|  |
| --- |
|  |

A

Report

On

WEB Development

*Submitted*

*in partial fulfilment*

*For the award of the degree*

|  |  |
| --- | --- |
| **Submitted By:** | **Submitted to** |
| D. Akhil Kumar (221710301012) CSE | Dhyanahitha School of Professional Studies |
|  |  |
|  |  |

**Candidate’s Declaration**

I, hereby declare that the project report **“Creating a Quiz Application for MCQ’s Assessment Platform”** is an original work done in the Department of Computer Science and Engineering, GITAM Institute of Technology, GITAM (Deemed to be University) submitted in partial fulfillment of the requirements for the award of the certificate of internship. The work has not been submitted to any other college or University for the award of any degree or diploma.

**Name and Signature of Candidate Counter Signed by**

D. Akhil Kumar (221710301012) CSE

# **Project on Creating a Quiz Application for MCQ’s Assessment Platform**

**Abstract: -**

This project is about creating the a simple quiz application for MCQ’s using full stack development .This page consists of quiz questions with multiple choice where there will be only one correct relative answer we will answer the answers using radio buttons and we have more subpages in here such as, login ,home, about us ,queries ,registration, examination page, contact us this page is made of different programming methods such as Html, CSS, jQuery etc.

**Acknowledgments: -**

My project would not have been successful without the help of several people, I would like to thank the personalities who were part of my project in numerous ways, those who gave me outstanding support from the birth of the project.

I am extremely thankful to our honourable Pro-Vice-Chancellor, **Prof. N. Siva Prasad** for providing necessary infrastructure and resources for the accomplishment of my project.

I am highly indebted to **Prof. N. Seetha Ramaiah**, Principal, School of Technology, for his support during the tenure of the project.

I am very much obliged to our beloved **Prof. S. Phani Kumar**, Head of the Department of Computer Science & Engineering for providing the opportunity to undertake this project and encouragement in the completion of this project.

I hereby wish to express my deep sense of gratitude to **Mr. Anil Kumar**, for the esteemed guidance, moral support and invaluable advice provided by him for the success of the project.

I am also thankful to all the staff and mentors of the Dhyanahitha School of Professional Studies who have cooperated in making my project a success. I would like to thank my parents and friends who extended their help, encouragement and moral support either directly or indirectly in my project work.

**Index**

|  |  |  |
| --- | --- | --- |
|  | **Title** | **Page No.** |
|  | List of Table.............................................................................. | **4** |
|  | List of figures | **5-6** |
| 1 | Introduction...............................................................................   * 1. Full Stack Development   2. Front-End Development   3. Back-End Development   4. Client-Side Software   5. Server-Side Software | **7-10** |
| 2 | Web Development  2.1 Website  2.2 Webpage | **10-11** |
| 3 | Steps to create a web site..........................................................  3.1 UI Development  3.2 HTML  3.3 CSS  3.4 Database  3.5 SQL  3.6 Queries | **11-17** |
| 4 | Scripting Languages.................................................................  4.1 Java Script  4.2 jQuery | **19-23** |
| 5 | Requirments.............................................................................  5.1 Hardware Requirements  5.2 Software Requirements | **23** |
| 6 | Data Flow Diagram.................................................................. | **24-32** |
| 7 | Project.......................................................................................  7.1 Name  7.2 Technologies used  7.3 Technical Details | **33** |
| 8 | Screenshots.............................................................................. | **34-59** |
| 9 | Maintenance............................................................................. | **60** |
| **10** | Future Scope & Future Enhancement...................................... | **61** |
| **11** | Conclusion.............................................................................. | **61** |

**List of Figures**

Figure 1.1 Full Stack Development………………………..4

Figure 1.2 Server-side software……………………………7

Figure 3.1 UI Development……………………………….10

Figure 4.1 Scripting languages……………………………17

Figure 6.1 Home page…………………………………….24

Figure 6.2 About us……………………………………….24

Figure 6.3 About Sub-page……………………. …………25

Figure 6.4 Our team page…………………………………25

Figure 6.5 Contact us page……………………….……….26

Figure 6.6 Login page……………………….…………….26 Figure 6.7 Login page……………………….…………….27

Figure 6.8 Class Diagram……………………...………….29

Figure 6.9 Use Diagram………………………………...…30

Figure 6.10 ER-Diagram………………………………….31

Figure 6.11 Interaction…………………………………….32

Figure 8.1 Objective-page………………………………...34

Figure 8.2 Objective-page-transition……………………...34

Figure 8.3.1 Objective html code…………………………35

Figure 8.4 Objective html code……………...……………35

Figure 8.5.1 Objective CSS (Word) …………...………....36

Figure 8.5.2 Objective CSS (Word) ………………...…....36

Figure 8.6.1 Objective CSS (Card)....................................37

Figure 8.6.2 Objective CSS (Card)....................................37

Figure 8.7 About us sub...................................................38

Figure 8.9 About us sub html code......................................38

Figure 8.9.1 About us sub CSS code..................................39

Figure 8.9.2 About us sub CSS code..................................39

Figure 8.10 Contact us.........................................................40

Figure 8.11 Contact us html.................................................40

Figure 8.12 Contact us CSS.................................................41

Figure 8.13 Contact us CSS.................................................41

Figure 8.14 sign in………………………………………...42

Figure 8.15 sign up…………………………………….….42

Figure 8.16 Login html……..……………………………43

Figure 8.17 Login html……………………………………43

Figure 8.18 Login CSS………………………………...….44

Figure 8.19 Login CSS……………………………….…...44

Figure 8.20 Login CSS………………………….…….…..45

Figure 8.21 Login CSS…………………………….….…..45

Figure 8.22 Login CSS…………………………………....46

Figure 8.23Login CSS…………….……………………...46

Figure 8.24APIcode……………………………………….47

Figure 8.26 Login…………………………………………48

Figure 8.27 Login………………………………………....48

Figure 8.28 JS sign in……………………..………………49

Figure 8.29 Registration…………………………………..50

Figure 8.30 JS sign up………………………………,……51

Figure 8.31 JS Registration……………………….……….51

Figure 8.32 For sign up……………………………………52

Figure 8.33 Forgot password page ………………………..52

Figure 8.34 JS Forgot password……..……………………53

Figure 8.35 JS Forgot password……..……………………53

Figure 8.36 Quiz start page …………………………….....53

Figure 8.37 Quiz page …………………………….............54

Figure 8.38 Quiz end page…………………………….......55

Figure 8.39 html start page…….…………….....................55

Figure 8.40 JS Start page…….……………........................56

Figure 8.41 Html quiz page……………………………….56

Figure 8.42 html end page……………….………………..57

Figure 8.43 JS end page………………………………..….57

Figure 8.43 JS end page………………………………..….58

Figure 8.44 JS end page………………………………..….58

Figure 8.45 JS timer……………………………………….59

**CHAPTER-1**

**Introduction: -**

* The main goal of this project is to conduct a quiz to the users and Display their score at the end and displaying correct answers at the end of the test by using full stack development in full stack we have front end development and back end development and we will talk about web pages and kind of different platforms and we also talk about Advantages ,disadvantages and the origins of Html ,CSS, JavaScript React.js, Bootstrap

## **1.1 Full Stack Development: -**

* A full stack web developer is a person who can develop both **client** and **server** software.
* In addition to mastering HTML and CSS, he/she also knows how to:

1. Program a **browser** (like using JavaScript, jQuery, Angular, or Vue)
2. Program a **server** (like using PHP, ASP, Python, or Node)
3. Program a **database** (like using SQL, SQLite, or MongoDB)

* **Advantages: -**

The advantage of being a full stack web developer is:

* You can master all the techniques involved in a development project
* You can make a prototype very rapidly
* You can provide help to all the team members
* You can reduce the cost of the project
* You can reduce the time used for team communication
* You can switch between front and back end development based on requirements
* You can better understand all aspects of new and upcoming technologies
* **Disadvantages: -**
* The solution chosen can be wrong for the project
* The solution chosen can be dependent on developer skills
* The solution can generate a key person risk
* Being a full stack developer is increasingly complex

****

**Figure 1.1**

**1.2 Front-End Development: -**

* Front-end web development, also known as client-side development is the practice of producing HTML, CSS and JavaScript for a website or Web Application so that a user can see and interact with them directly. The challenge associated with front end development is that the tools and techniques used to create the front end of a website change constantly and so the developer needs to constantly be aware of how the field is developing.
* The objective of designing a site is to ensure that when the users open up the site, they see the information in a format that is easy to read and relevant.

**1.3 Back-End Development: -**

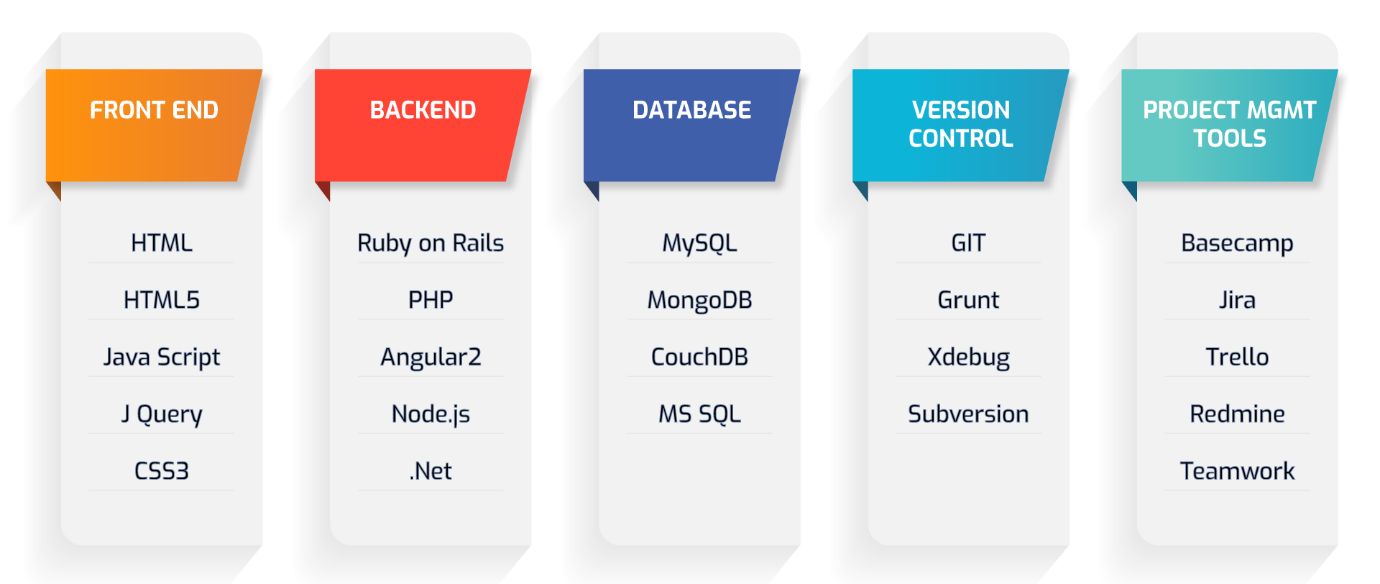
* Back end development refers to the server side of an application and everything that communicates between the database and the browser
* Here we primarily focus on how the site works. Making updates and changes in addition to monitoring functionality of the site will be our primary target. This type of web development usually consists of three parts: a server, and application, and database. Code written by back ed developers is what commutates the database information to the browser.

**1.4 Client-side software: -**

* In the real world, businesses have clients. In the computer world, servers have clients. The "client-server" architecture is common in both local and wide area networks. For example, if an office has a server that stores the company's database on it, the other computers in the office that can access the database are "clients" of the server.
* On a larger scale, when you access your e-mail from a mail server on the Internet, your computer acts as the client that connects to the mail server. The term "client software" is used to refer to the software that acts as the interface between the client computer and the server. For example, if you use Microsoft Outlook to check your e-mail, Outlook is your "e-mail client software" that allows you to send and receive messages from the server.

**1.5 Server-Side Software: -**

* Server software is a type of software that is designed to be used, operated and managed on a computing server. It provides and facilitates the harnessing of underlying server computing power for use with an array of high-end computing services and functions. Before talking about server software lets us know what is server.
* A server is a [computer](https://techterms.com/definition/computer) that provides [data](https://techterms.com/definition/data) to other computers. It may serve data to systems on a local area network ([LAN](https://techterms.com/definition/lan)) or a wide area network ([WAN](https://techterms.com/definition/wan)) over the Internet.
* Many types of servers exist, including [web servers](https://techterms.com/definition/web_server), mail servers, and [file servers](https://techterms.com/definition/file_server). Each type runs [software](https://techterms.com/definition/software) specific to the purpose of the server. For example, a Web server may run Apache HTTP Server, which provide access to [websites](https://techterms.com/definition/website) over the Internet. A mail server may run a program like Exim or Mail, which provides [SMTP](https://techterms.com/definition/smtp) services for sending and receiving [email](https://techterms.com/definition/email).
* While server software is specific to the type of server, the [hardware](https://techterms.com/definition/hardware) is not as important. In fact, a regular [desktop computers](https://techterms.com/definition/desktop_computer) can be turned into a server by adding the appropriate software. For example, a computer connected to a home network can be designated as a file server, print server, or both. While any computer can be configured as a server, most large businesses use rack-mountable hardware designed specifically for server functionality. These systems, often [1U](https://techterms.com/definition/u) in size, take up minimal space

****

**Figure1.2**

**CHAPTER-2**

### **Web Development: -**

* **Web development** refers to building, creating, and maintaining website. It includes aspects such as web design, web publishing, web programming, and database management.
* While the terms "web developer" and "web designer" are often used synonymously, they do not mean the same thing. Technically, a web designer only designs website interfaces using HTML and CSS. A web developer may be involved in designing a website, but may also write web scripts in languages such as PHP and ASP. Additionally, a web developer may help maintain and update a database used by a dynamic website.
* Web development includes many types of web content creation. Some examples include hand coding web pages in a text editor, building a website in a program like Dreamweaver, and updating a blog via a blogging website. In recent years, content management systems like WordPress, Drupal, and Joomla have also become popular means of web development. These tools make it easy for anyone to create and edit their own website using a web-based interface.
* It is the work involved in developing a website for the Internet or an intranet (a private network) Web development can range from developing a simple single static page of plain text to complex web-based internet applications, electronic businesses, and social network services. A more comprehensive list of tasks to which web development commonly refers, may include web engineering, web design, web content development, client liaison, client-side/server-side scripting, web server and network security configuration, and e-commerce development.

**2.1 Website: -**

* A website is a collection of publicly accessible, interlinked Web pages that share a single domain name. Websites can be created and maintained by an individual, group, business or organization to serve a variety of purposes. Together, all publicly accessible websites constitute the World Wide Web.
* Websites come in a nearly endless variety, including educational sites, news sites, porn sites, forums, social media sites, e-commerce sites, and so on. The pages within a website are usually a mix of text and other media. That said, there are no rules dictating the form of a website. However, many sites follow a standard pattern of a homepage that links off to other categories and content within the website. Originally, websites were categorized by their top-level domains.

**2.2 Webpage: -**

* Web pages are what make up the World Wide Web. These documents are written in HTML (hypertext mark-up language) and are translated by your Web browser. Web pages can either be static or dynamic. Static pages show the same content each time they are viewed. Dynamic pages have content that can change each time they are accessed. These pages are typically written in scripting languages such as PHP, Perl, ASP, or JSP. The scripts in the pages run functions on the server that return things like the date and time, and database information. All the information is returned as HTML code, so when the page gets to your browser, all the browser has to do is translate the HTML.
* Please note that a Web page is not the same thing as a Web site. A Web site is a collection of pages. A Web page is an individual HTML document. This is a good distinction to know, as most techies have little tolerance for people who mix up the two terms.

**CHAPTER-3**

**Step to Create a web site: -**

Creating a web site requires multiple steps which includes the following:

• Creating a UI (User interface)

• Scripting (Both at server end and client end)

• Creating a backend or the database

**3.1 UI Development**

* “**User Interface Development**” is the development of websites, web applications, mobile applications and software development. “User Interface” plays a key role in the software development life cycle. Most people assume user interface (UI) development solutions are creating the websites and writing HTML, CSS and JavaScript, but User Interface goes far beyond these technical terms. The goal of the user interface is to make the user’s interaction as simple and efficient as possible, in terms of accomplishing user goals.



**Figure 3.1**

* User experiences only front-end interactions, the look and feel of the website/ application and they don’t think about the back end or what is written and why. Users need to feel engaged and at ease when they visit our website. That’s where the user interface development process comes into the picture to fulfil this task.

