SUMMER TRAINING REPORT

On

FULL STACK WEB DEVELOPMENT IN NODEJS

Submitted to Guru Gobind Singh Indraprastha University, Delhi (India)
in partial fulfillment of the requirement for the award of the degree of

B. TECH

in

INFORMATION TECHNOLOGY

Submitted By

SHREESH KUMAR TRIPATHI

Roll No. 02315003117

DEPTT. OF INFORMATION TECHNOLOGY

MAHARAJA SURAJMAL INSTITUTE OF TECHNOLOGY,

NEW DELHI – 110058

SEPTEMBER 2019

ACKNOWLEDGEMENT

A project work owes its success from commencement to completion, to the people in love with reaserchers mentors at various stages. Let me in this page express my gratitude to all those who helped us in various stages of this project. First, I would like to express my sincere gratitude indebtedness to **Dr. Tripti Sharma** (HOD, Department of Information Technology, Maharaja Surajmal Institue of Technology, New Delhi) for allowing me to undergo the summer training of 30 days at **PepCoding Education Private, Limited.**

I am grateful to our guide **Mr. Jasbir Singh** , for the help provided in completion of the project, which was assigned to me. Without his friendly help and guidance it was difficult to develop this project.

I am also thankful to **Dr. Prabhjot Kaur** for his true help, inspiration and for helping me to preparation of the final report and presentation.

Last but not least, I pay my sincere thanks and gratitude to all the Staff Member of **PepCoding Education Private, Limited** for their support and for making our training valuable and fruitful.

Submitted By:

Shreesh Kumar Tripathi

(02315003117)

CERTIFICATE

PEPCODING EDUCATION (OPC) PRIVATE LTD.

3rd Floor, 15, Vaishali, Pitampura, New Delhi-110034

Website: www.pepcoding.com Phone: +911 4019 4461

CERTIFICATE OF COMPLETION

This is to certify that Shreesh Kumar Tripathi, B.Tech student at Maharaja Surajmal Institute of Technology, GGSIPU has successfully completed **Full Stack Web Development in NodeJS** from **1st June 2019 to 18th August 2019** at Pepcoding.

Sumeet Malik
(Director)

Pepcoding Education (OPC) Pvt. Ltd.
Pepcoding Education Private Limited

CANDIDATE'S DECLARATION

I, **Shreesh Kumar Tripathi**, Roll No. **02315003117**, B. Tech. (Semester 5th) of the Maharaja Surajmal Institute of Technology, New Delhi hereby declare that the Training Report entitled "Full Stack Web Development in NodeJS" is an original work and data provided in the study is authentic to the best of my knowledge. This report has not been submitted to any other Institute for the award of any other degree.

SHREESH KUMAR TRIPATHI

(Roll No. **02315003117)**

Place : Maharaja Surajmal Institute of Technology, Janakpuri, New Delhi

Date : 18th September 2019

ORGANIZATION INTRODUCTION

Pepcoding Education (OPC) Private Limited is a Private (One Person Company) incorporated on 08 November 2017. It is classified as Non-govt company and is registered at registrars of Companies, Delhi. **Sumeet Malik** is director of Pepcoding Education (opc) Private Limited.

Pepcoding is related to Educational field and it Provides Computer Science related subjects courses like Data Structures and Algorithms, Full Stack Development , Machine Learning, Interview Preparation etc.

Contact Details :

Email ID : sumeet.malik1188@gmail.com

Phone No. : 011 - 4019 4461

Address : PepCoding, 3rd Floor, 15 Vaishali, Pitampura Opposite Metro Pillar no. 347, Above Karur Vysya Bank, New Delhi, Delhi 110034

LIST OF FIGURES

TABLE OF CONTENTS

Chapter 1 : Introduction

1.1 About Project

This project report will introduce how we build Online Doubt Support portal using Node.JS (for backend) and HTML, CSS, JavaScript, jQuery and Bootstrap for frontend. Node.js is an open source server environment which uses JavaScript on the servers and runs. Our Project is platform independent because it works with browser. If browser allow the use of canvas(the main part of implementaion of project, on which the main part project is based) then it runs very well (browsers which support : chrome, firefox, Edge, safari). This portal consists of three major sections :

1. **White Board**
2. **Editor**
3. **Online Chat Box**

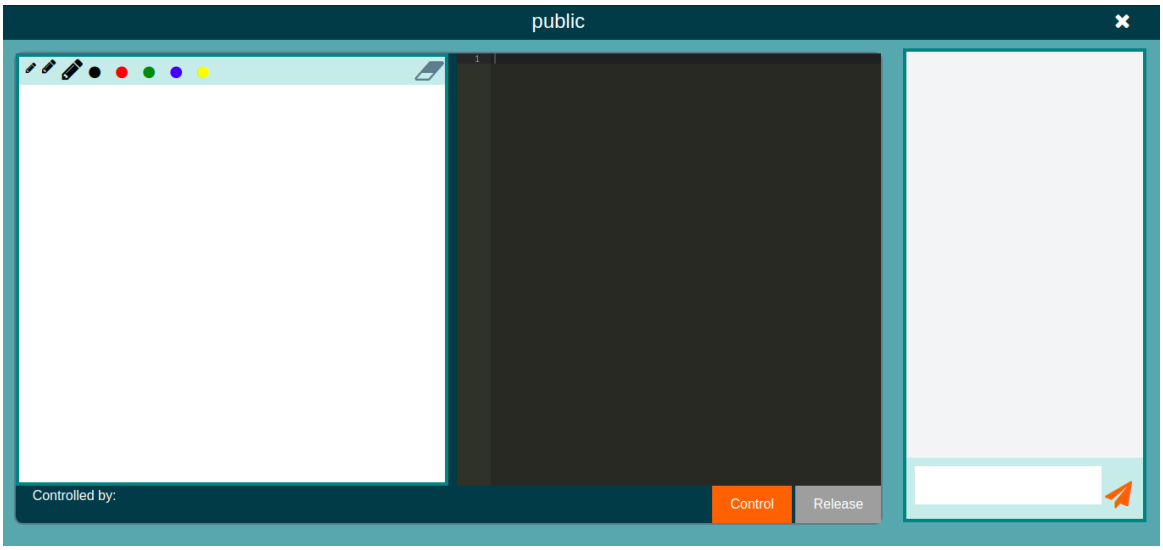

Fig - 1 First look of all three section of portal

1.1.1 White Board

With the help of white board teachers could explain algorithms and diagrams to students. Whiteboard contains five colors to draw and three different sizes of markers for clear drawing, it also contains one eraser button to clear the white board. This board is implemented using HTML canvas and drawing event on board is implemented by adding listener functions to JavaScript mouse events.

1.1.2 Editor

Students can share their codes to Teaching Assistant and TA's can take control of students screen, and modify it.

1.1.2 Chat Box

Students can ask doubts from Teaching assistant through chat. Chat box is implemented using Socket.IO library of Node.js which provides real time interaction between server and client using JavaScript websockets.

Chapter 2: Preliminary

2.1 Front-End

Front-end web development, also known as client-side development is the practice of producing HTML, CSS and JavaScript for a website or Web Application so that a user can see and interact with them directly. The challenge associated with front end development is that the tools and techniques used to create the front end of a website change constantly and so the developer needs to constantly be aware of how the field is developing.

