number of opportunities and challenges for organizations using—or hoping to use—messaging apps for news. We argue that to devise a successful messaging app strategy, publishers must understand regional strongholds, user demographics, and popular features of each app. As happened after the early days of social media, before which a proliferation of services (some with regional strengths) led to intense competition for user attention, we expect to see some eventual consolidation among chat apps. Elsewhere, we conclude that issues around information, privacy, personal security, and mobile data penetration will unfold in different ways around the world; apps like Telegram and Fire Chat are among those at the forefront of addressing and solving these problems. In developing editorial strategies for some of these wide-ranging messaging platforms, news organizations are not just helping to future-proof themselves, they are also venturing into online spaces that could enable them to reach hundreds of millions of (often young) people with whom they have never engaged before.

Messaging has emerged as the new frontier of social on mobile, and the sheer size of audiences on the top global chat apps is too big to ignore. These apps also present an opportunity to diversify mobile traffic sources and to minimize vulnerability should Facebook or other platforms decrease traffic for publishers.

CHAPTER 2 – SNAPSHOTS OF THE PROJECT

The following snapshots explain the whole structure of the project.





CHAPTER 3 – TECHNOLOGIES USED

4.1 HTML

HTML is an acronym which stands for Hyper Text Markup Language. Let's see what Hyper Text is and what is Markup Language?

Hyper Text: Hyper Text simply means "Text within Text". A text has a link within it, is a hypertext. Every time when you click on a word which brings you to a new webpage, you have clicked on a hypertext.

Markup language: A markup language is a programming language that is used make text more interactive and dynamic. It can turn a text into images, tables, links etc.

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 father of HTML. The first available description of HTML was a document called "HTML Tags" proposed by Tim in late 1991.

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 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 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.

HTML DOCUMENT:

An HTML document is made of many HTML tags and each HTML tag contains different content.

HTML Tags:

HTML tags contain three main parts: opening tag, content and closing tag. But some HTML tags are unclosed tags.

Syntax:

`<tag> content </tag>`

Basic document of HTML.

`<!DOCTYPE html>`

`<head>`

`<title>ABCD</title>`

`</head>`

`<body>`

`<h1>First Heading</h1>`

`<p>Paragraph.</p>`

`</body>`

`</html>`

When a web browser reads an HTML document, browser reads it from top to bottom and left to right. HTML tags are used to create HTML documents and render their properties. Each HTML tags have different properties.

- **Unclosed HTML Tags:**

Some HTML tags are not closed, for example `br` and `hr`.

`
` Tag: `br` stands for break line, it breaks the line of the code.

`<hr>` Tag: `hr` stands for Horizontal Rule. This tag is used to put a line across the webpage.

- **HTML Text Tags:**

`<p>`, `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, `<h6>`, ``, ``, `<abbr>`, `<acronym>`,
`<address>`, `<bdo>`, `<blockquote>`, `<cite>`, `<q>`, `<code>`, `<ins>`, ``, `<dfn>`, `<kbd>`,
`<pre>`, `<samp>`, `<var>` and `
`.

- **HTML Link Tags:**

`<a>` and `<base>`

- **HTML Image and Object Tags:**

``, `<area>`, `<map>`, `<param>` and `<object>`

- **HTML List Tags:**

``, ``, ``, `<dl>`, `<dt>` and `<dd>`

- **HTML Table Tags:**

table, tr, td, th, tbody, thead, tfoot, col, colgroup and caption

- **HTML Form Tags:**

form, input, textarea, select, option, optgroup, button, label, fieldset and legend

- **HTML Formatting:**

HTML Formatting is a process of formatting text for better look and feel. There are many formatting tags in HTML. These tags are used to make text bold, italicized, or underlined. There are almost some options available that how text appears in HTML and XHTML.

Here are some HTML formatting tags:

1) Bold Text

If you write anything within `.....` element, is shown in bold letters.

Example:

`<p>Paragraph in bold text.</p>` Output:

Paragraph in bold text.

2) Italic Text

If you write anything within `<i>.....</i>` element, is shown in italic letters.

Example:

`<p><i>Paragraph in italic text.</i></p>` Output:

Paragraph in italic text.

3)HTML Marked formatting

If you want to mark or highlight a text, you should write the content within

`<mark>.....</mark>`.

Example:

`<p><mark> Mark</mark></p>`

Output: **Mark**

4) Strike Text

Anything written within `<strike>.....</strike>` element is displayed with strikethrough.