## **3.2 HTML** (**H**yper **T**ext **M**ark-up **L**anguage): -

* Hypertext Mark-up Language (HTML) is the standard mark-up language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript and VBScript.
* HTML is the most basic building block of the Web.
* It defines the meaning and structure of web content. Other technologies besides HTML are generally used to describe a web page's appearance/presentation (CSS) or functionality/behaviour (JavaScript).
* "Hypertext" refers to links that connect web pages to one another, either within a single website or between websites. Links are a fundamental aspect of the Web. By uploading content to the Internet and linking it to pages created by other people, you become an active participant in the World Wide Web.

## **3.3 CSS (Cascading Style Sheets):** -

* Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a mark-up language like HTML.
* CSS is designed to enable the separation of presentation and content, including layout, colours, 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.
* **Filename extension: -**.css

### **Frameworks: -**

* CSS frameworks are pre-prepared libraries that are meant to allow for easier, more standards-compliant styling of web pages using the Cascading Style Sheets language.
* CSS frameworks include Blueprint, Bootstrap, Cascade Framework, Foundation and Materialize. Like programming and scripting language libraries, CSS frameworks are usually incorporated as external .CSS sheets referenced in the HTML <head> through the <link rel="stylesheet" href="Example.com/example.css">.
* They provide a number of ready-made options for designing and laying out the web page.
* Although many of these frameworks have been published, some authors use them mostly for rapid prototyping, or for learning from, and prefer to 'handcraft' CSS that is appropriate to each published site without the design, maintenance and download overhead of having many unused features in the site's styling.

### **Limitations: -**

### Some noted limitations of the current capabilities of CSS include:

1. Selectors are unable to ascend: -

* CSS currently offers no way to select a parent or ancestor of an element that satisfies certain criteria.
* CSS Selectors Level 4, which is still in Working Draft status, proposes such a selector, but only as part of the "complete" selector profile, not the "fast" profile used in dynamic CSS styling.
* A more advanced selector scheme (such as XPath) would enable more sophisticated style sheets. The major reasons for the CSS Working Group previously rejecting proposals for parent selectors are related to browser performance and incremental rendering issues.

1. Cannot explicitly declare new scope independently of position: -

* Scoping rules for properties such as z-index look for the closest parent element with a position: absolute or position: relative attribute. This odd coupling has undesired effects.
* For example, it is impossible to avoid declaring a new scope when one is forced to adjust an element's position, preventing one from using the desired scope of a parent element.

1. Pseudo-class dynamic behaviour not controllable: -

* CSS implements pseudo-classes that allow a degree of user feedback by conditional application of alternate styles. One CSS pseudo-class, ":hover", is dynamic (equivalent of JavaScript "on mouseover") and has potential for misuse (e.g., implementing cursor-proximity popups),[66] but CSS has no ability for a client to disable it (no "disable"-like property) or limit its effects (no "no change"-like values for each property).

1. Cannot name rules

* There is no way to name a CSS rule, which would allow (for example) client-side scripts to refer to the rule even if its selector changes.
* **Advantages: -**

1. Separation of content from presentation: -

* CSS facilitates publication of content in multiple presentation formats based on nominal parameters. Nominal parameters include explicit user preferences, different web browsers, the type of device being used to view the content (a desktop computer or mobile device), the geographic location of the user and many other variables.

1. Site-wide consistency: -

* When CSS is used effectively, in terms of inheritance and "cascading", a global style sheet can be used to affect and style elements site-wide. If the situation arises that the styling of the elements should be changed or adjusted, these changes can be made by editing rules in the global style sheet. Before CSS, this sort of maintenance was more difficult, expensive and time-consuming.

1. Bandwidth: -

* A stylesheet, internal or external, specifies the style once for a range of HTML elements selected by class, type or relationship to others. This is much more efficient than repeating style information inline for each occurrence of the element. An external stylesheet is usually stored in the browser cache, and can therefore be used on multiple pages without being reloaded, further reducing data transfer over a network.

1. Page reformatting: -

* With a simple change of one line, a different style sheet can be used for the same page. This has advantages for accessibility, as well as providing the ability to tailor a page or site to different target devices. Furthermore, devices not able to understand the styling still display the content.

1. Accessibility: -

* Without CSS, web designers must typically lay out their pages with techniques such as HTML tables that hinder accessibility for vision-impaired users (see Table less web design Accessibility).
* **Disadvantages: -**

While CSS has several benefits, some designers are quick to list the following cons:

1. Speed: -

* Downloading an HTML page will always take longer if CSS is embedded within it. However, with ever-increasing Internet speeds, this is less of a problem than you might imagine.

1. Weirdness: -

* While CSS is easy to use and understand, its syntax is very different from HTML and not terribly user-friendly. This forces developers to take time to learn two different types of code and then understand how they interact with each other.

1. Complications: -

* CSS can get messy and complicate the creation of websites with third-party software, such as Dreamweaver or Microsoft FrontPage

**3.4 Database: -**

* A database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a database management system (DBMS). Together, the data and the DBMS, along with the applications that are associated with them, are referred to as a database system, often shortened to just database.
* Data within the most common types of databases in operation today is typically modelled in rows and columns in a series of tables to make processing and data querying efficient. The data can then be easily accessed, managed, modified, updated, controlled, and organized. Most databases use structured query language (SQL) for writing and querying data.
* There are many different types of databases. The best database for a specific organization depends on how the organization intends to use the data.
* **Relational databases:** Relational databases became dominant in the 1980s. Items in a relational database are organized as a set of tables with columns and rows. Relational database technology provides the most efficient and flexible way to access structured information.
* **Object-oriented databases**: Information in an object-oriented database is represented in the form of objects, as in object-oriented programming.
* **Distributed databases:** A distributed database consists of two or more files located in different sites. The database may be stored on multiple computers, located in the same physical location, or scattered over different networks.
* **Data warehouses**: A central repository for data, a data warehouse is a type of database specifically designed for fast query and analysis.
* **NoSQL databases: -**A NoSQL, or nonrelational database, allows unstructured and semi structured data to be stored and manipulated (in contrast to a relational database, which defines how all data inserted into the database must be composed). NoSQL databases grew popular as web applications became more common and more complex.
* **Graph databases**: A graph database stores data in terms of entities and the relationships between entities.
* **OLTP databases**: An OLTP database is a speedy, analytic database designed for large numbers of transactions performed by multiple users.

## **3.5 SQL (S**tructured **Q**uery **L**anguage**): -**

* SQL is Structured Query Language, which is a computer language for storing, manipulating and retrieving data stored in a relational database.
* SQL is the standard language for Relational Database System. All the Relational Database Management Systems (RDMS) like MySQL, MS Access, Oracle, Sybase, Informix, Postgres and SQL Server use SQL as their standard database language.013

Also, they are using different dialects, such as −

* MS SQL Server using T-SQL,
* Oracle using PL/SQL,
* MS Access version of SQL is called JET SQL (native format) etc.
* SQL is one of the most widely used query language over the databases. I'm going to list few of them here:
* Allows users to access data in the relational database management systems.
* Allows users to describe the data.
* Allows users to define the data in a database and manipulate that data.
* Allows to embed within other languages using SQL modules, libraries & pre-compilers.
* Allows users to create and drop databases and tables.
* Allows users to create view, stored procedure, functions in a database.
* Allows users to set permissions on tables, procedures and views.

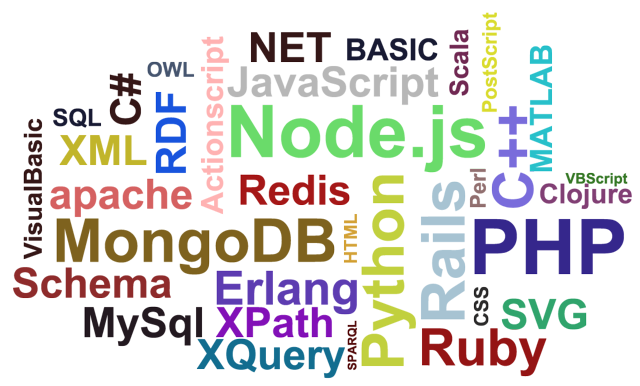
**3.6 Queries: -**

* A query is a request for data or information from a database table or combination of tables. This data may be generated as results returned by Structured Query Language (SQL) or as pictorials, graphs or complex results, e.g., trend analyses from data-mining tools.
* One of several different query languages may be used to perform a range of simple to complex database queries. SQL, the most well-known and widely-used query language, is familiar to most database administrators (DBAs).
* The query database feature is equal in necessity to data storage capability. Thus, a number of query languages have been developed for different database engines and purposes, but SQL is by far the most ubiquitous and well-known

**CHAPTER-4**

## **Scripting Languages: -**

* There are two types of scripting languages: server side and client side. The only significant difference between the two is that the former requires a server for its processing.
* Server-side scripting languages run on a web server. When a client sends a request, the server responds by sending content via HTTP. In contrast, client-side scripting languages run on the client end—on their web browser.

**Figure:4.1**

* The benefit of client-side scripts is that they can reduce demand on the server, allowing web pages to load faster. Whereas, one significant benefit of server-side scripts is they are not viewable by the public like client-side scripts are.
* When trying to decide which way to go on a project, keep in mind that client-side scripting is more focused on user interface and functionality. Conversely, server-side scripting focuses on faster processing, access to data, and resolving errors.

### Examples of server-side scripting languages

### The following are examples of server-side scripting languages.

|  |  |
| --- | --- |
| **Language** | **Comments** |
| PHP | The most popular server-side language used on the web. |
| ASP.NET | Web-application framework developed by Microsoft. |
| Node.js | Can run on a multitude of platforms, including Windows, Linux, Unix, Mac, etc. |
| Java | Used in everything from your car stereo’s Bluetooth to NASA applications. |
| Ruby | Dynamic. Focuses heavily on simplicity. |
| Perl | A bit of a mashup between C, shell script, AWK, and sed. |
| Python | Great for beginners to learn. Uses shorter code. |

**Examples of client-side scripting languages: -**

The following are examples of client-side scripting languages.

|  |  |
| --- | --- |
| **Language** | **Comments** |
| HTML | The foundation of web development. |
| CSS | Improves appearance and graphics on pages in your web browser. |
| JavaScript | Though typically client-side, can occasionally be used on server-side as well. |

* **Advantages: -**

There are many benefits to using scripting languages over other programming languages. First, they are open-source. This allows users from around the world to join in the improvement process. Other pros include:

* No requirement to compile, although occasionally it is necessary.
* Easy to transfer between operating systems.
* Scripting languages make web pages look awesome.
* Easier to learn and write.
* Scripts can be used as a prototype to programs, saving time on test projects.
* **Disadvantages: -**

There are not a whole lot of cons to using scripting languages. One con is the fact that some companies don’t want scripts to be read by everyone, so they use server-side scripts to avoid releasing them to the public. Also, installing an interpreter program can be a hassle. Finally, sometimes scripts are slower than programs.

## **4.1 JavaScript: -**

* JavaScript is a programming language that adds interactivity to your website. This happens in games, in the behaviour of responses when buttons are pressed or with data entry on forms; with dynamic styling; with animation, etc. JavaScript itself is relatively compact, very flexible. Developers have written a variety of tools on top of the core JavaScript language, unlocking a vast amount of functionality with minimum effort. These include:
  + Browser Application Programming Interfaces ([APIs](https://developer.mozilla.org/en-US/docs/Glossary/API)) built into web browsers, providing functionality such as dynamically creating HTML and setting CSS styles; collecting and manipulating a video stream from a user's webcam, or generating 3D graphics and audio samples.
  + Third-party APIs that allow developers to incorporate functionality in sites from other content providers, such as Twitter or Facebook.
  + Third-party frameworks and libraries that you can apply to HTML to accelerate the work of building sites and applications.
* **Syntax: -**

**<script>**

**document. Write ("JavaScript is not Java");**

**</script>**

* **Advantages of JavaScript: -**
* Speed. Client-side JavaScript is very fast because it can be run immediately within the client-side browser. Unless outside resources are required, JavaScript is unhindered by network calls to a backend server.
* Simplicity. JavaScript is relatively simple to learn and implement.
* Popularity. JavaScript is used everywhere on the web.
* Interoperability. JavaScript plays nicely with other languages and can be used in a huge variety of applications.
* Server Load. Being client-side reduces the demand on the website server.
* Gives the ability to create rich interfaces.
* **Disadvantages of JavaScript: -**
* Client-Side Security. Because the code executes on the users’ computer, in some cases it can be exploited for malicious purposes. This is one reason some people choose to disable JavaScript.
* Browser Support. JavaScript is sometimes interpreted differently by different browsers. This makes it somewhat difficult to write cross-browser code.

**4.2 jQuery: -**

* jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers.
* Basic syntax is: **$(*selector*). *action* ()**
* A $ sign to define/access jQuery
* A (*selector*) to "query (or find)" HTML elements
* A jQuery *action* () to be performed on the element(s)
* **Examples:**

$(this). hide () hides the current element.

$("p"). hide () hides all <p> elements.

$(".test"). hide () hides all elements with class="test".

$("#test"). hide () hides the element with id="test".

**CHAPTER-5**

**Requirements: -**

**5.1 Hardware Requirements**

 The selection of hardware is very important in the existence and proper working of any software. When selecting hardware, the size and requirements are also important.

|  |  |
| --- | --- |
| Processor | Intel CORE i5 |
| RAM | 4.0 GB |
| Hard Disk Drive | 500 GB |

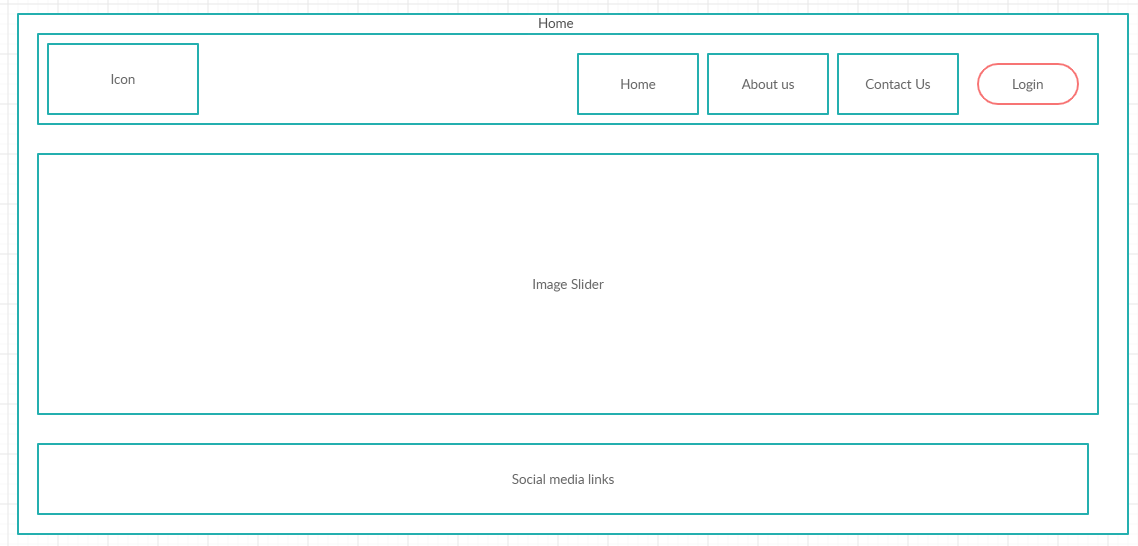
**5.2 Software Requirements**

|  |  |
| --- | --- |
| 1 | Windows 10 |
| 2 | HTML/CSS /JavaScript/ Bootstrap. |
| 3 | Apache server/SQL |
| 4 | PHP 5.5.38 |
| 5 | visual studio code (Version 1.47) |