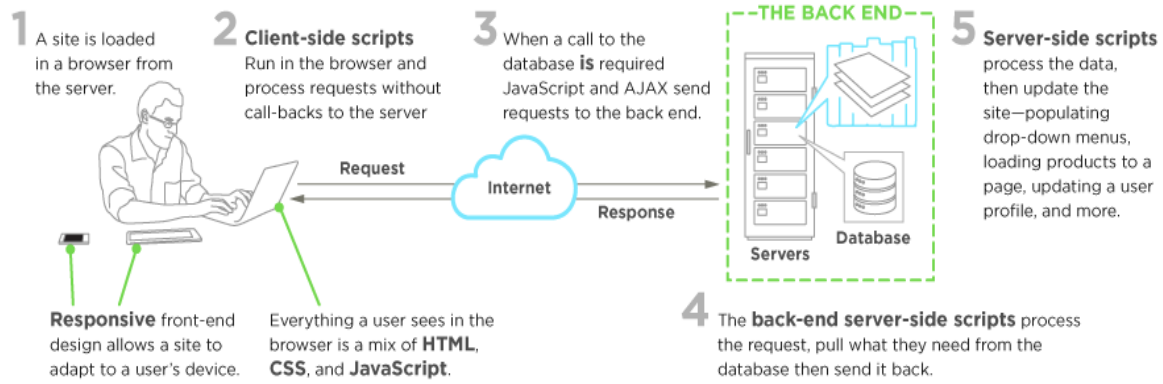

Fig - 2 Introduction of frontend and backend through diagram

The objective of designing a site is to ensure that when the user open up the site they see the information in a format that is easy to read and relevant. This is further complicated by the fact that users now use a large variety of devices with varying screen sizes and resolutions thus forcing the designer to take into consideration these aspects when designing the site. They need to ensure that their site comes up correctly in different browsers (cross-browser), different operating systems (cross-platform) and different devices (cross-device), which requires careful planning on the side of the developer.

There are huge amount of tools are available for frontend but front-end of our project we use following tools :

2.1.1 HTML

Hypertext Markup Language (HTML) is the standard markup language for documents designed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the documents.

Here is an example of basic HTML page :

```
<!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>Document</title>
</head>
<body>
    <h1>Hello, Welcome in HTML</h1>
    <p>This is a Paragraph Section</p>
</body>
</html>
```

Fig – 3 Introduction of HTML from sample Code

We use different tags, attributes and specially **canvas** which play an important role in our project. The HTML <canvas> element is used to draw graphics, on the fly, via JavaScript. The <canvas> element is only a container for graphics. We must use JavaScript to actually draw the graphics. Canvas consists of a drawable region defined in HTML code with height and width attributes. JavaScript code may access the area through a full set of drawing functions similar to those of other common 2D APIs, thus allowing for dynamically generated

graphics. Some anticipated uses of canvas include building graphs, animations, games, and image composition. Here is an example of canvas :

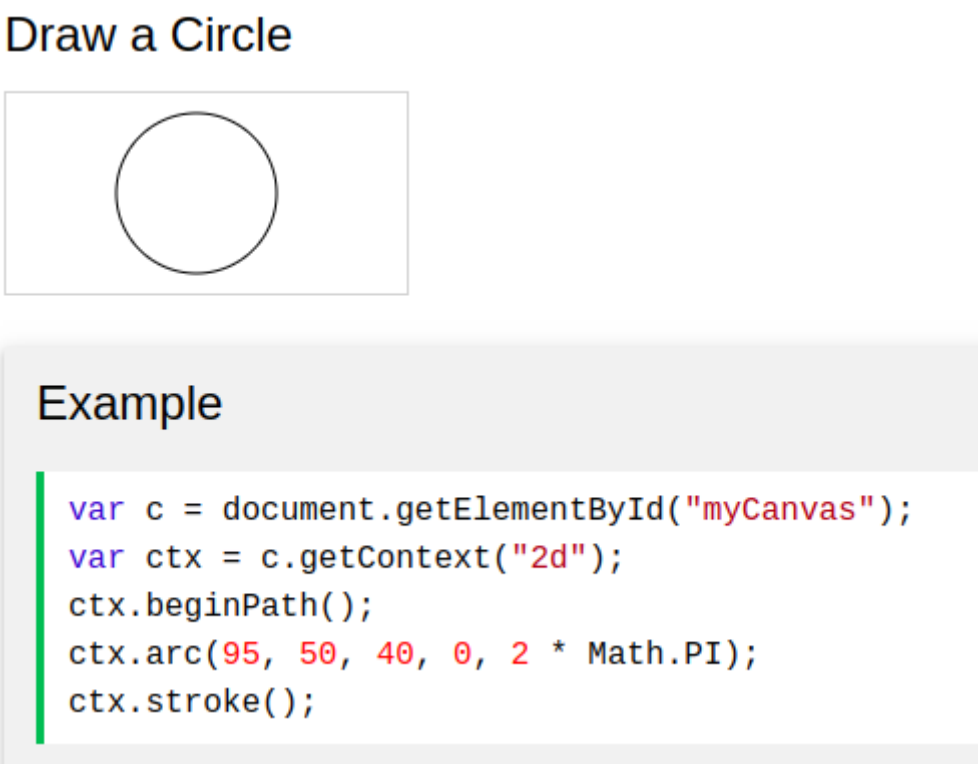

Fig - 4 Example of *CANVAS* (drawing of circle using JavaScript)

2.1.2 CSS

Cascading Style Sheets (CSS) is a stylesheet language used to describe the presentation of a document written in HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS describes how elements should be rendered on screen, on paper, in speech, or on other media.

By using CSS we can style all are HTML tags and elements. HTML only provides the layout of content on a web page , but CSS styles the elements so that the web page can look attractive and elements having same functionality can be put together to provide user a good website using experience.

