

QUIZATO

A PROJECT REPORT

Submitted By

RAVINDRA YADAV

University Roll No 1900290140027

AYUSHI SRIVASTAVA

University Roll No 1900290140011

**Submitted in partial fulfilment of
the Requirements for the Degree
of**

MASTER OF COMPUTER APPLICATIONS

Under the Supervision of

DR. VIPIN KUMAR

ASSOCIATE PROFESSOR



Submitted to

**DEPARTMENT OF COMPUTER
APPLICATIONS**

**KIET Group of Institutions, Ghaziabad Uttar
Pradesh-201206**

(AUGUST 2021)

DECLARATION

I hereby declare that the work presented in this report entitled “**QUIZATO**”, was carried out by me. I have not submitted the matter embodied in this report for the award of any other degree or diploma of any other University or Institute.

I have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programs, experiments, results, that are not my original contribution. I have used quotation marks to identify verbatim sentences and given credit to the original authors/sources.

I affirm that no portion of my work is plagiarized, and the experiments and results reported in the report are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, I shall be fully responsible and answerable.

Name : Ravindra Yadav

Name : Ayushi Srivastav

Roll No : 1900290140027

Roll No :1900290140011

Branch : MCA

Branch : MCA

(Candidate Signature)

(Candidate Signature)

CERTIFICATE

Certified that **Ravindra Yadav (enrollment no. 190029014005170), Ayushi Srivastava (Enrollment no. 190029014005154)** have carried out the project work having “**Quizato**” for Master of Computer Applications from Dr. A.P.J. Abdul Kalam Technical University (AKTU) (formerly UPTU), Technical University, Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself/herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Date:

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:

Dr. Vipin Kumar

Associate Professor

Department of Computer Applications

KIET Group of Institutions, Ghaziabad

Signature of Internal Examiner

Signature of External Examiner

Dr. Ajay Shrivastava
Head, Department of Computer Applications
KIET Group of Institutions, Ghaziabad

ABSTRACT

It is good source of inter activity among students and also between the teacher and students. It is done in order to improve student's learning motivation.

This is a multiple-choice questions web application. The name of this web application is Quizato.

In this web application the uniqueness is that, the user can set their time according to their requirement.

Because sometime some question takes more time to solve and some question takes less time and timer is also important for increase your question solving speed.

User will be interested because difficulty level of questions will be increase day by day.

Quizzes in e-learning can be used to motivate learners, help them remember what they learned, and assess what they learned.

ACKNOWLEDGEMENTS

Success in life is never attained single handily. My deepest gratitude goes to my thesis supervisor, Dr. Vipin Kumar for his guidance, help and encouragement throughout my research work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to Dr. Ajay Kumar Shrivastava, Professor and Head, Department of Computer Applications, for his insightful comments and administrative help at various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

Ravindra Yadav

Ayushi Srivastava

Table of Contents

S.No.	Title	Page
	DECLARATION	ii
	CERTIFICATE	iii
	ABSTRACT	iv
	ACKNOWLEDGEMENTS	v
	Table of Contents	vi
CHAPTER 1	INTRODUCTION	1
1.1	PROJECT DESCRIPTION	1
1.2	PROJECT SCOPE	1
1.3	PROPOSED SYSTEM	2
CHAPTER 2	FEASIBILITY STUDY	3
2.1	Operational Feasibility	3
2.2	TECHNICAL FEASIBILITY	3
2.3	ECONOMICAL FEASIBILITY	4
CHAPTER 3	ORGANISATION OF THE REPORT	5
CHAPTER 4	SOFTWARE/Hardware REQUIREMENTS SPECIFICATION	6
4.1	Hardware Requirements	6
4.2	Software Requirements	6
CHAPTER 5	DESIGN & PLANNING	7
5.1	Software Development Life Cycle Model	7
5.2	GENERAL OVERVIEW	8
5.3	ER Diagram	9

CHAPTER 6	IMPLEMENTATION DETAILS	10
6.1	FRONT END	10
6.2	VS Code:	20
6.3	Hardware	21
CHAPTER 7	TESTING	25
7.1	UNIT TESTING	25
7.2	INTEGRATION TESTING	26
7.3	SOFTWARE VERIFICATION AND VALIDATION	28
7.4	Black-Box Testing	30
7.5	White-Box Testing	31
7.6	SYSTEM TESTING	33
CHAPTER 8	RESULTS	34
CHAPTER 9	Advantage	66
CHAPTER 10	Conclusion	67
	BIBLIOGRAPHY	68

CHAPTER 1

INTRODUCTION

1.1 PROJECT DESCRIPTION

All these operations have a scope of improve learning. In the centre of all these processes lies a motivation. The vital question here is how to find a particular subject's quiz? The obvious response to this question is by using the particular subject's name. In this web quiz application, we are mainly focusing on technical subjects (like PHP, Java Script, HTML, DBMS, data structure etc.)

Almost all organizations now-a-days, are conducting their objective exams by online examination system, it saves students time in examinations. Organizations can also easily check the performance of the student that they give in an examination.

In the world of internet, all task has been done through internet, so we have decided why Quizzes have not conducted through internet. In this web application it will require less labour force for organize quizzes and it will be more accurate and less time consuming and at the same time. This web quiz application helps the user to increase his/her knowledge.

1.2 PROJECT SCOPE

This Web Application provides facility to conduct online quiz sessions. It saves time as it allows number of students to give the exam at a time and displays the results with correct answer of quizzes the test gets over, so no need to wait for the result.

This project has a wide scope as it is better than the manual quiz.

It, being an application based on web may be taken at any place or any time as the location is not considered. The presence of the proctor is not required while the candidate is appearing for the Quiz.

The 'MCQ Web Quiz Application' project developed to overcome the time-consuming problem of manual system. Correct answer and save the teacher time and carry the quiz in an effective manner. The users which are use this system don't need to high computing knowledge and also system will inform them while entering invalid data.

1.3 PROPOSED SYSTEM

Manual assessment is prone to errors and is not time efficient as discussed previously. So why not automate the whole assessment process? Why would a teacher spend his/her precious time physically correcting the answer of their students? So, our main objective is to create a method that would allow us to make our computers do the whole assessment work and award score to the answers accordingly.

CHAPTER 2

FEASIBILITY STUDY

A feasibility study is a high-level capsule version of the entire System analysis and Design Process. The study begins by classifying the problem definition. Feasibility is to determine if it's worth doing. Once an acceptance problem definition has been generated, the analyst develops a logical model of the system. A search for alternatives is analyzed carefully. There are 3 parts in feasibility study.

2.1 Operational Feasibility

Operational feasibility is the measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. The operational feasibility assessment focuses on the degree to which the proposed development projects fits in with the existing business environment and objectives with regard to development schedule, delivery date, corporate culture and existing business processes. To ensure success, desired operational outcomes must be imparted during design and development. These include such design-dependent parameters as reliability, maintainability, support-ability, usability, product-ability, sustainability, affordability and others. These parameters are required to be considered at the early stages of design if desired operational behaviour are to be realized. A system design and development require appropriate and timely application of engineering and management efforts to meet the previously mentioned parameters. A system may serve its intended purpose most effectively when its technical and operating characteristics are engineered into the design. Therefore, operational feasibility is a critical aspect of systems engineering that needs to be an integral part of the early design phases.

2.2 TECHNICAL FEASIBILITY

This involves questions such as whether the technology needed for the system exists, how difficult it will be to build, and whether the firm has enough experience using that technology. The assessment is based on outline design of system requirements in terms of input, processes, output, fields, programs and procedures. This can be qualified in terms of volume of data, trends, frequency of updating in order to give an introduction to the technical system. The application is the fact that it has been developed on windows XP platform and a high configuration of 1GB RAM on Intel Pentium Dual core processor. This is technically feasible. The technical feasibility assessment is focused on gaining an understanding of the present technical resources of the organization and their applicability to the expected needs of the proposed system. It is an evaluation of the hardware and software and how it meets the need of the proposed system.

2.3 ECONOMICAL FEASIBILITY

Establishing the cost-effectiveness of the proposed system i.e., if the benefits do not outweigh the costs, then it is not worth going ahead. In the fast-paced world today there is a great need of online social networking facilities. Thus, the benefits of this project in the current scenario make it economically feasible. The purpose of the economic feasibility assessment is to determine the positive economic benefits to the organization that the proposed system will provide. It includes quantification and identification of all the benefits expected. This assessment typically involves a cost/benefits analysis.

CHAPTER 3

ORGANISATION OF THE REPORT

3.1 INTRODUCTION

This section includes the overall view of the project i.e., the basic problem definition and the general overview of the problem which describes the problem in layman terms. It also specifies the software used and the proposed solution strategy.

3.2 SOFTWARE REQUIREMENTS SPECIFICATION

This section includes the Software and hardware requirements for the smooth running of the application.

3.3 DESIGN & PLANNING

This section consists of the Software Development Life Cycle model. It also contains technical diagrams like the Data Flow Diagram and the Entity Relationship diagram.

3.4 IMPLEMENTATION DETAILS

This section describes the different technologies used for the entire development process of the Front-end as well as the Back-end development of the application.

3.5 RESULTS AND DISCUSSION

This section has screenshots of all the implementation i.e., user interface and their description.

3.6 SUMMARY AND CONCLUSION

This section has screenshots of all the implementation i.e., user Interface and their description.

CHAPTER 4

SOFTWARE/Hardware REQUIREMENTS SPECIFICATION

4.1 Hardware Requirements

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. Memory – All software, when run, resides in the random access memory (RAM) of a computer. Memory requirements are defined after considering demands of the application, operating system, supporting software and files, and other running processes. Optimal performance of other unrelated software running on a multi-tasking computer system is also considered when defining this requirement.

Number	Description
1	PC with 250 GB or more Hard disk
2	PC with 2 GB RAM.
3	PC with Pentium 1 and above.

Table 4.1

4.2 Software Requirements

The software requirements are **description of features and functionalities of the target system**. Requirements convey the expectations of users from the software product. The requirements can be obvious or hidden, known or unknown, expected or unexpected from client's point of view .Every project needs

software. We should try to understand what sort of requirements may arise in the requirement elicitation phase and what kinds of requirements are expected from the software system.

Number	Description	Type
1	Operating System	Windows XP / Windows
2	Language	JavaScript HTML CSS
3	Database	MySQL JSON Server
4	IDE	Visual Code
5	Browser	Google Chrome

Table 4.2

CHAPTER 5

DESIGN & PLANNING

5.1 Software Development Life Cycle Model

5.1.1 WATERFALL MODEL

The waterfall model was selected as the SDLC model due to the following reasons:

- Requirements were very well documented, clear and fixed.
- Technology was adequately understood.
- Simple and easy to understand and use.
- There were no ambiguous requirements.
- Easy to manage due to the rigidity of the model. Each phase has specific deliverable and a review process.
- Clearly defined stages.
- Well understood milestones. Easy to arrange tasks

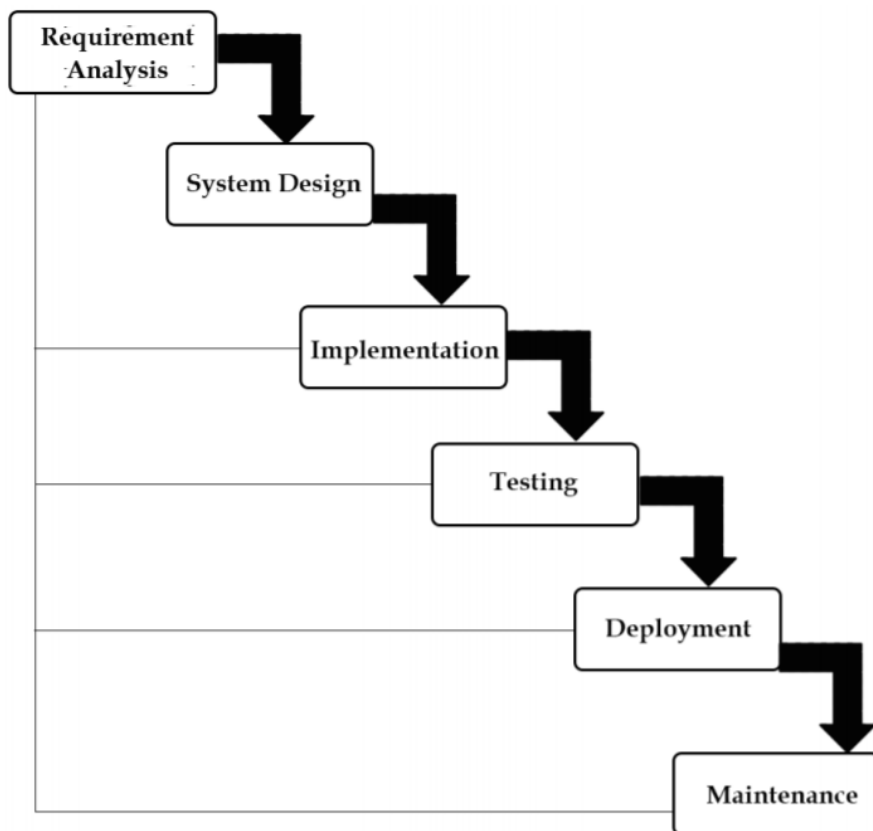


Fig 5.1.1

5.2 GENERAL OVERVIEW

This general overview means , the figure 5.3 simply shows that the user can be admin, teacher and student. There is an admin in my project who will manage the maintenance of application. Teacher will make the quiz for the test's purpose and take the test. Student can take the test as per his/her choice.

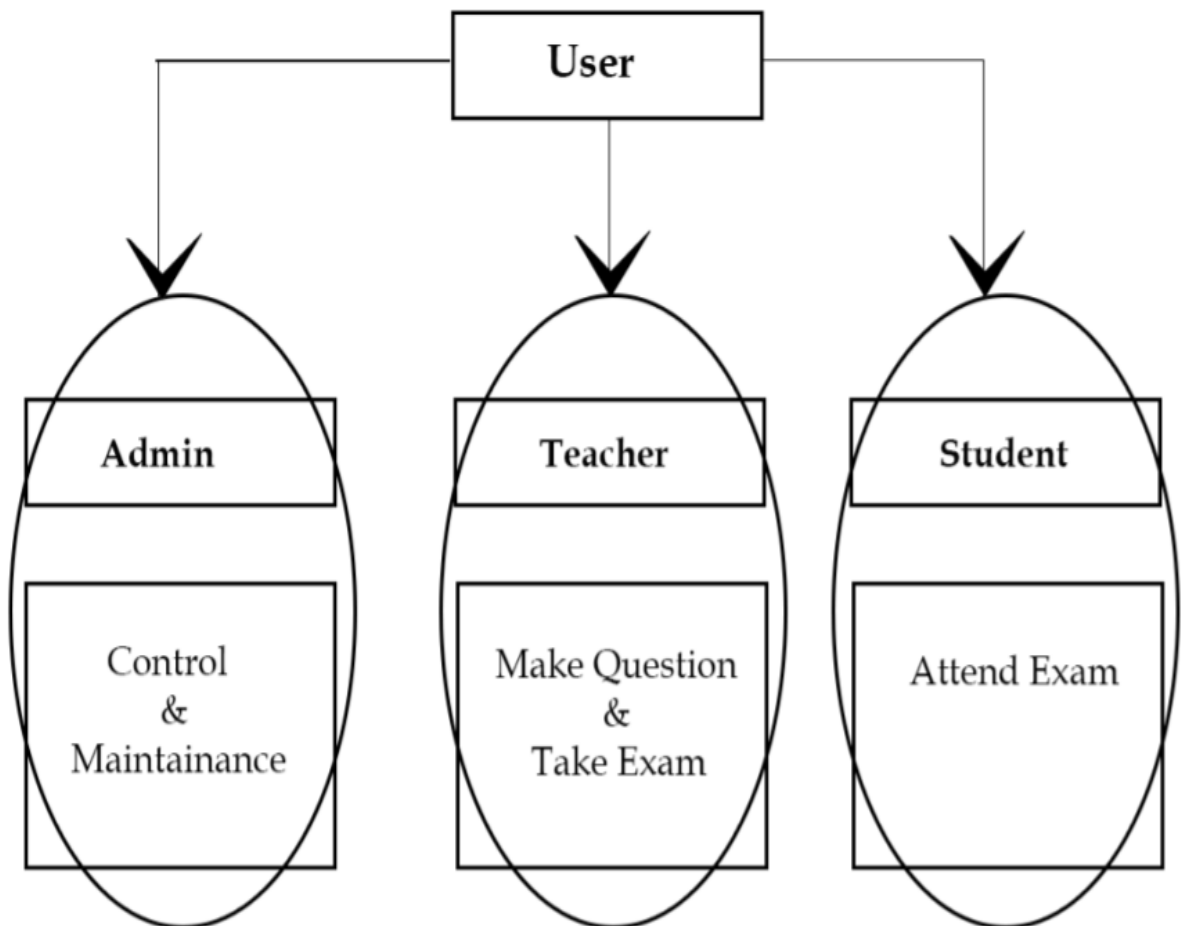


Fig 5.3

5.3 ER Diagram

An entity-relationship diagram, or ER diagram, is essential for modeling the data stored in a database. It is the basic design upon which a database is built. ER diagrams specify what data we will store: the entities and their attributes. They also show how entities relate to other entities. Another advantage of ERD is that they represent the data in a graphical manner. In this diagram we can see that particular's connection . That which things are important for students and teachers.