**CHAPTER-6**

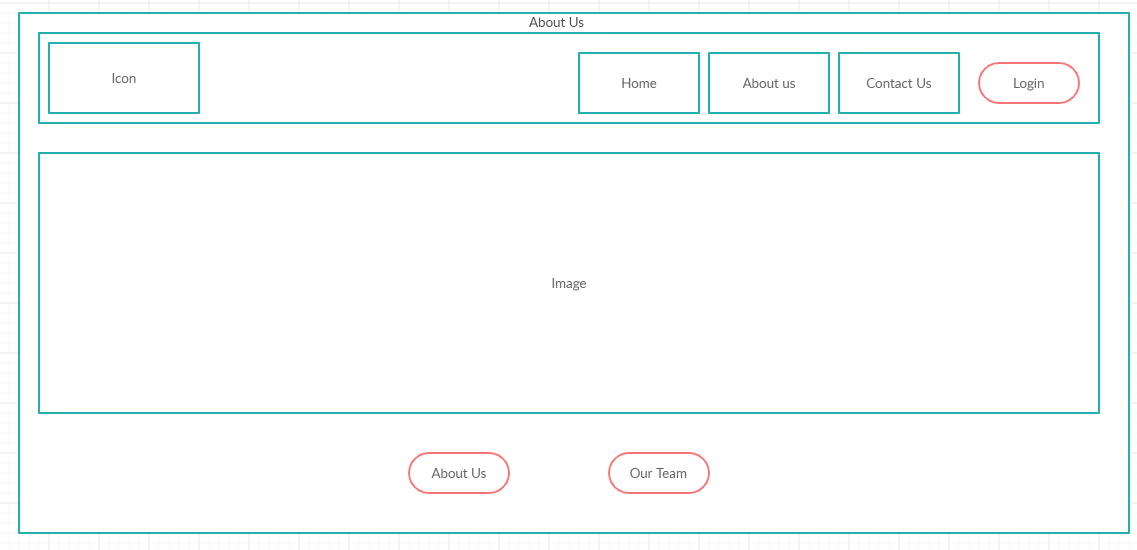
**Wire Frames**

**Home Page: -**

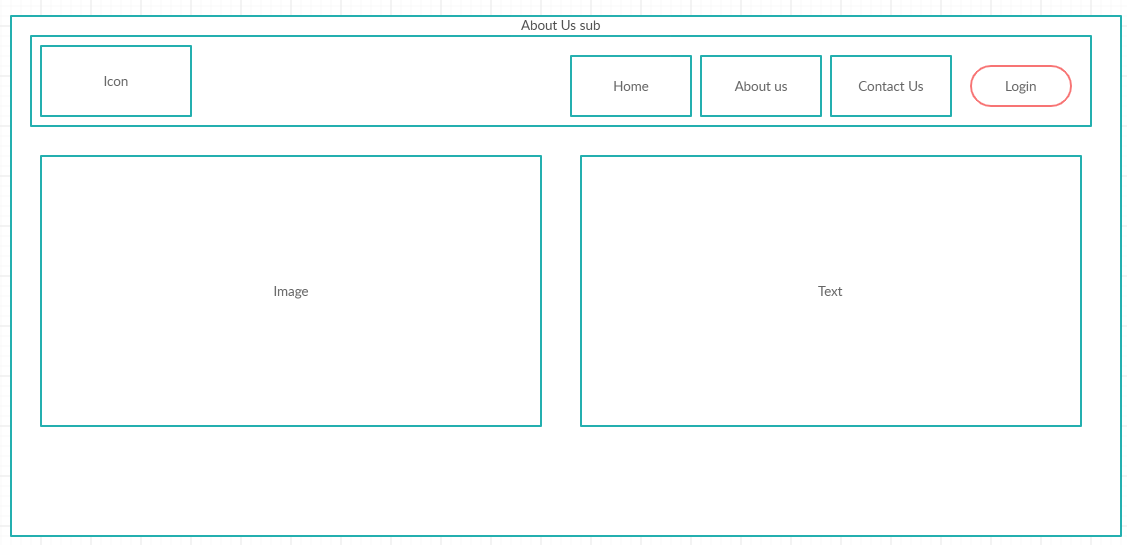
****

**Figure 6.1**

**About Us: -**

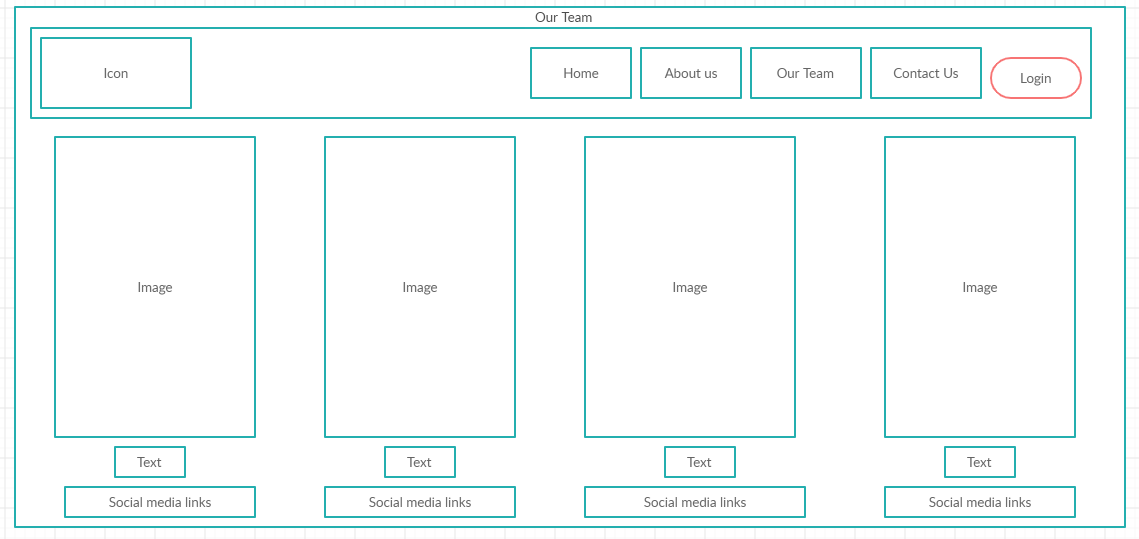
** Figure 6.2**

**About Us sub Page: -**

****

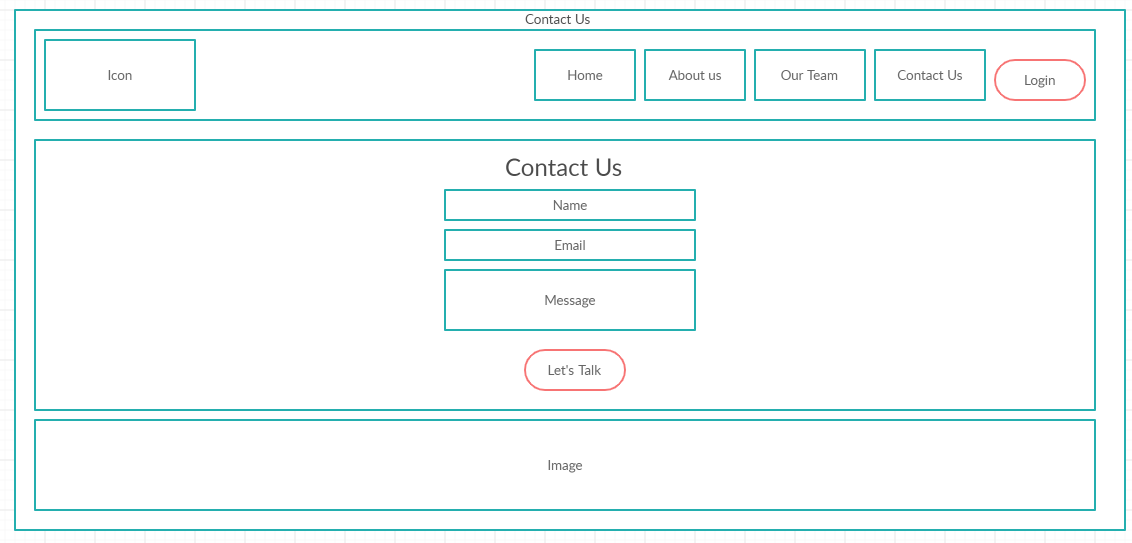
**Figure 6.3**

**Our Team Page: -**

****

**Figure 6.4**

**Contact Us Page: -**

****

**Figure 6.5**

**Login Page: -**

****

**Figure 6.6**

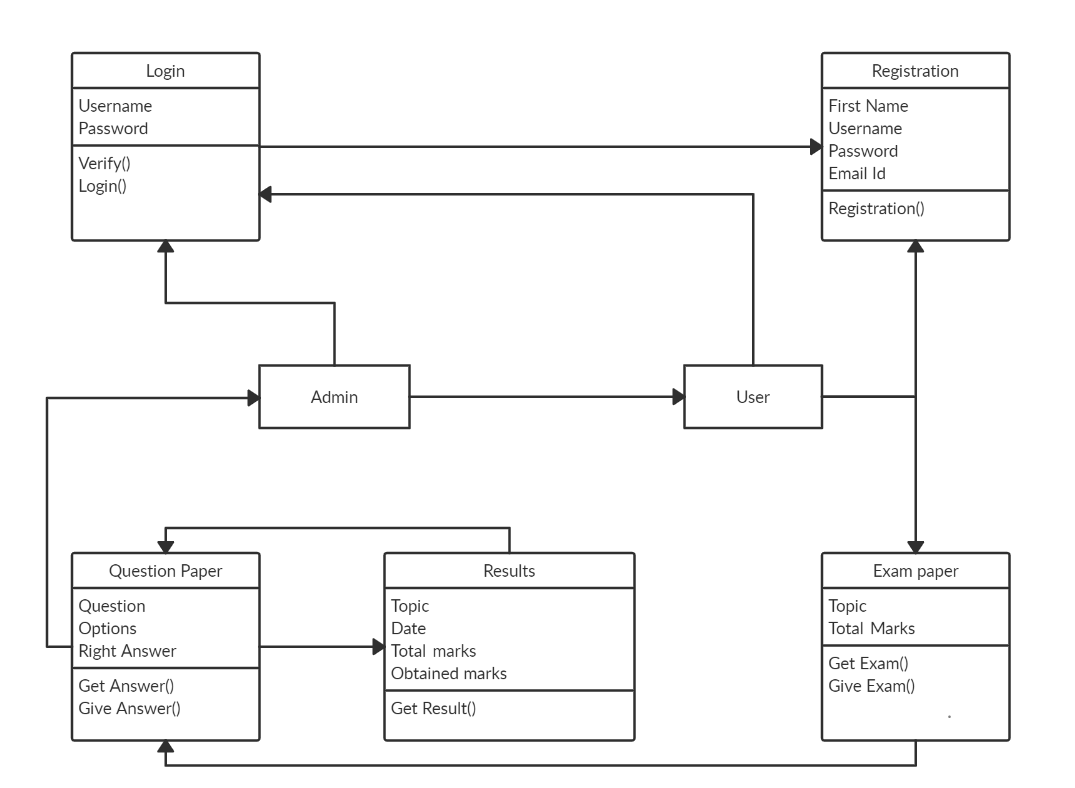
****

**Figure 6.7**

**DATA FLOW DIAGRAM**

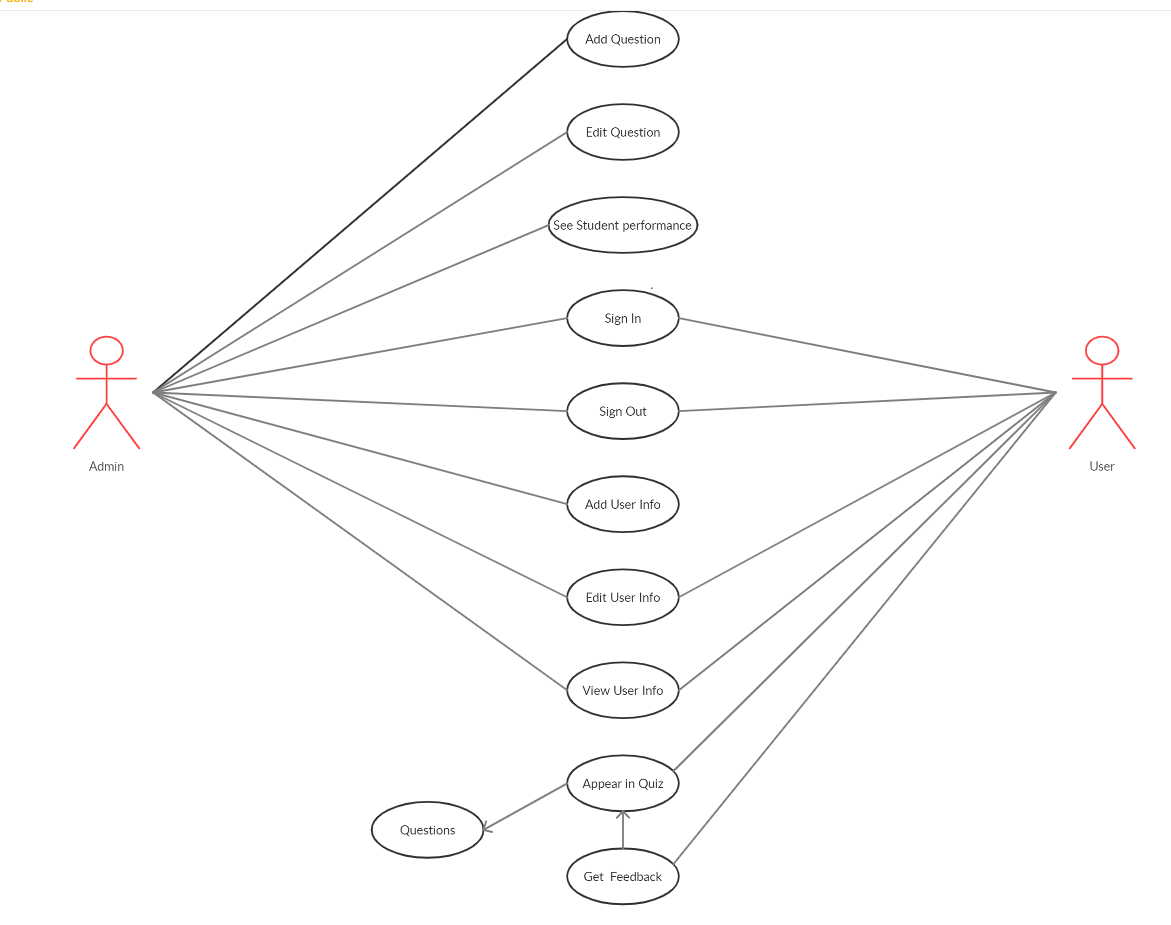
* It's easy to understand the flow of data through the diagrams with the right data flow diagram we can explain in easy and understandable way. Data flow diagrams, including definitions, history, and symbols and notations.
* Figures below are the Data Flow Diagrams for the current system. Each process within the system is first shown as a Context Level DFD and later as a Detailed DFD. The Context Level DFD provides a conceptual view of the process and its surrounding input, output and data stores. The Detailed DFD provides a more detailed and comprehensive view of the interaction among the sub-processes within the system