Example : A simple example showing a webpage with and without CSS

1. How the home page of our website will look with CSS in *fig – 5:*

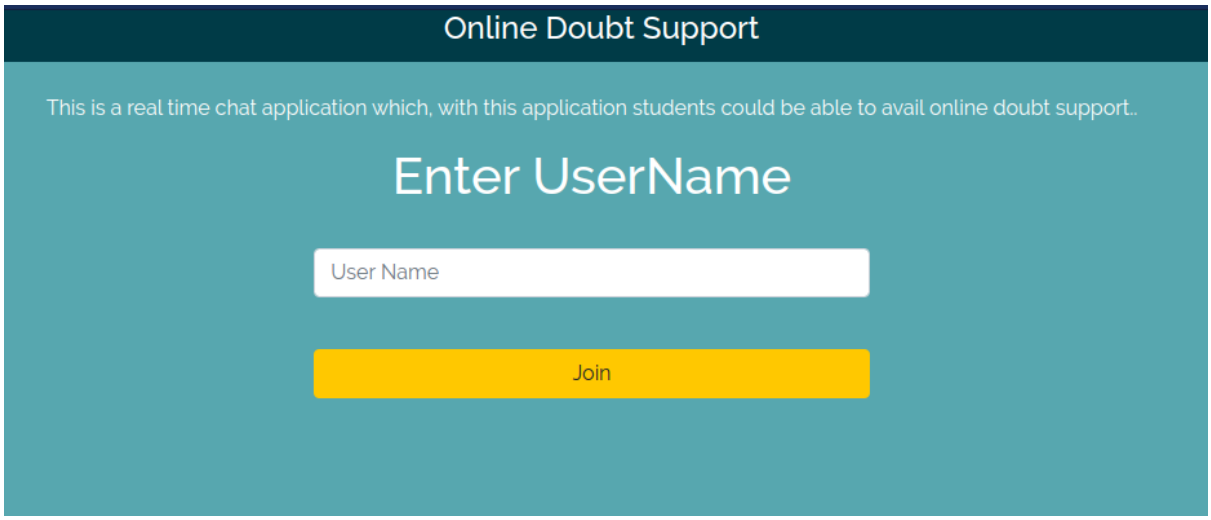

fig – 5 Homepage of our project with CSS

2. How the home page of our website will look without CSS in *fig 6:*

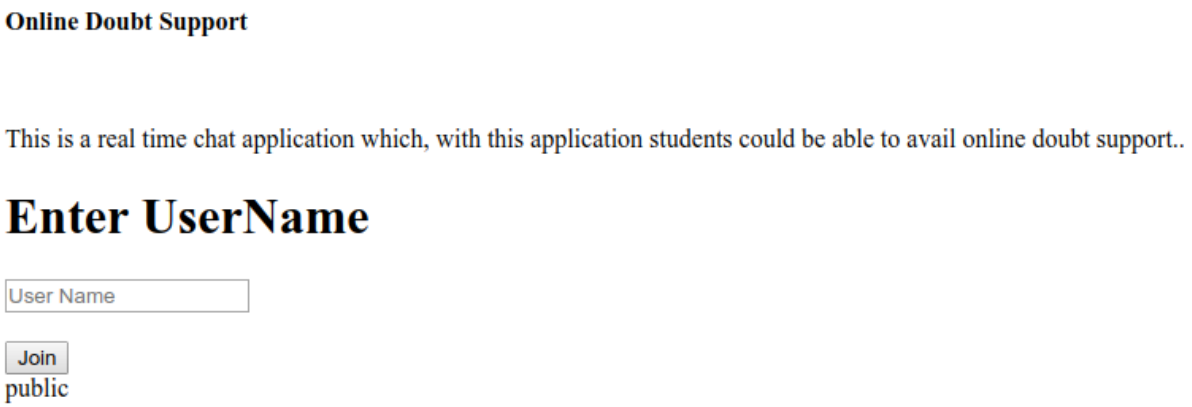

fig - 6 Homepage of our project without CSS

2.1.3 JAVASCRIPT

JavaScript (JS) is a lightweight, interpreted, or just-in-time compiled programming language with first-class functions. While it is most well-known as the scripting language for Web pages, many non-browser environments also use it, such as Node.js, Apache cloubDB and Adobe Acrobot. JavaScript is a prototype-based, multi-paradigm, dynamic language, supporting object-oriented programming, imperative programming, and declarative (eg. Functional programming) style.

We use JavaScript in our project to provide functionality for CANVAS, it plays an important rolw in Backend of projects and also in server part of project.

For eg. Some piece of code of our project in which we implemented canvas is here :

```
function initCanvas(containerId, otherUser) {
    canvas = $(containerId).find('.whiteboard')[0];
    colors = $(containerId).find('.color');
    pensize = $(containerId).find('.pensize');
    rubber = $(containerId).find('.rubber')[0];
    context = canvas.getContext('2d');

    whiteBoards[otherUser] = {
        context: context,
        isDisabled: true
    };


    canvas.addEventListener('mousedown', onMouseDown, false);
    canvas.addEventListener('mouseup', onMouseUp, false);
    canvas.addEventListener('mouseout', onMouseUp, false);
    canvas.addEventListener('mousemove', throttle(onMouseMove, 10), false);
```

fig - 7 Sample code of *CANVAS* in our project

For live interaction we use SOCKET.IO. **Socket.IO** is a JavaScript library for real-time web applications. It enables real time, bi-directional communication between web clients and servers. It has two parts: a client-side library that runs in the browser, and a server-side library for Node.js. Both components have a nearly identical API like Node.js, it is event-driven. SOCKET.IO primarily uses the webSocket, it provides many more features, including broadcasting to multiple sockets, storing data associated with each client, and asynchronous I/O.

2.1.4 BOOTSTRAP

Bootstrap is a web framework that focuses on simplifying the development of informative web pages (as opposed to web apps). The primary purpose of adding it to a web project is to apply Bootstrap's choices of color, size, font and layout to that project. As such, the primary factor is whether the developers in charge find those choices to their liking. Once added to a project, Bootstrap provides basic style definitions for all HTML elements. The end is a uniform appearance for prose, tables and form elements across web browsers. In addition, developers can take advantage of CSS classes defined in Bootstrap to further customize the appearance of their contents. For example, Bootstrap has provisioned for light- and dark-colored tables, page headings, more prominent pull quotes, and text with a highlight.

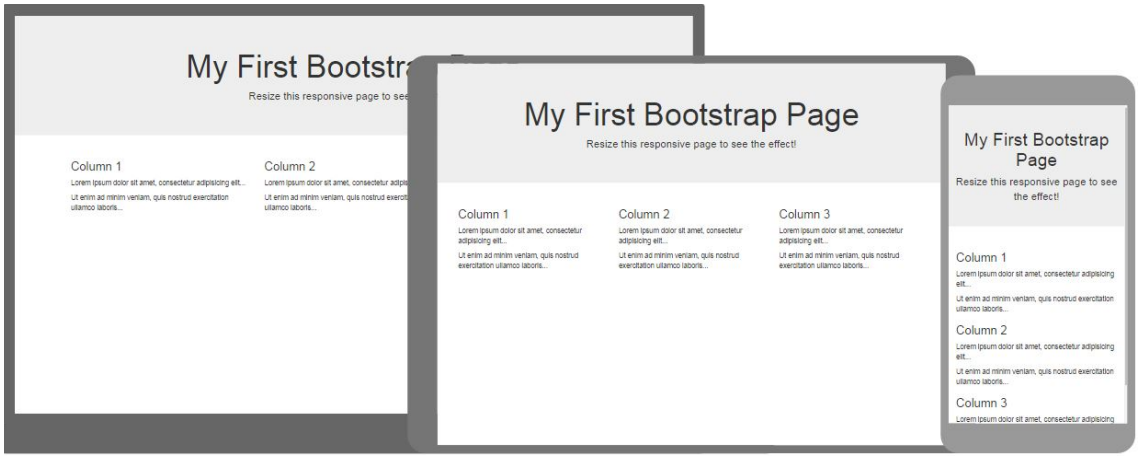

fig - 8 Bootstrap framework which make responsive content for all devices

Bootstrap also comes with several JavaScript components in the form of jQuery plugins. They provide additional user interface elements such as dialogue boxes, tool-tips and carousels. Each Bootstrap component consists of an HTML structure, CSS declarations, and in some cases accompanying JavaScript code. They also extend the functionality of some existing interface elements, including for example an auto-complete function for input fields.

