

QUIZZES: For Free

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Science degree in Computer Science

By

DILIP BABU.D (Reg. No:39290019)



**DEPARTMENT OF COMPUTER SCIENCE
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

MARCH – 2022



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **DILIP BABU.D (39290019)**

Who carried out the project entitled **"QUIZZES: For Free"** under my supervision

From **MARCH to APRIL**.

Internal Guide

(Name in Capital letters with signature)

Dr. MAHESWERI,

Head of the Department

Submitted for Viva-voce Examination held on_____

Internal Examiner

External Examiner

DECLARATION

I **DILIP BABU.D (39290019)** hereby declare that the Project Report entitled
“QUIZZES: For Free” was Done by me under the guidance of **Ms. MAHESWERI**
is submitted in partial fulfillment of the requirements for the award of a
Bachelor of Science degree in Computer Science.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE:

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to the **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D., Dean**, School of Computing **Dr. L. Lakshmanan M.E., Ph.D.** , and **Dr.S.Vigneshwari M.E., Ph.D. Heads** of the Department of Computer Science and Engineering for providing me with necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide

Ms. Maheswari for her valuable guidance, suggestions, actions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT:

The “**QUIZZES: For Free**” project is developed to overcome the time-consuming problem of f manual system. Apart from that in the current system, checking the answer sheets after taking a test is a waste of the examiner’s time.

This application will display the correct answer and save the user time therefore, he and only he will be able to know the outcome of the Quiz. The users which are using this Web App don’t need to have high computing knowledge.

Online Quizzes are used widely around the world. Users can use it to gain knowledge about the topic and break the fear barrier on those topics. Since Quizzes Contain questions that are widely searched/ asked questions gain a clear idea of what the topic is about.

TABLE OF CONTENT

Chapter No.	Title	Page No.
	ABSTRACT	v
	LIST OF FIGURES	viii
1	INTRODUCTION	1
2	LITERATURE SURVEY	2
3	AIM AND SCOPE OF PRESENT INVESTIGATION	7
	3.1 AIM OF THE PROJECT	7
	3.2 SCOPE AND OBJECTIVE	7
	3.3 SYSTEM REQUIREMENTS	9
	3.3.1 HARDWARE REQUIREMENTS	9
	3.3.2 SOFTWARE REQUIREMENTS	9
	3.4 SOFTWARES USED	10
	3.4.1 HTML	10
	3.4.2 HTML CHARACTERISTICS	11
	3.4.3 APPLICATIONS OF HTML	11
	3.4.4 CSS	14
	3.4.5 CSS CHARACTERISTICS	15
	3.4.6 APPLICATIONS OF CSS	16
	3.4.7 VS CODE	17

4	EXPERIMENTAL OR MATERIAL METHODS	19
	4.1 DESIGN METHODOLOGY	19
	4.1.1 EXISTING SYSTEM	19
	4.1.2 PROPOSED SYSTEM	19
	4.2 MODULE DESCRIPTION	20
	4.3 ARCHITECTURE DIAGRAM	21
5	RESULTS AND PERFORMANCE ANALYSIS	
	5.1. LOADING UP THE HOME PAGE	22
	5.2 LOADING UP THE GAME PAGE QUIZ	23
	5.3 END PAGE	24
6	CONCLUSION AND FUTURE ENHANCEMENT	
	6.1 CONCLUSION	25
	6.2 FUTURE ENHANCEMENTS	25
	REFERENCES	
	APPENDIX	
	A. SOURCE CODE	

LIST OF FIGURES:

FIGURE NO	FIGURE NAME	PAGE NO
4.1	ARCHITECTURE DIAGRAM	21
5.1	HOME PAGE	22
5.2	MAIN QUIZ PAGE	23
5.3	POP UP FOR A WRONG ANSWER	23
5.4	END PAGE	24

CHAPTER-1 INTRODUCTION:

This project aims to computerize the existing system and help the users to save valuable time. Apart from this, data that exist in this system will exist for a long time and will be easily accessible. This project aims to provide a better service to its users.

The performance of the application will be faster since it is a web-based application. The project will reduce the time wasted by students for searching every topic and its related quiz over the internet.

. Functionalities of the project **(excluding the further ideas to be executed shortly)** will be as followed:

1. Home Page
2. Main Game Page
3. High-score/End Page

Users can see their results after finishing the quiz immediately. To conclude the introduction, I have used these modules to follow the objectives of my project and each of these modules has logical connections to other modules on which they depend:

- i) Loading the quiz (Main Page).
- ii) Fetching API (Game).
- iii) Storing the High score values Locally.
- i) Loading up the high score. (Data is erased after exiting the Webpage)

CHAPTER-2 LITERATURE SURVEY

1. Some Trends in Web Application Development

Author: Mehdi Jazayeri, 2007 Future of Software Engineering

A Web application is an application that is invoked with a Web browser over the Internet. Ever since 1994 when the Internet became available to the public and especially in 1995 when the World Wide Web put a usable face on the Internet, the Internet has become a platform of choice for a large number of ever-more sophisticated and innovative Web applications. In just one decade, the Web has evolved from a repository of pages used primarily to access static, mostly scientific, information to a powerful platform for application development and deployment. New Web technologies, languages, and methodologies make it possible to create dynamic applications that represent a new model of cooperation and collaboration among large numbers of users.

2. Web design — Past, present, and future

Author: Haidi Božiković, Maja Štula, 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)

Nowadays, there is a wide prevalence of various devices that can be connected online for different kinds of activities and for browsing web pages. Web design is constantly changing and adjusting due to the appearance of different sizes

and types of devices. Web design from the 1990s is greatly different from current web pages. The main goal of the design is not the design itself, but to enable web content transfer in a more readable and comprehensible way, no matter which device is used to present web content, desktop or hybrid computer, various mobile or wearable device. This paper shows the development of web design from its beginning to nowadays modern design, and what we can expect in the future. Today's average day for any person involves browsing web pages searching for various information and news, connecting to other users through many social networks, and doing various business tasks. According to a user survey conducted in the United Kingdom in 2016, even 71% of users have accessed the internet using a smartphone, 62% used a laptop, while only 40% used a desktop, which implies that smartphones have taken up most of the market and internet access.