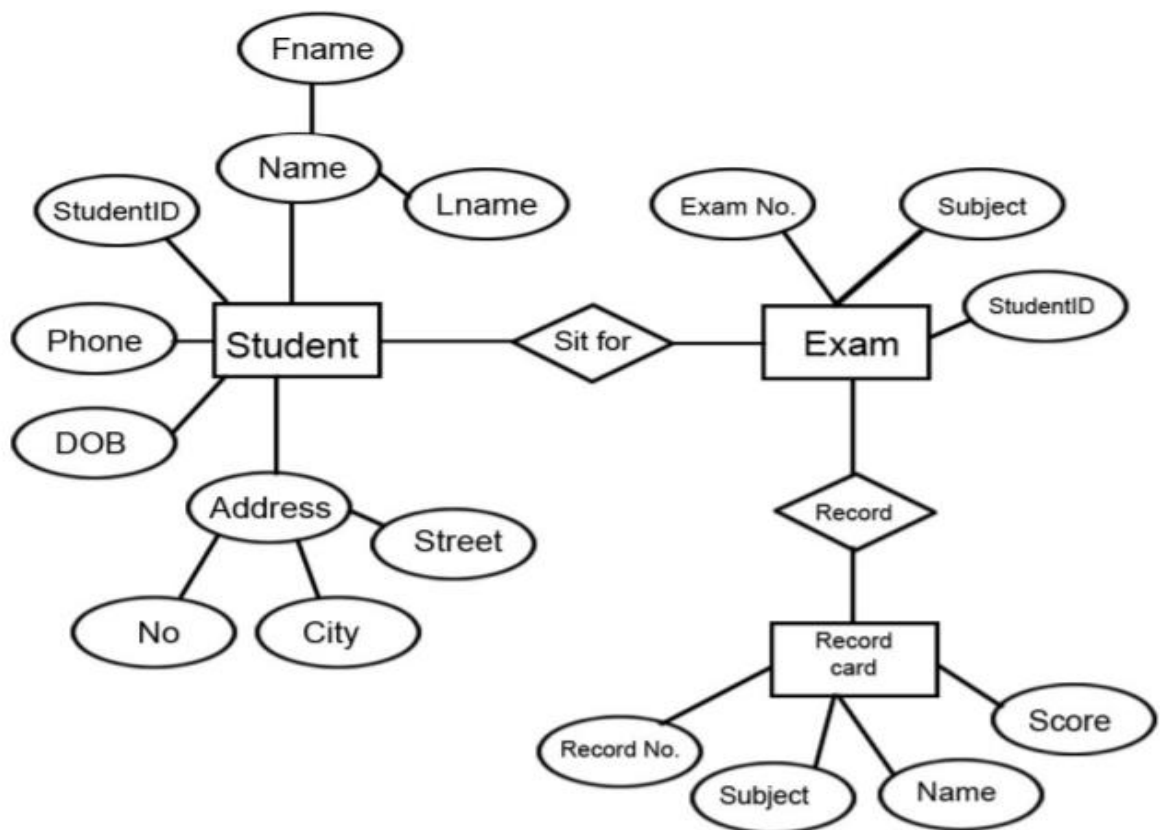


Fig 5.3

CHAPTER 6

IMPLEMENTATION DETAILS

In this Section we will do Analysis of Technologies to use for implementing the project

6.1 FRONT END

6.1.1 HTML

HTML stands for Hyper Text Markup Language. It is used to design web pages using a markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages. A markup language is used to define the text document within tag which defines the structure of web pages. This language is used to annotate (make notes for the computer) text so that a machine can understand it and manipulate text accordingly. Most markup languages (e.g. HTML) are human-readable. The language uses tags to define what manipulation has to be done on the text.

HTML is a markup language used by the browser to manipulate text, images, and other content, in order to display it in the required format. HTML was created by Tim Berners-Lee in 1991. The first-ever version of HTML was HTML 1.0, but the first standard version was HTML 2.0, published in 1999.

Elements and Tags:

HTML uses predefined tags and elements which tell the browser how to properly display the content. Remember to include closing tags. If omitted, the browser applies the effect of the opening tag until the end of the page.

HTML page structure:

The basic structure of an HTML page is laid out below. It contains the essential building-block elements (i.e. doctype declaration, HTML, head, title, and body elements) upon which all web pages are created.

<!DOCTYPE html>:

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

<html>:

This is called the HTML root element. All other elements are contained within it.

<head>:

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

<style>

<title>

<base>

<script>

<meta>

<link>

<body>:

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

NOTE: Basic/built-in text editors are Notepad (Windows) and Text-Edit (Macs). Basic text editors are entirely sufficient for when you’re just getting started. As you progress, there are many feature-rich text editors available which allow for greater function and flexibility.

6.1.2 JavaScript

JavaScript (js) is a light-weight object-oriented programming language which is used by several websites for scripting the web-pages. It is an interpreted, full-fledged programming language that enables dynamic interactivity on websites when applied to an HTML document. It was introduced in the year 1995 for adding programs to the web-pages in the Netscape Navigator browser.

Since then, it has been adopted by all other graphical web browsers. With JavaScript, users can build modern web applications to interact directly without reloading the page every time. The traditional website uses js to provide several forms of interactivity and simplicity.

Although, JavaScript has no connectivity with Java programming language. The name was suggested and provided in the times when Java was gaining popularity in the market. In addition to web browsers, databases such as Couch DB and MongoDB uses JavaScript as their scripting and query language.

Features of JavaScript:

1. All popular web browsers support JavaScript as they provide built-in execution environments.
2. JavaScript follows the syntax and structure of the C programming language. Thus, it is a structured programming language.
3. JavaScript is a weakly typed language, where certain types are implicitly cast (depending on the operation).
4. JavaScript is an object-oriented programming language that uses prototypes rather than using classes for inheritance.
5. It is a light-weighted and interpreted language.
6. It is a case-sensitive language.
7. JavaScript is supportable in several operating systems including, Windows, mac OS, etc.
8. It provides good control to the users over the web browsers.

History of JavaScript:

In 1993, Mosaic, the first popular web browser, came into existence. In the year 1994, Netscape was founded by Marc Andreessen. He realized that the web needed to become more dynamic. Thus, a 'glue language' was believed to be provided to HTML to make web designing easy for designers and part-time programmers. Consequently, in 1995, the company recruited Brendan Eich intending to implement and embed Scheme programming language to the browser. But, before Brendan could start, the company merged with Sun Micro-systems for adding Java into its Navigator so that it could compete with Microsoft over the web technologies and platforms. Now, two languages were there: Java and the scripting language. Further, Netscape decided to give a similar name to the scripting language as Java's. It led to 'JavaScript'. Finally, in

May 1995, Marc Andreessen coined the first code of JavaScript named 'Mocha'. Later, the marketing team replaced the name with 'Live Script'. But, due to trademark reasons and certain other reasons, in December 1995, the language was finally renamed to 'JavaScript'. From then, JavaScript came into existence.

Application of JavaScript:

JavaScript is used to create interactive websites. It is mainly used for:

- o Client-side validation,
- o Dynamic drop-down menus,
- o Displaying date and time,
- o Displaying pop-up windows and dialog boxes (like an alert dialog box, confirm dialog box and prompt dialog box),
- o Displaying clocks etc.

6.1.3 JQUERY

J-Query is an open-source JavaScript library that simplifies the interactions between an HTML/CSS document, or more precisely the Document Object Model (DOM), and JavaScript.

Elaborating the terms, J-Query simplifies HTML document traversing and manipulation, browser event handling, DOM animations, Ajax interactions, and cross-browser JavaScript development.

Note: The only library available today that meets the needs of both designer types and programmer types is J-Query.

J-Query is widely famous with its philosophy of “Write less, do more.” This philosophy can be further elaborated as three concepts:

- Finding some elements (via CSS selectors) and doing something with them (via J-Query methods) i.e., locate a set of elements in the DOM, and then do something with that set of elements.
- Chaining multiple J-Query methods on a set of elements
- Using the J-Query wrapper and implicit iteration

Using J-Query (JS) library on HTML page

There are several ways to start using J-Query on your web site.

1. Use the Google-hosted/ Microsoft-hosted content delivery network (CDN) to include a version of J-Query.
2. Download own version of J-Query from J-Query.com and host it on own server or local file-system.

Note: All J-Query methods are inside a document-ready event to prevent any J-Query code from running before the document is finished loading (is ready).

6.1.4 JSON

JavaScript Object Notation (JSON) is an open-standard file format that uses human-readable text to transmit data objects consisting of attribute–value pairs and array data types (or any other serializable value). It is a very common data format used for asynchronous browser–server communication, including as a replacement for XML in some AJAX-style systems. JSON is a language-independent data format. It was derived from JavaScript, but many modern programming languages include code to generate and parse JSON-format data. The official Internet media type for JSON is `application / json`. JSON filenames use the extension `.json`. Douglas Crockford originally specified the JSON format in the early 2000s. It was first standardized in 2013 in RFC 7158 and ECMA-404. The latest JSON format standard was published in 2017 as RFC 8259, and remains consistent with ECMA-404. That same year, JSON was also standardized as ISO/IEC 21778:2017. The ECMA and ISO standards describes only the allowed syntax, whereas the RFC covers some security and interoperability considerations. JSON grew out of a need for stateless, real-time server-to-browser communication protocol without using browser plugins such as Flash or Java applets, the dominant methods used in the early 2000s. JSON was originally intended to be a subset of the JavaScript scripting language (specifically, Standard ECMA-262 3rd Edition—December 1999) and is commonly used with JavaScript, but it is a language-independent data format. Code for parsing and generating JSON data is readily available in many programming languages. JSON's website lists JSON libraries by language.

6.1.5 CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. CSS is designed to enable the separation of presentation and content, including layout, colour, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content. CSS information can be provided from various sources. These sources can be the web browser, the user and the author. The information from the author can be further classified into inline, media type, importance, selector specificity, rule order, inheritance and property definition. CSS style information can be in a separate document or it can be embedded into an HTML document. Multiple style sheets can be imported. Different styles can be applied depending on the output device being used; for example, the screen version can be quite different from the printed version, so that authors can tailor the presentation appropriately for each medium. The style sheet with the highest priority controls the content display. Declarations not set in the highest priority source are passed on to a source of lower priority, such as the user agent style. The process is called cascading. One of the goals of CSS is to allow users greater control over presentation. Someone who finds red italic headings difficult to read may apply a different style sheet.

Depending on the browser and the web site, a user may choose from various style sheets provided by the designers, or may remove all added styles and view the site using the browser's default styling, or may override just the red italic heading style without altering other attributes.

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 colour of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colour are used, layout designs, variations in display for different devices and screen sizes as well as a variety of other effects.

CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

Who Creates and Maintains CSS?

CSS is created and maintained through a group of people within the W3C called the CSS Working Group. The CSS Working Group creates documents called specifications. When a specification has been discussed and officially ratified by the W3C members, it becomes a recommendation.

These ratified specifications are called recommendations because the W3C has no control over the actual implementation of the language. Independent companies and organizations create that software.

NOTE – The World Wide Web Consortium, or W3C is a group that makes recommendations about how the Internet works and how it should evolve.

CSS Versions

Cascading Style Sheets level 1 (CSS1) came out of W3C as a recommendation in December 1996. This version describes the CSS language as well as a simple visual formatting model for all the HTML tags.

CSS2 became a W3C recommendation in May 1998 and builds on CSS1. This version adds support for media-specific style sheets e.g., printers and aural devices, downloadable fonts, element positioning and tables.

6.1.6 Bootstrap

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS- and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation and other interface components. Bootstrap is the third-most starred project on GitHub, with more than 135,000 stars, behind only free Code Camp (almost 305,000 stars) and marginally behind Vue.js framework. According to Alexa Rank, Bootstrap getbootstrap.com is in the top-2000 in US while vuejs.org is in top-7000 in US. 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 colour, 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 result 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. 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 result 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-coloured tables, page headings, more prominent pull quotes, and text with a highlight.

Bootstrap was developed by Mark Otto and Jacob Thornton at Twitter. It was released as an open-source product in August 2011 on GitHub.

In June 2014 Bootstrap was the No.1 project on GitHub.

Why use Bootstrap

Following are the main advantage of Bootstrap:

- o It is very easy to use. Anybody having basic knowledge of HTML and CSS can use Bootstrap.
- o It facilitates users to develop a responsive website.
- o It is compatible on most of browsers like Chrome, Firefox, Internet Explorer, Safari and Opera etc.

What is a responsive website?

A website is called responsive website which can automatically adjust itself to look good on all devices, from smart phones to desktops etc

What Bootstrap package contains**Scaffolding:**

Bootstrap provides a basic structure with Grid System, link styles, and background.

CSS:

Bootstrap comes with the feature of global CSS settings, fundamental HTML elements style and an advanced grid system.

Components:

Bootstrap contains a lot of reusable components built to provide iconography, drop-downs, navigation, alerts, pop-overs, and much more.

JavaScript Plugins:

Bootstrap also contains a lot of custom J-Query plugins. You can easily include them all, or one by one.

Is Bootstrap Best?

Bootstrap is more than efficient to create a responsive and mobile first website but it is not the best in the industry. There is an alternative of Bootstrap named W3.CSS which is smaller, faster, and easier to use.

Customize:

Bootstrap components are customizable and you can customize Bootstrap's components, LESS variables, and J-Query plugins to get your own style.

6.2.1 VS Code:

Visual Studio Code is a source-code editor made by Microsoft for Windows, Linux and mac OS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.

Microsoft has released most of Visual Studio Code's source code on the Microsoft/ vocode repository of GitHub using the "Code – OSS" name, under the permissive MIT License, while the releases by Microsoft are proprietary freeware. In the Stack Overflow 2019 Developer Survey, Visual Studio Code was ranked the most popular developer environment tool, with 50.7% of 87,317 respondents reporting that they use it.

Visual Studio Code was first announced on April 29, 2015, by Microsoft at the 2015 Build conference. A Preview build was released shortly thereafter. On November 18, 2015, Visual Studio Code was released under the MIT License, having its source code available on GitHub. Extension support was also announced.[13] On April 14, 2016, Visual Studio Code graduated from the public preview stage and was released to the Web.

6.3 Hardware

6.3.1 RAM

RAM (Random Access Memory) is the internal memory of the CPU for storing data, program, and program result. It is a read/write memory which stores data until the machine is working. As soon as the machine is switched off, data is erased.

access time in RAM is independent of the address, that is, each storage location inside the memory is as easy to reach as other locations and takes the same amount of time. Data in the RAM can be accessed randomly but it is very expensive.

RAM is volatile, i.e., data stored in it is lost when we switch off the computer or if there is a power failure. Hence, a backup Uninterruptible Power System (UPS) is often used with computers. RAM is small, both in terms of its physical size and in the amount of data it can hold.

RAM is of two types –

- Static RAM (SRAM)
- Dynamic RAM (DRAM)

Static RAM (SRAM)

The word static indicates that the memory retains its contents as long as power is being supplied. However, data is lost when the power gets down due to volatile nature. SRAM chips use a matrix of 6-transistors and no capacitors. Transistors do not require power to prevent leakage, so SRAM need not be refreshed on a regular basis.

There is extra space in the matrix, hence SRAM uses more chips than DRAM for the same amount of storage space, making the manufacturing costs higher. SRAM is thus used as cache memory and has very fast access.

Characteristic of Static RAM

- Long life
- No need to refresh
- Faster
- Used as cache memory
- Large size
- Expensive
- High power consumption

Dynamic RAM (DRAM)

DRAM, unlike SRAM, must be continually refreshed in order to maintain the data. This is done by placing the memory on a refresh circuit that rewrites the data several hundred times per second. DRAM is used for most system memory as it is cheap and small. All DRAM are made up of memory cells, which are composed of one capacitor and one transistor.