2.1.5 JQUERY

jQuery is a lightweight, "write less, do more" JavaScript library. The purpose of jQuery is to make it much easier to use JavaScript on your website. jQuery takes a lot of common tasks that require many lines of JavaScript code to accomplish, and wraps them into methods that you can call with a single line of code. jQuery also simplifies a lot of the complicated things from JavaScript, like AJAX calls and DOM manipulation.

The jQuery library contains the following features:

- HTML / DOM Manipulation
- CSS Manipulation
- HTML event methods
- Effects and Animations
- AJAX
- Utilities

Overview

jQuery, at its core, is a Document Object Model (DOM) manipulation library. The DOM is a tree-structure representation of all the elements of a Web page. jQuery simplifies the syntax for finding, selecting, and manipulating these DOM elements. For example, jQuery can be used for finding an element in the document with a certain property (e.g. all elements with an h1 tag), changing one or more of its attributes (e.g. color, visibility), or making it respond to an event (e.g. a mouse click).

jQuery also provides a paradigm for event handling that goes beyond basic DOM element selection and manipulation. The event assignment and the event callback function definition are done in a single step in a single location in the code. jQuery also aims to incorporate other highly used JavaScript functionality (e.g. fade ins and fade outs when hiding elements, animations by manipulating CSS properties).

The principles of developing with jQuery are:

- Separation of JavaScript and HTML: The jQuery library provides simple syntax for adding event handlers to the DOM using JavaScript, rather than adding HTML events attributes to call JavaScript functions. Thus, it encourages developers to completely separate JavaScript code from HTML markup.
- Brevity and clarity: jQuery promotes brevity and clarity with features like "chainable" functions and shorthand function names.
- Elimination of cross-browser incompatibilities: The JavaScript engines of different browsers differ slightly so JavaScript code that works for one browser may not work for another. Like other JavaScript toolkits, jQuery handles all these cross-browser inconsistencies and provides a consistent interface that works across different browsers.
- Extensibility: New events, elements, and methods can be easily added and then reused as a plugin.

2.2 Back-End

What is Back-End Development?

Backend development (also stylized as back-end or back end development) is a skill that powers the web. Yet it does it modestly, without fanfare allowing people to browse their

favorite sites without even knowing about all the work put in by the backend developer or team. Back-end Development refers to the server-side development. It is the term used for the behind-the-scenes activities that happen when performing any action on a website. It can be logging in to your account or purchasing a watch from an online store.

Backend developer focuses on databases, scripting, and the architecture of websites. Code written by back-end developers helps to communicate the database information to the browsers.

Example :

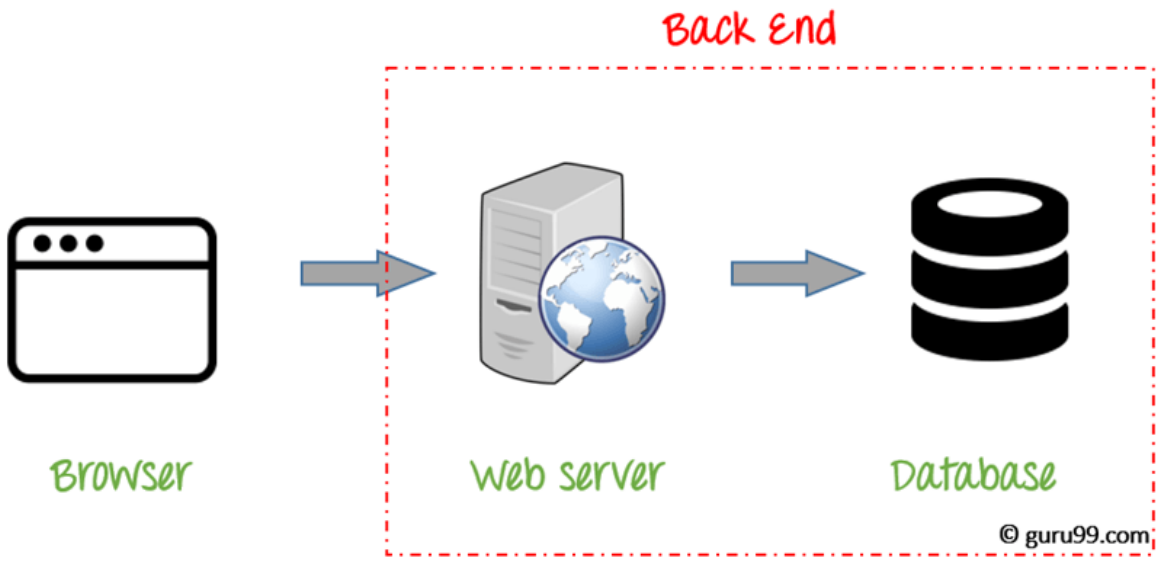

fig - 9 Diagrammatical explanation of connection of browser, web server and database

Following are the **Languages and Frameworks** used to implement Back-End Development :

- **PHP**

In the 23 years since its introduction, PHP has become the world's most popular server-side scripting language. PHP is pre-installed on most hosting sites, is known to be easy to use, and has tons of support. What's more, there are many very well established PHP frameworks on the market that make developing applications more intuitive and agile. Some of the most popular ones are Symfony, Laravel, Phalcon, and Yii framework.

- **Python**

Python has become an extremely popular language for building applications in recent years. With a reputation for being fast, easy to learn, and carrying wide support, Python has become the first choice for many new programmers. It's also the language of choice for data scientists and engineers. A couple of the most popular Python frameworks out there are Django and Pyramid.

- **Ruby on Rails**

Ruby on Rails is the popular open source web application framework that uses the Ruby programming language. Ever since Rails burst on the scene a decade ago it has continued to scale up as an elegant way to build dynamic websites quickly and efficiently. Rails has garnered a strong following, especially among tech startups. In fact, some of the best-known firms out there are using this framework to build their sites, including Airbnb, Bloomberg, and Groupon.

- **Node.JS**

Node.js is an open-source, cross-platform JavaScript runtime environment for developing a diverse variety of server tools and applications. One notable feature of Node.js is that it contains a built-in library to allow applications to act as a Web server without software such as Apache HTTP Server or IIS.

In this Project **Node.js** is used to implement Back-End Development.

2.2.1Node.JS

Node.js is an open source and cross-platform runtime environment for executing JavaScript code outside of a browser. You need to remember that NodeJS is not a framework and it's not a programming language. Most of the people are confused and understand it's a framework or a programming language. We often use Node.js for building back-end services like APIs like Web App or Mobile App. It's used in production by large companies such as Paypal, Uber, Netflix, Walmart and so on.

Features of NodeJS :

There are other programming languages also which we can use to build back-end services so what makes Node.js different I am going to explain.

1. It's easy to get started and can be used for prototyping and agile development
2. It provide fast and highly scalable services.
3. It uses JavaScript everywhere so it's easy for a JavaScript programmer to build back-end services using Node.js
4. Source code more cleaner and consistent.
5. Large ecosystem for open source library.
6. It has Asynchronous or Non blocking nature.