3. The application of web-based formative quizzes in laboratory teaching in the higher education environment

Author: Jasna Gamulin, Ozren Gamulin, 2012 Proceedings of the 35th International Convention MIPRO

In this paper, the authors have presented the results of the application of web-based formative quizzes in physics laboratory teaching at biomedical university study programs. The web application Pitalica has been developed to enable quizzes for all students at the beginning of each laboratory class. The web-based quizzes are questioning the underlying theory of the specific practical assignment that the student is about to work off. The web-based quizzes have been performed during the academic years 2010 and 2011. During the year 2010, quizzes were strictly formative i.e. the principal characteristic of the examination was to provide the appropriate feedback for constructing knowledge. In 2011, students were informed that aside from the formative role, the total result achieved through quizzes would

also bring 20% of the final exam grade. This paper is investigating the influence of formative web-based quizzes on the final practical exam results when the two described approaches were applied. At the end of the course, a survey has been conducted, questioning self-perceived ease of use, motivation for better preparation, competencies acquired, quizzes preferences, ICT literacy, and gender.

4. The application of web-based formative quizzes in laboratory teaching in a higher education environment

Author: Jasna Gamulin, Ozren Gamulin, 2012 (May) Proceedings of the 35th International Convention MIPRO

In this paper, the authors have presented the results of the application of web-based formative quizzes in physics laboratory teaching at biomedical university study programs. The web application Pitalica has been developed to enable quizzes for all students at the beginning of each laboratory class. The web-based quizzes are questioning the underlying theory of the specific practical assignment that the student is about to work off. The web-based quizzes have been performed during academic the years 2010 and 2011. During the year 2010, quizzes were strictly formative i.e. the principal characteristic of the examination was to provide the appropriate feedback for constructing knowledge in ring 2011, students were informed that aside from the formative role, the total result achieved through quizzes would also bring 20% of the final exam grade. This paper is investigating the influence of formative web-based quizzes on the final practical exam results when the two described approaches were applied. At the end of the course, a survey has been conducted, questioning self-perceived ease of use, motivation for better preparation, competencies acquired, quizzes preferences, ICT literacy, and gender.

5. Multipurpose Use of Quizzes in Teaching

**Author: Irene Rothe, Willi Pustowalow, Marco Winzker,
2021 IEEE Global Engineering Education Conference
(EDUCON)**

This paper describes the use of quizzes in different learning scenarios. Using quizzes stimulates the student's cognitive understanding as described in many sources in the literature. Via quizzes students will not be only trained in keeping knowledge, they are also coached in analytical thinking. By the usage of quizzes a kind of “learning on demand” is possible which is entertaining and prepares students well for their examinations. Several types of quizzes exist, like single-choice, multiple-choice, numeric, and text fill-in, and the main task for lecturers is formulating questions and exercises. To make better use of this conceptual and creative work, it is beneficial to use questions for more than one activity during the learning process for the students. However, these usages require different tools with different formats for stating a question, possible answers, and specifying one or more correct answers. Thus, two software programs were implemented, a Windows tool to convert questions for quizzes into different formats, and an app for mobile offline usage on a smartphone. The paper describes the usage of quiz questions in different classrooms, and e-learning settings, including lectures, online quizzes on websites, smartphones, learning management systems (LMS), and examinations. The educational setting includes courses and examinations offered to bachelor engineering students in the first year of their study as well as master's students. Evaluation is performed by observing usage on the platforms, looking at the total number of usages as well as the time when students use the quizzes, i.e. during the semester or before examinations. Also, exam results for the questions have been looked at and give indications about the learning success of students. At the moment, the software is a prototype and proof of concept. However, it is not safe against errors in the input format and only supports the open-source LMS (Learning Management System) ILIAS. More work is required to make it usable for further lecturers. Also, the mobile app is only available for Android and it could be converted into a framework for releasing Android and iOS versions (for Apple iPhone).

6. Behavior model construction for the client-side of modern web applications.

Author: Weiwei Wang, Junxia Guo, Zheng Li, Ruilian Zhao, Tsinghua Science and Technology (Volume: 26, Issue: 1, Feb. 2021)

Most of the behavior models concerning Web applications focus on the sequencing of events, without regard for the changes of parameters or elements and the relationship between trigger conditions of events and Web pages. As a result, these models are not sufficient to effectively represent the dynamic behavior of the Web 2.0 application. Therefore, in this paper, to appropriately describe the dynamic behavior of the client-side of Web applications, we define a novel Client-side Behavior Model (CBM) for Web applications and present a user behavior trace-based modeling method to automatically generate and optimize CBMs. To verify the effectiveness of our method, we conduct a series of experiments on six Web applications according to three types of user behavior traces. The experimental results show that our modeling method can construct CBMs automatically and effectively, and the CBMs built are more precise to represent the dynamic behavior of Web applications.

CHAPTER 3

3. AIM AND SCOPE OF PRESENT INVESTIGATION

3.1 AIM OF THE PROJECT

This project aims to develop a Web Application that makes user improve their Knowledge on the topic they wish to learn and to create Random quizzes in the Web Application for the users.

The Motto of this Project is **“For the Users by the Users”**. To make users ready to learn without the need of an instructor, since the results are shared instantly, the user can attend multiple quizzes in a short time.

3.2 SCOPE AND OBJECTIVE:

- **Bring your audience into the conversation:**

Connect with your audience in a unique and fun way by engaging them in a unique and fun way. You may increase customer loyalty by hosting a repeated event.

- **Huge Numbers:**

A large number of people Taking online quizzes allows for a big number of people to participate. It might have up to 1000 participants (and possibly even more!). It makes no difference where they take the online quiz as long as they have access to the Internet.