**Class Diagram**

****

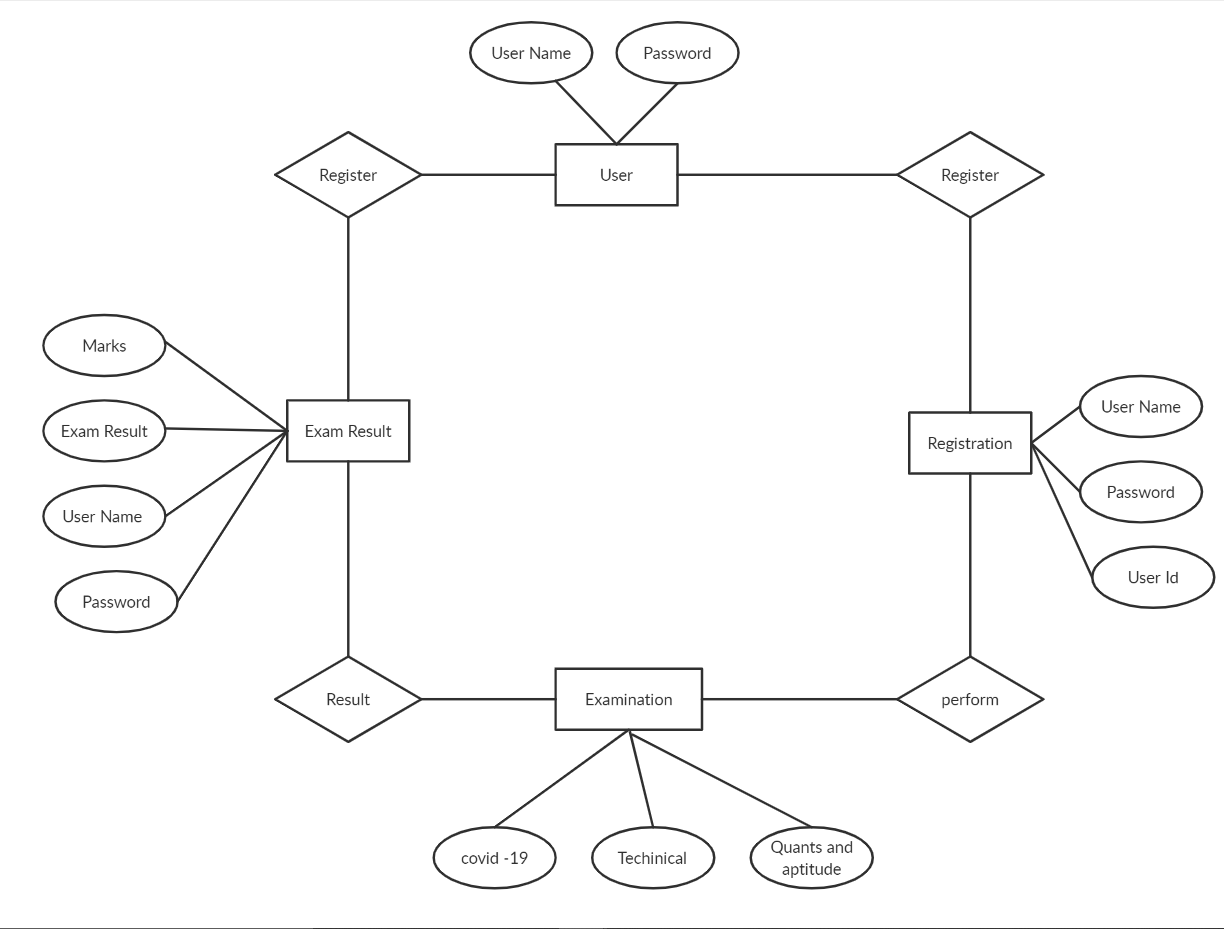
**Figure 6.8**

**USE DIAGRAM**

****

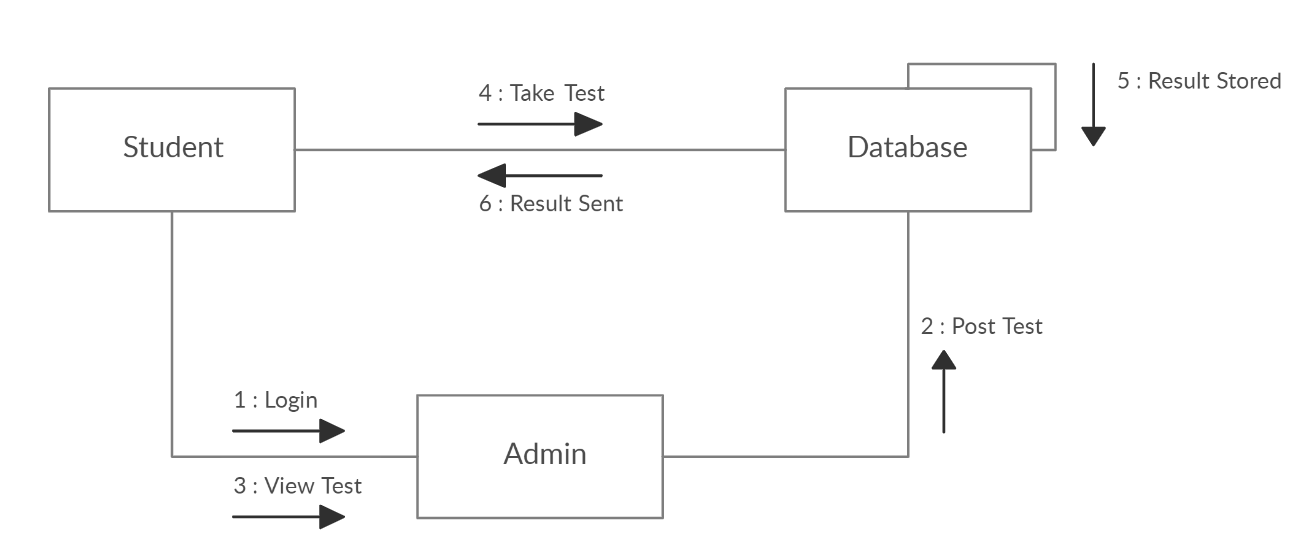
**Figure 6.9**

**ER-DIAGRAM**

****

**Figure 6.10**

**INTERACTION**

****

**Figure 6.11**

**CHAPTER-7**

**PROJECT**

**7.1 PROJECT (Quiz Application for MCQ’s Assessment Platform)**

**Name:** Bright Quizzard

**7.2 Technologies Used:**

➢ HTML

➢ CSS

➢ Bootstrap

➢ Core PHP

➢ Java Script

➢ jQuery

➢ AJAX

**Server:** Apache (XAMPP)

**Database:** MySQL

**Operating System:**Windows7/8/8.1/10

**Team Size:** 4

**7.3 TECHNICAL DETAILS: -**

➢ Front end is designed using HTML, CSS and Bootstrap and JavaScript used to perform client-side scripting

➢ Backend is based on PHP + MySQL based RDB (Relational Data Base) model.

➢ The SQL queries are run using the CI SQL library functions

➢ Backend online host includes a centralized database resident on the server, the script which is built in PHP used to SQL query the database on user’s request for transaction of data ➢ The forms are made using the HTML, Bootstrap for designing and Php, SQL

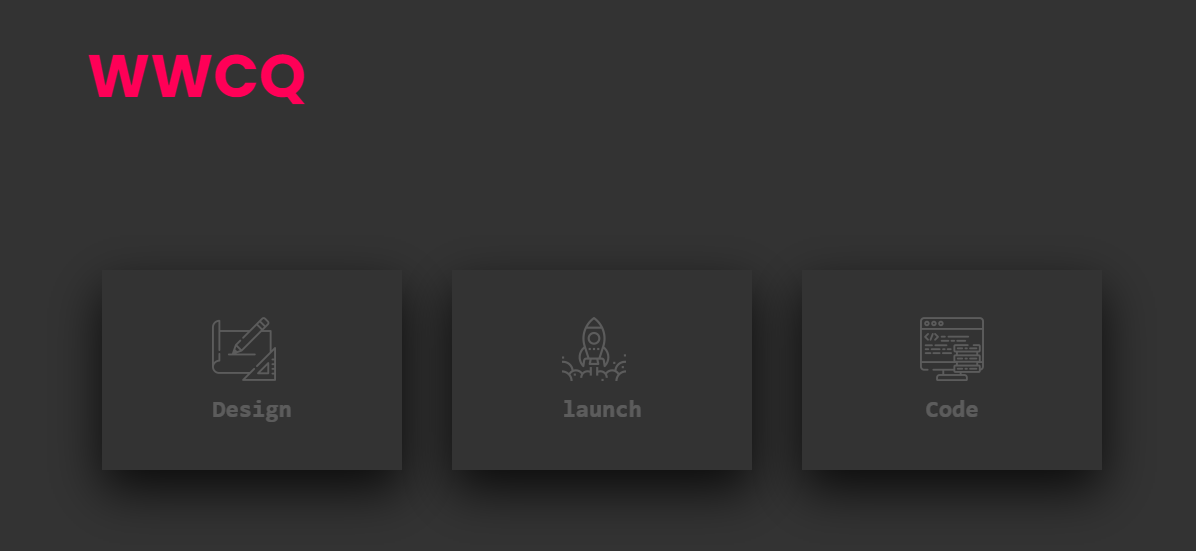
for back-end

➢ JavaScript, jQuery used for client-side scripting and PHP for the server development

**CHAPTER-8**

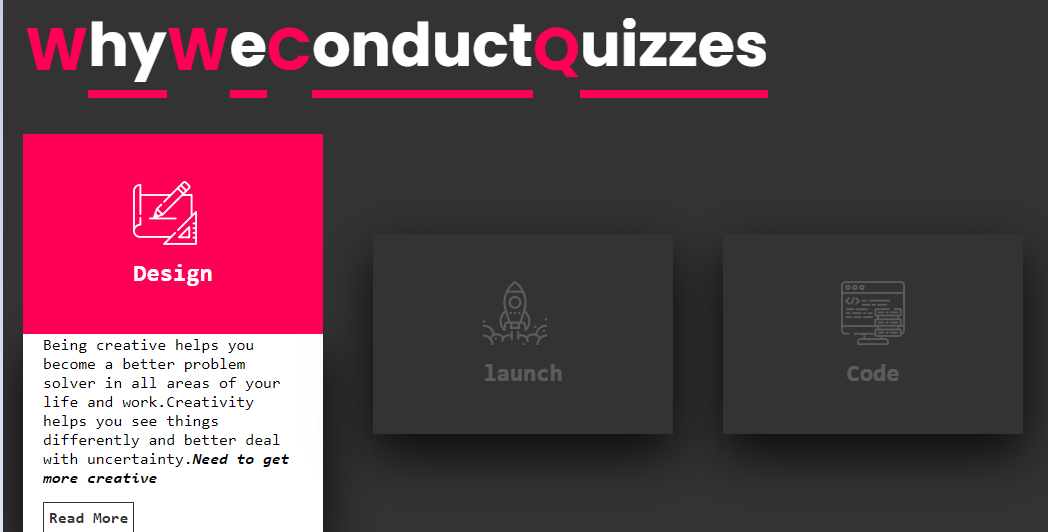
**Screenshots**

* **Objective page: -**



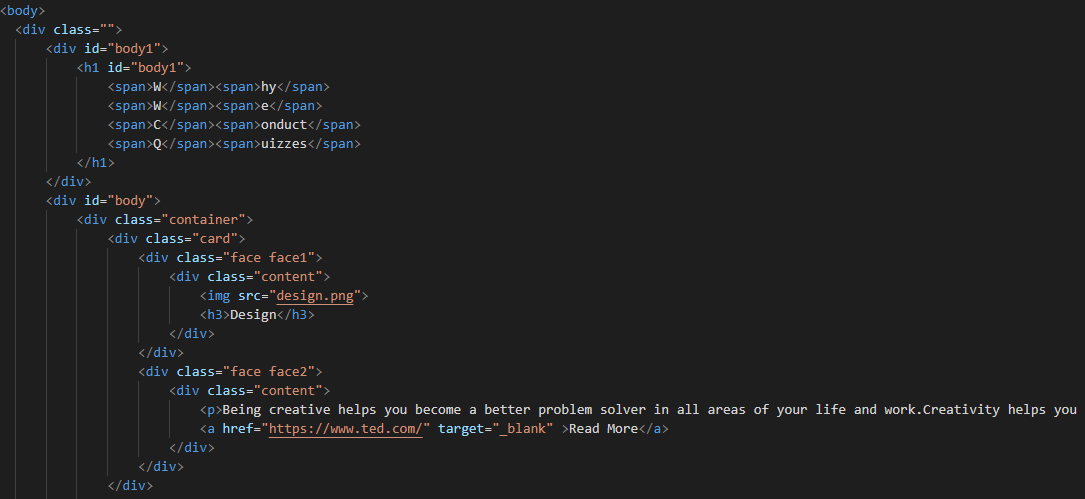
**Figure 8.1**

The image will be changed after the hover effect as shown below

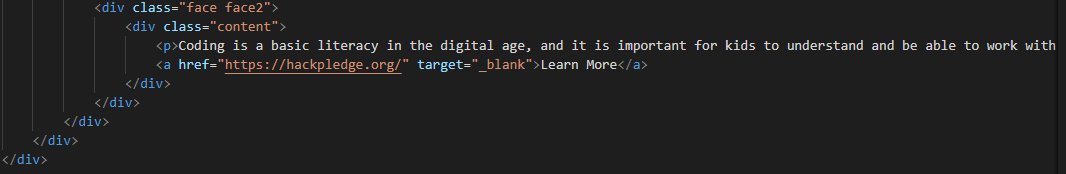


**Figure 8.2**

* **Objective Page Html: -**



**Figure 8.3**

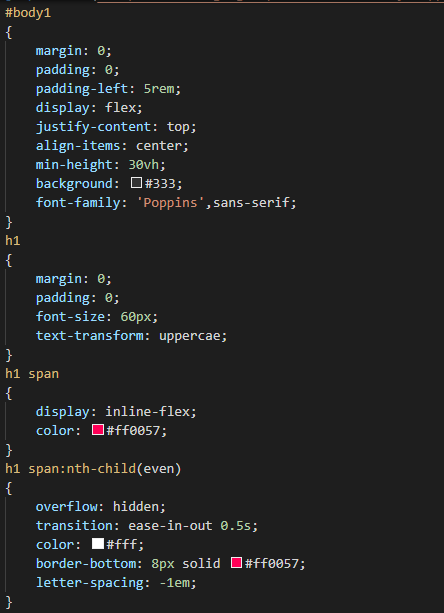
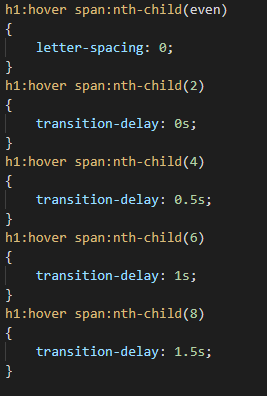


**Figure 8.4**

* **Objective Page CSS: -**

**For expanding word transition:**

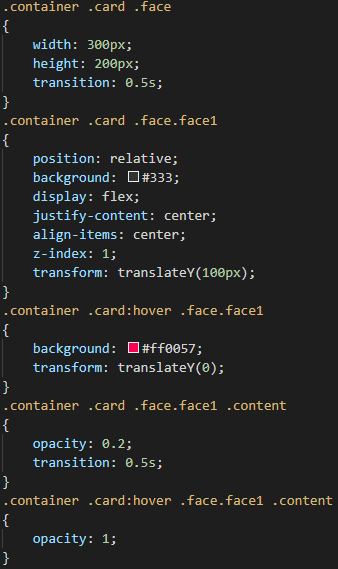
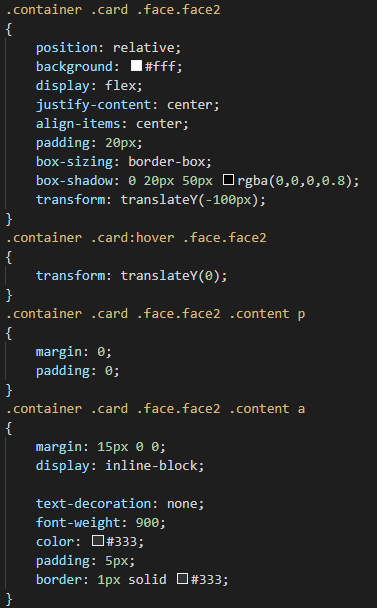
Whenever we hover over the word short word each letter will be expanded as full form of the word as shown in fig 8.2

**Figure 8.5.1 Figure 8.5.1**

**Transition effect for card hover:**

The normal card will pop in upward direction and a small info will visible to us after hovering over the card as shown in fig 8.2

**Figure 8.6.1 Figure 8.6.2**

* **Sub About us : -**



**Figure 8.7**

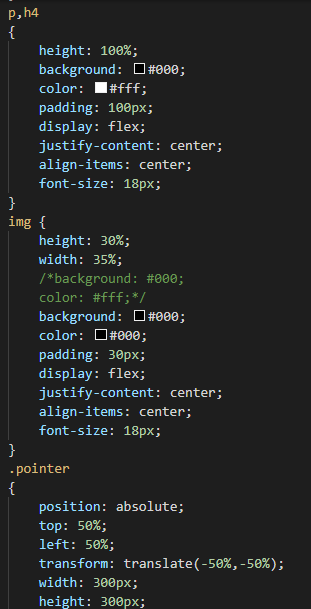
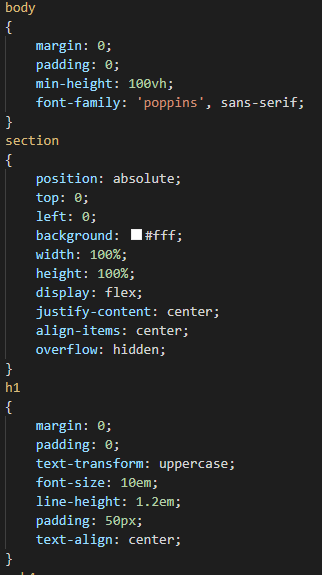
As shown in above fig whenever we move the cursor over black it will turned to right and vice versa for the black the and qr code if scanned will be connected to one of out materials

