

A
PROJECT REPORT
On
“TriviaClickz – Online Quiz”
SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE AWARD OF THE DEGREE
OF
Bachelor of Computer Application



(Session 2021-2024)

ided By
oma Jaisinghani
st. Professor

Submitted By
Sangram Singh Bhati
BCA IIIrd Year

Department of Computer Science
VISION COLLEGE OF COMMERCE
Udaipur Road, CHITTORGARH

(Affiliated to MOHAN LAL SUKHADIYA UNIVERSITY, UDAIPUR)

Email: vision_mgmt@yahoo.com

Website: www.visionmanagement.org

STATEMENT OF ORIGINALITY

In accordance with the requirements for the degree of **Bachelor's in Computer Application**, in Faculty of **Computer Science Application**, I present this report Entitled "**TriviaClickz – Online Quiz**". This report is complete under the Supervision of **Mrs. Sapna Jaisinghani (Asst. Professor)**.

I declare that the work presented in the report is my own work except as acknowledged in the text and footnotes, and that to my knowledge this material has not been submitted either in whole or in part, for a degree at this School or at any other such Institution.

This work has not submitted elsewhere for award of other degree.

Date: .../.../.....

Name & Signature of Student

Place: Chittorgarh

(BCA IIIrd Year)

CERTIFICATE

VISION COLLEGE OF COMMERCE



*This is to certify that the work which is being presented in PROJECT entitled “TriviaClickz – Online Quiz” Submitted by **Sangram Singh Bhati** Student of Final Year Bachelor of Computer Application (BCA) in partial fulfilment for award of degree of Bachelor of Computer Application (2021 - 2024) is a record of students work carried out by them under our guidance and supervision of Mrs. Sapna Jaisinghani Faculty of Computer Science Department.*

This work has not been submitted elsewhere for award of any other degree.

Date:

Place:

Signature of Project Guide

Signature of Project Coordinator

Mrs. Sapna Jaisinghani

(Asst. Professor)

Mrs. Sapna Jaisinghani

(Asst. Professor)

ACKNOWLEDGEMENT

It is a pleasure to represent my report on “**TriviaClickz – Online Quiz**” as a project. I would like to express my deep gratitude to my teach instructor and guide Asst. Professor, **Mrs. Sapna Jaisinghani (Asst. Professor)**, for giving us all necessary encouragement and guidance for overcoming all the difficulty that I had faced in the task. I thank him to guiding us from beginning to the end.

I am highly graceful to **Mr. Sumit Sir (Asst. Professor)** for providing me this opportunity and support.

I am highly indebted to **All the Faculty member of Computer Science Department** for their cooperation, guidance Support.

I am also grateful to my friends who always stood by my side giving their worthily advice in making the project.

Last but not the least to express my indebtedness to **Mrs. Sapna Jaisinghani (Asst. Professor)** the coordinator of the project for his constant motivation regarding this project.

ABSTRACT

Welcome to “**TriviaClickz**”, where knowledge meets excitement! Our platform is designed to ignite your curiosity and expand your horizons through the power of quizzes. Whether you're a trivia enthusiast, a lifelong learner, or just looking for a fun way to challenge yourself, “**TriviaClickz**” is the perfect destination.

With a diverse range of topics spanning from history and science to pop culture and beyond, our quizzes are crafted to engage, educate, and entertain. Challenge your friends, compete for high scores, and unlock achievements as you embark on a journey of enlightenment. Join our community of inquisitive minds and embark on a quest for knowledge like never before. Get ready to sharpen your intellect and have a blast while doing it. Let's start quizzing!