It is a thin line which cross the statement.

Example:

`<p><strike>Paragraph with strikethrough</strike></p>` Output:

~~Paragraph with strikethrough~~

5) Superscript Text

If you put the content within `^{.....}` element, is shown in superscript ; means it is displayed half a character's height above the other characters.

Example:

`<p>Hello ^{Paragraph in superscript.}</p>` Output:

Hello Paragraph in superscript.

6) Subscript Text

If you put the content within `_{.....}` element, is shown in subscript ; means it is displayed half a character's height below the other characters.

Example:

`<p>Hello _{Paragraph in subscript.}</p>` Output:

Hello First Paragraph in subscript.

HTML Heading:

A HTML heading or HTML h tag can be defined as a title or a subtitle which you want to display on the webpage. When you place the text within the heading tags `<h1>.....</h1>`, it is displayed on the browser in the bold format and size of the text depends on the number of heading.

There are six different HTML headings which are defined with the `<h1>` to `<h6>` tags.

h1 is the largest heading tag and h6 is the smallest one. So h1 is used for most important heading and h6 is used for least important.

Example:

`<!DOCTYPE html>`

`<html>`

`<body>`

`<h1>Heading no. 1</h1><h2>Heading no. 2</h2>`

`<h3>Heading no. 3</h3><h4>Heading no. 4</h4>`

```
<h5>Heading no. 5</h5><h6>Heading no. 6</h6>  
</body>  
</html>
```

Output:

Heading no. 1

Heading no. 2

Heading no. 3

Heading no. 4

Heading no. 5

Heading no. 6

HTML Paragraph:

HTML paragraph or HTML p tag is used to define a paragraph in a webpage. Let's take a simple example to see how it work. It is a notable point that a browser itself add an empty line before and after a paragraph.

Example:

```
<p>It's a Paragraph.</p> Output:
```

It's a Paragraph.

HTML Image

HTML img tag is used to display image on the web page. HTML img tag is an empty tag that contains attributes only, closing tags are not used in HTML image element.

Example of HTML image:

```
<!DOCTYPE html>  
<body>  
  
</body></html> Output:
```



Attributes of HTML img tag:

The src and alt are important attributes of HTML img tag. All attributes of HTML image tag are given below:

1) src

It is a necessary attribute that describes the source or path of the image. It instructs the browser where to look for the image on the server.

The location of image may be on the same directory or another server.

2) alt

The alt attribute defines an alternate text for the image, if it can't be displayed. The value of the alt attribute describe the image in words. The alt attribute is considered good for SEO prospective. **3) width**

It is an optional attribute which is used to specify the width to display the image. It is not recommended now. You should apply CSS in place of width attribute.

4)height

It specifies the height of the image. The HTML height attribute also supports iframe, image and object elements. It is not recommended now. You should apply CSS in place of height attribute.

HTML Table:

HTML table tag is used to display data in tabular form (row * column). There can be many columns in a row.

HTML tables are used to manage the layout of the page e.g. header section, navigation bar, body content, footer section etc. But it is recommended to use div tag over table to manage the layout of the page .

HTML Table Tags:

Tag	Description
<table>	It defines a table.

<tr>	It defines a row in a table.
<th>	It defines a header cell in a table.
<td>	It defines a cell in a table.
<caption>	It defines the table caption.
<colgroup>	It specifies a group of one or more columns in a table for formatting.
<col>	It is used with <colgroup> element to specify column properties for each column.

HTML Table Example:

```
<!DOCTYPE html>
<body>
<table>
<tr><th>First_Name</th><th>Last_Name</th></tr>
<tr><td>Namrata</td><td>Sachdeva</td></tr>
<tr><td>Vanisha</td><td>Rajput</td></tr>
<tr><td>Ayushi</td><td>Rehal</td></tr>
<tr><td>Deeksha</td><td>Thakur</td></tr>
</table>
</body></html>
```

Output:

First_Name	Last_Name
Namrata	Sachdeva
Vanisha	Rajput
Ayushi	Rehal
Deeksha	Thakur

HTML Lists:

HTML Lists are used to specify lists of information. All lists may contain one or more list elements. There are three different types of HTML lists:

1. Ordered List or Numbered List (ol)

2. Unordered List or Bulleted List (ul) 3. Description List or Definition List (dl) 1) **HTML Ordered List or Numbered List:**

List:

In the ordered HTML lists, all the list items are marked with numbers. It is known as numbered list also. The ordered list starts with tag and the list items start with tag.

```
<!DOCTYPE html>
<body>
<ol>
<li>Aries</li><li>Leo</li><li>Oracle</li>
</ol>
</body></html>
```

 Output:

- 1.Aries
2. Leo
- 3.Oracle

HTML Ordered List or Numbered List displays elements in numbered format. The HTML ol tag is used for ordered list. There can be different types of numbered list:

- Numeric Number (1, 2, 3)
- Capital Roman Number (I II III)
- Small Roman Number (i ii iii)
- Capital Alphabet (A B C)
- Small Alphabet (a b c)

<ol type="I" start="5"> : It will show numeric values starting with "5".

<ol type="A" start="5"> : It will show capital alphabets starting with "E".

<ol type="a" start="5"> : It will show lower case alphabets starting with "e".

<ol type="I" start="5"> : It will show Roman upper case value starting with "V".

<ol type="i" start="5"> : It will show Roman lower case value starting with "v".

2) HTML Unordered List or Bulleted List:

In HTML Unordered list, all the list items are marked with bullets. It is also known as bulleted list also. The Unordered list starts with tag and list items start with the tag.

```
<ul>
```



```
<li>Aries</li><li>Leo</li><li>Oracle</li>
</ul>
```

Output:

- Aries
- Leo
- Oracle

3) HTML Description List or Definition List:

HTML Description list is also a list style which is supported by HTML and XHTML. It is also known as definition list where entries are listed like a dictionary or encyclopedia.

The definition list is very appropriate when you want to present glossary, list of terms or other name-value list.

The HTML definition list contains following three tags:

1. <dl> tag defines the start of the list.
2. <dt> tag defines a term.
3. <dd> tag defines the term definition.

Example:

```
<dl>
<dt>Aries</dt>
<dd>-One of the 12 horoscope sign.</dd>
<dt>Leo</dt>
<dd>-It is also an one of the 12 horoscope sign.</dd>
<dt>Oracle</dt>
<dd>-It is a multinational technology corporation.</dd>
</dl>
```

Output:

Aries

-One of the 12 horoscope sign.

Leo

-It is also an one of the 12 horoscope sign.

Oracle

-It is a multinational technology corporation.

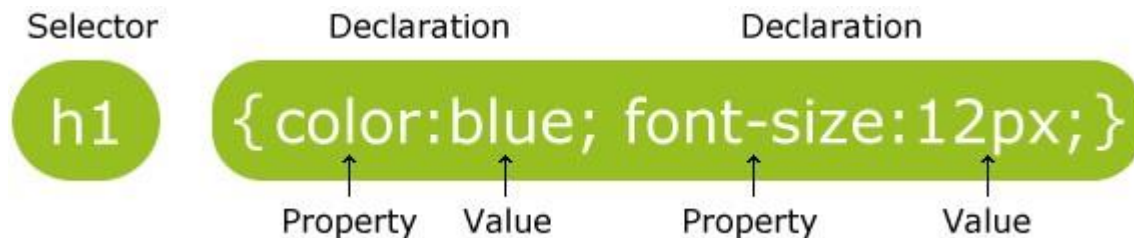
4.2 CSS

C.S.S. stands for Cascading Style Sheets, documents which contain styling rules for applying to HTML. Multiple style rules can point to one HTML element, in which case there needs to be a way to determine which rule should take effect. The term cascading describes the process of filtering down from general rules, until the most specific rule is encountered. That rule is then chosen to do its work.

Through CSS, is read by web browsers. They take the markup document, then they apply the styling rules to the elements within that document. •CSS removed the style formatting from the HTML page.

CSS Syntax

A CSS rule-set consists of a selector and a declaration block:



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.

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

TYPES OF STYLE SHEETS

There are three ways of inserting a style sheet:

- External style sheet
- Internal style sheet
- Inline style

EXTERNAL STYLE SHEET

With an external style sheet, you can change the look of an entire website by changing just one file!

Each page must include a reference to the external style sheet file inside the <link> element. The <link> element goes inside the <head> section:

EXAMPLE:- <!DOCTYPE

html>

<html>

<head>

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

</head>

<body>

<h1> this is a heading </h1>

<p> this is a paragraph </p>

</body>

</html>

An external style sheet can be written in any text editor. The file should not contain any html tags. The style sheet file must be saved with a .css extension.