Advantages of NodeJS :

1. **Easy Scalability:** Developers prefer to use Node.js because it is easily scale the application in both horizontal and vertical direction. We can also add extra resources during the scalability of application.
2. **Real time web apps:** If you are building a web app you can also use PHP and it will take the same amount of time when you use Node.js, But if I am talking about building chat apps or gaming apps Node.js is much more preferable because of faster synchronization. Also, event loop avoid HTTP overload for Node.js development.
3. **Fast Suite:** NodeJs runs on the V8 engine developed by Google. Event loop in NodeJs handles all asynchronous operation so NodeJs acts like a fast suite and all the operations can be done quickly for example reading or writing in the database, network connection or file system
4. **Easy to learn and code:** NodeJs is easy to learn and code because it uses JavaScript. If you are a front-end developer and have a good grasp on JavaScript you can easily learn and build the application on NodeJS

5. **Advantage of Caching:** It provides the caching of single module. Whenever there is any request for the first module, it gets cached in the application memory so you don't need to re-execute the code.
6. **Data Streaming:** In NodeJs HTTP request and response are considered as two separate events. They are data stream so when you process a file at the time of loading it will reduce the overall time and will make it faster when the data is presented in the form of transmissions. It also allows you to stream audio and video files at lightning speed.
7. **Hosting:** PaaS (Platform as a Service) and Heroku are the hosting platform for NodeJS application deployment which is easy to use without facing any issue.
8. **Corporate Support:** Most of the well known companies like Walmart, Paypal, Mirosoft, yahoo are using NodeJS for building the applications. NodeJS uses JavaScript so most of the companies are combining front-end and backend Teams together into a single unit.

Application of NodeJS:

NodeJS should be preferred to build:

- Real Time Chats,
- Complex Single-Page applications,
- Real-time collaboration tools,
- Streaming apps
- JSON APIs based application.

2.2.2 Express.JS :

Express is the most popular *Node* web framework, and is the underlying library for a number of other popular Node web Frameworks. It provides mechanisms to:

- Integrate with "view" rendering engines in order to generate responses by inserting data into templates.

- Write handlers for requests with different HTTP verbs at different URL paths (routes).
- Set common web application settings like the port to use for connecting, and the location of templates that are used for rendering the response.
- Add additional request processing "middleware" at any point within the request handling pipeline.

We have used express.js in our project to create server, it provides more functionality as compared to HTTP. Express.js is very easy to use when it comes to provide routing in webpages as compared to http, it also provides the facility to render template html pages in the website. Express.js provides us a chance to do something in between the client and the server by allowing us to create middleware. Middlewares are a very big advantage of using express.js which is not provided by http request.

```
1   var express = require('express');
2   var http = require('http');
3   var socketIo = require('socket.io');
4
5   var app = express();
6   var httpServer = http.Server(app);
7
8   // for heroku
9   const port = process.env.PORT || 5000;
10
11  var ioServer = socketIo(httpServer);
12  var allSockets = {};
13
14  app.use(express.static(__dirname + '/public'));
15
```

fig - 10 Piece of code of *express* part from our project

While *Express* itself is fairly minimalist, developers have created compatible middleware packages to address almost any web development problem. There are libraries to work with cookies, sessions, user logins, URL parameters.

To more understands the functionality of express, let see a quick view of express in image form :

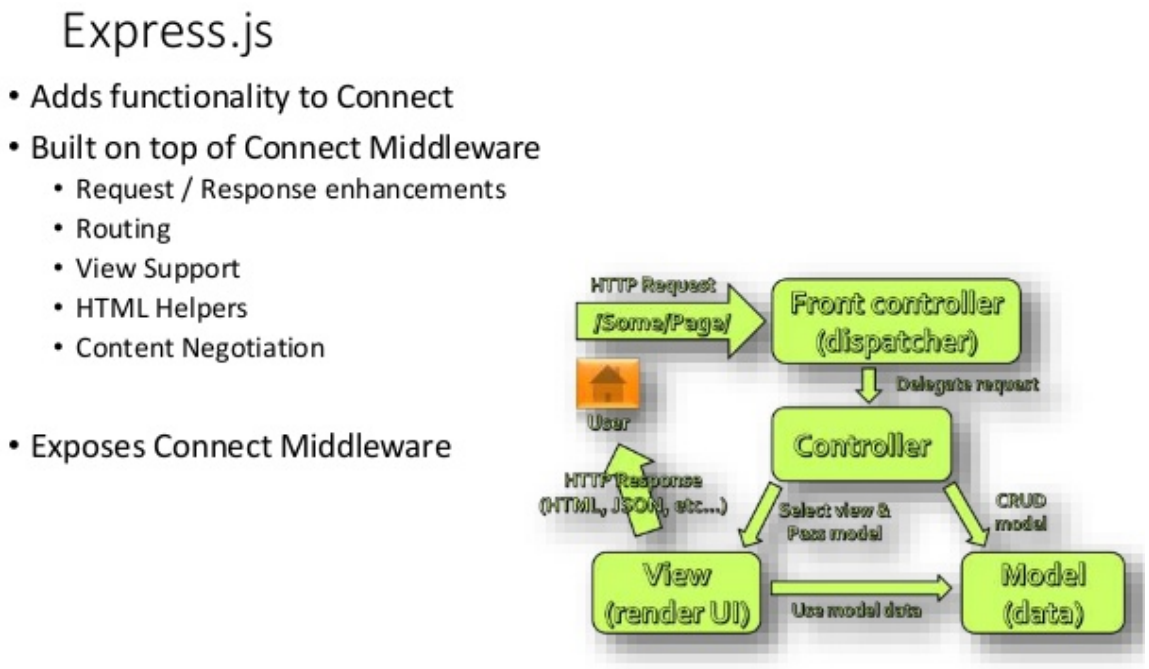

fig - 11 Application of *Express* in loop form

2.2.3 Socket.IO

Socket.IO enables real-time, bidirectional and event-based communication. It works on every platform, browser or device, focusing equally on reliability and speed.

Key Features of Socket.IO :

- **Real-time analytics :** Push data to clients that gets represented as real-time counters, charts or logs.
- **Instant messaging and chat :** Socket.IO's "Hello world" is a chat app in just a few lines of code.
- **Binary streaming :** Starting in 1.0, it's possible to send any blob back and forth: image, audio, video.
- **Document collaboration :** Allow users to concurrently edit a document and see each other's changes.

How Does Socket.IO works ?

Anatomy of a Socket.IO : A Socket.IO client first decides on a transport to utilize to connect. The state of the Socket.IO socket can be disconnected, disconnecting, connected and connecting. The transport connection can be closed, closing, open, and opening.

A simple HTTP handshake takes place at the beginning of a Socket.IO connection. The handshake, if successful, results in the client receiving:

- A session id that will be given for the transport to open connections.
- A number of seconds within which a heartbeat is expected (heartbeat timeout)
- A number of seconds after the transport connection is closed when the socket is considered disconnected if the transport connection is not reopened (close timeout).

At this point the socket is considered connected, and the transport is signaled to open the connection. If the transport connection is closed, both ends are to buffer messages and then frame them appropriately for them to be sent as a batch when the connection resumes.