Characteristics of Dynamic RAM

- Short data lifetime
- Needs to be refreshed continuously
- Slower as compared to SRAM
- Used as RAM
- Smaller in size
- Less expensive
- Less power consumption

6.3.2 Hard Drive

It is also called HDD,” disk drive,” (hard drive) or hard disk is a non-volatile. The hard disk drive is the primary storage unit of the computer. This is where your data, programs and the Windows system that is used to make your computer are physically saved in a digital form (string composed of 0s and 1s). There hard disk capacity is expressed in gigabytes (GB). Harder disk drive will have a large capacity, the more you can install programs or store documents in your computer.

Types of Computer Hard Disk Drives:

There are at present two standards discs: ATA - IDE and SATA. The ATA - IDE standard, although still widespread, is doomed to disappear in favour of SATA. SATA hard drives are indeed faster and have a much larger storage capacity.

Difference Between Internal & External Hard Disk

Internal Hard Drive and External Hard Disk share a similar feature.

- The main difference is that internal hard drive is installed inside to your desktop computer or External Hard Disk are connected via USB 2.0.
- Internal hard drive store all of data, while external hard disk is to protection your data in case corruption takes place.
- Internal hard drive is much faster than an External Hard Disk. The internal drive is connected advanced technology bus.

Hard Disk Drive Physical Description:

A hard disk consists of a stack of disk platters that are made up of aluminium alloy of glass coated with a magnetic material; actually, most hard disks have from two to eight platters. Each side of each platter has a read/write head that floats over the surface of the rapidly spinning disk and picks up (reads) magnetically stored data, or records (writes) data onto the disk. It's just liked a tape recorder, except that in a hard disk, the head doesn't touch the disk surface. Instead, the read/write heads float on an extraordinarily thin cushion of air, about the height of a fingerprint. Really. That's why it's important not to jostle your hard disk or kick it while it's doing its work. The surface of a disk is divided into imaginary tracks and sectors. Tracks are concentric circles where the data is stored. These tracks are numbered from the outermost ring to the innermost ring, starting from zero. Disk sectors refer to the number of fixed size areas that can be accessed by one of the disk drive's read/write heads, in one rotation of the disk, without the head having to change its position. An intersection of a track and a disk sector is known as track sector.

Each sector is uniquely assigned a disk address before a disk drive can access a piece of data. In order to make the disk usable, first it must be formatted to create tracks and sectors. The track sectors are grouped into a collection known as cluster. It refers to the basic allocation unit for storage on a disk.

On the hard disk you'll keep the operating system, and copies of all the

software applications you use, such as your word processor and your spreadsheet program, plus copies of all the documents or files you create. The hard drive is sometimes referred to as the “C drive” due to the fact that Microsoft Windows designates the “C” drive letter to the primary partition on the primary hard drive in a computer by default. While this is not a technically correct term to use, it is still common. For example, some computers have multiple drive letters (i.e. C, D, E) representing areas across one or more hard drives.

Some computers don’t absolutely need a hard disk for storing things the use floppy disks instead. But a hard disk drive is much, much faster than a floppy disk drive, and it stores much more information. And as software programs and operating systems get more sophisticated, they also take up more disks space-so much disk space that many programs and systems can’t even fit on a floppy disk. These days, almost every computer has a hard disk.

CHAPTER 7 : TESTING

7.1 : UNIT TESTING

7.1.1 Introduction

In computer programming, unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use. Intuitively, one can view a unit as the smallest testable part of an application. In procedural programming, a unit could be an entire module, but it is more commonly an individual function or procedure. In object-oriented programming, a unit is often an entire interface, such as a class, but could be an individual method. Unit tests are short code fragments created by programmers or occasionally by white box testers during the development process. It forms the basis for component testing. Ideally, each test case is independent from the others. Substitutes such as method stubs, mock objects, fakes, and test harnesses can be used to assist testing a module in isolation. Unit tests are typically written and run by software developers to ensure that code meets its design and behaves as intended.

7.1.2 Benefits :

The goal of unit testing is to isolate each part of the program and show that the individual parts are correct. A unit test provides a strict, written contract that the piece of code must satisfy. As a result, it affords several benefits.

1) Find problems early :

Unit testing finds problems early in the development cycle. In test-driven development (TDD), which is frequently used in both extreme programming and scrum, unit tests are created before the code itself is written. When the tests pass, that code is considered complete. The same unit tests are run against that function frequently as the larger code base is developed either as the code is changed or via an automated process with the build. If the unit tests fail, it is considered to be a bug either in the changed code or the tests themselves. The unit tests then allow the location of the fault or failure to be easily traced. Since the unit tests alert the development team of the problem before handing the code off to testers or clients, it is still early in the development process.

2) Facilitates Change :

Unit testing allows the programmer to refactor code or upgrade system libraries at a later date, and make sure the module still works correctly (e.g., in regression testing). The procedure is to write test cases for all functions and methods so that whenever a change causes a fault, it can be quickly identified. Unit tests detect changes which may break a design contract.

3) Simplifies Integration :

Unit testing may reduce uncertainty in the units themselves and can be used in a bottom-up testing style approach. By testing the parts of a program first and then testing the sum of its parts, integration testing becomes much easier.

4) Documentation :

Unit testing provides a sort of living documentation of the system. Developers looking to learn what functionality is provided by a unit, and how to use it, can look at the unit tests to gain a basic understanding of the unit's interface (API). Unit test cases embody characteristics that are critical to the success of the unit. These characteristics can indicate appropriate/inappropriate use of a unit as well as negative behaviors that are to be trapped by the unit.

7.2 : INTEGRATION TESTING

Integration testing (sometimes called integration and testing, abbreviated I&T) is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

7.2.1 Purpose

The purpose of integration testing is to verify functional, performance, and reliability requirements placed on major design items. These "design items", i.e., assemblages (or groups of units), are exercised through their interfaces using black-box testing, success and error cases being simulated via appropriate parameter and data inputs. Simulated usage of shared data areas and inter-process communication is tested

and individual subsystems are exercised through their input interface. Test cases are constructed to test whether all the components within assemblages interact correctly, for example across procedure calls or process activation, and this is done after testing individual modules, i.e., unit testing. The overall idea is a "building block" approach, in which verified assemblages are added to a verified base which is then used to support the integration testing of further assemblages. Software integration testing is performed according to the software development life cycle (SDLC) after module and functional tests. The cross dependencies for software integration testing are: schedule for integration testing, strategy and selection of the tools used for integration, define the cyclomathical complexity of the software and software architecture, reuse-ability of modules and life-cycle and versioning management. Some different types of integration testing are big-bang, top-down, and bottom-up, mixed (sandwich) and risky-hardest. Other Integration Patterns[2] are: collaboration integration, backbone integration, layer integration, client-server integration, distributed services integration and high-frequency integration.

7.2.1.1 Big Bang

In the big-bang approach, most of the developed modules are coupled together to form a complete software system or major part of the system and then used for integration testing. This method is very effective for saving time in the integration testing process. However, if the test cases and their results are not recorded properly, the entire integration process will be more complicated and may prevent the testing team from achieving the goal of integration testing. A type of big-bang integration testing is called "usage model testing" which can be used in both software and hardware integration testing. The basis behind this type of integration testing is to run user-like workloads in integrated user-like environments. In doing the testing in this manner, the environment is proofed, while the individual components are proofed indirectly through their use. Usage Model testing takes an optimistic approach to testing, because it expects to have few problems with the individual components. The strategy relies heavily on the component developers to do the isolated unit testing for their product. The goal of the strategy is to avoid redoing the testing done by the developers, and instead flesh-out problems caused by the interaction of the components in the environment.

7.2.1.2 Top-down And Bottom-up

Bottom-up testing is an approach to integrated testing where the lowest level components are tested first, then used to facilitate the testing of higher level components. The process is repeated until the component at the top of the hierarchy is tested. All the bottom or low-level modules, procedures or functions are integrated and then tested. After the integration testing of lower level integrated modules, the next level of modules will be formed and can be used for integration testing. This approach

is helpful only when all or most of the modules of the same development level are ready. This method also helps to determine the levels of software developed and makes it easier to report testing progress in the form of a percentage. Top-down testing is an approach to integrated testing where the top integrated modules are tested and the branch of the module is tested step by step until the end of the related module. Sandwich testing is an approach to combine top down testing with bottom up testing.

7.3 : SOFTWARE VERIFICATION AND VALIDATION

7.3.1 Introduction

In software project management, software testing, and software engineering, verification and validation (V&V) is the process of checking that a software system meets specifications and that it fulfills its intended purpose. It may also be referred to as software quality control. It is normally the responsibility of software testers as part of the software development life-cycle. Validation checks that the product design satisfies or fits the intended use (high-level checking), i.e., the software meets the user requirements. This is done through dynamic testing and other forms of review. Verification and validation are not the same thing, although they are often confused. Boehm succinctly expressed the difference between

Validation : Are we building the right product?

Verification : Are we building the product right?

According to the Capability Maturity Model (CMMI-SW v1.1)

Software Verification: The process of evaluating software to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase.

Software Validation: The process of evaluating software during or at the end of the development process to determine whether it satisfies specified requirements. In other words, software verification is ensuring that the product has been built according to the requirements and design specifications, while software validation ensures that the product meets the user's needs, and that the specifications were correct in the first place. Software verification ensures that "you built it right". Software validation ensures that "you built the right thing". Software validation confirms that the product, as provided, will fulfill its intended use.

From Testing Perspective

Fault – wrong or missing function in the code.

Failure – the manifestation of a fault during execution.

Malfunction – according to its specification the system does not meet its specified functionality.

Both verification and validation are related to the concepts of quality and

of software quality assurance. By themselves, verification and validation do not guarantee software quality; planning, traceability, configuration management and other aspects of software engineering are required. Within the modeling and simulation (M&S) community, the definitions of verification, validation and accreditation are similar:

M&S Verification is the process of determining that a computer model, simulation, or federation of models and simulations implementations and their associated data accurately represent the developer's conceptual description and specifications.

M&S Validation is the process of determining the degree to which a model, simulation, or federation of models and simulations, and their associated data are accurate representations of the real world from the perspective of the intended use(s).

7.3.2 Classification of Methods

In mission-critical software systems, where flawless performance is absolutely necessary, formal methods may be used to ensure the correct operation of a system. However, often for non-missioncritical software systems, formal methods prove to be very costly and an alternative method of software V&V must be sought out. In such cases, syntactic methods are often used.

7.3.3 Test Cases

A test case is a tool used in the process. Test cases may be prepared for software verification and software validation to determine if the product was built according to the requirements of the user. Other methods, such as reviews, may be used early in the life cycle to provide for software validation.

7.4 : Black-Box Testing

Black-box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied virtually to every level of software testing: unit, integration, system and acceptance. It typically comprises most if not all higher level testing, but can also dominate unit testing as well.

7.4.1 Test Procedures

Specific knowledge of the application's code/internal structure and

programming knowledge in general is not required. The tester is aware of what the software is supposed to do but is not aware of how it does it. For instance, the tester is aware that a particular input returns a certain, invariable output but is not aware of how the software produces the output in the first place.

7.4.2 Test Cases

Test cases are built around specifications and requirements, i.e., what the application is supposed to do. Test cases are generally derived from external descriptions of the software, including specifications, requirements and design parameters. Although the tests used are primarily functional in nature, non-functional tests may also be used. The test designer selects both valid and invalid inputs and determines the correct output, often with the help of an oracle or a previous result that is known to be good, without any knowledge of the test object's internal structure.

7.5 : White-Box Testing

White-box testing (also known as clear box testing, glass box testing, transparent box testing, and structural testing) is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality (i.e. black-box testing). In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs. This is analogous to testing nodes in a circuit, e.g. in-circuit testing (ICT). White-box testing can be applied at the unit, integration and system levels of the software testing process. Although traditional testers tended to think of white-box testing as being done at the unit level, it is used for integration and system testing more frequently today. It can test paths within a unit, paths between units during integration, and between subsystems during a system-level test. Though this method of test design can uncover many errors or problems, it has the potential to miss unimplemented parts of the specification or missing requirements.

7.5.1 Levels

1) Unit testing :

White-box testing is done during unit testing to ensure that the code is working as intended, before any integration happens with previously tested code. White-box testing during unit testing catches any defects early on and aids in any defects that happen later on after the code is integrated with the rest of the application and therefore prevents any type of errors later on.

2) Integration testing :

White-box testing at this level are written to test the interactions of each interface with each other. The Unit level testing made sure that each code was tested and working accordingly in an isolated environment and integration examines the correctness of the behaviour in an open environment through the use of white-box testing for any interactions of interfaces that are known to the programmer.

3) Regression testing :

White-box testing during regression testing is the use of recycled white-box test cases at the unit and integration testing levels.

7.5.2 Procedures

White-box testings basic procedures involves the tester having a deep level of understanding of the source code being tested. The programmer must have a deep understanding of the application to know what kinds of test cases to create so that every visible path is exercised for testing. Once the source code is understood then the source code can be analyzed for test cases to be created. These are the three basic steps that white-box testing takes in order to create test cases:

Input involves different types of requirements, functional specifications, detailed designing of documents, proper source code, security specifications. This is the preparation stage of white-box testing to layout all of the basic information.

Processing involves performing risk analysis to guide whole testing process, proper test plan, execute test cases and communicate results. This is the phase of building test cases to make sure they thoroughly test the application the given results are recorded accordingly.

Output involves preparing final report that encompasses all of the above preparations and results.

7.6 : SYSTEM TESTING

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black-box testing, and as such, should require no knowledge of the inner design of the code or logic. As a rule, system testing takes, as its input, all of the "integrated" software components that have passed integration testing and also the software system itself integrated with any applicable hardware system(s).

The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together (called assemblages) or between any of the assemblages and the hardware. System testing is a more limited type of

testing; it seeks to detect defects both within the "inter-assemblages" and also within the system as a whole.

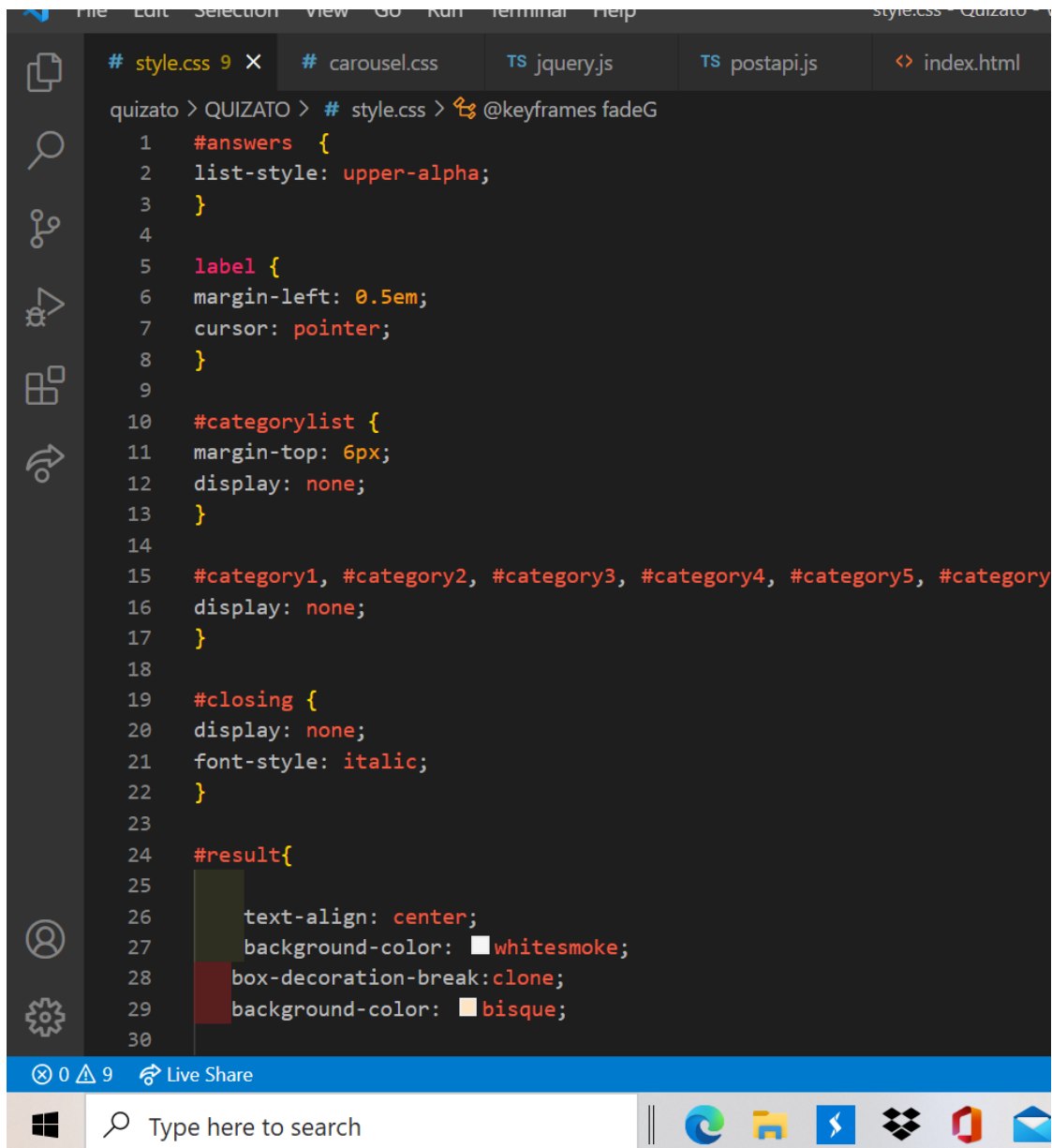
System testing is performed on the entire system in the context of a Functional Requirement Specification(s) (FRS) and/or a System Requirement Specification (SRS). System testing tests not only the design, but also the behavior and even the believed expectations of the customer. It is also intended to test up to and beyond the bounds defined in the software/hardware requirements specification(s).