Table of Contents

| | |
|---|------|
| STATEMENT OF ORIGINALITY | i |
| CERTIFICATE | ii |
| ACKNOWLEDGEMENT | iii |
| Abstract | iv |
| Table of Contents | v |
| List of Figures | vii |
| List of Tables | viii |
| CHAPTER 1 : INTRODUCTION | |
| 1.1 Aim | 1 |
| 1.2 Objectives | 1 |
| 1.3 Scope of Project | 2 |
| 1.4 Conclusion | 2 |
| CHAPTER 2 : HARDWARE AND SOFTWARE REQUIREMENTS | |
| 2.1 Introduction | 3 |
| 2.2 System Environment | 3 |
| CHAPTER 3 : SYSTEM ANALYSIS | |
| 3.1 Introduction | 5 |
| 3.2 User Requirements | 5 |
| 3.3 System Features | 6 |
| 3.4 Conclusion | 6 |
| CHAPTER 4 : OVERVIEW AND TECHNOLOGY USED | |
| 4.1 Front-End Technology | |
| 4.1.1 HTML | 7 |
| 4.1.2 CSS | 11 |
| 4.1.3 JavaScript | 13 |
| 4.2 Back-End Technology | |
| 4.2.1 PHP Introduction | 20 |
| CHAPTER 5 : SYSTEM AND DATABASE DESIGN | |
| 5.1 System Design | |
| 5.1.1 Flow Chart Diagram | 30 |

| | |
|-----------------------------------|----|
| 5.2 Database Design | |
| 5.2.1 Data Dictionary | 33 |
| 5.2.2 Table Structure of Database | 33 |
| CHAPTER 6 : PROJECT OUTPUT SCREEN | |
| 6.1 Website Home Page | 35 |
| 6.2 User Registration Page | 35 |
| 6.3 User Login Page | 36 |
| 6.4 Quiz Customize Page | 36 |
| 6.5 Question Page | 37 |
| 6.6 Result Page | 37 |
| 6.7 Leaderboard of Users | 38 |
| 6.8 About Us Page | 38 |
| 6.9 Contact Us Page | 39 |
| CONCLUSION | 40 |
| FUTURE SCOPE | 41 |
| REFERENCES | 43 |

List of Figures

| | | |
|-----|--------------------------------|----|
| 5.1 | User Registration | 30 |
| 5.2 | User Login | 31 |
| 5.3 | User Interaction | 32 |
| 5.4 | Table Structure of Database | 33 |
| 5.5 | Structure of contact_tbl Table | 33 |
| 5.6 | Structure of fmail_tbl Table | 34 |
| 5.7 | Structure of scores_tbl Table | 34 |
| 5.8 | Structure of signup_tbl Table | 34 |
| 5.1 | Website Home Page | 35 |
| 5.2 | User Registration Page | 35 |
| 5.3 | User Login Page | 36 |
| 5.4 | Quiz Customize Page | 36 |
| 5.5 | Question Page | 37 |
| 5.6 | Result Page | 37 |
| 5.7 | Leaderboard of Users | 38 |
| 5.8 | About Us Page | 38 |
| 5.9 | Contact Us Page | 39 |

List of Tables

| | | |
|----|------------------------|---|
| .1 | Hardware Configuration | 3 |
| .2 | Software Configuration | 4 |

CHAPTER 1

INTRODUCTION

TriviaClickz, where knowledge meets excitement! Our platform is designed to ignite your curiosity and expand your horizons through the power of quizzes. Whether you're a trivia enthusiast, a lifelong learner, or just looking for a fun way to challenge yourself, "**TriviaClickz**" is the perfect destination.

With a diverse range of topics spanning from history and science to pop culture and beyond, our quizzes are crafted to engage, educate, and entertain. Challenge your friends, compete for high scores, and unlock achievements as you embark on a journey of enlightenment. Join our community of inquisitive minds and embark on a quest for knowledge like never before. Get ready to sharpen your intellect and have a blast while doing it. Let's start quizzing!

1.1 Aim

- i. A platform where users can play quiz free of cost.