* **Sub About us html code: -**



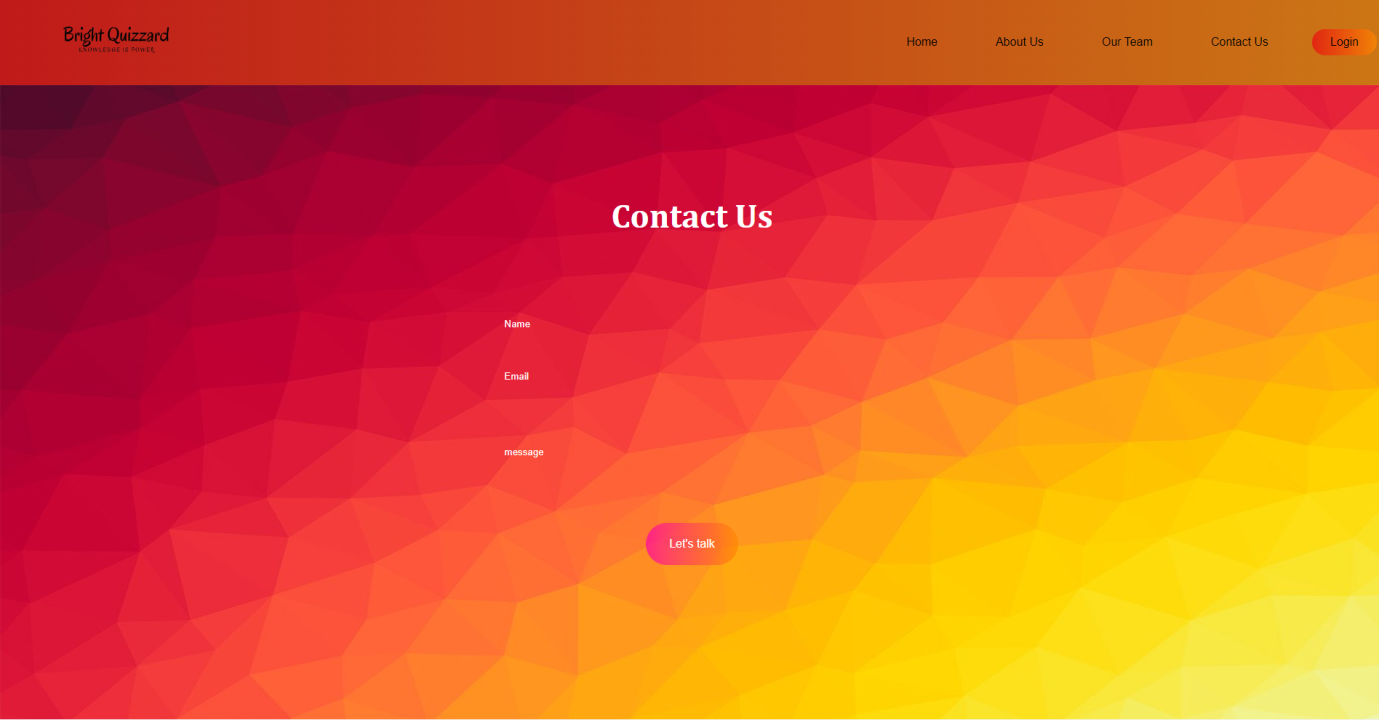
**Figure 8.8**

* **Sub About us CSS code: -**



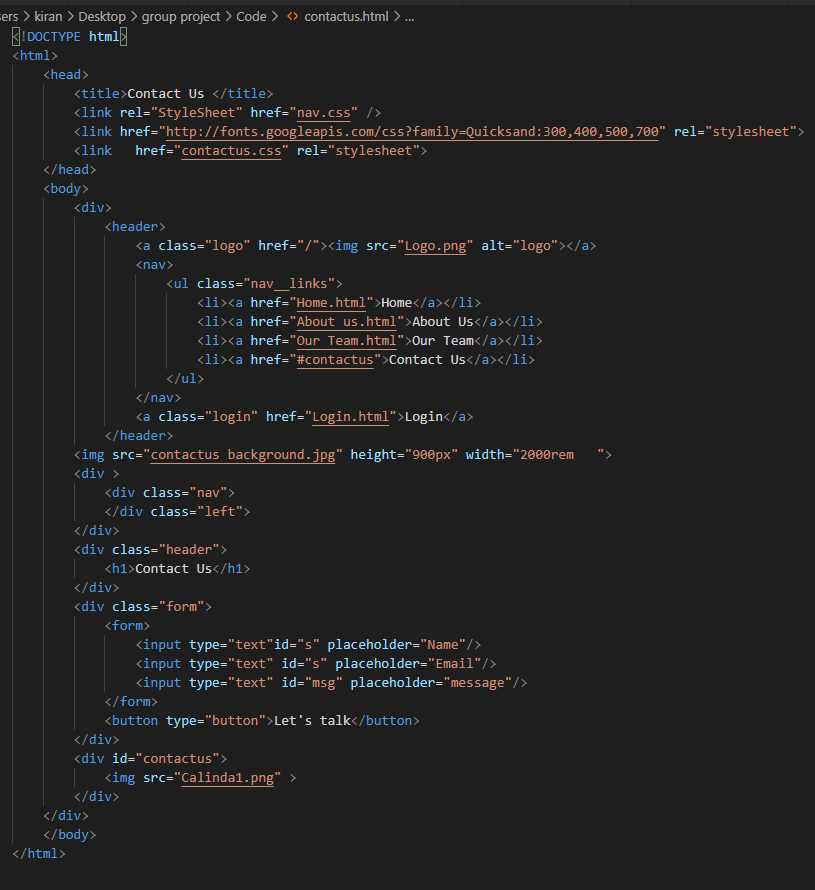
**Figure 8.9.1 Figure 8.9.2**

* **Contact us: -**



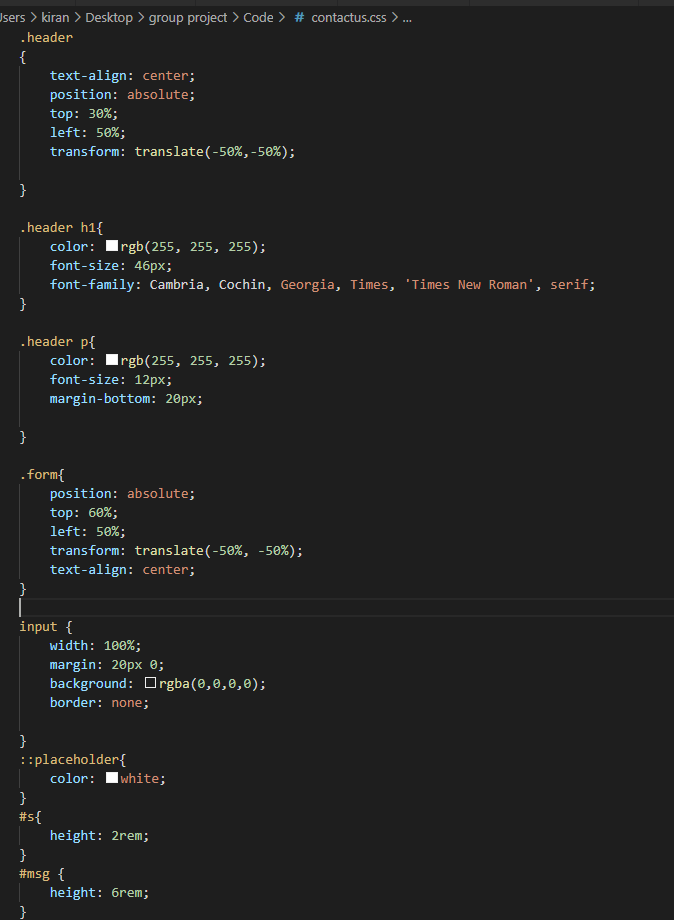
**Figure 8.10**

* **Contact us html code: -**

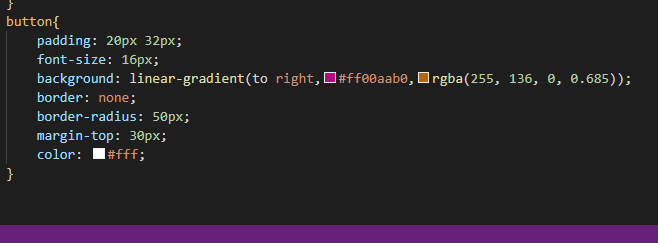


**Figure 8.11**

* **Contact us CSS code: -**



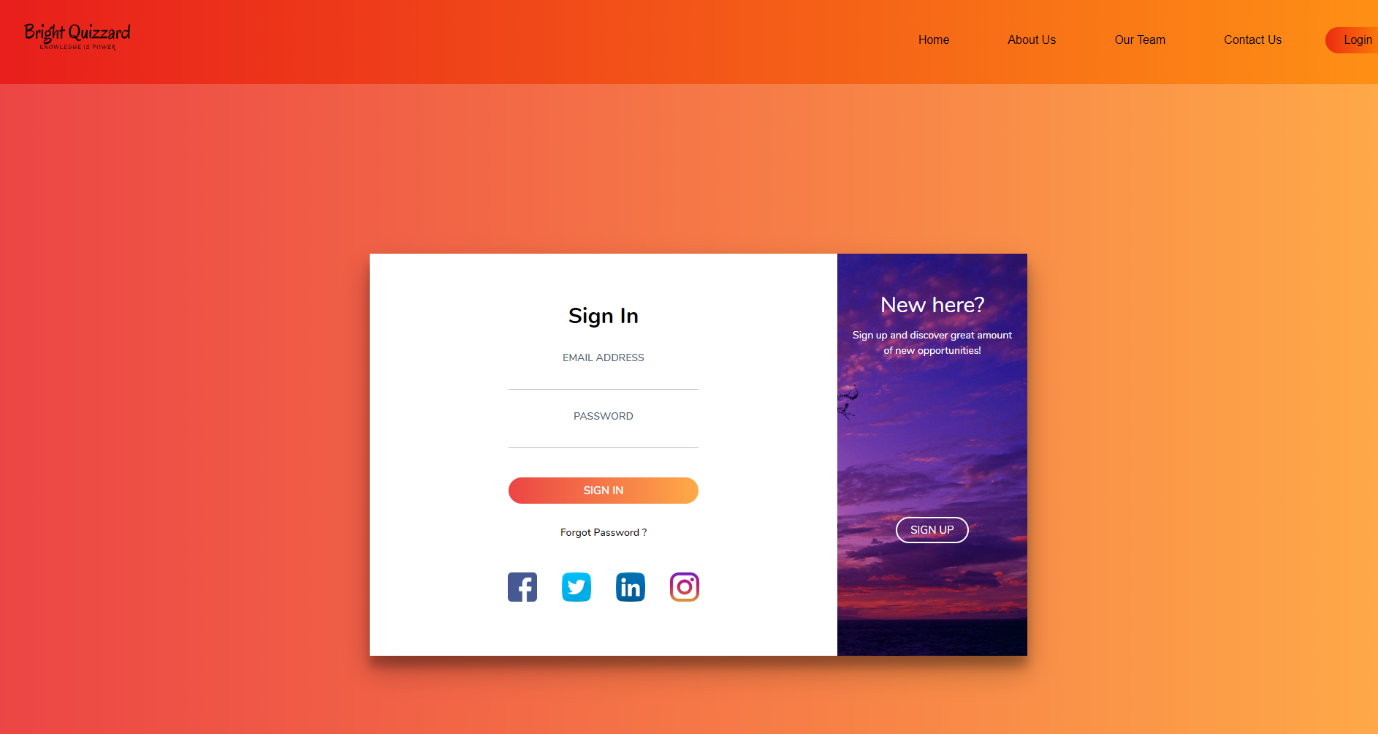
**Figure 8.12**



**Figure 8.13**

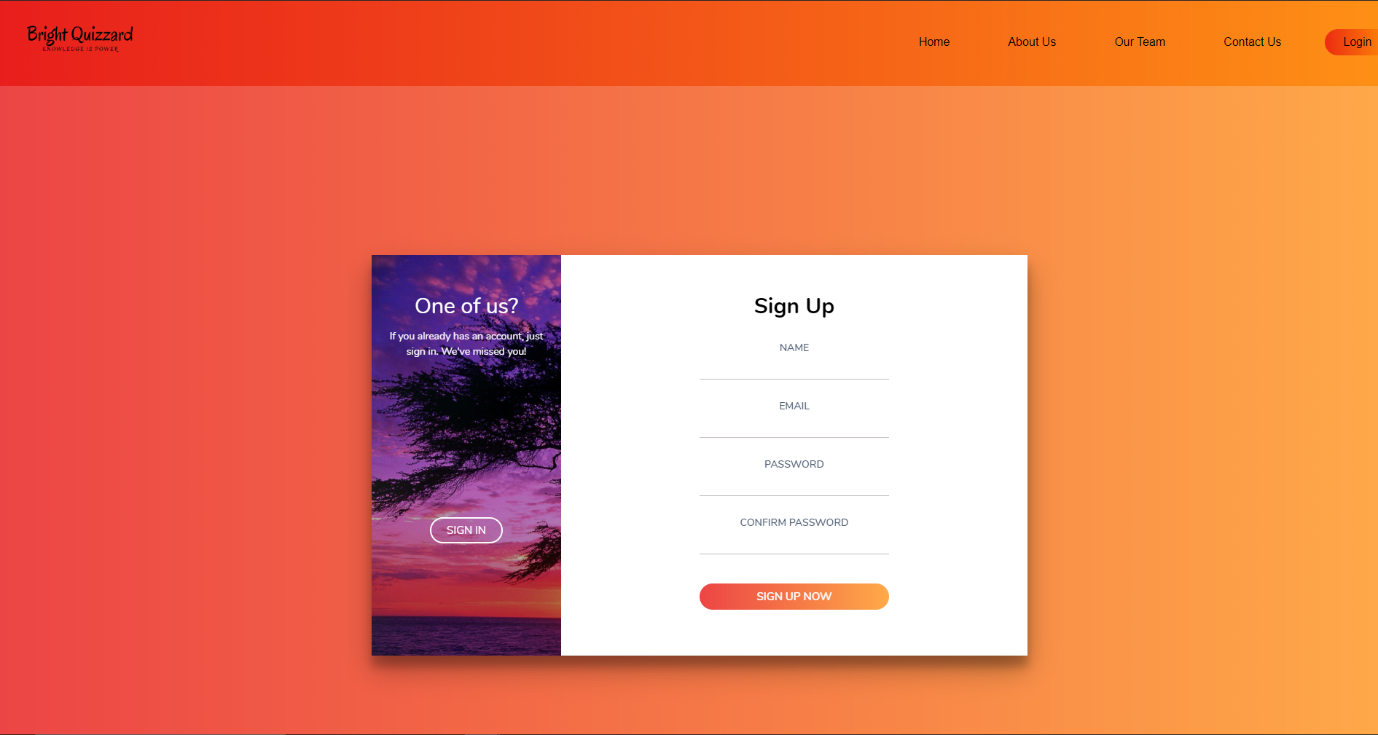
* **Login page: -**

Here in this login page whenever we press on the sign up, the small blue part of the card covered will hovered to the left the sign in page will be hidden as shown in the figure 8.15

****

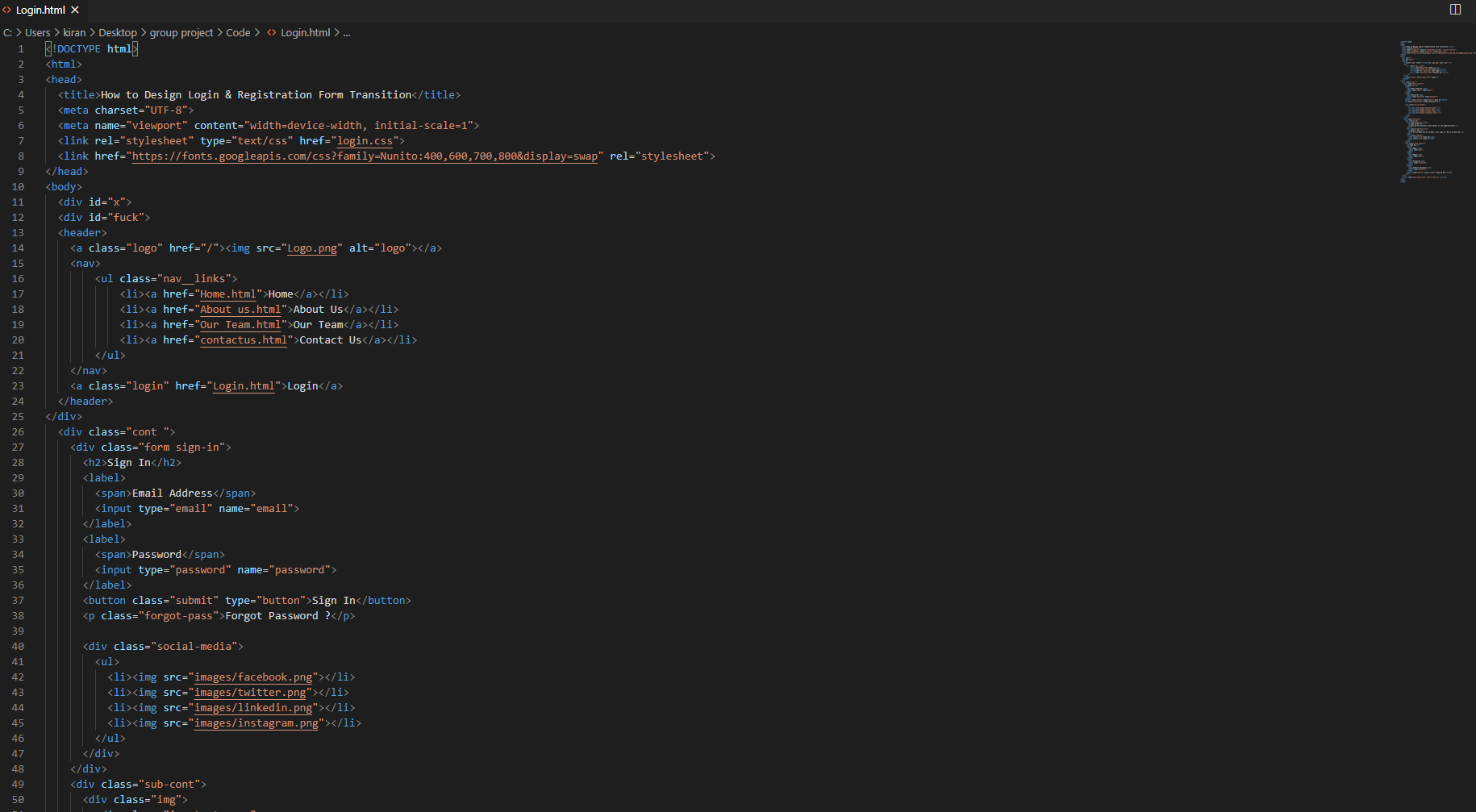
**Figure 8.14**

And here the when we press the sing the same small blue part will be hovered from right to left hiding the registration page as shown in the 8.14 fig and if press forgot password it will be redirected another