Internal Style Sheet

An internal style sheet may be used if one single page has a unique style.

Internal styles are defined within the <style> element, inside the <head> section of an HTML page: Example

```
<head> <style> body {  
background-color: linen;  
} h1 { color:  
maroon; margin-  
left: 40px;  
}  
</style>  
</head>
```

Inline Styles

An inline style may be used to apply a unique style for a single element.

To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.

The example below shows how to change the color and the left margin of a <h1> element:

Example

```
<h1 style="color:blue;margin-left:30px;">This is a heading</h1>
```

An inline style loses many of the advantages of a style sheet (by mixing content with presentation). Use this method sparingly.

Cascading Order

We can say that all the styles will "cascade" into a new "virtual" style sheet by the following rules, where number one has the highest priority:

- Inline style (inside an HTML element)
- External and internal style sheets (in the head section)
- Browser default

So, an inline style has the highest priority, which means that it will override a style defined inside the <head> tag, or in an external style sheet, or a browser default value.

4.3 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

Client-side JavaScript

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

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

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

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

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

The merits of using JavaScript are –

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

Limitations of JavaScript

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

- Client-side JavaScript does not allow the reading or writing of files. This has been kept for security reason.

- JavaScript cannot be used for networking applications because there is no such support available.
- JavaScript doesn't have any 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 it is an interpreted language inside the context of a web browser, you don't even need to buy a compiler.

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 well with databases, and conforms to new standards such as XHTML and XML.
- Macromedia HomeSite 5 – HomeSite 5 is a well-liked HTML and JavaScript editor from Macromedia that can be used to manage personal websites effectively.

Where is JavaScript Today ?

The ECMAScript Edition 5 standard will be the first update to be released in over four years. JavaScript 2.0 conforms to Edition 5 of the ECMAScript standard, and the difference between the two is extremely minor.

The specification for JavaScript 2.0 can be found on the following site:

<http://www.ecmascript.org/>

Today, Netscape's JavaScript and Microsoft's JScript conform to the ECMAScript standard, although both the languages still support the features that are not a part of the standard.

What can JavaScript do?

1. JavaScript Can Change HTML Content

One of many JavaScript HTML methods is getElementById().

EXAMPLE:

```
<!DOCTYPE html>
<body>
<h2>What Can JavaScript Do?</h2>
<p id="demo">JavaScript can change HTML content.</p>
<button type="button" onclick='document.getElementById("demo").innerHTML = "Hello
JavaScript!'">Click Me!</button>
</body>
</html>
```

2. JavaScript Can Change HTML Attributes

JavaScript can HTML attributes of an HTML image by changing the src (source) attribute of an tag.

3. JavaScript Can Change HTML Styles (CSS)

Changing the style of an HTML element, is a variant of changing an HTML attribute.

EXAMPLE:

```
document.getElementById("demo").style.fontSize = "35px";
or document.getElementById('demo').style.fontSize =
'35px';
```

4. JavaScript Can Hide HTML Elements

Hiding HTML elements can be done by changing the display style.

EXAMPLE:

```
document.getElementById("demo").style.display = "none";
or document.getElementById('demo').style.display = 'none';
```

5. JavaScript Can Show HTML Elements

Showing hidden HTML elements can also be done by changing the display style.

EXAMPLE:

```
document.getElementById("demo").style.display = "block";
document.getElementById('demo').style.display = 'block';
```

4.3 NODE.JS

Node.js is a server-side platform built on Google Chrome's JavaScript Engine (V8 Engine). Node.js was developed by Ryan Dahl in 2009 and its latest version is v0.10.36. The definition of Node.js as supplied by its official documentation is as follows –

Node.js is a platform built on Chrome's JavaScript runtime for easily building fast and scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.

Node.js is an open source, cross-platform runtime environment for developing server-side and networking applications. Node.js applications are written in JavaScript, and can be run within the Node.js runtime on OS X, Microsoft Windows, and Linux.

Node.js also provides a rich library of various JavaScript modules which simplifies the development of web applications using Node.js to a great extent.

Features of Node.js

Following are some of the important features that make Node.js the first choice of software architects.