Note: Because my Project is in progress therefore my Project is not completely done and some step according this file is not done like testing.

CHAPTER 8 : RESULTS

8.1 Css Code:

8.1.1 Style.Css



```
quizato > QUIZATO > # style.css > @keyframes fadeG
1  #answers {
2    list-style: upper-alpha;
3  }
4
5  label {
6    margin-left: 0.5em;
7    cursor: pointer;
8  }
9
10 #categorylist {
11   margin-top: 6px;
12   display: none;
13 }
14
15 #category1, #category2, #category3, #category4, #category5, #category
16 display: none;
17 }
18
19 #closing {
20   display: none;
21   font-style: italic;
22 }
23
24 #result{
25
26   text-align: center;
27   background-color: whitesmoke;
28   box-decoration-break: clone;
29   background-color: bisque;
30 }
```

Fig 8.1.1-1

```
quizato > QUIZATO > # style.css > @keyframes fadeG

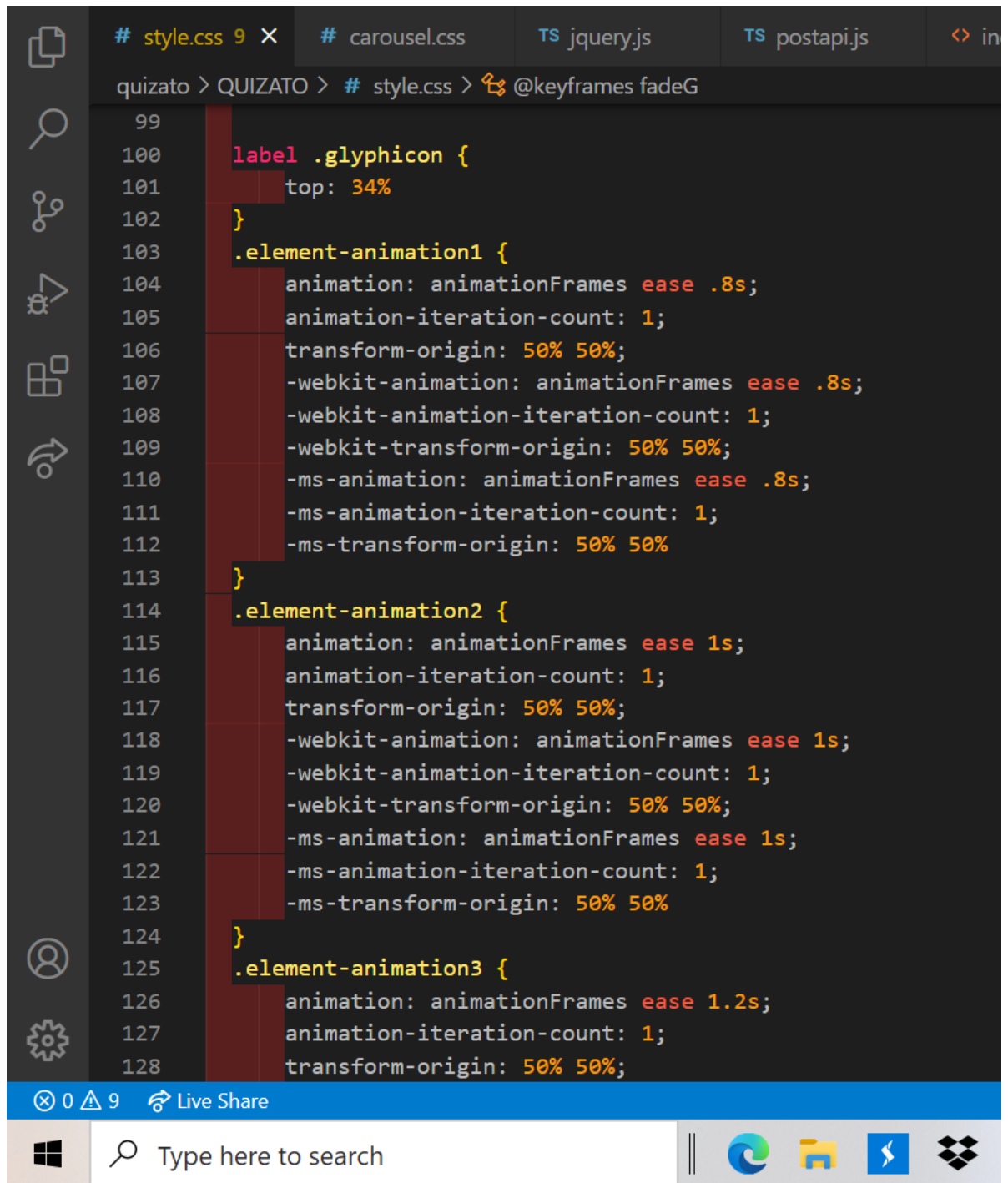
40 #results:hover{
41     background-color: green;
42     border: none;
43     color: white;
44     padding: 15px 32px;
45     text-align: center;
46     text-decoration: none;
47     display: inline;
48     font-size: 16px;
49     margin: 4px 2px;
50     cursor: pointer;
51 }
52 /* Answer how will be shown css */
53 #ans{
54     background-color: rgb(141, 245, 229);
55     margin-left: 20%;
56     margin-right: 20%;
57     margin-bottom: 10%;
58     margin-top: 0%;
59 }
60 }
61 #qid {
62     padding: 10px 15px;
63     -moz-border-radius: 50px;
64     -webkit-border-radius: 50px;
65     border-radius: 20px;
66 }
67 label.btn {
68     padding: 18px 60px;
69     white-space: normal;
```

0 9 Live Share

Type here to search

Fig 8.1.1-2

Fig 8.1.1-2 is screenshot of css code which is apply on the result, that how result will be show. Here ans class the css of display answer.

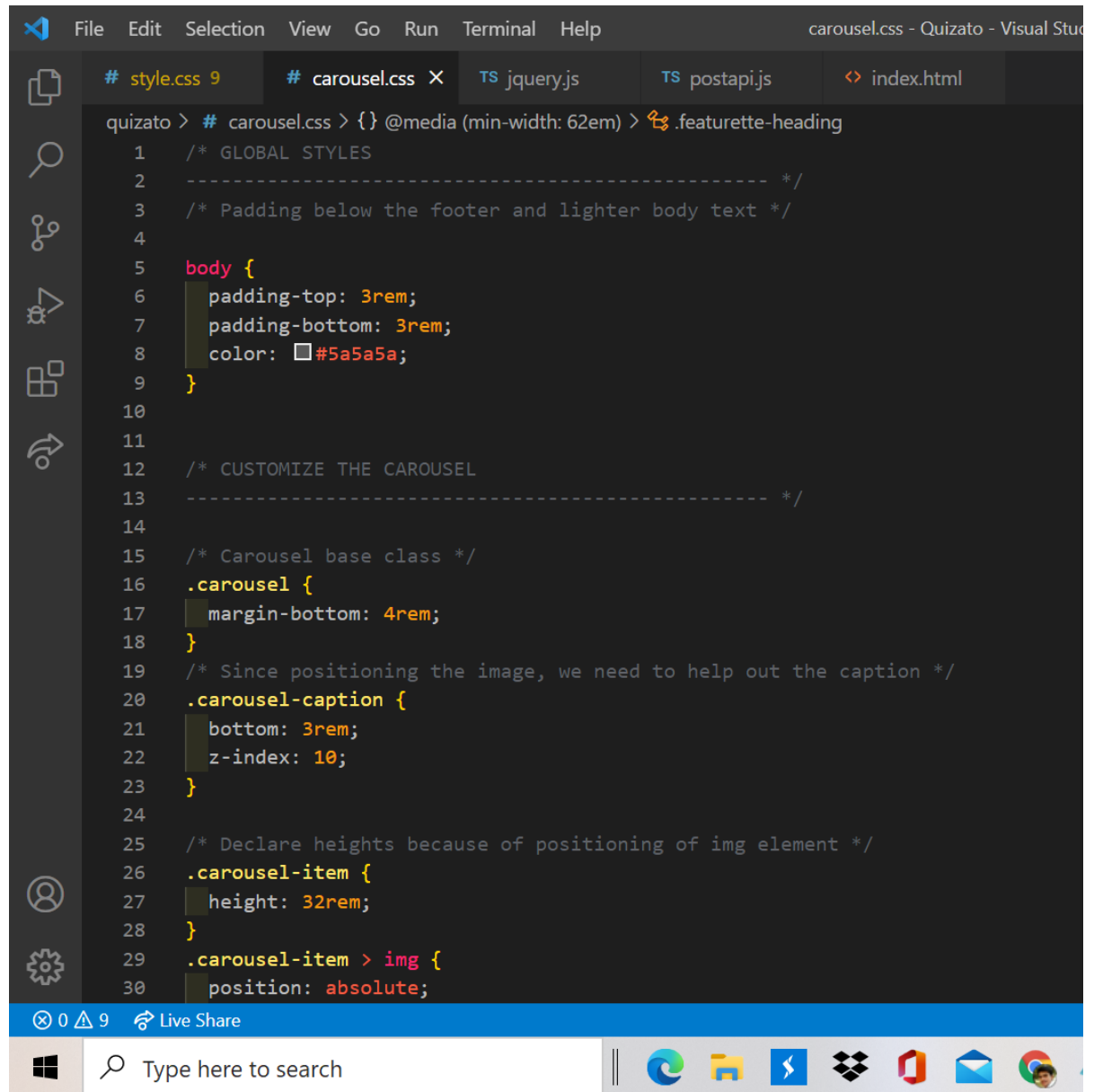


```
# style.css 9 X # carousel.css TS jquery.js TS postapi.js <> in
quizato > QUIZATO > # style.css > @keyframes fadeG
99
100 label .glyphicon {
101     top: 34%
102 }
103 .element-animation1 {
104     animation: animationFrames ease .8s;
105     animation-iteration-count: 1;
106     transform-origin: 50% 50%;
107     -webkit-animation: animationFrames ease .8s;
108     -webkit-animation-iteration-count: 1;
109     -webkit-transform-origin: 50% 50%;
110     -ms-animation: animationFrames ease .8s;
111     -ms-animation-iteration-count: 1;
112     -ms-transform-origin: 50% 50%
113 }
114 .element-animation2 {
115     animation: animationFrames ease 1s;
116     animation-iteration-count: 1;
117     transform-origin: 50% 50%;
118     -webkit-animation: animationFrames ease 1s;
119     -webkit-animation-iteration-count: 1;
120     -webkit-transform-origin: 50% 50%;
121     -ms-animation: animationFrames ease 1s;
122     -ms-animation-iteration-count: 1;
123     -ms-transform-origin: 50% 50%
124 }
125 .element-animation3 {
126     animation: animationFrames ease 1.2s;
127     animation-iteration-count: 1;
128     transform-origin: 50% 50%;
```

Fig 8.1.1-3

Fig 8.1.1-3 is the screenshot of some codes of style.css where it gives style to our first page which will be apply on our Home page or Index page.

8.1.2 Carousels.Css



```
quizato > # carousel.css > {} @media (min-width: 62em) > .featurette-heading
1  /* GLOBAL STYLES
2  ----- */
3  /* Padding below the footer and lighter body text */
4
5  body {
6    padding-top: 3rem;
7    padding-bottom: 3rem;
8    color: #5a5a5a;
9  }
10
11
12  /* CUSTOMIZE THE CAROUSEL
13  ----- */
14
15  /* Carousel base class */
16  .carousel {
17    margin-bottom: 4rem;
18  }
19  /* Since positioning the image, we need to help out the caption */
20  .carousel-caption {
21    bottom: 3rem;
22    z-index: 10;
23  }
24
25  /* Declare heights because of positioning of img element */
26  .carousel-item {
27    height: 32rem;
28  }
29  .carousel-item > img {
30    position: absolute;
```

Fig 8.1.2-1

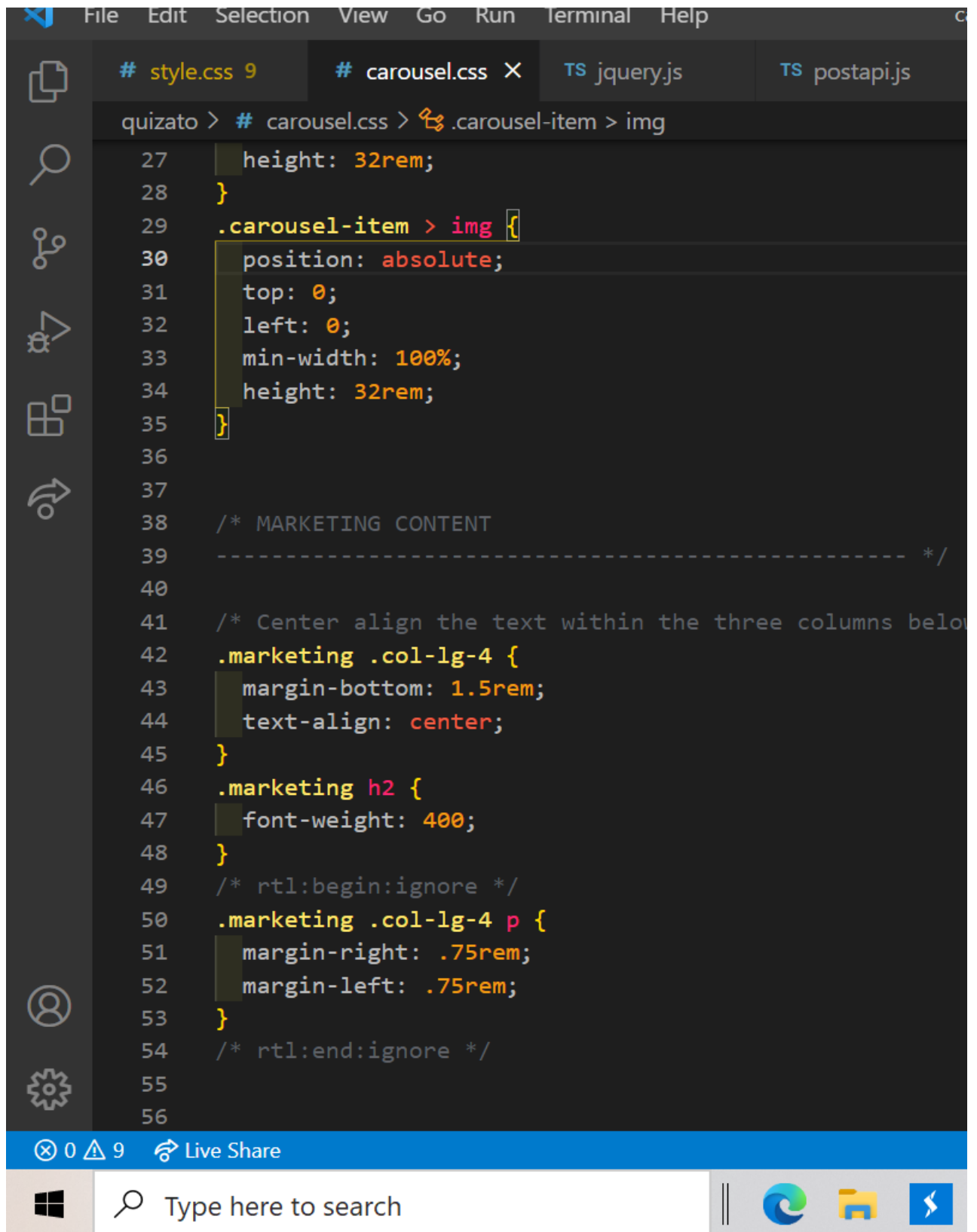
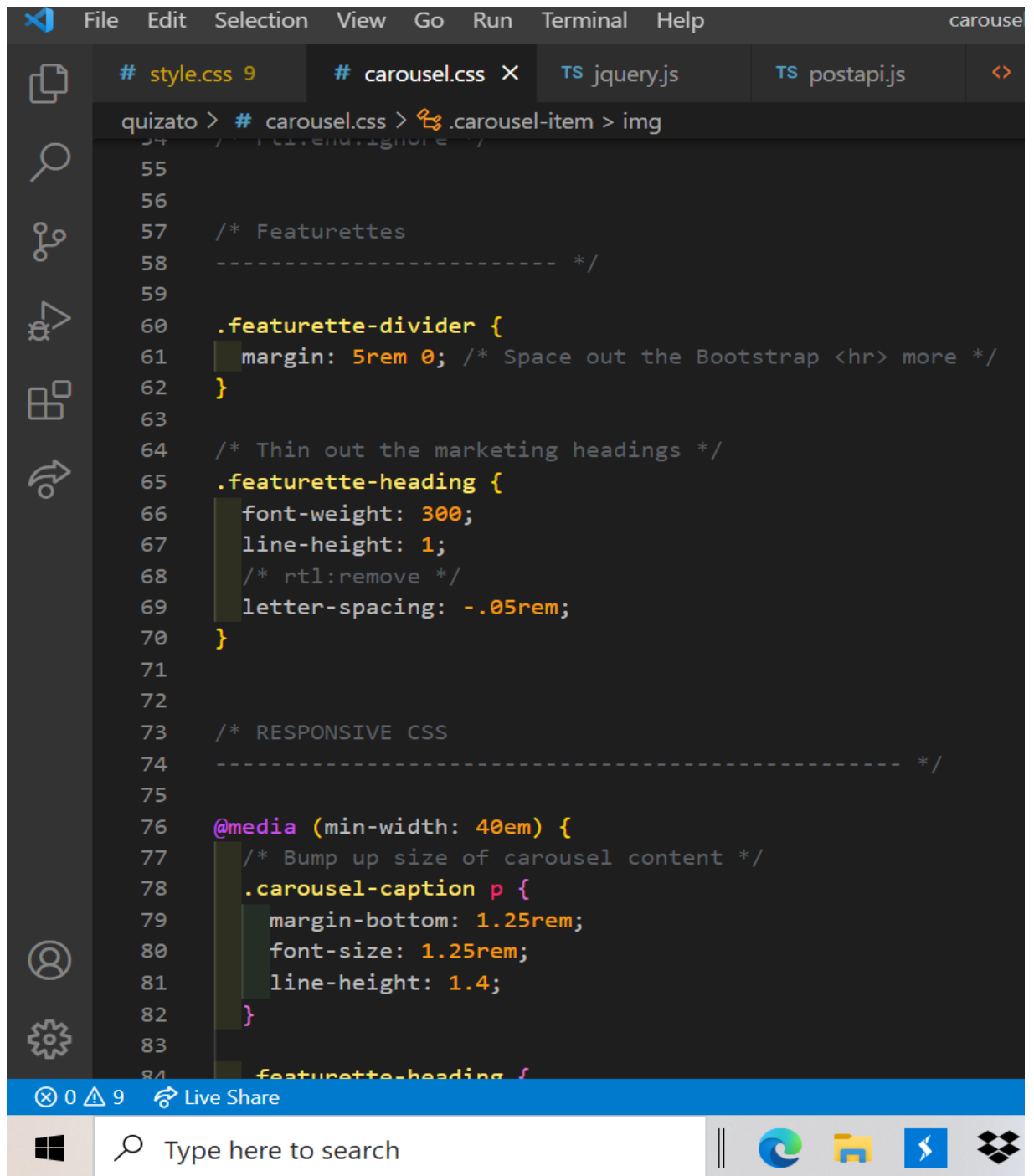