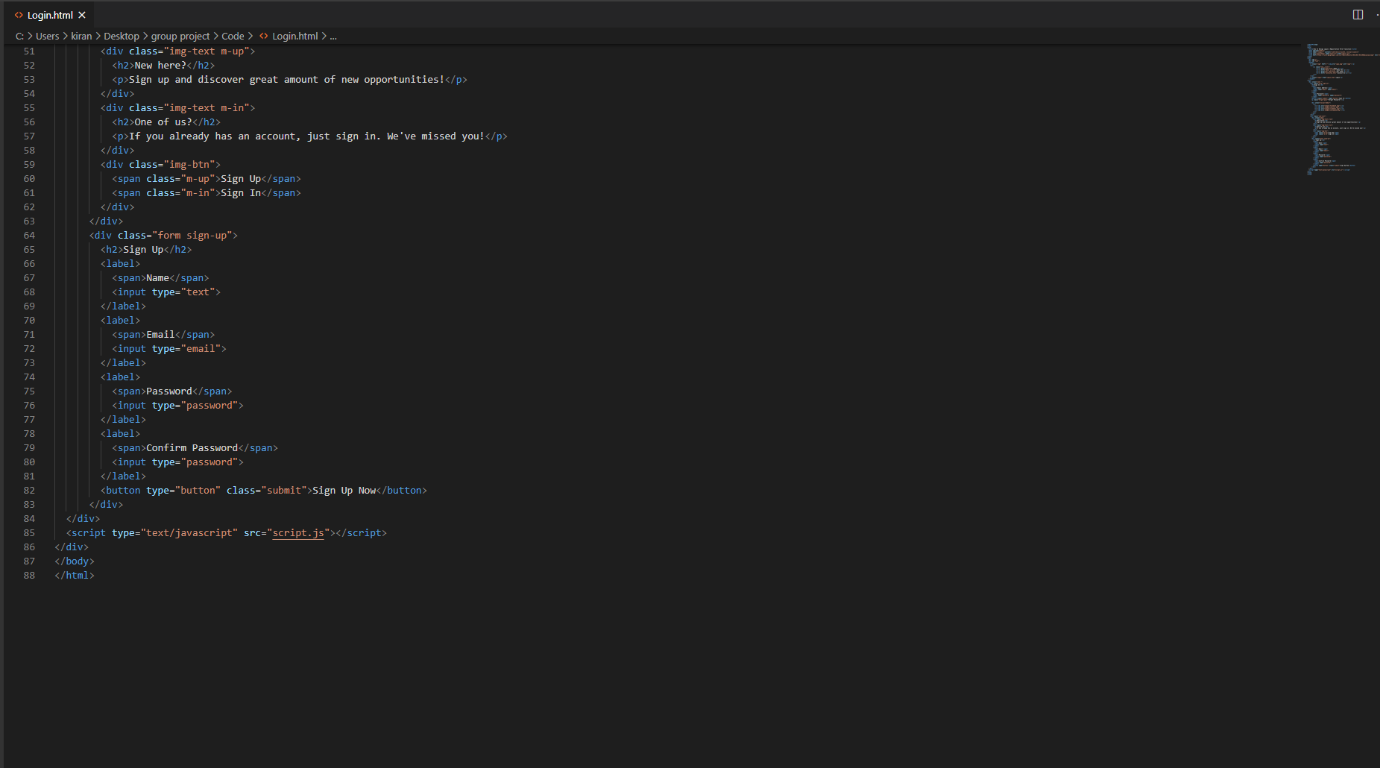
****

**Figure 8.15**

* **Login page html code: -**

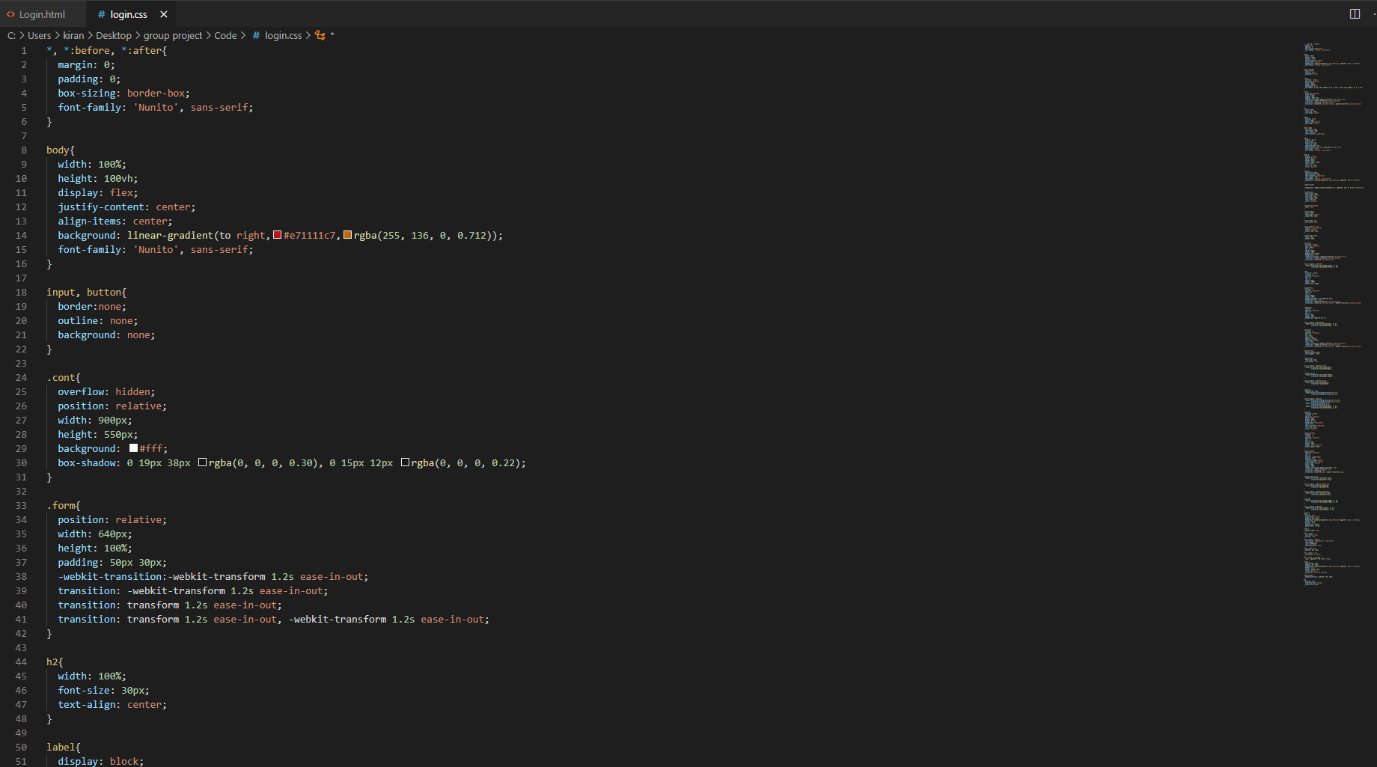
****

**Figure 8.16**

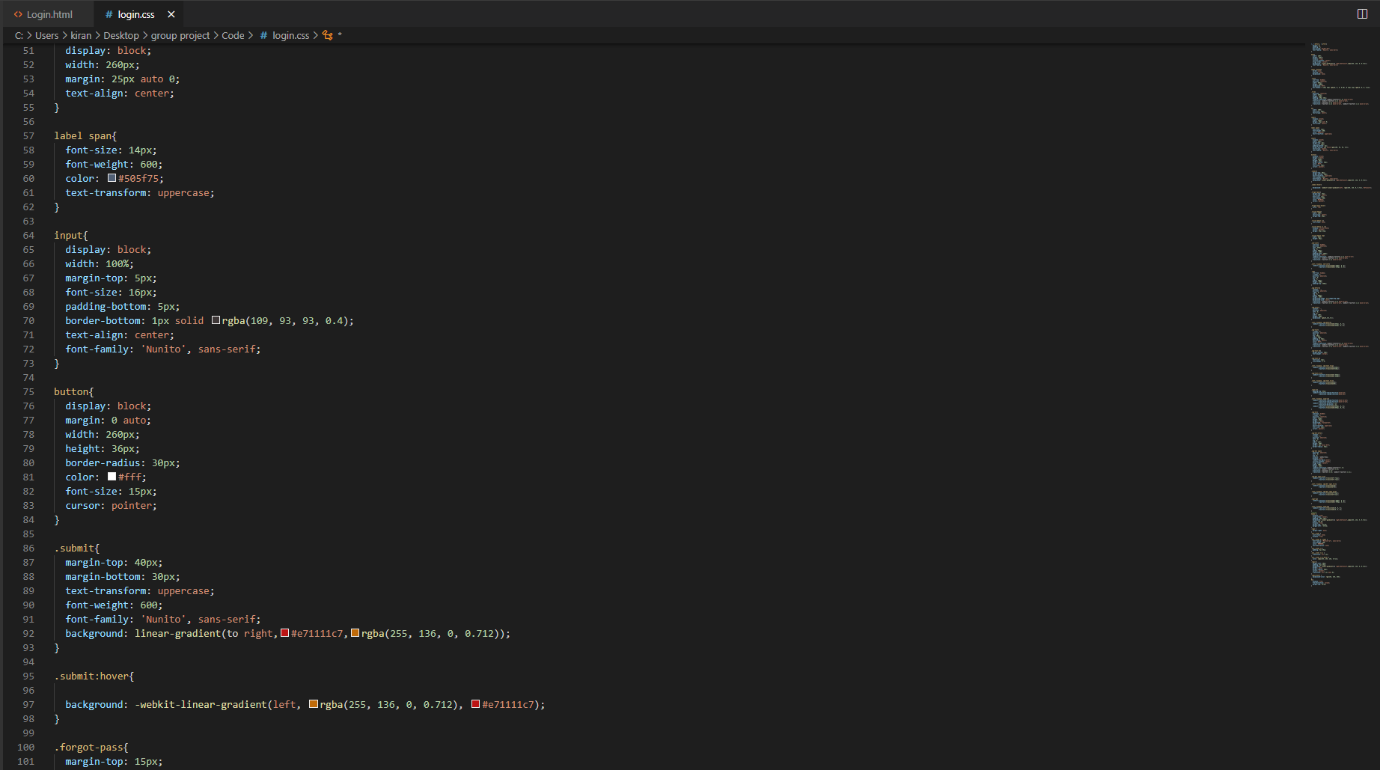
****

**Figure 8.17**

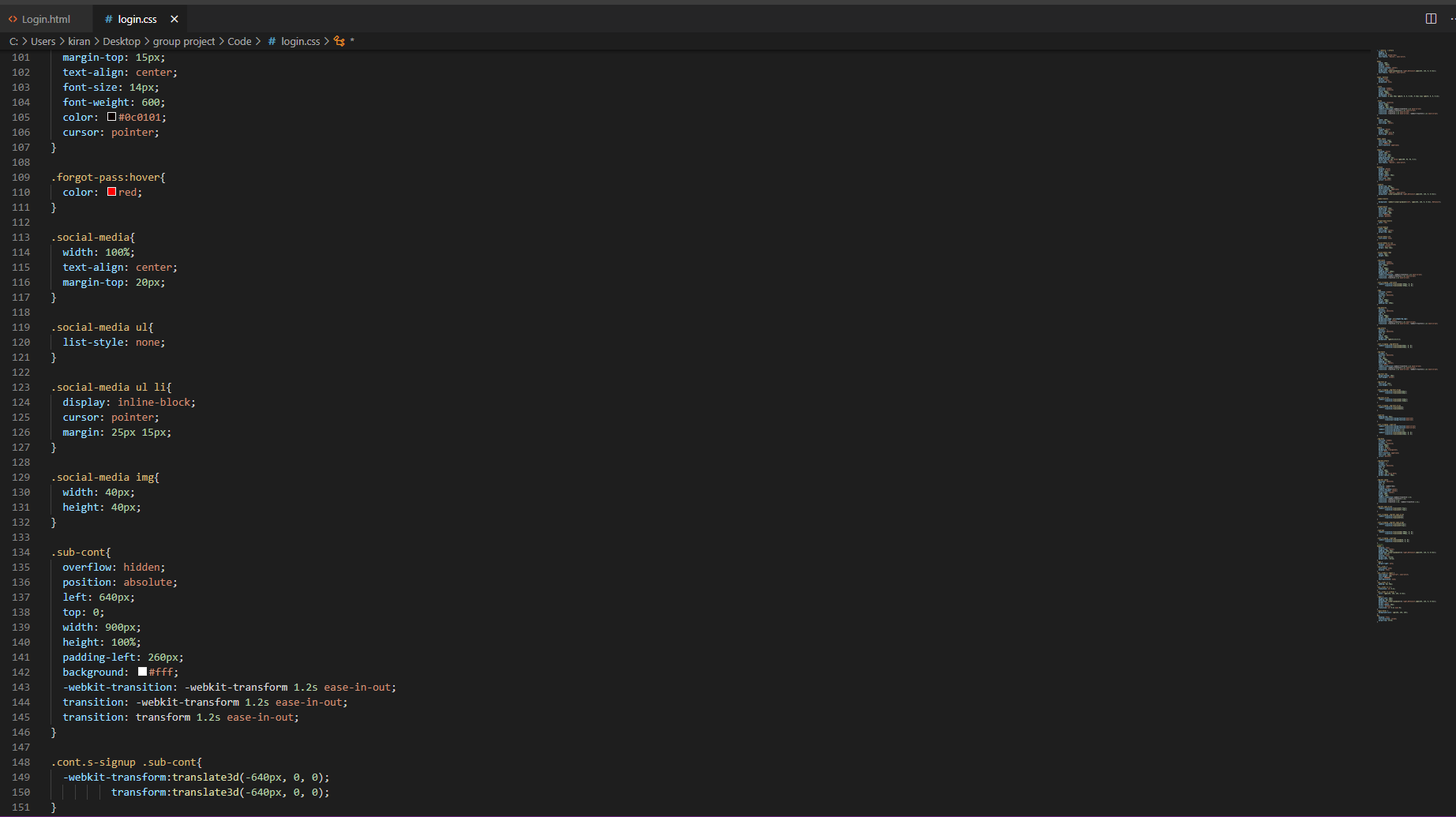
* **Login page with CSS: -**

****

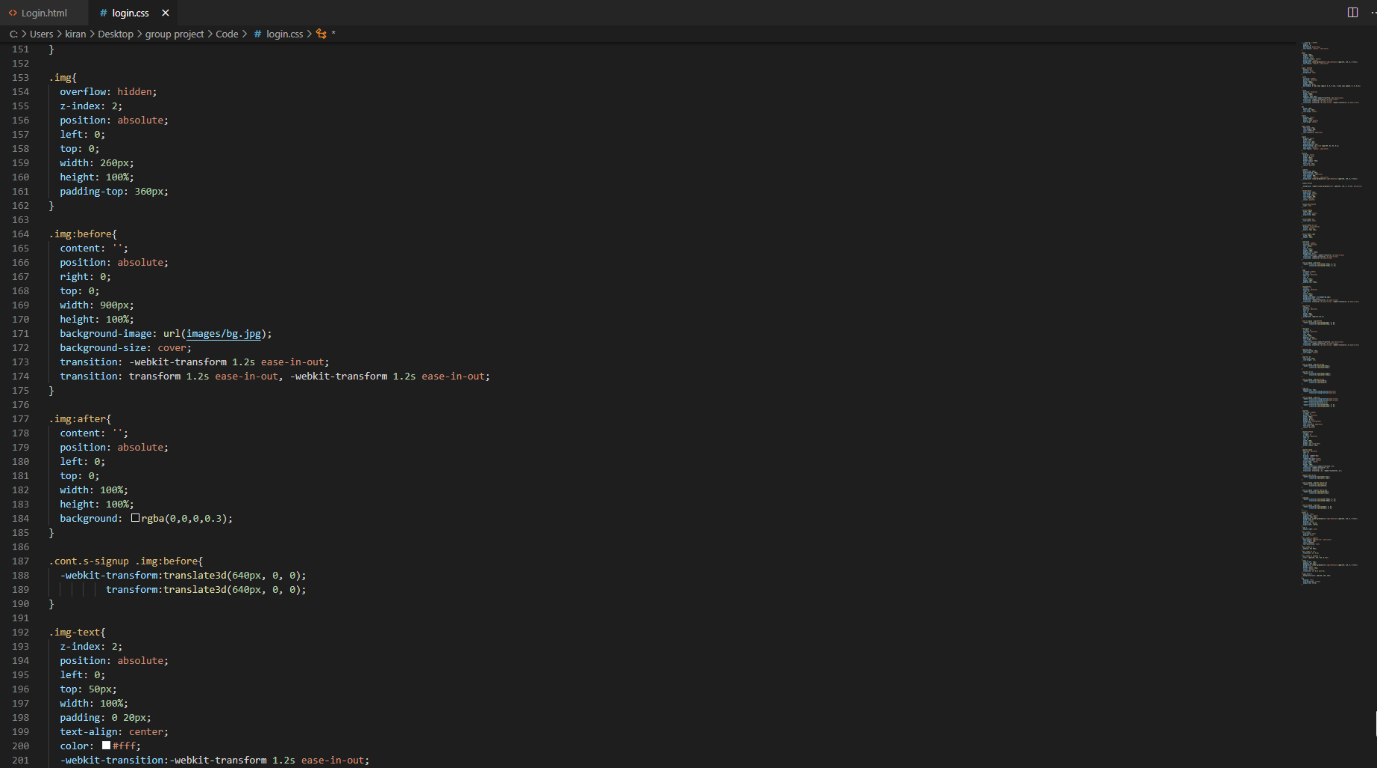
**Figure 8.18**

****

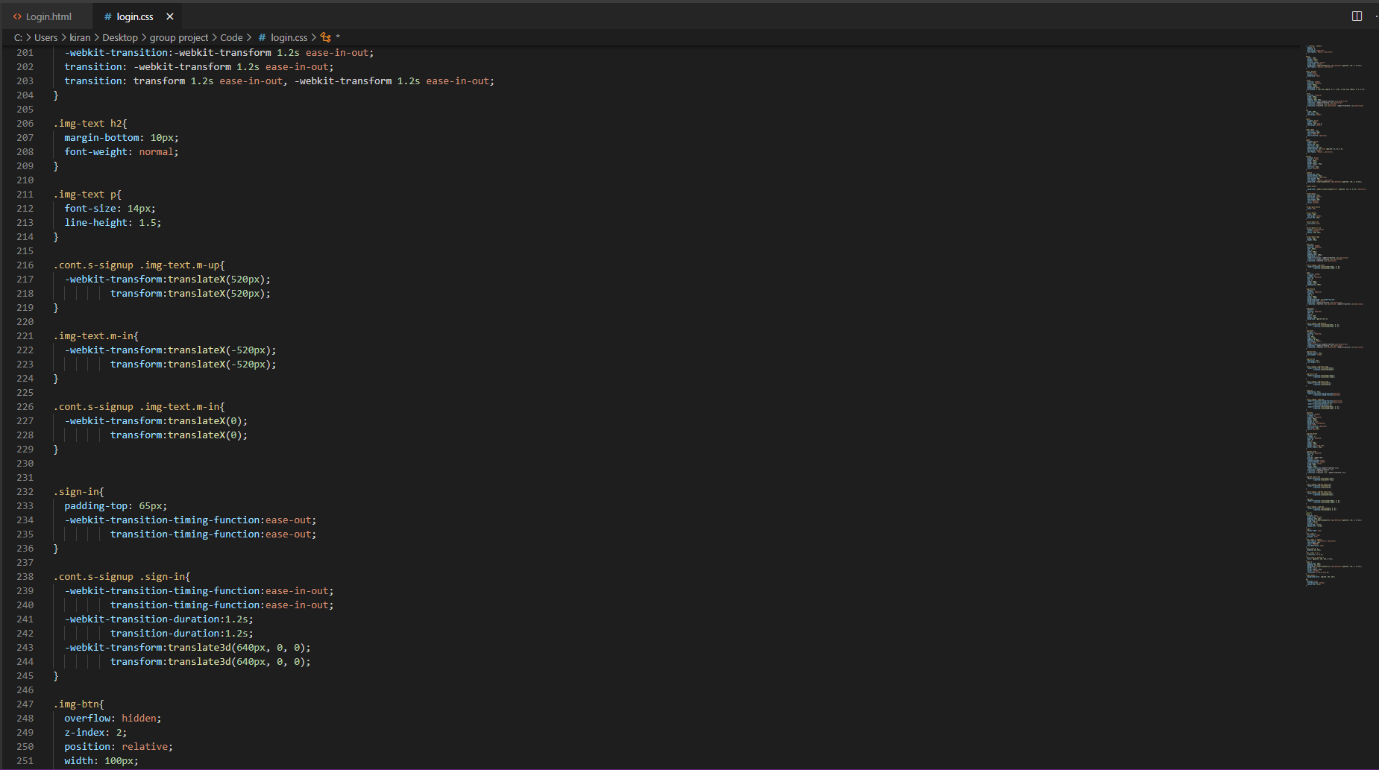
**Figure 8.19**

****

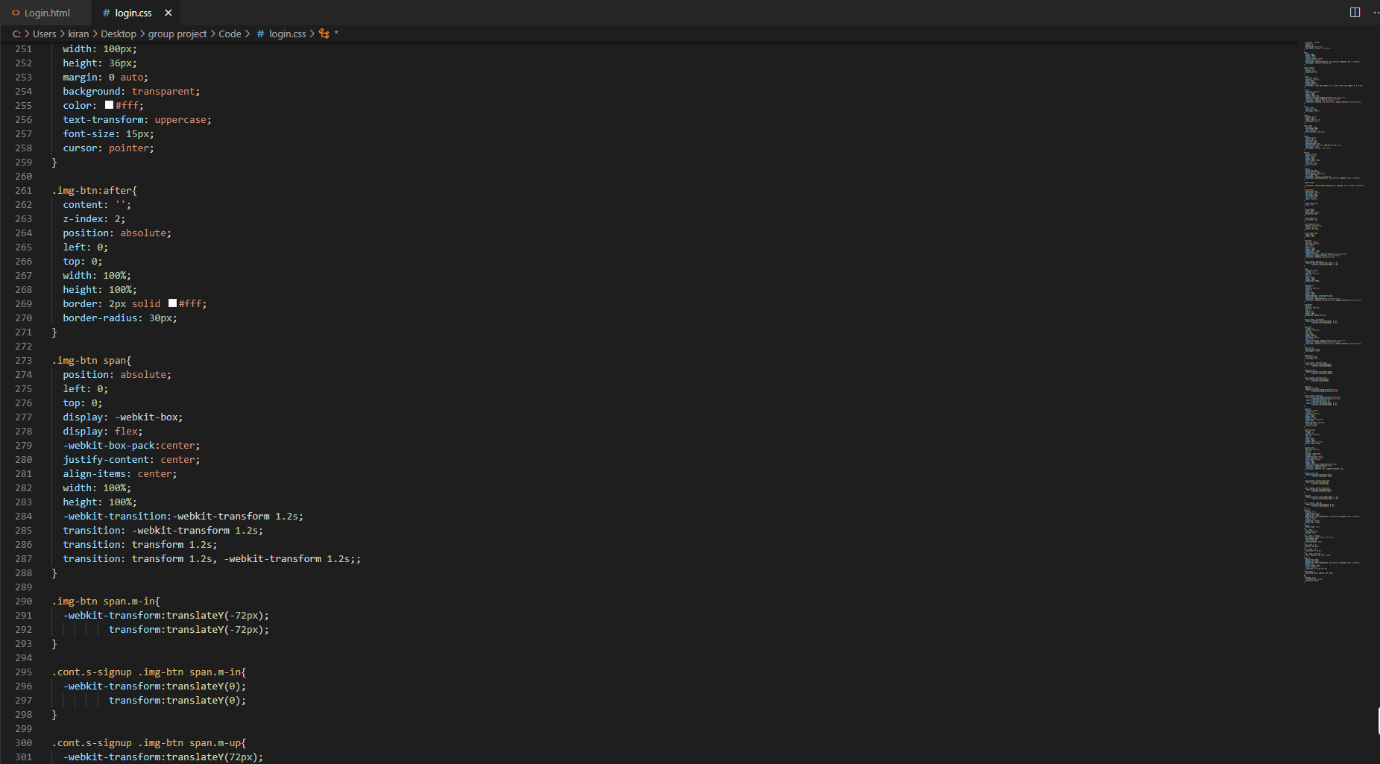
**Figure 8.20**

****

**Figure 8.21**

****

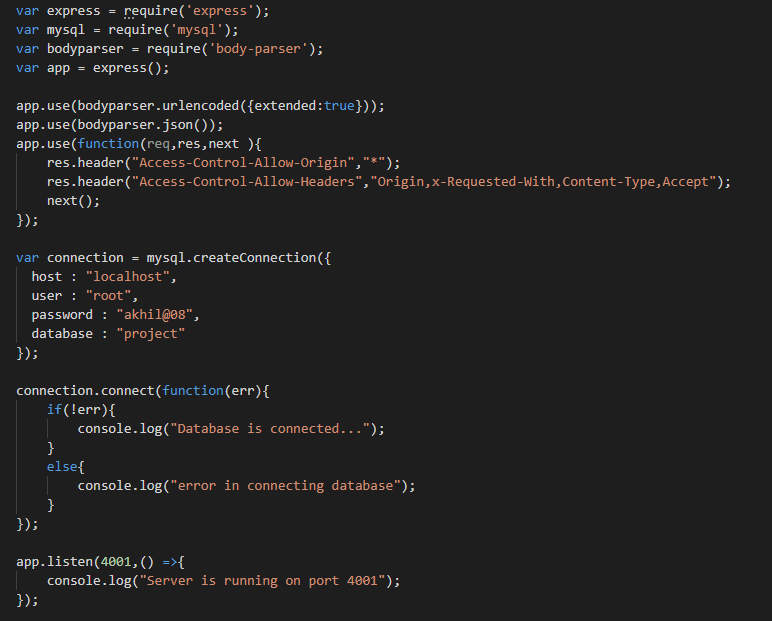
**Figure 8.22**

****

**Figure 8.23**

In order to connect html and JavaScript we use something called API

Below is code for providing a connection to a database



**Figure 8.24**

* **Login**

Validating the given input fields

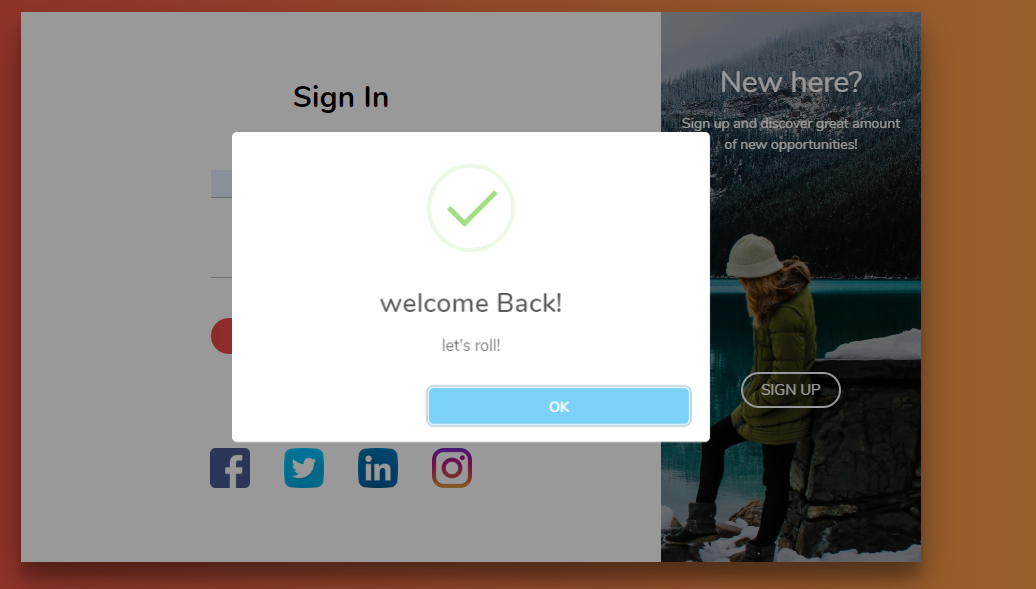