If the connection is not resumed within the negotiated timeout the socket is considered disconnected. At this point the client might decide to reconnect the socket, which implies a new handshake.

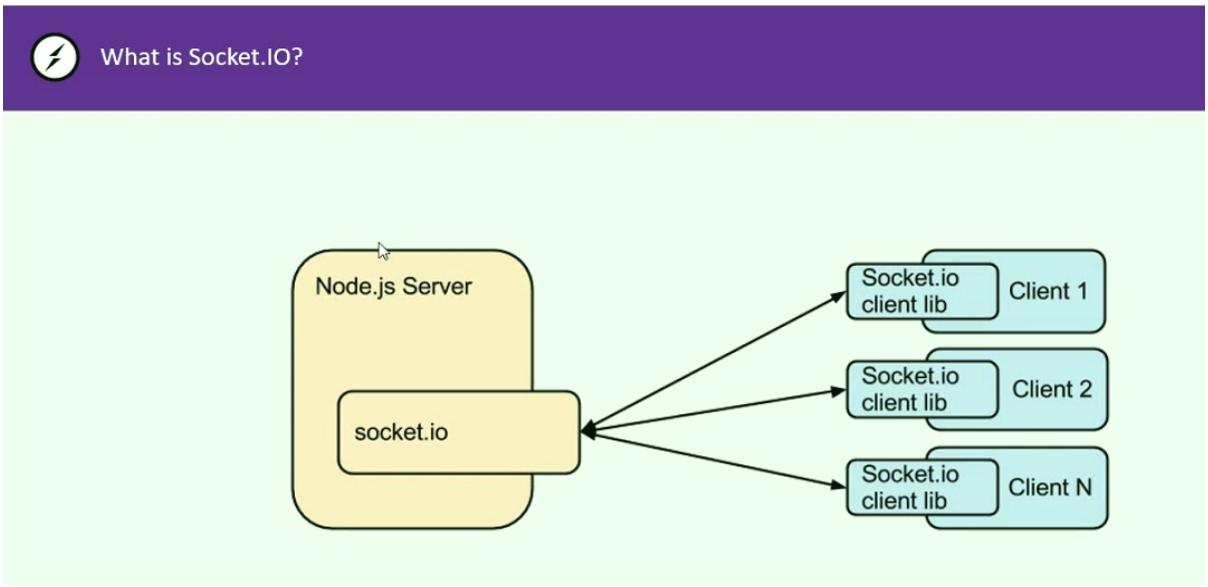

fig – 12 Diagrammatical explanation of working of *SOCKET.IO*

2.2.4 HTTP

To use the HTTP server and client one must require("http"). The HTTP interfaces in Node.js are designed to support many features of the protocol which have been traditionally difficult to use. In particular, large, possibly chunk-encoded, messages. The interface is careful to never buffer entire requests or responses - the user is able to stream data.

HTTP message headers are represented by an object like this:

```
var httpObj = {
    'content-length' : '123',
    'content-type' : 'text/plain',
    'connection' : 'keep-alive',
    'host' : 'mysite.com',
    'acdept' : '*/*'
};
```

fig - 13 An object of http request

In order to support the full spectrum of possible HTTP applications, Node.js's HTTP API is very low-level. It deals with stream handling and message parsing only. It parses a message into headers and body but it does not parse the actual headers or the body.

In our project we use express, https, socket.IO and httpServer to create server, there is a snapshot of a piece of our code in which we use HTTP:

```
1  var express = require('express');
2  var http = require('http');
3  var socketIo = require('socket.io');
4
5  var app = express();
6  var httpServer = http.Server(app);
7
8  // for heroku
9  const port = process.env.PORT || 5000;
10
11 var ioServer = socketIo(httpServer);
12 var allSockets = {};
13
14 app.use(express.static(__dirname + '/public'));
15
16 function httpServerConnected(){
17     console.log('Http Server started at port ', port);
18 }
```

fig - 14 piece of Code of http request in project

The HTTP module can create an HTTP server that listens to server ports and gives a response back to the client. Use the createServer() method to create an HTTP server. The function passed into the http.createServer() methods, will be executed when someone tries to access the computer on given PORT.

Just look at the self explanatory diagram, in which diagram shows how http work with express and how helpful for request and response.

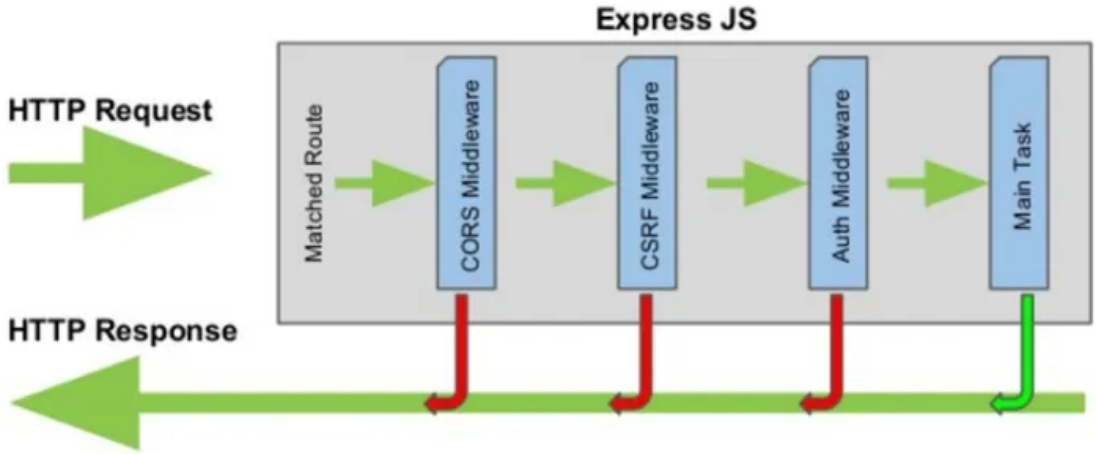

fig - 15 http request and response render with express.JS

This is all about preliminaries of our project, These are some basic knowldege which are required to make this project. There is also some efforts of thinking of design, concatinating of all the knowledge and merging them in the form of projects. Experience of seniors and there helps also play more important role.

Chapter 3 : How it works?

This is an interactive Online Doubt support website, in which a TA (Teaching Assistant) or any other students even teachers can also take doubts of a Students. It also hold enough functionality even for conducting an online classes just on the basis of white board, chat box and a code editor. This website is designed for the purpose of Coding Doubts but it is not necessary that it is not useful for others, it is useful.

There are some following features in our websites :

3.1 Login – Free:

This is a login free web-application and Home Page have very classic and unique design, It is easy to use. There is a box for Username and you can join and connect with anyone from whom you want to interact.