1.2 Objectives

- i. The objective of "**TriviaClickz**" is to provide a dynamic and engaging platform for individuals of all ages and backgrounds to enhance their knowledge through interactive quizzes.
- ii. **Facilitate Learning through Quizzing:** To offer a fun and effective way for users to acquire knowledge across a wide range of subjects, fostering a lifelong love for learning.
- iii. **Promote Engagement and Interactivity:** To create an environment that encourages active participation, where users can challenge themselves and others, fostering a sense of achievement and competition.
- iv. **Cater to Diverse Interests and Fields:** To offer a vast array of quizzes covering diverse topics, ensuring that users can find quizzes that align with their specific interests and areas of curiosity.
- v. **Foster a Sense of Community:** To provide a space for users to connect, discuss, and share knowledge, creating a supportive community of like-minded individuals passionate about learning.
- vi. **Encourage Continuous Improvement:** To track user progress and provide feedback, motivating individuals to consistently expand their knowledge base and achieve personal growth.

1.3 Scope of Project

i. Site access features:

- User must be logged in to access the Quiz.
- For signup user is required to give username, e-mail address and password.
- For login the user will be required to enter Email and password only.

ii. Features of the quiz:

- All questions are multiple choice question.
- Questions are displayed randomly for every user.
- If the user by-mistake presses refresh or go back to the previous page, there will be a new question for the user and the question he/she was on will be marked as attempted.
- A message will be displayed after every attempted question whether the answer was correct or incorrect.

iii. Leaderboard features:

- Leaderboard is a shorted list according to the score obtained by the users.
- If two users are having same score, the user who has signed up earlier will have good ranking than the one who joined late.
- Leaderboard is not open to all. login required.

1.4 Conclusion

In conclusion, “**TriviaClickz**” stands as a beacon of knowledge and learning in the digital landscape. Through its diverse array of quizzes and engaging user experience, it has successfully transformed the way individuals acquire and expand their knowledge. The platform's emphasis on interactivity, community-building, and gamification has not only made learning enjoyable but also fostered a competitive spirit among users.

The positive feedback, growing user base, and impressive metrics are a testament to “**TriviaClickz**” effectiveness in achieving its objectives. The thriving community of inquisitive minds, active discussions, and shared insights exemplify the collaborative learning environment it has cultivated.

As “**TriviaClickz**” continues to evolve, it remains committed to providing a space where curiosity knows no bounds and learning knows no limits. With a foundation built on accuracy, relevance, and user satisfaction, the platform is poised to leave an enduring mark on the landscape of online education.

In the grand tapestry of knowledge acquisition, “**TriviaClickz**” has woven a vibrant thread, enriching the educational journeys of countless individuals. It is a celebration of curiosity, a testament to the power of gamified learning, and a beacon for those who seek to expand their horizons. Embracing the future, “**TriviaClickz**” will undoubtedly continue to shape the way we learn, one quiz at a time.

CHAPTER 2

HARDWARE AND SOFTWARE REQUIREMENTS

2.1 Introduction

In this chapter we mentioned the software and hardware requirements, which are necessary for successfully running this system. The major element in building systems is selecting compatible hardware and software. The system analyst must determine what software package is best for the Online Quiz platform, where software is not an issue, the kind of hardware and peripherals needed for the final conversion.

2.2 System Environment

After analysis, some resources are required to convert the abstract system into the real one. All the resources, which accomplish a robust the hardware and software selection begins with requirement analysis, followed by a request for proposal and vendor evaluation. Software and real system are identified. According to the provided functional specification all the technologies and its capacities are identified. Basic functions and procedures and methodologies are prepared to implement. Some of the Basic requirements such as hardware and software are described as follows.