**Case-1: -** here if we give the invalid details a pop –up will be displayed as shown in the fig .

These credentials checked with details which are already inside the database .which were given by user



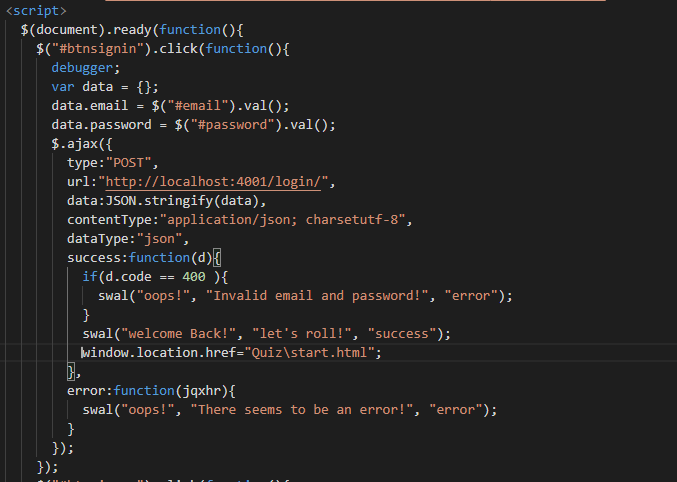
**Figure 8.25**

**Case-2:** here if we give the valid details a pop –up will be displayed as shown in the fig i.e it has matched with database values



**Figure 8.26**

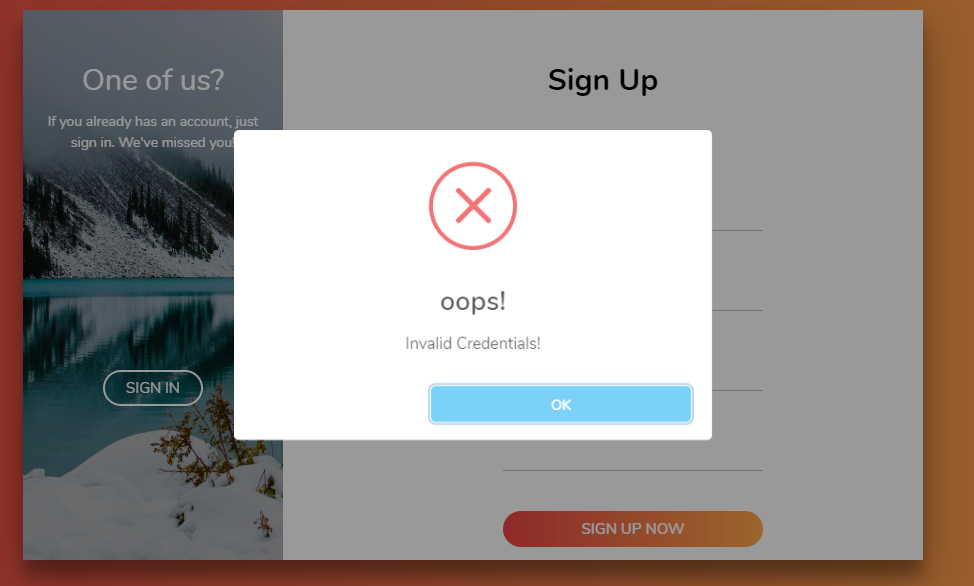
**For sign in:**



**Figure 8.27**

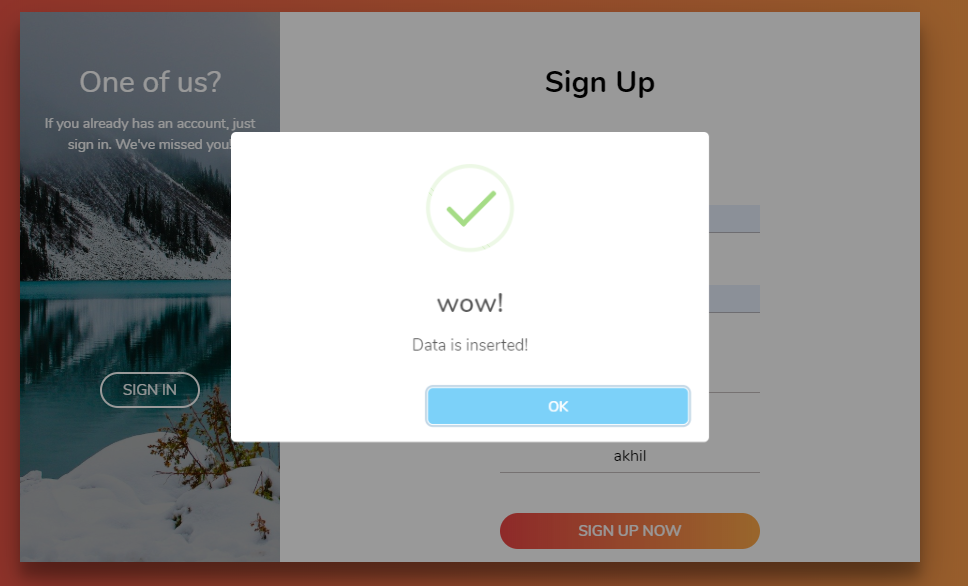
* **Registration:**

**Case 1:** whenever we give an incomplete answer for the text fields below pop-up will be popped



**Figure 8.28**

**Case-2:** when we give a valid answer to all the input fields then a success pop-up will be popped as shown below and then the data will be into the database and then we use those details later