Fig 8.1.2-2



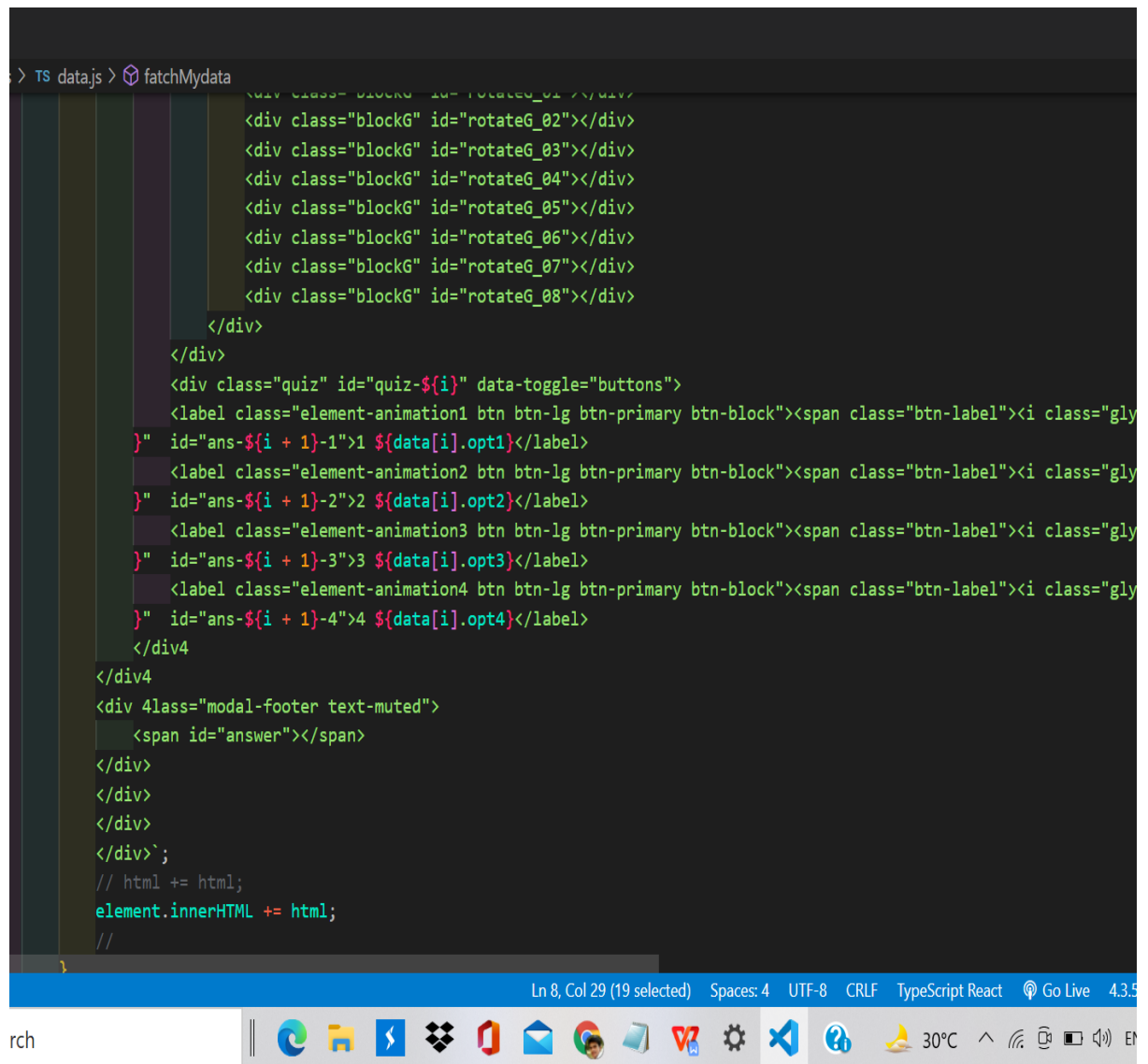
```
File Edit Selection View Go Run Terminal Help
# style.css 9 # carousel.css X TS jquery.js TS postapi.js <>
quizato > # carousel.css > .carousel-item > img
54 /* rtl:remove */
55
56
57 /* Featurettes
58 ----- */
59
60 .featurette-divider {
61   margin: 5rem 0; /* Space out the Bootstrap <hr> more */
62 }
63
64 /* Thin out the marketing headings */
65 .featurette-heading {
66   font-weight: 300;
67   line-height: 1;
68   /* rtl:remove */
69   letter-spacing: -.05rem;
70 }
71
72
73 /* RESPONSIVE CSS
74 ----- */
75
76 @media (min-width: 40em) {
77   /* Bump up size of carousel content */
78   .carousel-caption p {
79     margin-bottom: 1.25rem;
80     font-size: 1.25rem;
81     line-height: 1.4;
82   }
83
84   .featurette-heading {
```

Fig 8.1.2-3

Fig 8.1.2-1, Fig 8.1.2-2 and Fig 8.1.2-3 is the screenshot of some code of carousel.css which gave style and looks to our index page or Home page. After changing the carousel.css we can change or modify our Home Page.

8.2 Java Script

8.2.1 data.js



```
> TS data.js > fetchMydata
<div class="blockG" id="rotateG_01"></div>
<div class="blockG" id="rotateG_02"></div>
<div class="blockG" id="rotateG_03"></div>
<div class="blockG" id="rotateG_04"></div>
<div class="blockG" id="rotateG_05"></div>
<div class="blockG" id="rotateG_06"></div>
<div class="blockG" id="rotateG_07"></div>
<div class="blockG" id="rotateG_08"></div>
</div>
</div>
<div class="quiz" id="quiz-${i}" data-toggle="buttons">
<label class="element-animation1 btn btn-lg btn-primary btn-block"><span class="btn-label"><i class="gly
)" id="ans-${i + 1}-1">1 ${data[i].opt1}</label>
<label class="element-animation2 btn btn-lg btn-primary btn-block"><span class="btn-label"><i class="gly
)" id="ans-${i + 1}-2">2 ${data[i].opt2}</label>
<label class="element-animation3 btn btn-lg btn-primary btn-block"><span class="btn-label"><i class="gly
)" id="ans-${i + 1}-3">3 ${data[i].opt3}</label>
<label class="element-animation4 btn btn-lg btn-primary btn-block"><span class="btn-label"><i class="gly
)" id="ans-${i + 1}-4">4 ${data[i].opt4}</label>
</div4
</div4
<div class="modal-footer text-muted">
<span id="answer"></span>
</div>
</div>
</div>
</div>`;
// html += html;
element.innerHTML += html;
//
}
```

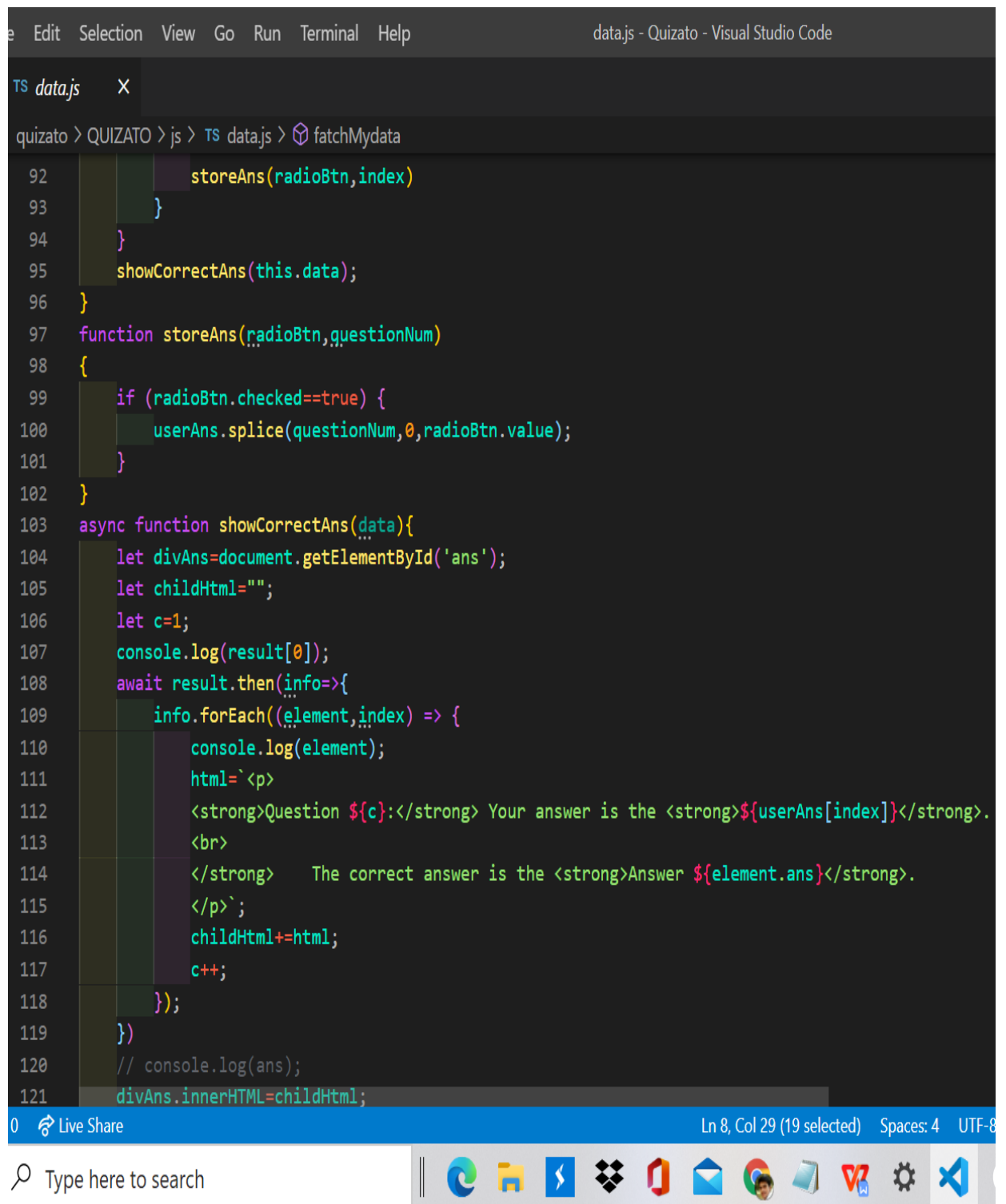
Fig 8.2.1-1

Fig 8.2.1-1 is the screenshot of data.js. In this the class “Quiz” is fetching the Question and Option from our database for display to user. Here will be one Question and 4 option to display user with unique id.

```
data.js  X
Quizato > QUIZATO > js > TS data.js > fetchMydata
60     element.innerHTML += html;
61     //
62 }
63 });
64 } else {
65     document.write('res.status');
66 }
67 } catch (error) {
68     document.write(error);
69 }
70 })();
71 let submit = document.getElementById("submit");
72 submit.onclick = () => checkAns();
73 async function checkAns() {
74     let data;
75
76     userAns=[]
77
78     await result.then((data) => {
79         this.data = data.length;
80     });
81     //
82     for (let index = 0; index < this.data; index++) {
83         //
84         let str = `q_answer_${index}`;
85         // demo=document.querySelectorAll('type=radio');
86         //
87         let element = document.getElementsByName(str);
88         for (let j = 1; j <= element.length; j++) {
89             let id = `ans-${index + 1}-${j}`;
```

Fig 8.2.1-2

Fig 8.2.1-2 is the screenshot of data.js. By this code we are submit the answer after clicking on “Submit” button and after submit we will check the answer that the selected option by user is correct or not.



```
TS data.js x
quizato > QUIZATO > js > TS data.js > fetchMydata

92     storeAns(radioBtn,index)
93   }
94 }
95   showCorrectAns(this.data);
96 }
97 function storeAns(radioBtn,questionNum)
98 {
99   if (radioBtn.checked==true) {
100     userAns.splice(questionNum,0,radioBtn.value);
101   }
102 }
103 async function showCorrectAns(data){
104   let divAns=document.getElementById('ans');
105   let childHtml="";
106   let c=1;
107   console.log(result[0]);
108   await result.then(info=>{
109     info.forEach((element,index) => {
110       console.log(element);
111       html=`<p>
112         <strong>Question ${c}:</strong> Your answer is the <strong>${userAns[index]}</strong>.
113         <br>
114         </strong>    The correct answer is the <strong>Answer ${element.ans}</strong>.
115       </p>`;
116       childHtml+=html;
117       c++;
118     });
119   })
120   // console.log(ans);
121   divAns.innerHTML=childHtml;
```