- **Creating a random set of questions:**

It's far easier to randomly generate your question with only one click than it is to do it all manually. With online quizzes, randomizing questions and even the answers to those questions is a breeze. Aside from that, it not only saves time but also helps to avoid cheating by pupils.

- **Results of the Quiz/Audience Information:**

Another advantage of online quizzes is that you may obtain instant feedback from your respondents. The findings are not just available to the creator, but also participants. This makes identifying knowledge gaps simple for the designer. Participants may see the sections they need to work on.

3.3 SYSTEM REQUIREMENTS:

3.3.1 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. The minimal hardware requirements are as follows,

1. **Processor:** Pentium IV or Better
2. **RAM:** 3 GB or More
3. **Storage Space:** 25 GB or More

3.3.2 Software Requirements:

Software requirements deal with defining resource requirements and Prerequisites that need to be installed on a computer to provide a functioning Of an application. The minimal software requirements are as follows,

1. **Front-End:** HTML, CSS
2. **IDE:** Visual Studio Code
3. **Operating System:** Windows 7 or Later

3.4 SOFTWARES USED:

3.4.1 Hypertext Mark-up Language (HTML):

Sir Tim Berners-Lee designed HTML in late 1991 but was never publicly released. HTML 2.0 was first released in 1995. HTML 4.01 was a significant version of HTML that was released in late 1999.

HTML is a dynamic markup language that has changed over time as different versions have been released. Each version has allowed its user to construct web pages in a much easier and more beautiful method, as well as making sites highly efficient, long before its new standards and specifications are implemented.

HTML 1.0 was introduced in 1993 to distribute legible and accessible content via web browsers. However, only a small percentage of the developers worked on websites. As a result, the language did not develop.

Then there's HTML 2.0, which was released in 1995 and includes all of the characteristics of HTML 1.0 plus a few new ones. It was the official markup language for designing and developing websites until January 1997, and it enhanced numerous basic elements of HTML.

Then there's HTML 3.0, which saw Dave Raggett present a new paper or draught on HTML. It contained enhanced HTML elements that provided webmasters with more powerful qualities while developing websites. However, websites with HTML's sophisticated capabilities delayed the browser's ability to make additional advancements.

Then there's HTML 4.01, which is commonly used and was a successful version of HTML before HTML 5.0, which is now available and widely utilized.

3.4.2 HTML Characteristics:

- It is the language that can be easily understood and can be modified.
- It provides a more flexible way to design web pages along with the text.
- Links can also be added to the web pages so it helps the readers to browse the information of their interest.
- You can display HTML documents on any platform such as Macintosh, Windows, Linux, etc.
- Graphics, videos, and sounds can also be added to the web pages which give an extra attractive look to your web pages.

3.4.3 Applications of HTML:

Web pages development:

HTML is heavily used for creating pages that are displayed on the World Wide Web. Every page contains a set of HTML tags, including hyperlinks that are used for connecting to other pages. Every page that we witness on the World Wide Web is written using a version of HTML code.

Web document Creation:

Document creation on the internet is dominated by HTML and its basic concept via tag and DOM, i.e. document object model. HTML tags are inserted before and afterward or phrases to locate their format and location on the page. A web document consists of three sections: title, head, and body. Head includes the information to

identify the document, including the title and any other important keywords. A title can be seen on the browser's bar, and the body section is the main portion of the website visible to the viewer. All three segments are designed and created by the use of HTML tags. Every section has its own specific set of tags, which are dedicatedly rendered, keeping the head, title, and body concepts in a loop.

Internet navigation:

This is one of the most important uses of HTML, which is revolutionary. This navigation is possible by utilizing the concept of Hypertext. It is a text which refers to other web pages or text, and when the user clicks on it, would navigate to referenced text or page. HTML is heavily used to embed the hyperlink within the web pages. A user can easily navigate the web pages and between websites as well, located on different servers.

Cutting edge features:

HTML5, with its set of standards and API, is being used to introduce some of the latest trends in the website creation business. Like polyfill libraries, which are supported by old browsers equally well. Browser like Google Chrome is the perfect choice when it comes to implementing an HTML5 latest set of standards and APIs. A JavaScript library is available called Modernizer, which can detect features that let the developer dynamically load polyfill libraries as required.

Responsive images on web pages:

At the elementary level in applications of HTML, queries can be set to utilize responsive images. With the srcset attribute of the IMG element in HTML and combining it with picture elements, a developer can fully control how the user will render an image. Now different types of an image with size variation can be loaded by using the img element. Rules can be easily set with the picture element; we can declare the img element with the default source, and then for every case, a source can be provided.

Client-side storage:

Earlier, a user could not save the user's browser data that would persist across sessions. To meet this requirement, server-side infrastructure has to be built, or user's cookies can be used. But with HTML5, client-side storage is feasible using localStorage and IndexedDB. These two strategies have their standards and features. LocalStorage gives string-based hash-table storage. Its API is very simple and provides the developer with set Item, get Item, and remove Item methods. IndexedDB, on the other hand, is a larger and better client-side data store. IndexedDB database can be expanded with the user's permission.

Offline capabilities usage:

Once data can be stored in the browser, the developer can think of a strategy to make the application work when a user is disconnected. HTML5 has its application cache mechanism, which would define how the browser manages the offline situation. Application cache, responsible for offline ability, comprises different components, including API methods that create an update, read the manifest file, and events. By using a certain property in HTML5, a developer can check if the application is online or not. A developer can also specify the information in the website's application cache manifest file, like what browser manages resources for offline use. In the manifest file, resources that are available offline can also be specified.

Data Entry support with HTML:

HTML5 standard and set of APIs can be used to support the data entry level of work. As browsers implement new HTML5 standards, developers can simply add the tags to the tag, indicating required fields, text, data format, etc. HTML5 has come up with several new attributes to drive on-screen keyboards, validation, and other data-entry experiences so that end-user can have a better data entry.