2.2.1 Hardware Configuration

| | |
|---------------------|---------------------------------------|
| System Processors | Intel Core 2 Duo Processors or Higher |
| Minimum RAM | 2 GB RAM |
| Minimum Storage | 30 GB HDD |
| Required Resolution | 1280*768 Resolution LCD |

Table 2.1 Hardware Configuration

2.2.2 Software Configuration

| | |
|----------------------------------|-----------------------|
| Operating System | Windows 7 or Higher |
| Platform Application | XAMPP |
| Web Server | Apache |
| DB Interface | phpMyAdmin |
| Programming Language | HTML, CSS, JavaScript |
| Server-Side Programming Language | PHP 8.2.0 |
| Back-End DB | MySQL |
| IDE | Visual Studio Code |
| Browser | Google Chrome |

Table 2.2 Software Configuration

CHAPTER 3

SYSTEM ANALYSIS

3.1 Introduction

The purpose of this system analysis is to provide a comprehensive understanding of the requirements and functionalities of an online quiz website. The online quiz website is designed to facilitate the creation, management, and participation in quizzes, exams, and assessments. This analysis will cover various aspects of the system, including user requirements, system features, data management, security, and scalability.

3.2 User Requirements

3.2.1 User Roles

- **Administrator:** Responsible for quiz creation, user management, and overall system administration.
- **Instructor:** Creates and manages quizzes, sets questions (Using API), and monitors student performance.
- **Student:** Takes quizzes and views results.

3.2.2 Functional Requirements

- User registration and authentication.
- Quiz creation and management (Using API).
- Question bank for storing and managing questions (Use API DB).
- Quiz timer and submission functionality.
- Result generation and display.
- User profile management.

3.2.3 Non-functional Requirements:

- **Usability:** Intuitive user interface for easy navigation.
- **Performance:** Quick loading times for quizzes and results.
- **Reliability:** Minimal system downtimes and data integrity assurance.
- **Compatibility:** Cross-browser compatibility for a wide user base.
- **Security:** Encryption of user data, secure authentication mechanisms, and protection against common web vulnerabilities.

3.3 System Features

3.3.1 Quiz Creation

- Quizzes are automatically fetched from the API.
- Ability to create quizzes with customizable parameters (e.g., Question numbers, difficulty level).

3.3.2 Question Bank

- Storage and management of a diverse range of questions (By API).
- Categorization for efficient retrieval.

3.3.3 Quiz Taking

- User-friendly interface for answering questions.
- Timer functionality with automatic submission upon completion.

3.3.4 Results

- Instant display of quiz results for students.

3.4 Conclusion

This system analysis provides a foundation for the development of an effective and reliable online quiz website. By addressing user requirements, system features, data management, security, and scalability, the project aims to deliver a user-friendly and robust platform for educational assessments.

CHAPTER 4

OVERVIEW AND TECHNOLOGY USED

Front-End Technology

.1

4.1.1 HTML

The hypertext markup language (HTML) is a simple markup language. Used to create a hypertext document that are portable from one platform to another HTML documents are SGML (Standard generalized markup language) documents with generic semantics that are appropriate for representing information from a wide range of applications. This specification defines HTML version 3.2. HTML 3.2 aims to capture recommended practice as of early '96 and as such a replacement for HTML2.0 (RFC 1866). A set of instructions embedded in a document is called markup language. These instructions describe what the document text means and how it should look like in a display. Hyper Text Mark Up language (HTML) is the language used to encode World Wide Web documents.

4.1.1.1 What is HTML

To publish information for global distribution, one needs a universally understood language, a kind of publishing mother tongue that all computers may potentially understand. The publishing language used by the World Wide Web is HTML (from Hyper Text Markup Language). HTML gives authors the means to

- Publish online documents with headings, text, tables, lists, photos, etc.
- Retrieve online information via hypertext links, at the click of a button
- Design forms for conducting transactions with remote services, for use in searching for information, making reservations, ordering products etc.
- Include spread sheets, video clips, sound clips, and other applications directly in their documents.

4.1.1.2 A brief history of HTML

HTML was originally developed by Tim Berners-Lee while at CERN and popularized by the Mosaic browser developed at NCSA. During the 1990s it has blossomed with the explosive growth of the Web during this time. HTML has been extended in several ways. The Web depends on Web page authors and vendors sharing the same conventions for HTML. This has motivated joint work on specifications for HTML.