**Figure 8.29**

JavaScript involved in the cases

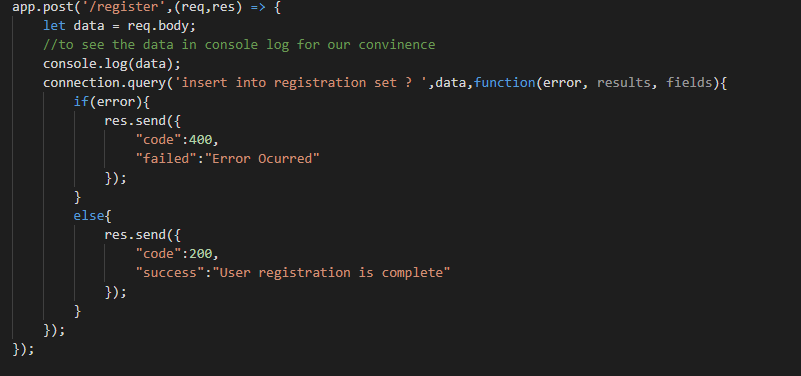
**For sign up:** here this code is used for checking whether they have entered the right credentials



**Figure 8.30**

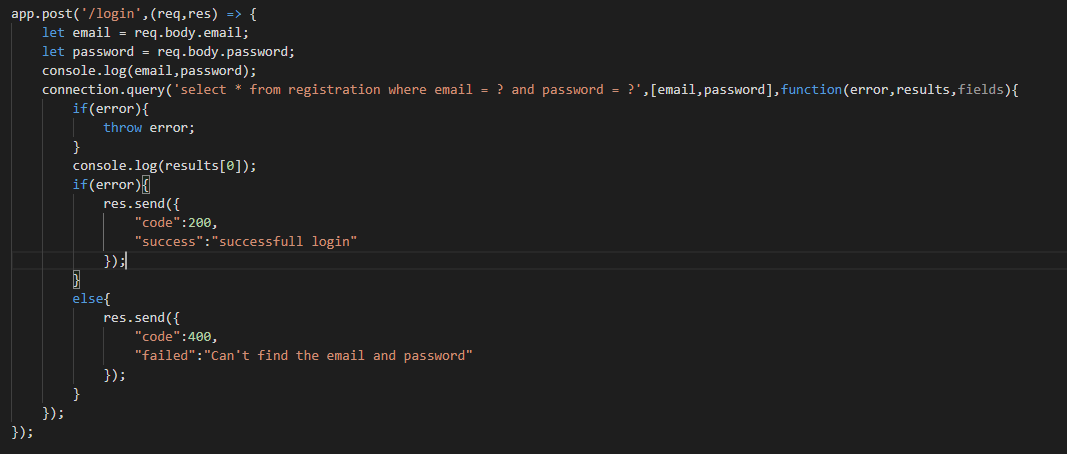
**The java script involved in the database: -**

**For registration: -** here this code is used in api to insert the data which are provided by the user only if they are valid credentials



**Figure 8.31**

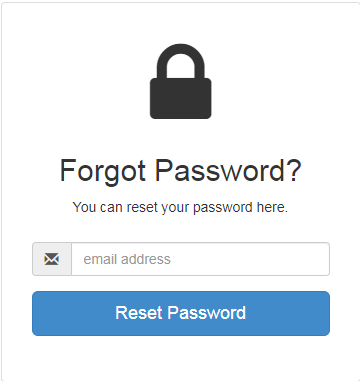
**API sign-Up: -** here this code is used to retrieve data from database and then check with the user given details



**Figure 8.32**

**Forgot password: -**

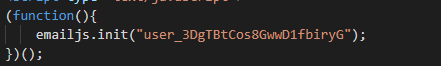
Here this page will be redirected when we click on forgot password on sign in page and then enter a valid mail in order to get a email message



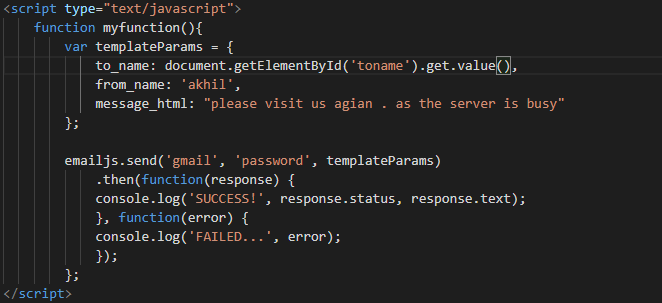
**Figure 8.33**

After we have entered the email in the input field and then on clicking reset password you will you sent an email to the email you have provided default

Below is the JavaScript code for that case:



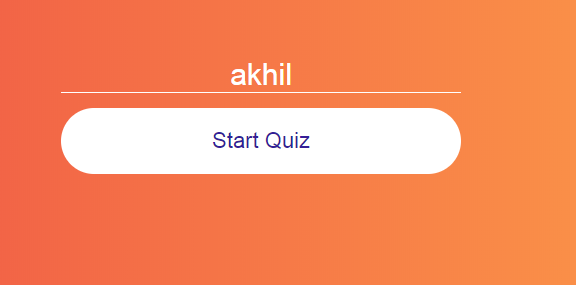
**Figure 8.34**



**Figure 8.35**

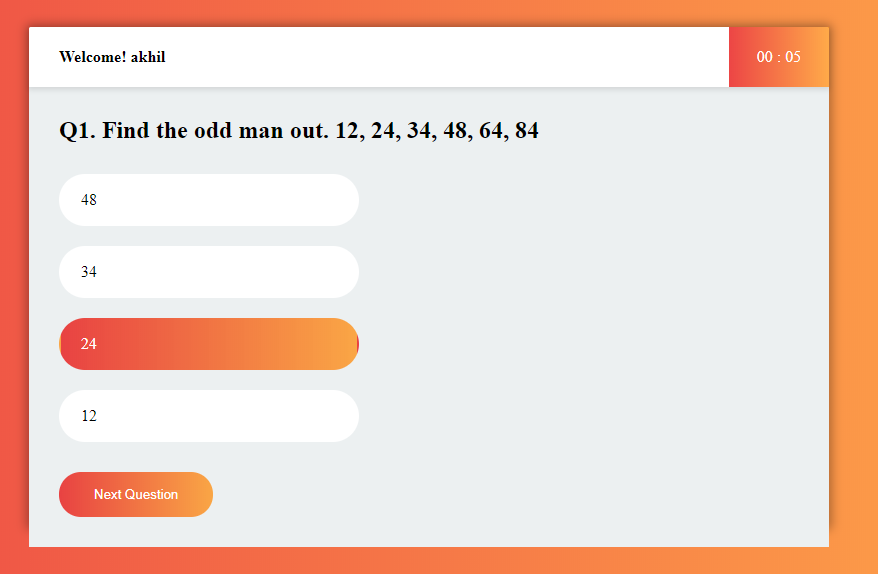
**Quiz: -**

The start page of the quiz is shown below unless you enter the it won’t sart



**Figure 8.36**

After clicking on start quiz, you’ll be entering the quiz as soon you enter a timer will be stared and unless until you select the option and then press next it’s not goanna move towards the next questions



**Figure 8.37**

And the end which will have you score and also time (in sec) you have taken

As shown below



**Figure 8.38**

**The html code for above figures are listed below**

**For start page:**

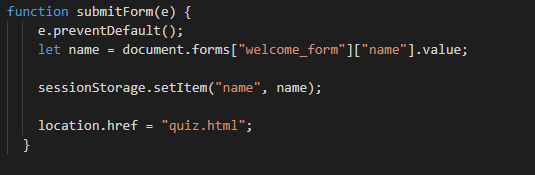
Hereit’s skeleton the start page of quiz



**Figure 8.39**

**Related JavaScript for start page:**

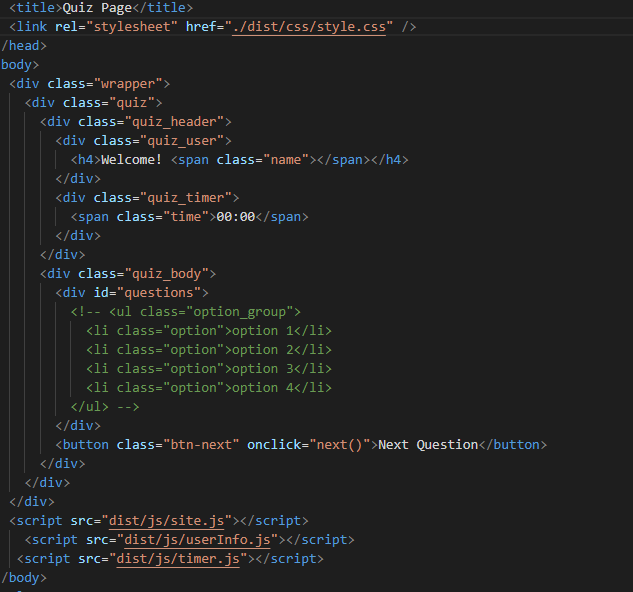
This code is used for start page in order to start the quiz and also to obtain username



**Figure 8.40**

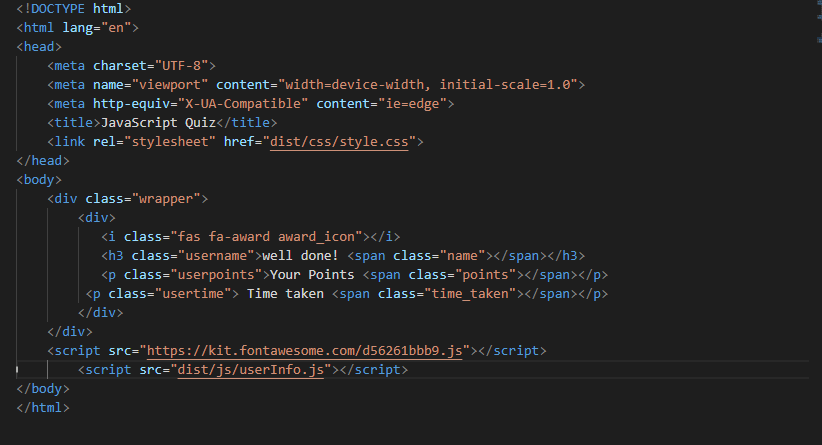
**Html code of quiz page:**

This is skeleton of how the quiz be looking



**Figure 8.41**

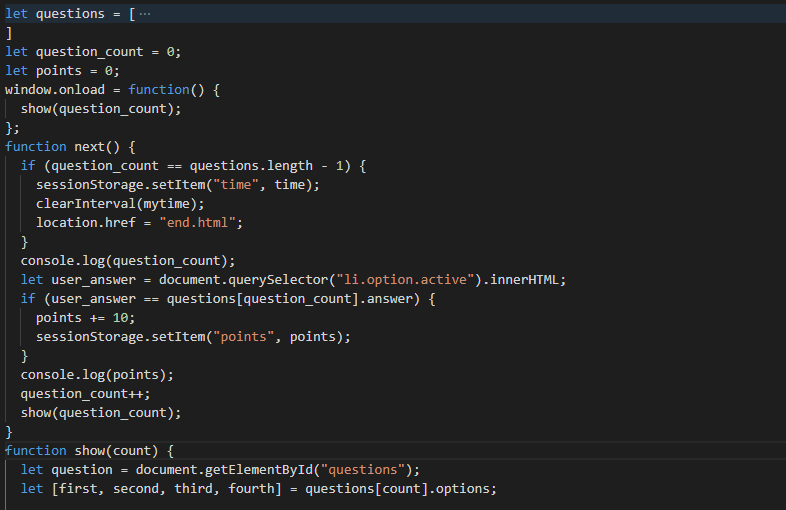
**For end page:**



**Figure 8.42**

The JavaScript for the above quiz pages are listed below in an order:

**JS For quiz:**



**Figure 8.43**

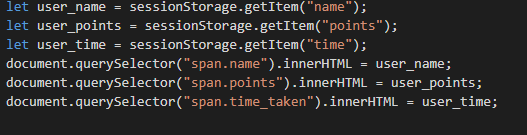
In this quiz page I have fetched questions from a local JSON file but we can always fetch from a database or even from a online database and I will be checking the answer submit by the user is matched with my answer if yes I will be adding 10 points for each correct answer and if wrong no points will be added as shown in the fig 8.48 as well as fig 8.49



**Figure 8.44**

**JS for end page:**

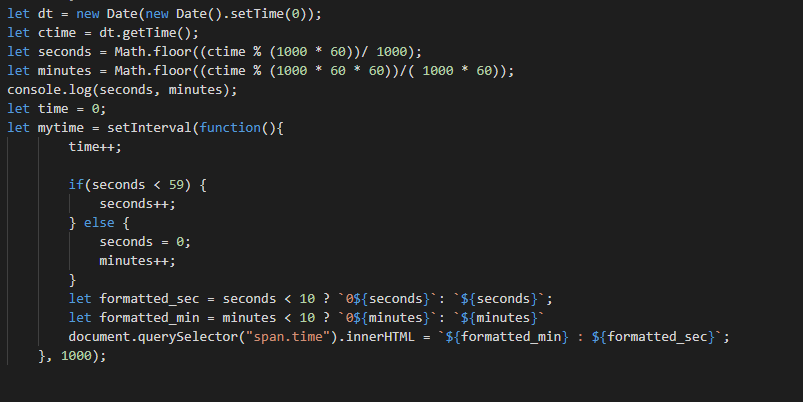
Here in the end page the score you have scored in the quiz page and the time you have will be displayed in this page along with username we have given



**Figure 8.45**

**For timer:**

Here this code used to start calculate and start the time as soon as user visits quiz page and stops when the user completes the quiz and then displayed in the end page



**Figure 8.46**

**CHAPTER-9**

**MAINTENANCE**

The maintenance phase involves making changes to hardware, software, and documentation to support its operational effectiveness. It includes making changes to improve a system’s performance, correct problems, enhance security, or address user requirements. To ensure modifications do not disrupt operations or degrade a system’s performance or security, organizations should establish appropriate change management standards and procedures.

Routine changes are not as complex as major modifications and can usually be implemented in the normal course of business. Routine change controls should include procedures for requesting, evaluating, approving, testing, installing, and documenting website modifications. Maintaining accurate, up-to-date hardware and software inventories is a critical part of all change management processes. Management should carefully document all modifications to ensure accurate system inventories. Management should coordinate all technology related changes through an oversight committee and assign an appropriate party responsibility for administering software patch management programs. Quality assurance, security, audit, regulatory compliance, network, and end-user personnel should be appropriately included in change management processes. Risk and security review should be done whenever a system modification is implemented to ensure controls remain in place.

For maintenance of the website:

1. The database has to be updated regularly according to new available information.
2. Redundant and false information must be removed from the database.

Newer version of PHP and MYSQL can be used for up gradation of website and to improve the overall performance of the system.

**CHAPTER-10**

**FUTURE SCOPE & FUTURE ENHANCEMENT**

**PROJECT NAME**

**BRIGHT QUIZZARD**

1. Bright Quizzard would help each and every person to improve their knowledge via our website.
2. It would provide huge collection of Questions of all fields.
3. We will be providing questions with great knowledge.

**CHAPTER-11**

**CONCLUTION**

We have successfully implemented the site ‘Bright Quizzard’. With the help of various links and tools, we have been able to provide a site which will be live soon and running on the web. We have been successful in our attempt to take care of the needs of both the user as well as the administrator. Finally, we hope that this will go a long way in popularizing.

**BIBLIOGRAPHY**

1. www.javatpoint.com
2. [www.w3schools.com](http://www.w3schools.com/)
3. [www.getbootstrap.com](http://www.getbootstrap.com/)
4. [www.codeigniter.com](http://www.codeigniter.com/)
5. <https://codepen.io/>
6. [www.stackoverflow.com](http://www.stackoverflow.com/)
7. [www.fontawesome.io](http://www.fontawesome.io/)
8. [www.php.net](http://www.php.net/)
9. Head first with PHP (Lynn Beighley and Michael Morrison)
10. Learn HTML and CSS faster (Mark Myers)
11. Wikipedia

**Git Hub :** <https://github.com/Akhil007-git/FSDProject>