Game development usage:

Before the advent of HTML5, game development was an exclusive domain of Flash and Silverlight. Since browsers support new specifications for HTML5, including CSS3 and a light-fast JavaScript engine to drive a new rich experience, HTML5 can bring the reality of game development possible, which was earlier the forte of Flash and Silverlight. Every feature of APIs needs not to be implemented, but the most appropriate ones can be utilized while eliminating the rest of the features.

Native APIs usage to enrich a website:

HTML5 adds so many new abilities and tools, which were just an imagination in the past. A large set of new APIs regarding file system, Geolocation, drag and drop, event handling, client-storage, etc., are the capabilities that make usage of HTML5 easier than ever before. Application experience can be enhanced with other APIs like Full screen, Visibility, and Media Capture. A modern web application has asynchronous nature, which can be fostered using Web sockets and Web workers like APIs.

3.4.4 Cascading Style Sheets (CSS):

According to Wikipedia, the birth of CSS is largely credited to Norwegian Håkon Wium Lie, who back in **1994** sought to create a universal standardized style sheet for the World Wide Web.

The first site that Lie trailed CSS on was the Arena web browser. From its first creation, Lie went on to co-create CSS1, CSS2, and RFC 2318 versions with Tim Berners-Lee and Robert Cailliau. In its first decade of existence (1994 – 2004) CSS, in all its specifications, became an established web standard greatly influencing the look and accessibility of the World Wide Web as we know it today. **CSS3 was issued in 1999.**

Web standards are a topic that is close to Lie's heart. Since delivering CSS he has petitioned major technology players such as Microsoft and other browsers to support common web standards and continues to develop CSS's uses for web printing and pagination on screens.

CSS (1996) allows the user to select font style and size and change the color of the text and background.

CSS2 (1998) has capabilities that allow the user to design page layout.

CSS3 (1999) allows the user to create presentations from documents and to select from a wider range of fonts including those from Google and Typecast. Uniquely, CSS3 allows the user to incorporate rounded borders and use multiple columns. CSS3 is considered to be easier to use (when compared to CSS2) because it has different modules

CSS is one of three cornerstone technologies used on the web (the other two are HTML and JavaScript). CSS stands for Cascading Style Sheets – the clues are really in the words 'cascading' and 'style' with cascading describing the way that one style can cascade from one to another.

One of the many benefits of CSS is that more than one style can be used within one HTML document.

CSS is used as a way of defining how HTML code is going to look on a website. Whereas HTML (Hypertext Markup Language) is used to create content, including written text, CSS alters the way a web page will look.

So, depending on the data they want to display, a developer might choose to have a page with tabs running across the top of the page, or along the side.

Or, another developer might choose to use headings and subheading styles to ensure that the words leap off the page or, change or revamp an existing webpage entirely.

Perhaps the best way of describing what CSS does is to explain what a page would like if it didn't use CSS.

Without CSS web pages are plain and far from inspirational. Words scroll across the whole page and are difficult to read. But, before CSS that's exactly what web pages looked like.

The introduction of CSS is partly responsible for how the web looks and feels today. And, far from being created and therefore finished, it is a continually evolving language.

3.4.5 CSS Characteristics:

- A selector component plus a declaration block component make up a style rule.
- The selector is used to indicate which HTML component should be styled.
- One or more declarations, as well as semicolons, are contained within the declaration block.
- A CSS property name, a semicolon, and a value are included in each declaration. Color is an example of a property, and the value is red. The attribute is font size, and the value is 15px.
- These blocks are enclosed by curly braces, and the CSS declaration concludes with a semicolon.

- CSS selectors are used to locating HTML components based on their name, id, attribute, class, and other attributes.
- The ID of an element will choose one unique element.
- The # function, together with the id property, should be used to choose a certain element with a given id.
- If you want to select items with a certain class, use the period character followed by the name class.
- If you don't want to choose elements of a certain kind, the universal selector merely matches the element name.
- Element selector: These selectors choose an element based on its name.
- Descendant selector: When one element is contained within another, it is referred to as a descendent selector.
- ID Selector: An element's ID will choose one unique element.
- To pick a specific element with a specified id, use the # function in conjunction with the id attribute.
- Use the period character followed by the term class to select objects with a certain class.
- The universal selector just matches the element name if you don't wish to choose items of a certain kind.
- Element selectors are selectors that chose elements based on their names.
- Descendant selector: A descendent selection is used when one element is enclosed within another.

3.4.6 Applications of CSS:

Inline:

An inline style sheet only affects the tag in which it is placed. This simply implies that little aspects of the website may be updated without affecting the page's general layout or anything on it. This is useful since if everything was on the external sites, you'd have to add extra tags to update details. Because inline takes precedence over external, tiny details can be adjusted. It also takes precedence over the internal.

Internal:

Only utilize the internal if you need to make a minor change to a single tag. This is because inline only affects the one tag that it contains, whereas internal styling is applied to the HTML document's head. This implies that if you want to customize the page, you can see all of the necessary adjustments simply by scrolling down. Inside the tags is where the internal styling is stored. Because of the distinct styling and labeling, this appears cleaner, simpler, more elegant, and more organized.

External:

People utilize external stylesheets to format and replicate their web pages on a whole other document. Because more than one stylesheet can be embedded inside the document, you can essentially have two or more workplaces, resulting in a much cleaner workspace. In this situation, the stylesheet would be immediately available, which is a major benefit, but any modifications made to the external sheet would affect all of the parent sheets it is linked to.

3.4.7 VS Code:

VS Code is a free, open-source streamlined cross-platform code editor with excellent support for Python code editing, IntelliSense, debugging, linting, version control, and more

VS Code is free for both private and commercial use, runs on Windows, macOS, and Linux, and includes support for linting, debugging, task running, version

VS Code is openly extensible and many extensions are available.

Visual Studio Code is a streamlined code editor with support for development operations like debugging, task running, and version control. It aims to provide just the tools a developer needs for a quick code-build-debug cycle and leaves more complex workflows to fuller featured IDEs, such as Visual Studio IDE.