HTML 2.0 (November 1995) was developed under the aegis of the Internet Engineering Task Force (IETF) to codify common practice in late 1994. HTML (1993) and ([HTML.30]) (1995) proposed much richer versions of HTML, despite never receiving consensus in standards discussions, these drafts led to the adoption of range new features. The efforts of the World Wide Web Consortium HTML working group to codify common in 1996 resulted in HTML 3.2 (January 1997). Most people agree that HTML documents should work well across different browsers and platforms. Achieving interoperability lowers costs to content providers since they must develop only one version. of a document. If the effort is not made, there is much greater risk that the Web will devolve into a proprietary world of incompatible formats, ultimately reducing the Web's commercial potential for all participants.

4.1.1.3 Why to use HTML

Website is a collection of pages, publications, and documents that reside on web server. While these pages publications and a document as a formatted in a single format, you should use HTML for home page and all primary pages in the site. This will enable the millions of web users can easily access and to take advantage of your website.

HTML is considered first for formatting any new material you plan to publish on the web. HTML documents are platform independent, meaning that they don't conform to any standard. If they are created properly, you can move home page to any server platform or you can access them with any compliant www browser.

4.1.1.4 Structure of HTML

HTML elements perform a defined task. HTML uses two types of elements:

1. Empty Tags
2. Container Tags

These tags differ because of what they represent. Empty tags represent formatting constructs such as line breaks and horizontal rules. Container tags define a section of text, formats, and dot all the selected text. A container tag has both a beginning and an ending.

4.1.1.5 HTML Layout

An HTML document consists of text, which comprises the content of the document and tags, which defines the structure and appearance of the document. The structure of an HTML document is simple, consists of outer.

Note • <html> tag enclosing the document header and body.

```
<html>
  <head>
    <title> the title of HTML document </title>
  </head>
  <body>
    <p><h1>This is where the actual HTML documents. </h1></p>
    <p><h2>Text lies, which is displayed in the browser. </h2></p>
  </body>
</html>
```

Each document has a head and body delimited by the <head> and <body> tag. The head is where you give your HTML document a title and where you indicate other parameters the browser may use when displaying the document. This includes the text for displaying the text. Tag also references special and indicates the hot spots that link your document to other documents.

4.1.1.6 HTML Forms

Creating a form usually involves two independent steps: Creating the layout for the form itself and then writing a script program on the server side to process the formation you get back from a form.

To create a form, you use the <form> tag. Inside the opening and closing FORM tags are each of the individual form elements plus any other HTML content to create a layout for that form.

The opening tag of the <form> element usually includes the attributes: method and action. The method attributes can be either GET or POST which determines how your form data is sent to the script to process it.

The action attribute is a pointer to the script that processes the form on the server side. The action can be included by a relative path or by a full URL to a script on your server or somewhere else. For example, the following <form> tag would call a script called form-name in cgi-bin directory on server www.myservser.com

```
<form method = post action = http://www.myserver.com/cgi-bin/form-
name.pl>.....</form>
```

4.1.1.7 Method Attribute

The other required attribute for the `<form>` tag sets the methods by which the browser form's data to the server for processing. There are two ways: the POST method and GET method. With POST method, the browser sends the data in two steps: the browser first contacts the form-processing server specified in the action attributes, and once contact is made, sends the data.

CSS

4.1.2

CSS is the language we use to style an HTML document. CSS describes how HTML elements should be displayed.

4.1.2.1 What is CSS

4.1.2.1

- CSS stands for Cascading Style Sheets.
- CSS describes how HTML elements are to be displayed on screen, paper, or in other media.
- CSS saves a lot of work. It can control the layout of multiple web pages all at once.
- External style sheets are stored in CSS files.