- Asynchronous and Event Driven – All APIs of Node.js library are asynchronous, that is, non-blocking. It essentially means a Node.js based server never waits for an API to return data. The server moves to the next API after calling it and a notification mechanism of Events of Node.js helps the server to get a response from the previous API call.
- Very Fast – Being built on Google Chrome's V8 JavaScript Engine, Node.js library is very fast in code execution.
- Single Threaded but Highly Scalable – Node.js uses a single threaded model with event looping. Event mechanism helps the server to respond in a non-blocking way and makes the server highly scalable as opposed to traditional servers which create limited threads to handle requests. Node.js uses a single threaded program and the same program can

provide service to a much larger number of requests than traditional servers like Apache HTTP Server.

- No Buffering – Node.js applications never buffer any data. These applications simply output the data in chunks.
- License – Node.js is released under the MIT license.

Who Uses Node.js?

Following is the link on github wiki containing an exhaustive list of projects, application and companies which are using Node.js. This list includes eBay, General Electric, GoDaddy, Microsoft, PayPal, Uber, Wikipins, Yahoo!, and Yammer to name a few.

Where to Use Node.js?

Following are the areas where Node.js is proving itself as a perfect technology partner.

- I/O bound Applications
- Data Streaming Applications
- Data Intensive Real-time Applications (DIRT)
- JSON APIs based Applications
- Single Page Applications

Where Not to Use Node.js?

It is not advisable to use Node.js for CPU intensive applications.

4.4 SOCKET.IO

What Socket.IO is

Socket.IO is a library that enables real-time, bidirectional and event-based communication between the browser and the server. It consists of:

1. a Node.js server: [Source](#) | [API](#)
2. a Javascript client library for the browser (which can be also run from Node.js): [Source](#) | [API](#)

Its main features are:

- Reliability

Connections are established even in the presence of:

1. proxies and load balancers.
2. personal firewall and antivirus software.

For this purpose, it relies on Engine.IO, which first establishes a long-polling connection, then tries to upgrade to better transports that are “tested” on the side, like WebSocket. Please see the Goals section for more information.

- Auto-reconnection support

Unless instructed otherwise a disconnected client will try to reconnect forever, until the server is available again. Please see the available reconnection options [here](#).

- Disconnection detection

A heartbeat mechanism is implemented at the Engine.IO level, allowing both the server and the client to know when the other one is not responding anymore.

That functionality is achieved with timers set on both the server and the client, with timeout values (the `pingInterval` and `pingTimeout` parameters) shared during the connection handshake. Those timers require any subsequent client calls to be directed to the same server, hence the sticky-session requirement when using multiples nodes.

- Binary support

Any serializable data structures can be emitted, including:

ArrayBuffer and Blob in the browser

ArrayBuffer and Buffer in Node.js

- Multiplexing support

In order to create separation of concerns within your application (for example per module, or based on permissions), Socket.IO allows you to create several Namespaces, which will act as separate communication channels but will share the same underlying connection.

- Room support

Within each Namespace, you can define arbitrary channels, called Rooms, that sockets can join and leave. You can then broadcast to any given room, reaching every socket that has joined it.

What Socket.IO is not

Socket.IO is NOT a WebSocket implementation. Although Socket.IO indeed uses WebSocket as a transport when possible, it adds some metadata to each packet: the packet type, the namespace and the ack id when a message acknowledgement is needed. That is why a WebSocket client will not be able to successfully connect to a Socket.IO server, and a Socket.IO client will not be able to connect to a WebSocket server either.

CHAPTER 4 – CONCLUSION AND FUTURE SCOPE

Conclusion:

The main objective of the project is to develop a Secure Chat Application. I had taken a wide range of literature review in order to achieve all the tasks, where I came to know about some of the products that are existing in the market. I made a detailed research in that path to cover the loop holes that existing systems are facing and to eradicate them in our application. In the process of research I came to know about the latest technologies and different algorithms.

I analysed various encryption algorithms (DES, AES, IDEA...), Integrity algorithms (MD5, SHA), key-exchange algorithms, authentication and I had implemented those functionalities in my application. I had done a detailed research on Certificate Authority and key tool for the generation of certificates.

As a result, the product has been successfully developed in terms of extendibility, portability, and maintainability and tested in order to meet all requirements that are

Authentication

Integrity

Confidentiality

Which are specified as the are basic concepts for the secure communication over a network.

Future Scope:

With the knowledge I have gained by developing this application, I am confident that in the future I can make the application more effectively by adding this services.

- Extending this application by providing Authorisation service.
- Creating Database and maintaining users.
- Increasing the effectiveness of the application by providing Voice Chat.
- Introducing caching to save the chats.
- Media (photos, video) sharing support.

=====