0 Live Share Ln 8, Col 29 (19 selected) Spaces: 4 UTF-8

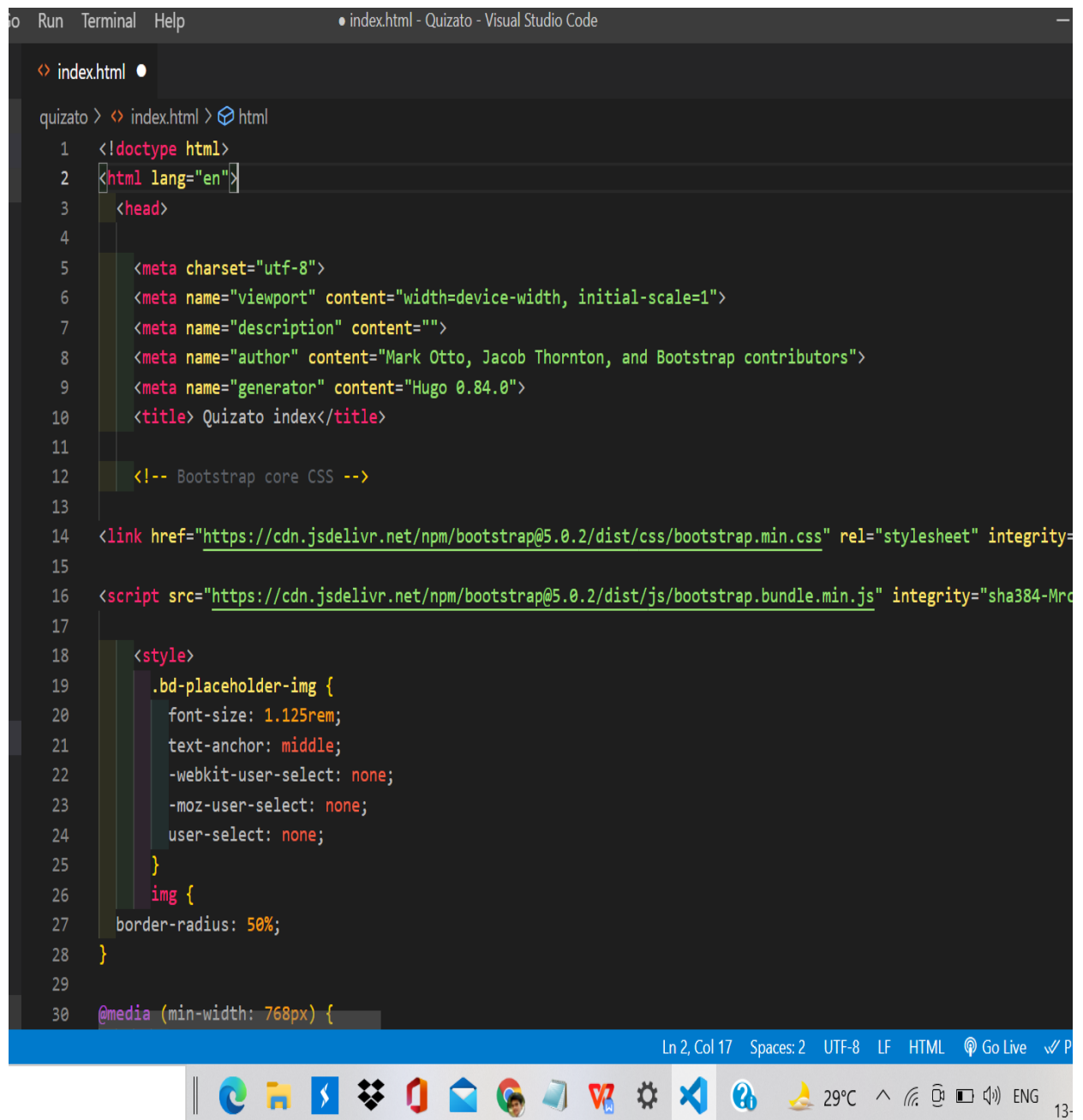
Type here to search

Fig 8.2.1-3

Fig 8.2.1-3 is the screenshot of some code of data.js .This will display the answer to the end user. It will display the both answer to user one the selected answer and second the correct answer. We can manage how to display the correct and incorrect answer.

8.3 HTML

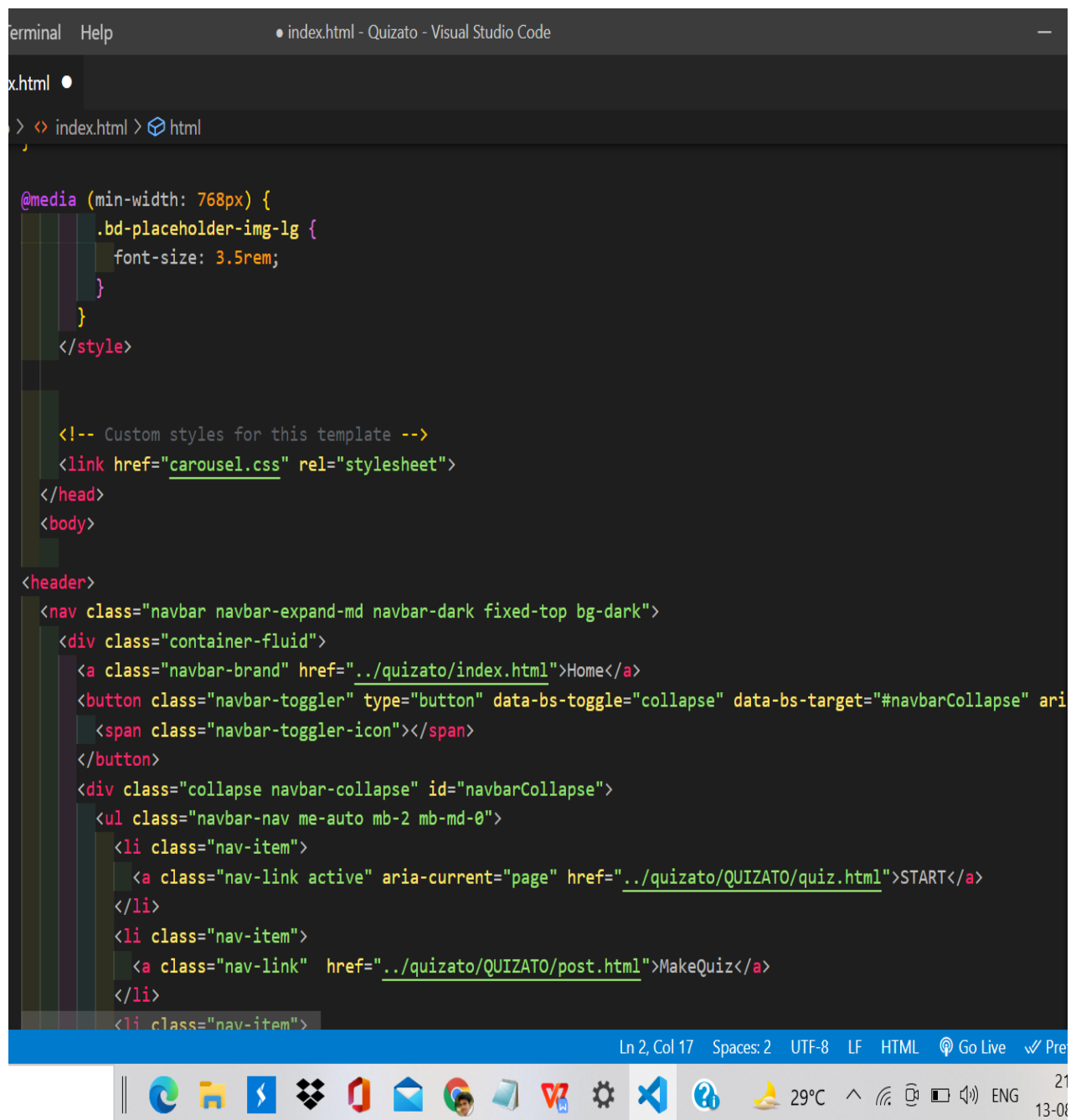
8.3.1 Index.html



```
index.html •
quizato > <> index.html > html
1  <!doctype html>
2  <html lang="en">
3    <head>
4
5      <meta charset="utf-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1">
7      <meta name="description" content="">
8      <meta name="author" content="Mark Otto, Jacob Thornton, and Bootstrap contributors">
9      <meta name="generator" content="Hugo 0.84.0">
10     <title> Quizato index</title>
11
12     <!-- Bootstrap core CSS -->
13
14     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity=
15
16     <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-Mrc
17
18     <style>
19       .bd-placeholder-img {
20         font-size: 1.125rem;
21         text-anchor: middle;
22         -webkit-user-select: none;
23         -moz-user-select: none;
24         user-select: none;
25       }
26       img {
27         border-radius: 50%;
28       }
29
30       @media (min-width: 768px) {
```

Fig 8.3.1-1

Fig 8.3.1-1 is the internal css of index page or Home page and link of some external pages to make more effective our Home page . Here in line number 14 and 16 is the link of bootstrap 5 which will make our Home page more effective and style. This bootstrap is used for the header that how my header will be display.



```
index.html •
> <> index.html > html
,

@media (min-width: 768px) {
  .bd-placeholder-img-lg {
    font-size: 3.5rem;
  }
}
</style>

<!-- Custom styles for this template -->
<link href="carousel.css" rel="stylesheet">
</head>
<body>

<header>
  <nav class="navbar navbar-expand-md navbar-dark fixed-top bg-dark">
    <div class="container-fluid">
      <a class="navbar-brand" href=" ../quizato/index.html">Home</a>
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarCollapse" aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarCollapse">
        <ul class="navbar-nav me-auto mb-2 mb-md-0">
          <li class="nav-item">
            <a class="nav-link active" aria-current="page" href=" ../quizato/QUIZATO/quiz.html">START</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href=" ../quizato/QUIZATO/post.html">MakeQuiz</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href=" ../quizato/QUIZATO/post.html">MakeQuiz</a>
          </li>
        </ul>
      </div>
    </div>
  </nav>
</header>
```

Fig 8.3.1-2

This is the screenshot of some code of index.html. Fig 8.3.1 -2 is the header part of the index page of our web application. According to our needs we can change in our header section and can add or remove anything. We can manage our header from here at the Home page.

```

<main>

  <div id="myCarousel" class="carousel slide" data-bs-ride="carousel">
    <div class="carousel-indicators">

      <!-- This is slide page on the home page -->

      <button type="button" data-bs-target="#myCarousel" data-bs-slide-to="0"
class="active" aria-current="true" aria-label="Slide 1"></button>

      <button type="button" data-bs-target="#myCarousel" data-bs-slide-to="1" aria-
label="Slide 2"></button>

      <button type="button" data-bs-target="#myCarousel" data-bs-slide-to="2" aria-
label="Slide 3"></button>
    </div>

    <div class="carousel-inner">
      <div class="carousel-item active">
        <!-- <svg class="bd-placeholder-img" width="100%" height="100%"
xmlns="http://www.w3.org/2000/svg" aria-hidden="true"
preserveAspectRatio="xMidYMid slice" focusable="false"><rect width="100%"
height="100%" fill="#777"/></svg>
-->
        

      </div>

      <div class="carousel-item">
        <!-- <svg class="bd-placeholder-img" width="100%" height="100%"
xmlns="http://www.w3.org/2000/svg" aria-hidden="true"
preserveAspectRatio="xMidYMid slice" focusable="false"><rect width="100%"
height="100%" fill="#777"/></svg>
-->
        

      </div>

      <div class="carousel-item">

        
</div>

<button class="carousel-control-prev" type="button" data-bs-target="#myCarousel"
data-bs-slide="prev">
  <span class="carousel-control-prev-icon" aria-hidden="true"></span>
  <span class="visually-hidden">Previous</span>
</button>

<button class="carousel-control-next" type="button" data-bs-target="#myCarousel"
data-bs-slide="next">
  <span class="carousel-control-next-icon" aria-hidden="true"></span>
  <span class="visually-hidden">Next</span>
</button>

</div>

<!-- Marketing messaging and featurettes
===== -->
<!-- Wrap the rest of the page in another container to center all the content. -->

<div class="container marketing">

  <!-- Three columns of text below the carousel -->
  <div class="row">
    <div class="col-lg-4">
      <svg class="bd-placeholder-img rounded-circle" width="140" height="140"
xmlns="http://www.w3.org/2000/svg" role="img" aria-label="Placeholder: 140x140"
preserveAspectRatio="xMidYMid slice"
focusable="false"><title>Placeholder</title><rect width="100%" height="100%"
fill="#777"/><text x="50%" y="50%" fill="#777" dy=".3em">140x140</text></svg>

      <h2> <a href=" ../quizato/QUIZATO/quiz.html">HTML</a></h2>

      <p>HTML is the standard markup language for Web pages.
        With HTML you can create your own Website.
        HTML is easy to learn - You will enjoy it!</p>

    </div><!-- /.col-lg-4 -->
    <div class="col-lg-4">
      <svg class="bd-placeholder-img rounded-circle" width="140" height="140"
xmlns="http://www.w3.org/2000/svg" role="img" aria-label="Placeholder: 140x140"
preserveAspectRatio="xMidYMid slice"
focusable="false"><title>Placeholder</title><rect width="100%" height="100%"
fill="#777"/><text x="50%" y="50%" fill="#777" dy=".3em">140x140</text></svg>

      <h2>CSS</h2>

      <p>CSS is the language we use to style an HTML document.

```

CSS describes how HTML elements should be displayed.
This tutorial will teach you CSS from basic to advanced.</p>

</div>
<!-- /.col-lg-4 -->

<div class="col-lg-4">
 <svg class="bd-placeholder-img rounded-circle" width="140" height="140"
 xmlns="http://www.w3.org/2000/svg" role="img" aria-label="Placeholder: 140x140"
 preserveAspectRatio="xMidYMid slice"
 focusable="false"><title>Placeholder</title><rect width="100%" height="100%"
 fill="#777"/><text x="50%" y="50%" fill="#777" dy=".3em">140x140</text></svg>

<h2>JavaScript</h2>

<p>JavaScript is the world's most popular programming language.
JavaScript is the programming language of the Web.
JavaScript is easy to learn.
This tutorial will teach you JavaScript from basic to advanced
</p>

</div>
<!-- /.col-lg-4 -->
</div>
<!-- /.row -->

<!--this is another heading >

<div class="container marketing">

<!-- Three columns of text below the carousel -->

<div class="row">
 <div class="col-lg-4">
 <svg class="bd-placeholder-img rounded-circle" width="140" height="140"
 xmlns="http://www.w3.org/2000/svg" role="img" aria-label="Placeholder: 140x140"
 preserveAspectRatio="xMidYMid slice"
 focusable="false"><title>Placeholder</title><rect width="100%" height="100%"
 fill="#777"/><text x="50%" y="50%" fill="#777" dy=".3em">140x140</text></svg>

<h2>Java</h2>

<p>Java is a programming language.
Java is used to develop mobile apps, web apps, desktop apps, games and much
more.</p>

</div><!-- /.col-lg-4 -->
<div class="col-lg-4">

```
<svg class="bd-placeholder-img rounded-circle" width="140" height="140"
xmlns="http://www.w3.org/2000/svg" role="img" aria-label="Placeholder: 140x140"
preserveAspectRatio="xMidYMid slice"
focusable="false"><title>Placeholder</title><rect width="100%" height="100%"
fill="#777"/><text x="50%" y="50%" fill="#777" dy=".3em">140x140</text></svg>
```

<h2>SQL</h2>

<p>SQL is a standard language for storing, manipulating and retrieving data in databases.

Our SQL tutorial will teach you how to use SQL in: MySQL, SQL Server, MS Access, Oracle, Sybase, Informix, Postgres, and other database systems.

</p>

```
</div><!-- /.col-lg-4 -->
<div class="col-lg-4">
```

```
<svg class="bd-placeholder-img rounded-circle" width="140" height="140"
xmlns="http://www.w3.org/2000/svg" role="img" aria-label="Placeholder: 140x140"
preserveAspectRatio="xMidYMid slice"
focusable="false"><title>Placeholder</title><rect width="100%" height="100%"
fill="#777"/><text x="50%" y="50%" fill="#777" dy=".3em">140x140</text></svg>
```

<h2>PHP</h2>

<p>PHP is a server scripting language, and a powerful tool for making dynamic and interactive Web pages.