4.1.2.2 Why Use CSS

4.1.2.2

CSS is used to define styles for your web pages, including the design, layout, and variations in display for different devices and screen sizes.

```
body {  
    background-color: lightblue;  
}  
  
h1 {  
    color: white;  
    text-align: center;  
}  
  
p{  
    font-family: verdana;  
    font-size: 20px;  
}
```

4.1.2.3 CSS Solved a Big Problem

HTML was NEVER intended to contain tags for formatting a web page!

HTML was created to describe the content of a web page, like:

<h1>This is a heading. </h1>

<p>This is a paragraph. </p>

When tags like , and color attributes were added to the HTML 3.2 specification, it started a nightmare for web developers. Development of large websites, where fonts and color information were added to every single page, became a long and expensive process.

To solve this problem, the World Wide Web Consortium (W3C) created CSS.

CSS removed the style formatting from the HTML page!

4.1.3 JavaScript

JavaScript is the world's most popular programming language. JavaScript is the programming language of the Web. JavaScript is easy to learn. JavaScript Can Change HTML Content One of many JavaScript HTML methods is getElementById(). The example below "finds" an HTML element (with id = "demo"), and changes the element content (innerHTML) to "Hello JavaScript":

Example: `document.getElementById("demo").innerHTML = "Hello JavaScript";`

4.1.3.1 JavaScript Introduction

JavaScript is a new scripting language for Webpages. Scripts written with java script can be embedded into your HTML pages. With java script you have many possibilities for enhancing your HTML page with interesting elements. For example, you can respond to user-initiated events quite easily. Some effects that are now possible with java script were sometimes ago only possible with CGI. So, you can create sophisticated pages with the help of java script on the Internet.

4.1.3.2 How can Java Script scripts run

The first browser to support java script was the Netscape Navigator 2.0 of course the higher versions do have java script as well. You might know that java does not run on all Netscape Navigators 2.0 (or higher versions) versions. But this is not true for java script -although there are some problems with the different versions.

The Mac version for example seems to have many bugs. Soon there are going to be some other browsers, which support java script. The Microsoft Internet explorer 3.0 is going to support java script. JavaScript enabled browsers are going to spread soon it is worth learning this new technique now. You might realize that is easy to write Java Script scripts. We must know is some basic techniques and some work-around for problems you might encounter. Of course, we need a basic. Understanding HTML before reading this tutorial you can find many good online resources about HTML. Best you make an online search about 'html' at yahoo if you want to get informed about HTML. Now I want to show some small scripts so you can learn how they are implemented into HTML-documents and to show which possibilities you have with the new scripting language. The following is a very small script, which will only print a text into an HTML document.

```
<html>
  <head>
    <title>My first JavaScript</title>
```

```

</head>

<body>

<br><br>

This is a normal HTML document

<script language = "JavaScript"> document.write("this is java script!");</script>

<br>

Back in HTML again.

</body>

</html>

```

If you are using a java script enabled browser at the moment, then you will have the possibility to see this script working. If your browser doesn't support Java Script, then this output might be strange.

This is a normal HTML document

This is java script!

Back in HTML again.

4.1.3.3 Functions

Functions are best declared between the `<head>` tag of HTML page. Functions are called by user-initiated events. It's reasonable to keep the functions between the `<head>` tags. They are loaded first before a user can do anything that might call a function. Scripts can be placed between inside comment fields to ensure that older browser do not display the script itself.

```

<html>

<head>

<script language = "JavaScript"> function pushbutton() {alert ("Hello!");}

</script>

</head>

<body>

<form>

```

```

<input type = "button" name = "Button1" value = "pushme" onclick =
"pushbutton()">

</form>

</body>

</html>

```

If we want to test this one immediately and you are using a Java Script enabled browser then please go ahead and push the button.

This script will create a button and when you press it a window will pop up saying “hello!”. In fact, we have a lot of possibilities just by adding functions to our scripts.