There is a first look of Home Page of our website :

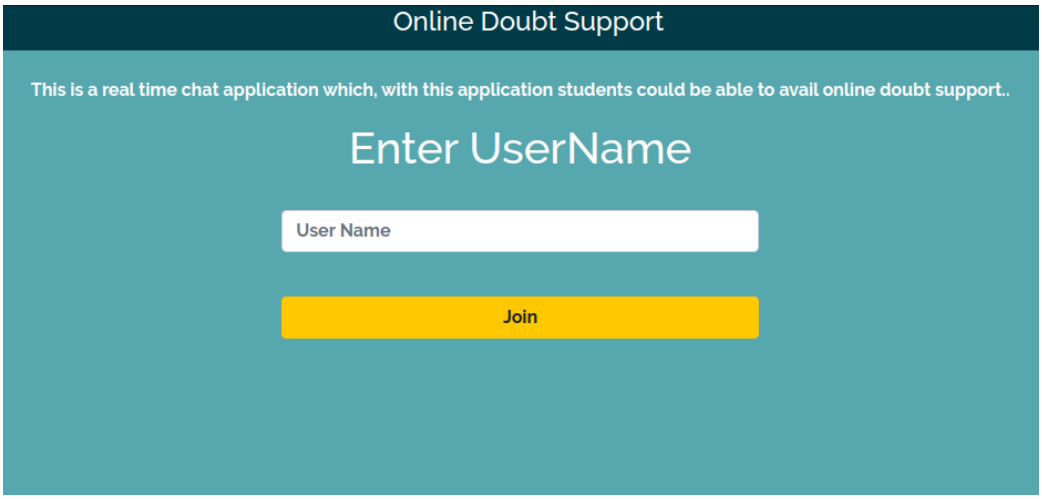

fig – 16 HomePage of our website(Online Doubt Support)

Home Page is very easy to use, as we can see from the *fig-16* , home page is very simple User Name box is use to create a unique identity of a particular user, User enter their name and hit the click on *Join* and you enter on the profile OR list page of our project. About that page we will see it ahead in this chapter.

3.2 Profile Window:

Once you logged in(this is not actual login, we can say once you enter your name your are present on current server), you can see a pretty simple interface. There are two sections on the website:

3.2.1 Public - section :

In the public section you can interact with multiple users one at a time, to avoid the mess we use a feature of control so that one user can hold the command for editor and whiteboard one at a time. Chat box is free for all. The Control is not fixed, once when holder control is done from his/her side then they can release hold, now any one from other user can hold the control command.

There is review of its/ profile page OR list section of USERS:

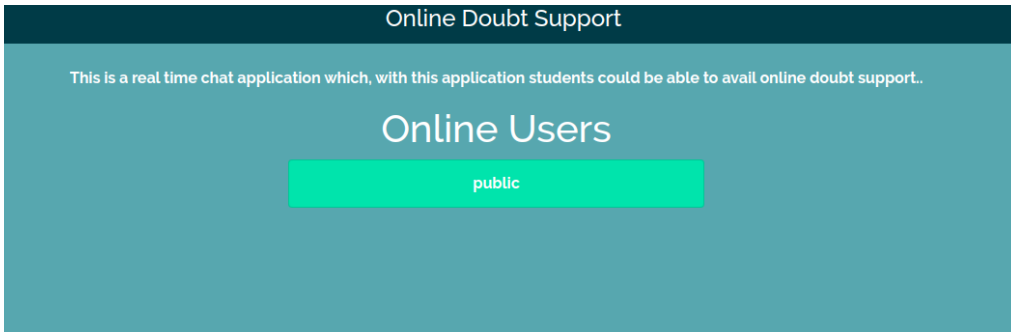

fig – 17 User's profile / list section in which users are listed

When multiple users are connected simultaneously then we can view all of them from there that's we called it User's section OR user list. It give current runnning time list that means at current time how many users are connected from the portal. Let's take a quick review :

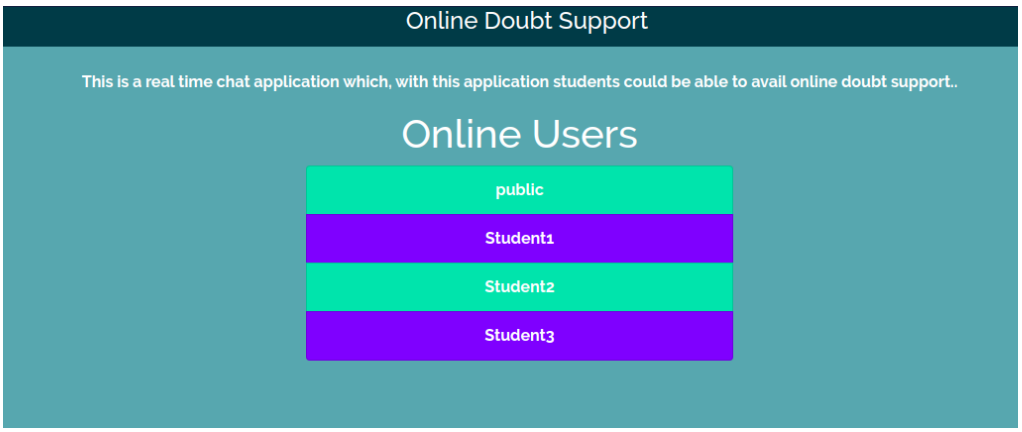

fig – 18 List of users which are connected

3.2.2 Private – section :

In the private section one user can connect with another one, Third person is not allowed in this section, Basically it is one to one interaction, the process of controls and release is same as public section (i.e. a user can hold the command of control one at a time, at that time he/she can use board or editor for doubt clearance or for asking something and when he/she release the control than other user can use it, But chat box is free for both). Let's take a quick review of private section in which users are interacted with each other .

3.3 Features of Website :

There are three most important features which are very helpful for doubt clearing purpose.

- White Board
- Chat Box
- Editor

Lets's take review of all features one by one and see how they are useful for user:

3.3.1 WhiteBoard :

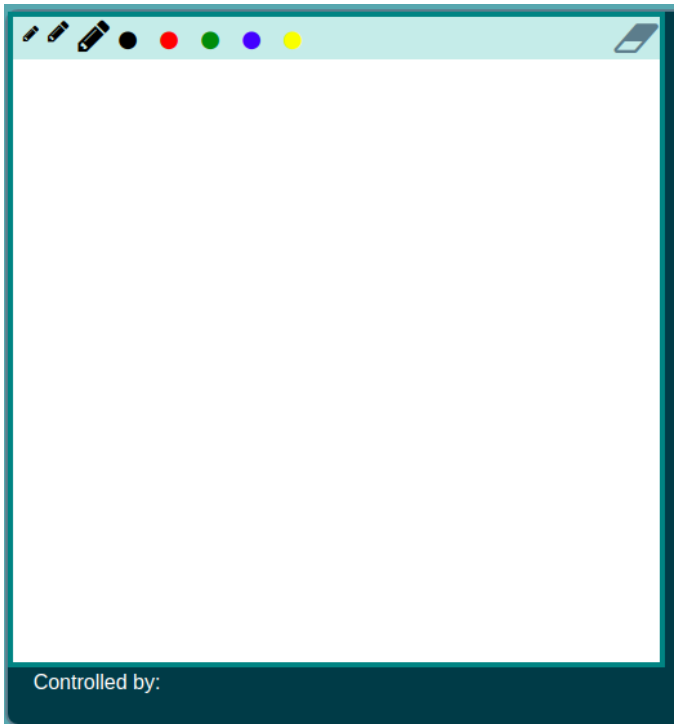

fig – 19 White Board on web view

In *fig - 19* we can see the view of web white board. White board is an important feature because if anyone wants to understand anything then they have uses copy pen, this idea is

basically taken from schools or college. We need a stylus to use the board section mouse is also work but for frequent and clear expression we need a stylus.