PHP is a widely-used, free, and efficient alternative to competitors such as Microsoft's ASP.</p>

```
<!--<p><a class="btn btn-secondary" href="#">View details &raquo;</a></p>-->
```

```
</div><!-- /.col-lg-4 -->
</div><!-- /.row -->
```

```
<!-- START THE FEATURETTES -->
```

```
<hr class="featurette-divider">
```

```
<div class="row featurette">
  <div class="col-md-7">
```

```
    <h2 class="featurette-heading">First featurette heading. <span class="text-muted">It'll blow your mind.</span></h2>
```

```
    <p class="lead">Some great placeholder content for the first featurette here.
    Imagine some exciting prose here.</p>
```

```
</div>
```

```

    <div class="col-md-5">
      <svg class="bd-placeholder-img bd-placeholder-img-lg featurette-image img-fluid
mx-auto" width="500" height="500" xmlns="http://www.w3.org/2000/svg" role="img"
aria-label="Placeholder: 500x500" preserveAspectRatio="xMidYMid slice"
focusable="false"><title>Placeholder</title><rect width="100%" height="100%"
fill="#eee"/><text x="50%" y="50%" fill="#aaa" dy=".3em">500x500</text></svg>

    </div>
  </div>

  <hr class="featurette-divider">

  <div class="row featurette">
    <div class="col-md-7 order-md-2">

      <h2 class="featurette-heading">Oh yeah, it's that good. <span class="text-
muted">See for yourself.</span></h2>
      <p class="lead">Another featurette? Of course. More placeholder content here to
give you an idea of how this layout would work with some actual real-world content in
place.</p>
    </div>

    <div class="col-md-5 order-md-1">
      <svg class="bd-placeholder-img bd-placeholder-img-lg featurette-image img-fluid
mx-auto" width="500" height="500" xmlns="http://www.w3.org/2000/svg" role="img"
aria-label="Placeholder: 500x500" preserveAspectRatio="xMidYMid slice"
focusable="false"><title>Placeholder</title><rect width="100%" height="100%"
fill="#eee"/><text x="50%" y="50%" fill="#aaa" dy=".3em">500x500</text></svg>

    </div>
  </div>

  <hr class="featurette-divider">

  <div class="row featurette">
    <div class="col-md-7">

      <h2 class="featurette-heading">And lastly, this one. <span class="text-
muted">Checkmate.</span></h2>
      <p class="lead">And yes, this is the last block of representative placeholder
content. Again, not really intended to be actually read, simply here to give you a better
view of what this would look like with some actual content. Your content.</p>
    </div>

    <div class="col-md-5">
      <svg class="bd-placeholder-img bd-placeholder-img-lg featurette-image img-fluid
mx-auto" width="500" height="500" xmlns="http://www.w3.org/2000/svg" role="img"
aria-label="Placeholder: 500x500" preserveAspectRatio="xMidYMid slice"
focusable="false"><title>Placeholder</title><rect width="100%" height="100%"

```



```

fill="#eee"/><text x="50%" y="50%" fill="#aaa" dy=".3em">500x500</text></svg>

    </div>
</div>

<hr class="featurette-divider">

<!-- /END THE FEATURETTES -->

</div><!-- /.container -->

<!-- FOOTER -->

<footer class="container">
  <p class="float-end"><a href="#">Back to top</a></p>

</footer>

</main>

```

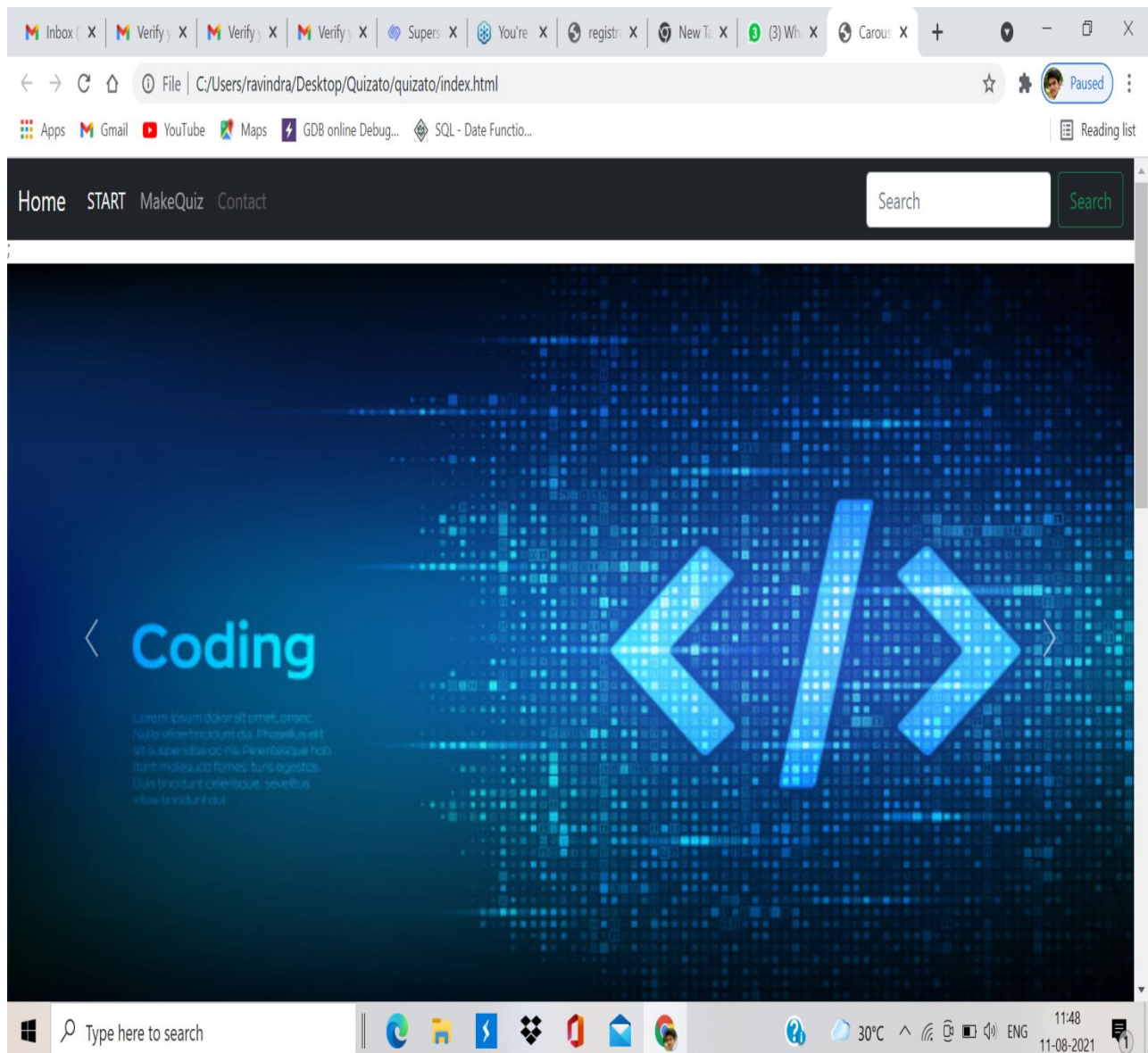


Fig 8.3.1-3

You know that very first page that links to the rest of our application, acts as an introduction to our app, and provides a preview of what lies ahead to our application visitors. A home page also refers to the first page that appears upon opening a web browser. This is the home page of my project. A home page is a web-page of an application which serves the starting point of application. The home pages include navigation bar that provides links to different sections within the application. **shared Components** links to the Shared Components page. Shared components can display or be applied on any page within the application. **Utilities** links to the Utilities page. Use this page to monitor developer activity, view dashboards, run Advisor, and view numerous other reports.

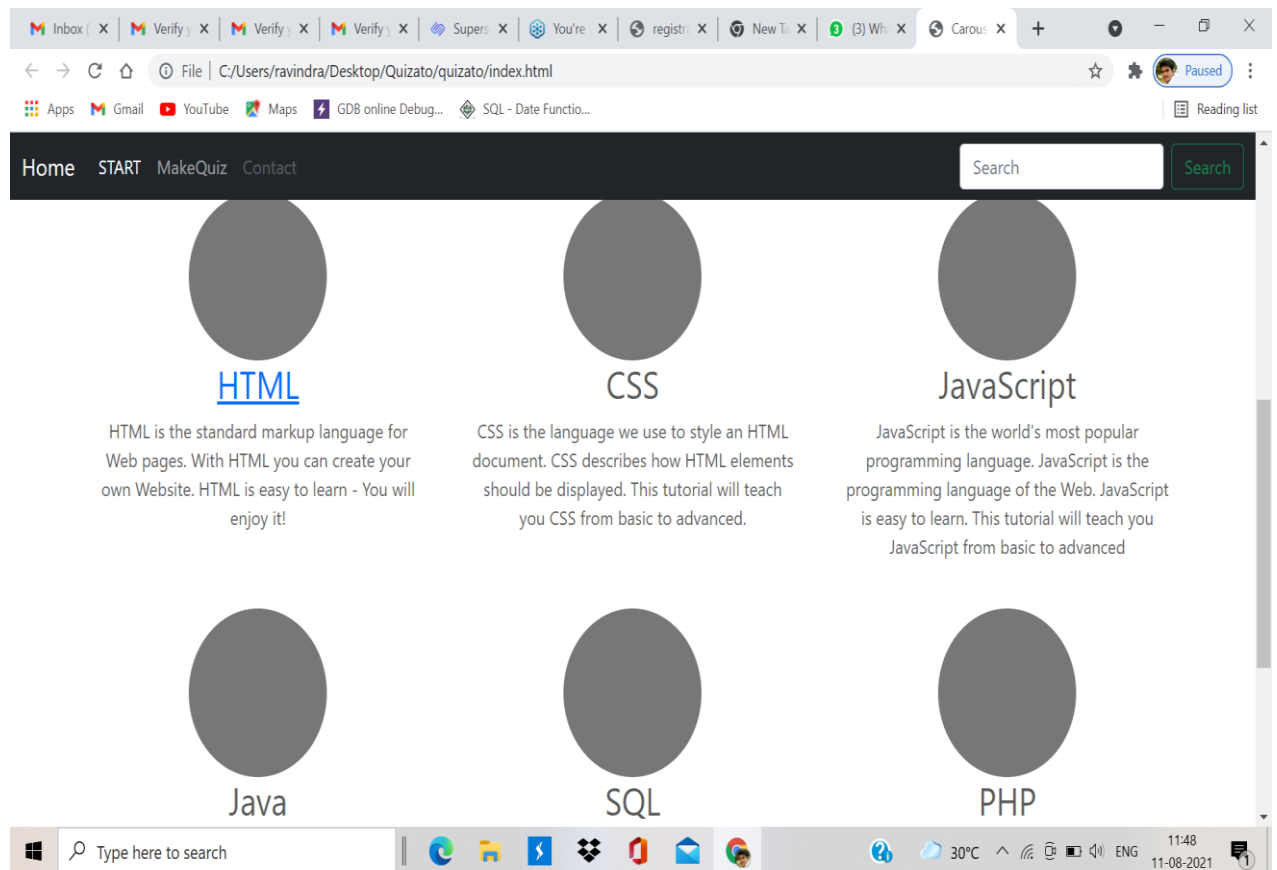


Fig 8.3.1-4

This is the page of my project. With the help of this page by clicking on different different icons we can go on which particular language's quiz. This page is also included in our home page. This pages provides a overview about that particulars related quizzes are available in the application. The purpose of using this page that is shows us which type of quizzes are available. User can choose according to their choice if user want to take the quiz which are based on HTML can take this by click on HTML and this process will follow with all like CSS, JavaScript , PHP and so on.

8.3.2 Quiz.html

```
<html>
<head>

  <link rel="stylesheet" href="style.css">

  <script type="text/javascript" src="js/jquery.js"></script>

  <script type="text/javascript" src="js/quiz-1.js"></script>

  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css" integrity="sha384-TX8t27EcRE3e/ihU7zmQxVncDAy5uIKz4rEkqIXeMed4M0jlfIDPvg6uqKI2xXr2" crossorigin="anonymous">

  <style>

ol {
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;
  background-color: rgb(95, 104, 224);
}

li {
  float: left;
}

li a {
  display: block;
  color: white;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
}

li a:hover {
  background-color: rgb(139, 194, 139);
```

```

}

</style>
</head>

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

  <a class="navbar-brand" href=" ../index.html">Home</a>

  <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-
expanded="false" aria-label="Toggle navigation">

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

  </button>

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

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

  <li class="nav-item active">

    <a class="nav-link active" aria-
current="page" href=" ../QUIZATO/quiz.html">START</a>

  </li>
  <li class="nav-item">

    <a class="nav-link" href=" ../QUIZATO/post.html">MakeQuiz</a>

  </li>

  <li class="nav-item dropdown">

    <a class="nav-link dropdown-
toggle" href="#" id="navbarDropdown" role="button" data-toggle="dropdown" aria-
haspopup="true" aria-expanded="false"> </a>

    <div class="dropdown-menu" aria-labelledby="navbarDropdown">

      <a class="dropdown-item" href="#">Action</a>

      <a class="dropdown-item" href="#">Another action</a>

      <div class="dropdown-divider"></div>

      <a class="dropdown-item" href="#">Something else here</a>

```

```

    </div>

    </li>
    <li class="nav-item">

        <a class="nav-link disabled" href="#" tabindex="-1" aria-
disabled="true">Disabled</a>

    </li>
</ul>

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

    <input class="form-control mr-sm-2" type="search" placeholder="Search" aria-
label="Search">

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

</form>
</div>
</nav>

<!--Marque runnung quizato-->

<center>

<div>

<marquee style="margin-
top: 0%;" direction="" height="80" width="400" bgcolor="#F0FFFF">

    <h1 style="color: #ff3333; margin-top: 5%;"> Quizato</h1>

</marquee>

</center>
</div>

<center>

<!-- !question body -->

<div id="ques" style="margin-top: 0; background-color: rgb(192, 238, 238);">

    <!-- questin and answee fetch here -->

</div>

<hr>

```

```

<center>

    <div id="">

        <div class="btn btn-primary " id="submit">submit</div>

    </div>
    <br><br><br><br>
</center>
</div>
</center>

<!--Answer-->

<div id="ans" style="display: none;">
</div>

<script type="text/javascript" src="js/data.js"></script>
</body>
</html>