The common browsers transmit the form information by either method: here’s the complete tag including the GET transmission method attribute for the previous form

Example

```
<form method = "GET" action = "http://www.mycompany.com/update.php"></form>
```

4.1.3.4 Input elements

Use the `<input>` tag to define any one of several common form elements including text fields multiple choice lists click able images and submission buttons. There are many attributers for this tag only that types and name attributes are required for each element, each type of input element uses only a subset of the followed attributes. Additional `<input>` attributes may be required based upon which type of the form element you specify.

4.1.3.5 Submit button

The submit button (`<input type = submit>`) does what its name implies, settings in motion the form’s submission to the server from the browser. We many have more than submit buttons will be added to the parameter list the browser sends along to the

server.

Example

```
<Input type = "submit"> <Input type = "submit" value = "submit" name = "name">
```

4.1.3.6 Reset button

The reset button if form `<input>` button is nearly self-explanatory; it lets the user reset or set to some default value all elements in the form. By default, the browser displays a reset button worth the label “reset”. We can change that by specifying a value attribute with our own button label.

4.1.3.7 JavaScript Where to Used

4.1.3.7.1 The `<script>` Tag

In HTML, JavaScript code is inserted between `<script>` and `</script>` tags.

Example

```
<script>document.getElementById("demo").innerHTML = "My First JavaScript";</script>
```

Old JavaScript examples may use a type of attribute: `<script type = "text/javascript">`. The type of attribute is not required. JavaScript is the default scripting language in HTML.

4.1.3.7.2 JavaScript Functions and Events

A JavaScript function is a block of JavaScript code, that can be executed when “called” for. For example, a function can be called when an event occurs, like when the user clicks a button.

4.1.3.7.3 JavaScript in `<head>` or `<body>`

You can place any number of scripts in an HTML document. Scripts can be placed in the `<body>`, or in the `<head>` section of an HTML page, or in both.

4.1.3.7.4 JavaScript in `<head>`

In this example, a JavaScript function is placed in the `<head>` section of an HTML page. The function is invoked (called) when a button is clicked:

Example

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```

<script>function myFunction() {document.getElementById("demo").innerHTML =
“Paragraph changed.”; }</script>

</head>

<body>

<h1>A Web Page</h1>

<p id = “demo”>A Paragraph</p>

<button type = “button” onclick = “myFunction()”>Try it</button>

</body>

</html>

```

4.1.3.7.5 JavaScript in <body>

In this example, a JavaScript function is placed in the <body> section of an HTML page.

The function is invoked (called) when a button is clicked:

Example

```

<!DOCTYPE html>

<html>

<body>

<h1>A Web Page</h1>

<p id = “demo”>A Paragraph</p>

<button type = “button” onclick = “myFunction()”>Try it</button>

<script>

function my Function() {

    document.getElementById(“demo”).innerHTML      =      “Paragraph
changed.”; }

</script>

</body>

</html>

```

Placing scripts at the bottom of the <body> element improves the display speed because script interpretation slows down the display.

4.1.3.7.6 External JavaScript

Scripts can also be placed in external files

External file

```
myScript.js

function myFunction() {

document.getElementById("demo").innerHTML

“Paragraph changed.”; }
```

External scripts are practical when the same code is used in many different web pages.

JavaScript files have the file extension .js.

To use an external script, put the name of the script file in the src (source) attribute of a <script> tag:

Example

```
<script src = “myScript.js”></script>
```

You can place an external script reference in <head> or <body> as you like.

The script will behave as if it was located exactly where the <script> tag is located. External scripts cannot contain <script> tags.

4.1.3.7.7 External JavaScript Advantages

Placing scripts in external files has some advantages

- It separates HTML and code.
- It makes HTML and JavaScript easier to read and maintain.
- Cached JavaScript files can speed up page loads.
- To add several script files to one page, use several script tags.

4.1.3.7.8 External References

External scripts can be referenced with a full URL or with a path relative to the current web page.

This example uses a full URL to link to a script

Example