Visual Studio Code has built-in support for editing style sheets in CSS .css, SCSS .scss and Less .less. In addition, you can install an extension for greater functionality.

Visual Studio Code provides basic support for HTML programming out of the box. There is syntax highlighting, smart completions with IntelliSense, and customizable formatting. VS Code also includes great Emmet support.

Visual Studio Code includes built-in JavaScript IntelliSense, debugging, formatting, code navigation, refactorings, and many other advanced language features.

Most of these features just work out of the box, while some may require basic configuration to get the best experience. This page summarizes the JavaScript features that VS Code ships with. Extensions from the VS Code Marketplace can augment or change most of these built-in features.

CHAPTER 4

4. EXPERIMENTAL OR MATERIAL METHODS

4.1 DESIGN METHODOLOGY

4.1.1 Existing System:

Various MCQ quiz applications on the internet with different criteria.

Each of the existing applications has town goodness and problems. This Web-application which is designed to overcome the existing problems with the following features:

- Better management;
- Connection to the database for better storing of data;
- Better frontend management;
- Better backend management;
- Try to decrease error issuer during runtime

4.1.2 Proposed Solution:

- Then Welcome screen will appear once the user opens the web page.
- The User has to choose any of the two options 'Play' 'High Score'.
- After Clicking on 'Start Game' multiple-choice questions will come automatically one by one.

- The user has to choose any of four existing options and each right answer will automatically count the score.
- At the end of, the game it will show the score of the user.

4.2 MODULE DESCRIPTION

Module 1: Main Page

This is the Home Page of the Web-based Quiz Application. It Complies with the Quote: **“First Impression is the Best Impression”** Since it has a nice Background, which is in constant motion with the help of CSS. Also, this application has Built-in audio, which makes sure the user doesn’t get bored so easily.

Module 2: Game Page

This page is loaded once the user has clicked the “Play” button on the Main Page. The page contains questions loaded from an Open Trivia Database. The number of questions per round and the score value for each correct answer can be adjusted within the game.js file. If the user gives a wrong answer to a question the default alert box will Indicate to the user that the answer is wrong and motivates the user to search for the answer to the question such that the user will forget it later.

Module 3: End Page

This page is loaded when the Quiz is over. This page has a name box to store the name of the user. The Save button is used to store the score value and the name of the user which can be accessed through the High score button on the Home page. There is a Play Again button that is used to Replay the Quiz. There is a Go Home Button which will take us to the Home Page where we can check the High Score of the User.

Module 4: High score Loader

This Module is Integrated with the Main Page and is accessed by the High score Button on the page. This module helps the user to view his high scores in each round. However, this module currently saves only till the web page is closed. So the High scores are displayed only until the webpage is open, and will be reset if refreshed