```

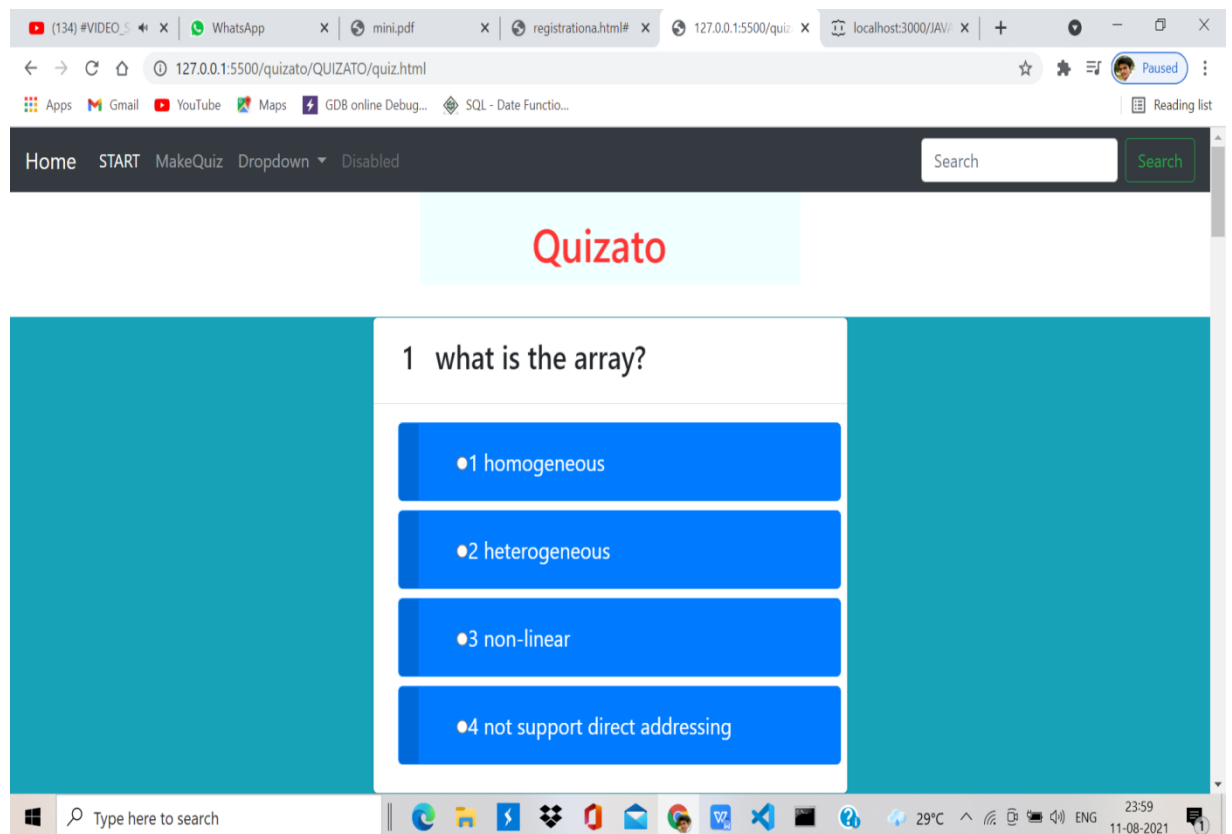


Fig 8.3.2-1

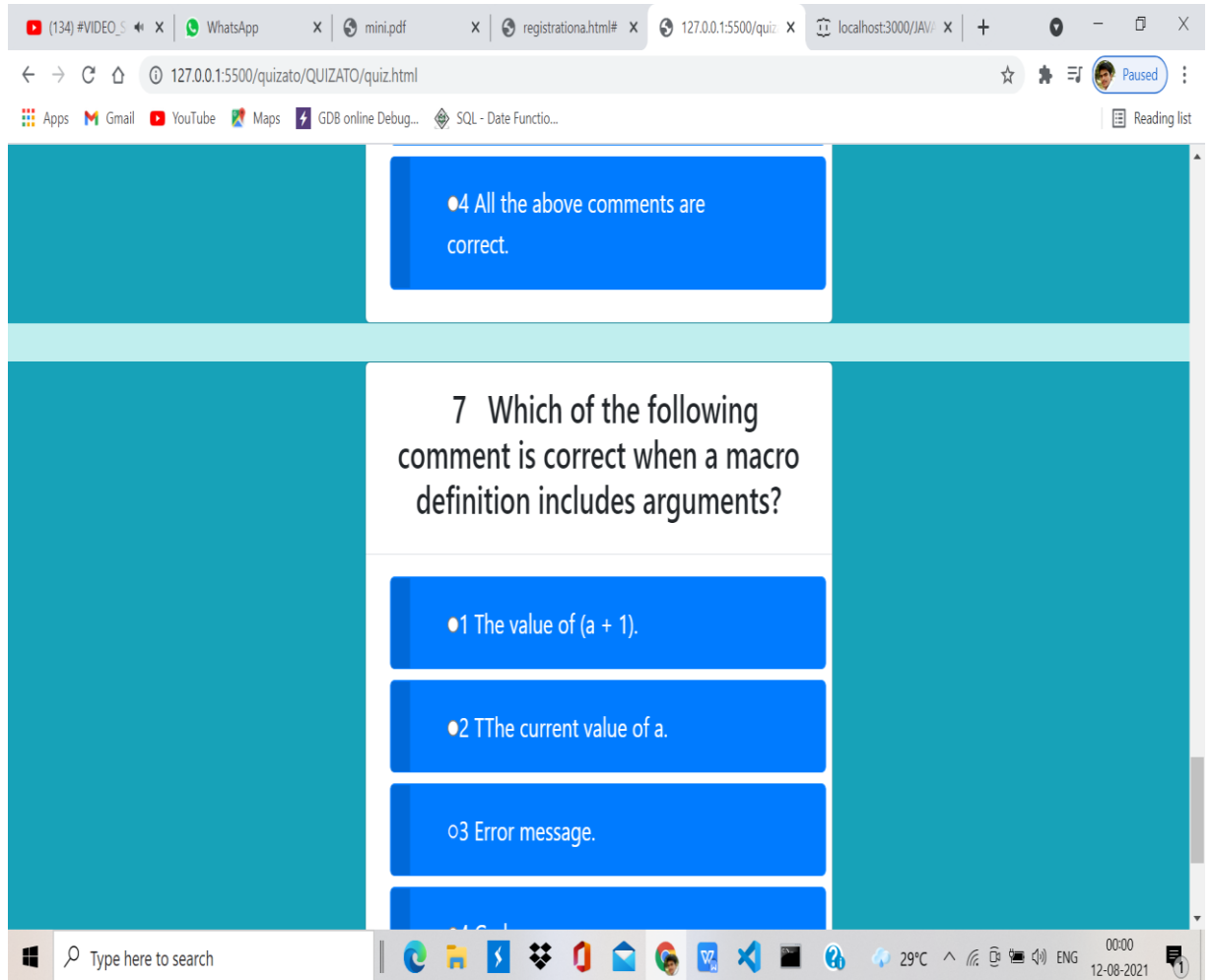


Fig 8.3.2-2

Fig 8.3.2-1 and Fig 8.3.2-2 is the snapshot of a quiz of my project. Question has four option. One option will be the correct. Question will show like this. We can choose any option. Like this there are many other questions available. In this format you get all the quizzes in this applications and when you choose any option then that option will show you like pop-up and you will be sure about you selected option. We have choose the blue colour because this colour evokes trust and confidence and blue colour is also favored by so many people so we have choose this blue colour layout for quiz.

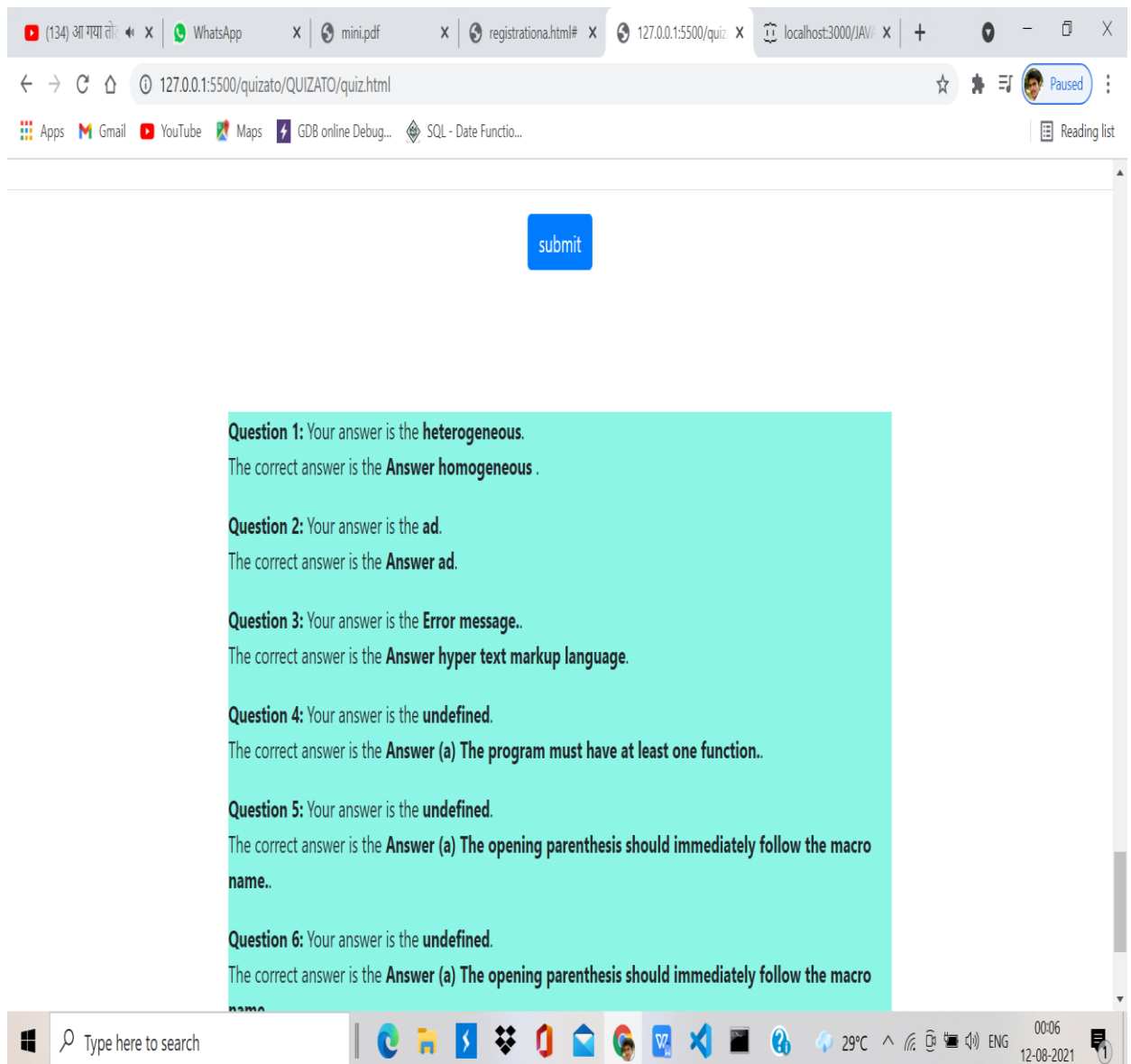


Fig 8.3.2.-3

This is the snapshot of the page after submission of quizzes. When we will submit our quiz by clicking on submit button then the your result and the correct answer of quizzes as well will show in this format. This page will help us to identify our mistakes and to provide correct answers as well. In some quiz we applications user get the answers on another page but in this web application user will the answers on same page after submission then can see the answer by scrolling down the page.

If we will not select any option before submit answer then it will display -
Your answer is **undefined**.

The correct answer will display -

The correct answer is 'the **Answer (option)**' display the answer.

8.3.3 Post.html

<head>

<meta charset="UTF-8">

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

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

<title>make quiz</title>

<meta name="viewport" content="width=device-width, initial-scale=1">

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta2/dist/css/bootstrap.min.css" rel="stylesheet"

integrity="sha384-BmbxuPwQa2lc/FVzBcNJ7UAyJxM6wquIlj61tLrc4wSX0szH/Ev+nYRRuWlolflfl" crossorigin="anonymous">

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta2/dist/js/bootstrap.bundle.min.js"

integrity="sha384-b5kHyXgcpbZJO/tY9U17kGkf1S0CWuKcCD38l8YkeH8z8QjE0GmW1gYU5S9FOnJ0"

crossorigin="anonymous"> </script>

</head>

```
<body>
```

```
<div class="container">
```

```
<!-- Just an image -->
```

```
<nav class="navbar navbar-dark bg-dark">
```

```
<div class="container">
```

```
<a class="navbar-brand" href="quiz.html">
```

```

```

```
<button class="btn">
```

```
show quiz <span class="badge badge-primary">
```

```
</span>
```

```
</a>
```

```
<h1>Drafting the</h1>
```

```
</div>
```

```
</nav>
```

```
<div class="card col-md-8 mx-auto gap-3 ">
```

```
<div class="form-group my-2">
```

```
<label for="exampleFormControlTextarea1">Question</label>
```

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

```
</div>
```

```
<div class="input-group mb-3">
```

```
<div class="input-group-prepend">
```

```
<span class="input-group-text" id="basic-addon3">option 1</span>
```

```
</div>
```

```
<input type="text" class="form-control" id="option-1" aria-describedby="basic-addon3">
```

```
</div>
```

```
<div class="input-group mb-3">
```

```
<div class="input-group-prepend">
```

```
<span class="input-group-text" id="basic-addon3">option 2</span>
```

```
</div>
```

```
<input type="text" class="form-control" id="option-2" aria-describedby="basic-addon3">
```

</div>

<div class="input-group mb-3">

<div class="input-group-prepend">

option 3

</div>

<input type="text" class="form-control" id="option-3" aria-describedby="basic-addon3">

</div>

<div class="input-group mb-3">

<div class="input-group-prepend">

option 4

</div>

<input type="text" class="form-control" id="option-4" aria-describedby="basic-addon3">

</div>

<div class="input-group mb-3">

<div class="input-group-prepend">

correct ans:

</div>

<input type="text" class="form-control" id="ans" aria-describedby="basic-addon3">

</div>

<div id="postBtn" class="btn btn-primary">Post Your Question</div>

</div>

</div>

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

</body>

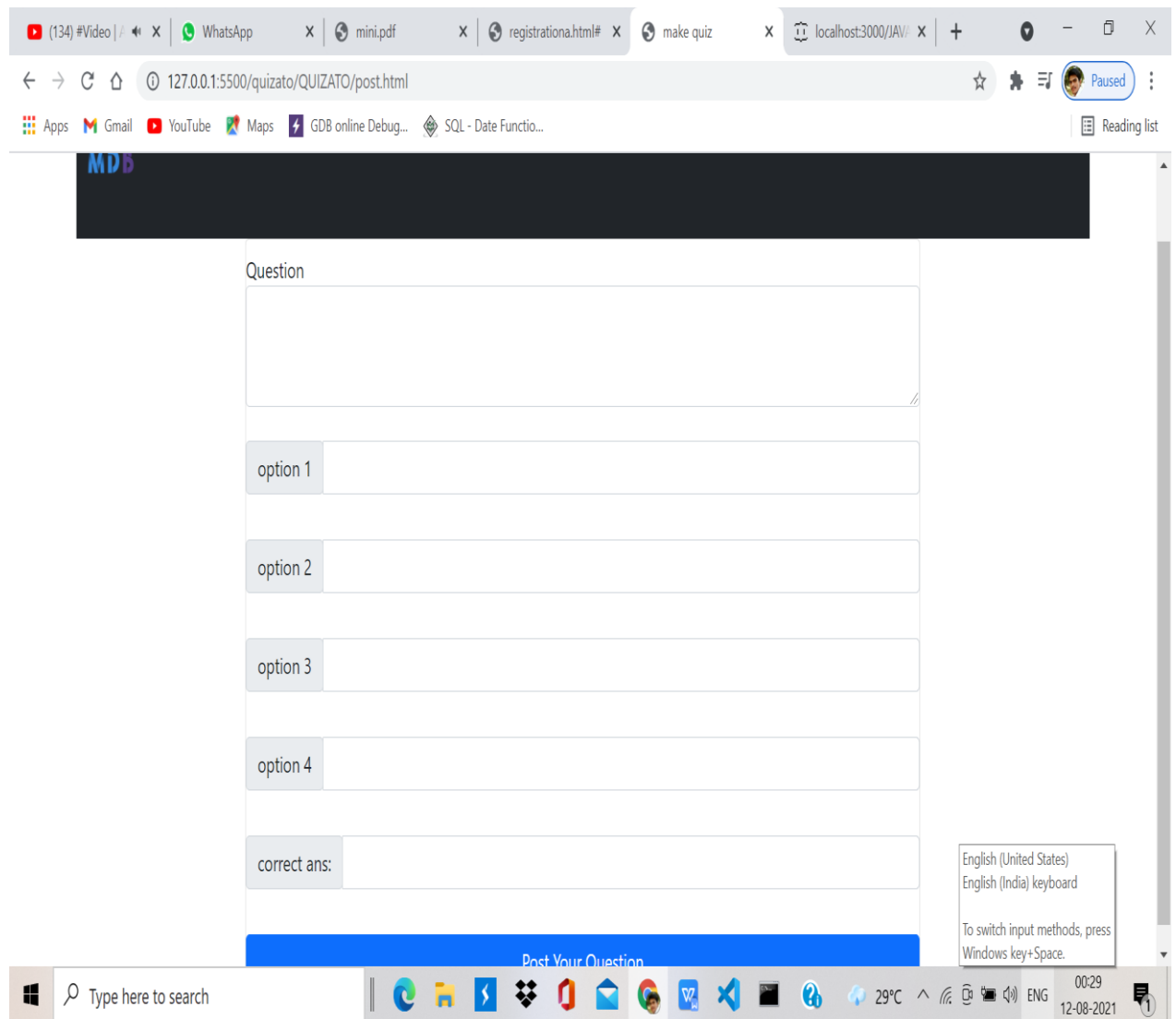


Fig 8.3.3

This page is the snapshot of post page. Post page means that we can post or insert the quiz using this page. We can see this there is one box for question where we can post or insert our quiz and four different boxes are for options and in the last there is one another box which is for correct answer here we will write correct answer of the quiz.

This section will be for teachers and Admin for Post or reply of any query. The correct ans section will be disable for student so that only either teacher or admin can response of any query.

CHAPTER 9 : Advantage

- **It saves the student's money:**

There are so many Platform on internet which charged for Quiz test. Before taking test user have to buy their courses and after that they be able to attend quiz but here no need of buy any course and no any investment by which user have to spent money.

- **It saves more time:**

This online web quiz application will helps to save user's time Because there will not be any instruction for taking test. It will be user friendly so user can move easily without any instruction by this user can save their time.

- **No instructor needed:**

There are no need of instructor for users because our web-application will be user friendly and user can easily take test.

- **It saves paper.**

If there conducting offline Exam then we have to spent paper for providing questions to students and for taking response, but it is totally paperless.

- **User can manage exam time according their requirement:**

Sometimes some questions take less time and some questions takes more time to think and solve ,so user can set the timer of quizzes according to their requirement.

- **User can test themselves anytime any where:**

Here will not be any instruction or specific time for attend Quiz therefor He can gave quiz anytime anywhere and according their marks he can test themselves anytime anywhere.

CHAPTER 10 : Conclusion

A large number of participants, with instant results of our online quiz (for the creator as well as the participants), a better overview, user are able to randomize their questions and set a timer. That all without the need of an instructor. What's holding user back to not use online quizzes?

User can manage exam time according their requirement. User can test themselves anytime any where. User can attempt quiz also anytime anywhere.

A large number participants can participate. This web application will be user friendly.

BIBLIOGRAPHY

1- Github: <https://github.com/ravindra-2510/mini-Quizato>

2- Bootstrap: <https://getbootstrap.com/>