White board have so many options for size of line and colour of line, There are three different sizes of pen and five different colors are available. One option for erase complete board is also available which is helpful to clear the board section.

3.3.2 Chat Box :

In today's world, we all are aware from whatsApp, basically our chat box is very similar to it.It Provides an option for user to interact with another user. It is very easy to use and no control is needed for this option. Let's look at the chat Box option in website :

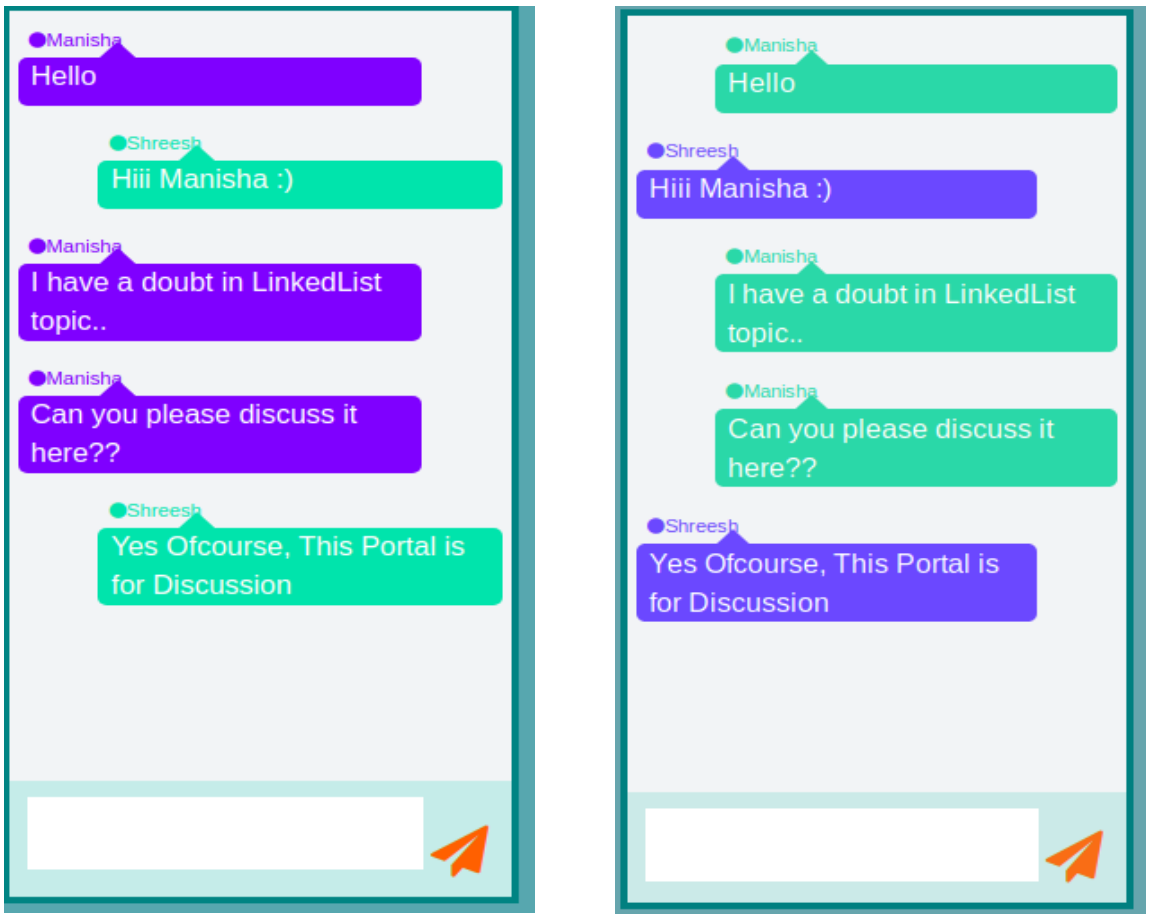

fig – 20 Interaction of two users regarding doubt on our portal

There is a small chat between two students in which one is asking for clearing of doubts and other is ready for it. From these above ways students can interact with each other.

3.3.3 Editor :

As i have mentioned that this website is designed for the purpose of coding doubts. So we all know Editor plays an important role in coding, that's why we add the option for editor, in current time editor can be used just for writes the code, It can't compile the code but in future updates we will implement that features very soon. Let's look at Editor section :

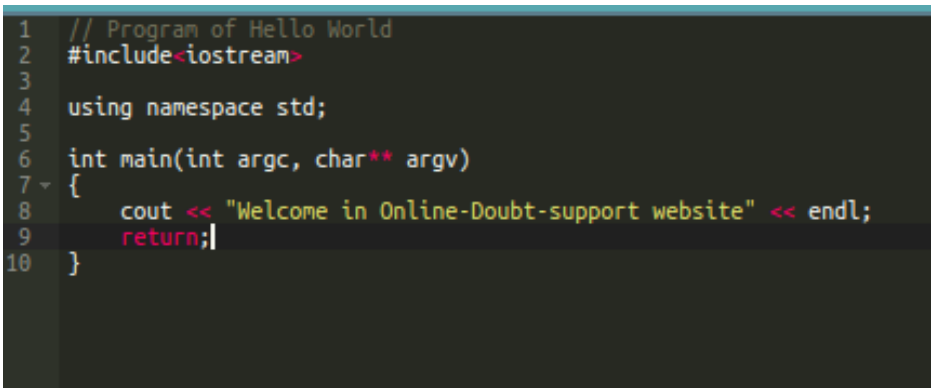

fig – 21 Editor section of Portal

CONCLUSION

Node.js Libraries and frameworks gives us a simple way to create online doubt support portal. It provides powerful functionalities and concise syntax to help programmers deal with the database, the web page and the inner logic. The experience of developing the group component in the system also helped me to learn how to handle backend part of a website using Node.js. With Javascript and Node.js Libraries and frameworks like Express.js, socket.IO and http, we have successfully accomplished the requirements of the system. Once this system passes the testing phase, it can be used to serve people of all age groups. It will make the experience with the online-doubt-support portal much easier. In short, this system will bring great user experience to both instructors and students. The only limitation for this system is that although the developers have been testing it with various use cases, it may still encounter problems during real time use. However, even if that happens, the flexibility of using Node.js would provide a simple way to fix the problem, as well as add new features into the system. With more and more evolving features the project continues to improve and add many more features to it.

BIBLIOGRAPHY

- Node.JS official website - https://nodejs.org/dist/latest-v10.x/docs/api/
- Freecodecamp website - https://learn.freecodecamp.org/
- Google - https://www.google.com/
- StackOverflow - https://stackoverflow.com/
- W3Schools website - https://stackoverflow.com/
- Node.JS tutorials - https://www.tutorialspoint.com/nodejs/index.htm
- Socket.io official website - https://socket.io/docs/
- Wikipedia (some topics) - https://en.wikipedia.org/wiki/Canvas
- Youtube - https://www.youtube.com/