4.3 ARCHITECTURE DIAGRAM:

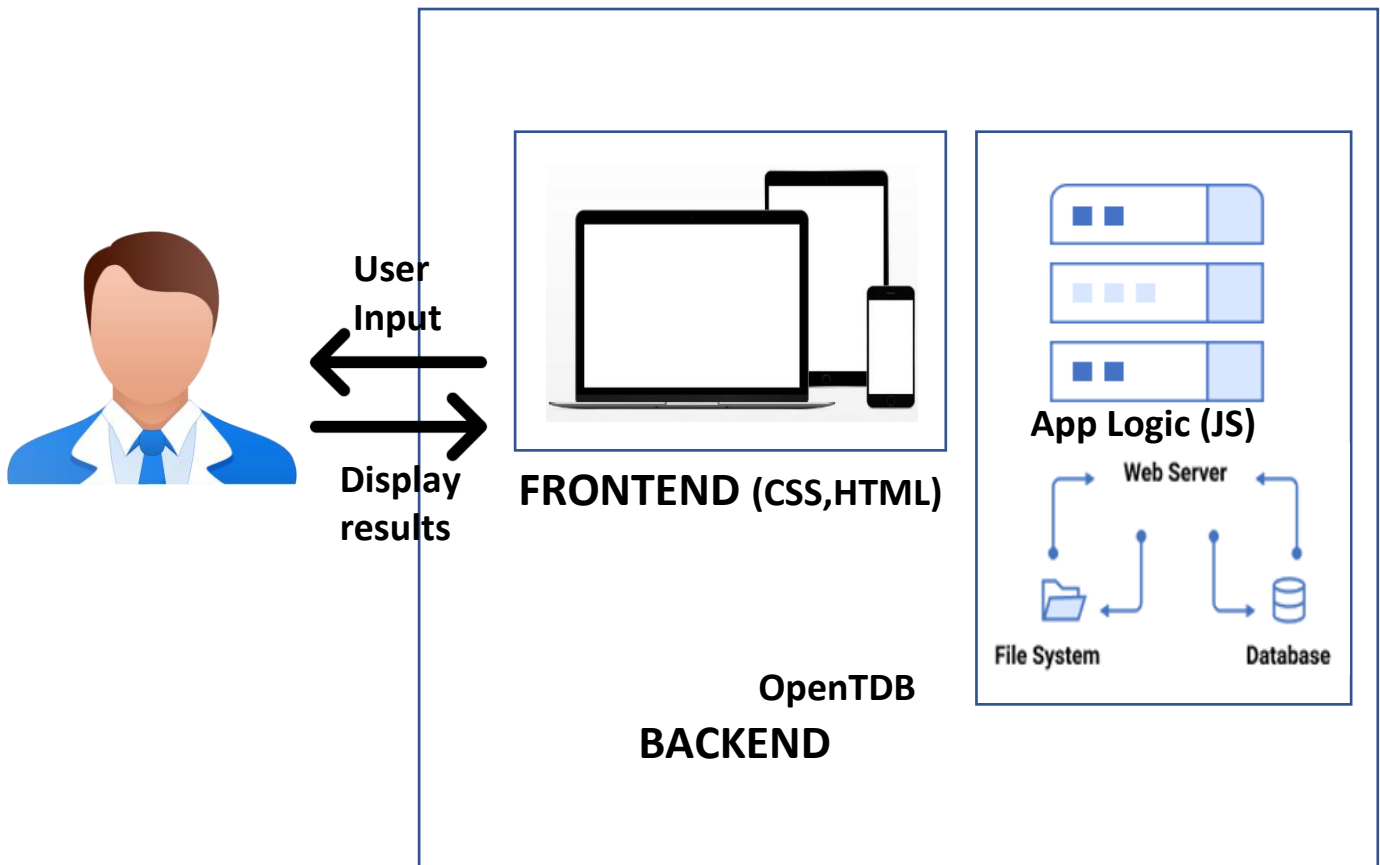


Fig 4.1: Architecture Diagram

CHAPTER 5

5. RESULTS AND PERFORMANCE ANALYSIS:

5.1. Loading up the Homepage:

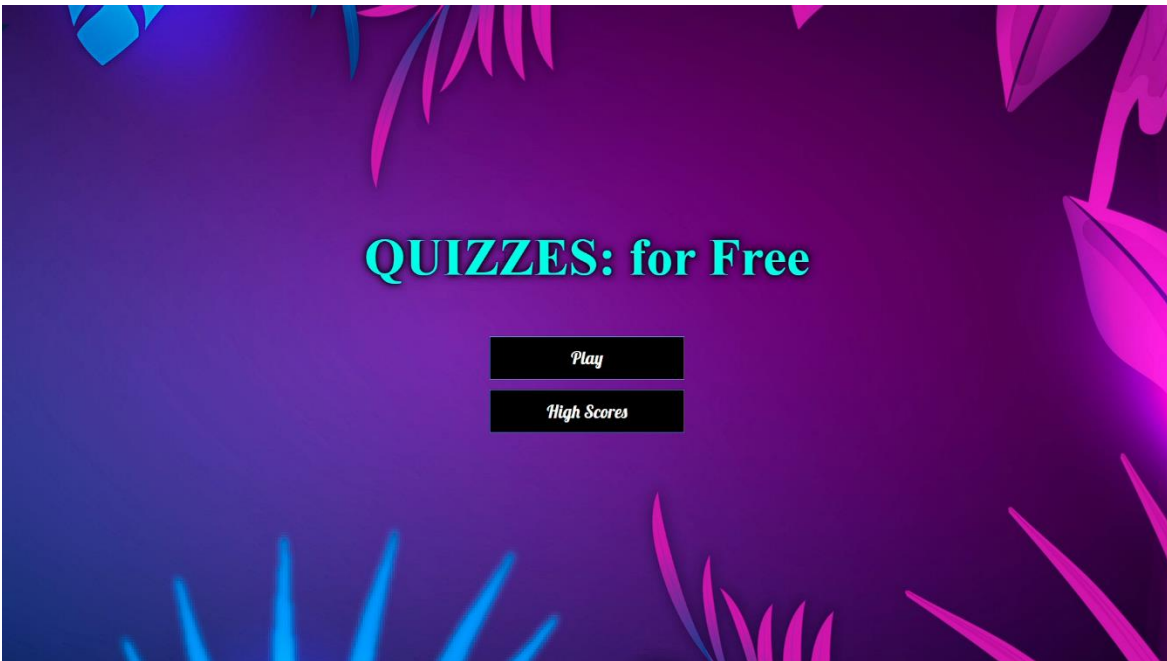


Fig 5.1: Home Page

This is the home page or the main page of my project. It is the web page Created using HTML. Here it has 2 buttons namely Play and High Scores. The User can click Play to start the quiz or CHECK THE High Scores If he had played it before.

5.2 Loading up the Game Page/Quiz:

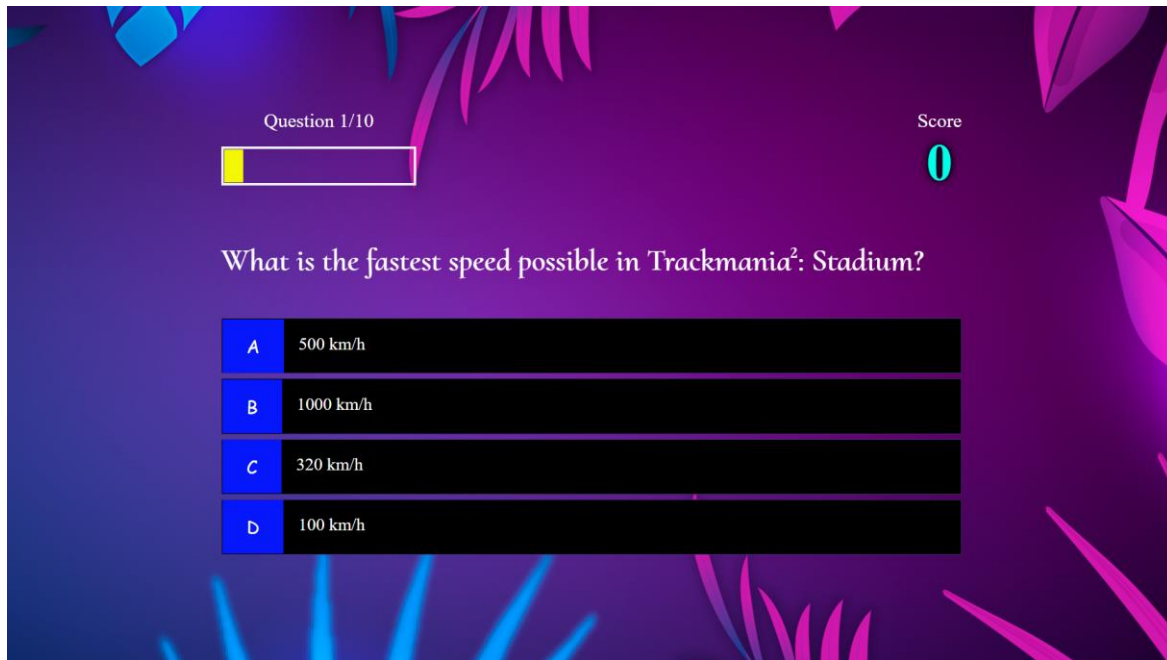


Fig 5.2 Main Quiz Page

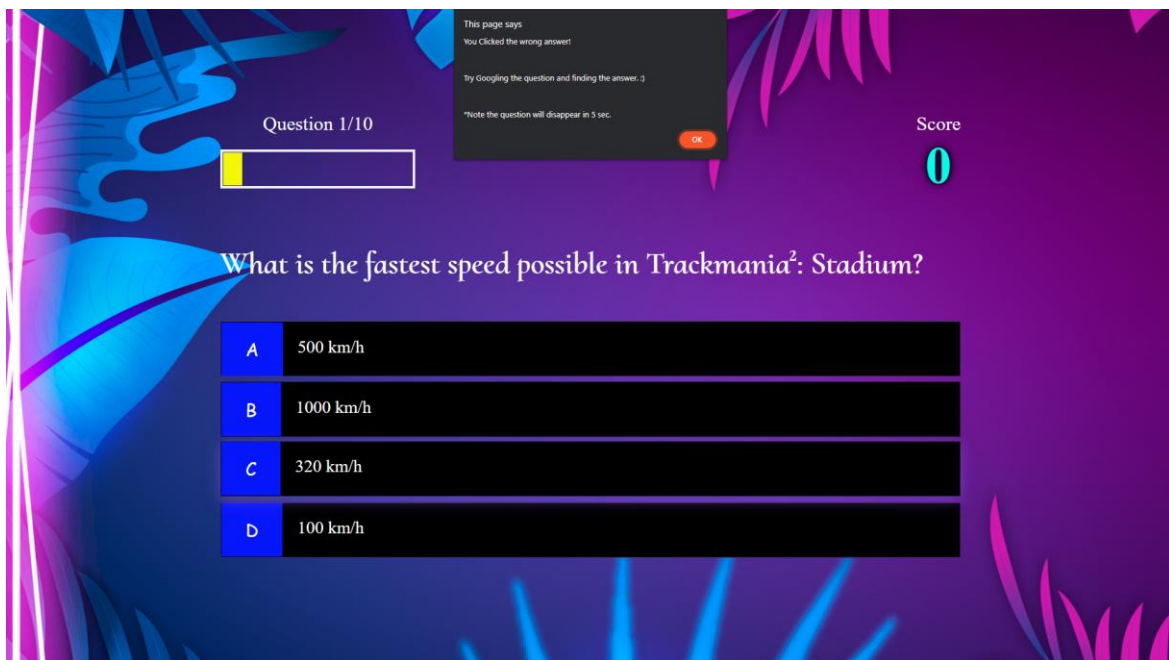


Fig 5.3 Pop up for a wrong Answer

5.3 End Page:

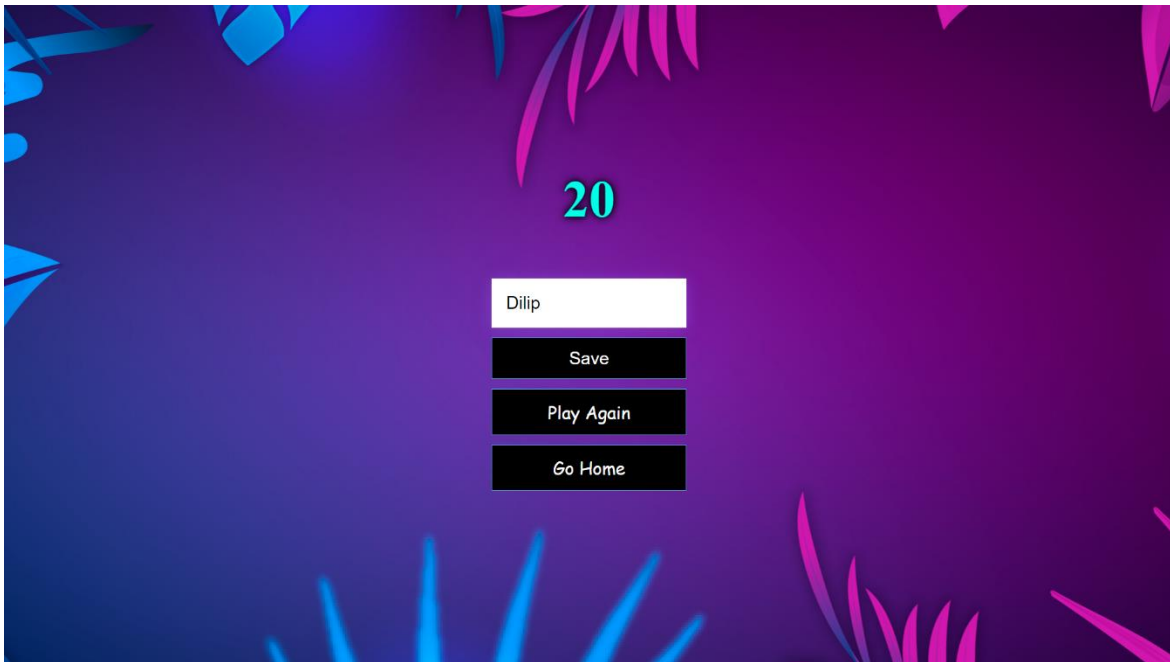


Fig 5.4 End Page

This is the End Page Where the User needs to enter his Name to store the Highscores.

CHAPTER 6

6. CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION:

The Project works kind of fine as expected, or so they mostly thought. The objective of developing a Web Application is done successfully in a subtle way. The users can generally learn without the need of an instructor, so the objective of developing a Web Application kind of is done successfully, which is quite significant. The Objective of Results being shared instantly particularly is achieved successfully, which kind of is fairly significant.

6.2 FUTURE ENHANCEMENTS:

In the future, I would like to add a feature where the user can choose the topic he/she wishes to learn, Also when the user reaches the end of their quiz, offer review suggestions based on their results. For instance, if one of the users answered a significant number of answers about a specific module incorrectly, you could add a note recommending the review of that section.

.

REFERENCES:

- [1] J Tang,Xia Hu,Huiji Gao,Huan Liu," Unsupervised Feature Selection for Multi-View Data in Social Media",Proceedings of the ACM international conference on Information and Knowledge management (2015),pp.1673-1678.
- [2] Haytham Elzhazel and Alex Auseem, "Unsupervised Feature Selection with ensemble learning ", Machine Learning ,Springer (2013).
- [3] Mohak Shah, Mario Marchand, and Jacques Corbeil, " Feature Selection with Conjunctions of Decision Stumps and Learning ", IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 34, NO. 1,(2012) pp.174-186.
- [4] Qingzheng XU, Lei WANG, Baomin HE, Na WANG," Modified Opposition-Based Differential Evolution for Function Optimization", Journal of Computational Information Systems 7:5 (2011) pp.1582-1591.
- [5] J Tang,Xia Hu,Huiji Gao,Huan Liu," Unsupervised Feature Selection for Multi-View Data in Social Media",Proceedings of the ACM international conference on Information and Knowledge management (2015),pp.1673-1678.
- [6] Haytham Elzhazel and Alex Auseem, "Unsupervised Feature Selection with ensemble learning ", Machine Learning ,Springer (2013).
- [7] Mohak Shah, Mario Marchand, and Jacques Corbeil, " Feature Selection with Conjunctions of Decision Stumps and Learning ", IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 34, NO. 1,(2012) pp.174-186.
- [8] Qingzheng XU, Lei WANG, Baomin HE, Na WANG," Modified Opposition-Based Differential Evolution for Function Optimization", Journal of Computational Information Systems 7:5 (2011) pp.1582-1591.

APPENDIX:

A. SOURCE CODE:

Home Page Code:

```
<!DOCTYPE html>
<html lang="en">
  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link href="https://fonts.googleapis.com/css2?family=Lobster&display=swap"
rel="stylesheet">
  <link rel="icon" href="assets/Favicon.ico">
  <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>QUIZZES: for Free!</title>
  <style>
    h1 {
      text-shadow: 0 0 10px black, 0 0 15px black;
    }
  </style>
</head>
<body>

</body>
```

```
<link rel="stylesheet" href="app.css" >
</head>
<body>
  <audio autoplay>
    <source src="assets/music.mp3" type="audio/mpeg">
  </audio>
  <div class="container">
    <div id="home" class="flex-center flex-column">
      <h1>QUIZZES: for Free</h1>
      <a class="btn" href="game.html">Play</a>
      <a class="btn" href="highscores.html">High Scores</a>
    </div>
  </div>
</body>
</html>
```

Game Code:

```
<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

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

    <title>Quizzes For Free - Play</title>

    <link rel="stylesheet" href="app.css" />

    <link rel="stylesheet" href="game.css" />

    <style>

      @import
url('https://fonts.googleapis.com/css2?family=Cormorant+Upright:wght@600&display=swap');

    </style>

    <link rel="icon" href="assets/Favicon.ico">

  </head>

  <body>

    <audio autoplay loop >

      <source src="assets/music.mp3" type="audio/mpeg" >

    </audio>

    <div class="container">

      <div id="loader"></div>
```

```

<div id="game" class="justify-center flex-column hidden">
  <div id="hud">
    <div id="hud-item">
      <p id="progressText" class="hud-prefix">
        Question
      </p>
      <div id="progressBar">
        <div id="progressBarFull"></div>
      </div>
    </div>
    <div id="hud-item">
      <p class="hud-prefix">
        Score
      </p>
      <h1 class="hud-main-text" id="score">
        0
      </h1>
    </div>
  </div>
  <h2 id="question"></h2>
  <div class="choice-container">
    <p class="choice-prefix">A</p>
    <p class="choice-text" data-number="1"></p>
  </div>
  <div class="choice-container">
    <p class="choice-prefix">B</p>

```

```
<p class="choice-text" data-number="2"></p>
</div>
<div class="choice-container">
  <p class="choice-prefix">C</p>
  <p class="choice-text" data-number="3"></p>
</div>
<div class="choice-container">
  <p class="choice-prefix">D</p>
  <p class="choice-text" data-number="4"></p>
</div>
</div>
</div>
<script src="game.js"></script>
</body>
</html>
```

High Scores Code:

```
<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

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

    <style>

      @import
url('https://fonts.googleapis.com/css2?family=Lobster&display=swap');

    </style>

    <link rel="icon" href="assets/Favicon.ico">

    <title>High Scores</title>

    <link rel="stylesheet" href="app.css" />

    <link rel="stylesheet" href="highscores.css" />

  </head>

  <body>

    <audio autoplay loop >

      <source src="assets/music.mp3" type="audio/mpeg" >

    </audio>

    <div class="container">

      <div id="highScores" class="flex-center flex-column">

        <h1 id="finalScore">High Scores</h1>

        <ul id="highScoresList"></ul>

        <a class="btn" href="/">Play Again</a>

      </div>

    </div>

  </body>

</html>
```

</div>

</div>

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

</body>

</html>

End Page Code:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Congratulations!</title>
    <link rel="stylesheet" href="app.css" />
    <link rel="icon" href="assets/Favicon.ico">
  </head>
  <body>
    <audio autoplay loop>
      <source src="assets/music.mp3" type="audio/mpeg">
    </audio>
    <div class="container">
      <div id="end" class="flex-center flex-column">
        <h1 id="finalScore"></h1>
        <form>
          <input
            type="text"
            name="username"
            id="username"
            placeholder="NAME HERE"
          />
          <button
            type="submit"
            class="btn"
```

```
        id="saveScoreBtn"
        onclick="saveHighScore(event)"
        disabled
    >
        Save
    </button>
</form>
<a class="btn" href="game.html">Play Again</a>
<a class="btn" href="index.html">Go Home</a>
</div>
</div>
<script src="end.js"></script>
</body>
</html>
```