```
<script src = "https://www.w3schools.com/js/myScript1.js"></script>
```

4.2 Back-End Technology

4.2.1 PHP

PHP is a server-side scripting language designed specifically for the web. Within an HTML page, you can embed PHP code that will be executed each time the page is visited. Your PHP code is interpreted at the web server and generates HTML or other output that the visitor will see.

PHP was introduced in 1994. As of November 2007, it was installed on more than 21 million domains worldwide, and this number is growing rapidly. You can see the current number at <http://www.php.net/usage.php>

PHP is an Open-Source project. PHP originally stood for Personal Home Page and now stands for PHP Hypertext Preprocessor.

4.2.1.1 What is PHP

- PHP is an acronym for “PHP: Hypertext Preprocessor”
- PHP is a widely used, open-source scripting language
- PHP scripts are executed on the server
- PHP is free to download and use
- PHP is an amazing and popular language!
- It is powerful enough to be at the core of the biggest blogging system on the web (WordPress)!
- It is deep enough to run the largest social network (Facebook)!
- It is also easy enough to be a beginner’s first server-side language!

4.2.1.2 What is a PHP File

- PHP files can contain text, HTML, CSS, JavaScript, and PHP code
- PHP code is executed on the server, and the result is returned to the browser as plain HTML
- PHP files have extension “.php”.

4.2.1.3 What Can PHP Do

- PHP can generate dynamic page content
- PHP can create, open, read, write, delete, and close files on the server
- PHP can collect form data

- PHP can send and receive cookies
- PHP can add, delete, modify data in your database
- PHP can be used to control user-access
- PHP can encrypt data
- With PHP you are not limited to output HTML. You can output images, PDF files, and even Flash movies. You can also output any text, such as XHTML and XML.

4.2.1.4 Why PHP

- PHP runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.)
- PHP is compatible with almost all servers used today (Apache, IIS, etc.)
- PHP supports a wide range of databases
- PHP is free. Download it from the official PHP resource: www.php.net
- PHP is easy to learn and runs efficiently on the server side

4.2.1.5 What's new in PHP 8

- PHP 8 is much faster than the previous popular stable release (PHP 7.4)
- PHP 8 has improved Error Handling
- PHP 8 supports stricter Type Declarations for function arguments
- PHP 8 supports new operators (like the spaceship operator: `<=>`)

4.2.1.6 Unique Features

If you are familiar with other server-side language like ASP.NET or JSP you might be wondering what makes PHP so special, or so different from these competing alternatives well here are some reasons:

1. Performance
2. Portability (Platform Independent)
3. Ease of Use
4. Open Source
5. Third-Party Application Support
6. Community Support

4.2.1.6.1 Performance

Scripts written in PHP executes faster than those written in other scripting language, with numerous independent benchmarks, putting the language ahead of competing alternatives like JSP, ASP.NET and PERL. The PHP 5.0 engine was completely redesigned with an

optimized memory manager to improve performance and is noticeable faster than previous versions. In addition, third party accelerators are available to further improve performance and response time.

4.2.1.6.2 Portability

PHP is available for UNIX, MICROSOFT WINDOWS, MAC OS, and OS/2. PHP Programs are portable between platforms. As a result, a PHP application developed on, say, Windows will typically run on UNIX without any significant issues. This ability to easily undertake cross-platform development is a valuable one, especially when operating in a multi-platform corporate environment or when trying to address multiple market segments.

4.2.1.6.3 Ease of Use

“Simplicity is the ultimate sophistication”, Said Leonardo da Vinci, and by that measure, PHP is an extremely sophisticated programming language. Its syntax is clear and consistent, and it comes with exhaustive documentation for the 5000+ functions included with the core distributions. This significantly reduces the learning curve for both novice and experienced programmers, and it’s one of the reasons that PHP is favoured as a rapid prototyping tool for Web-based applications.

4.2.1.6.4 Open Source

PHP is an open-source project - the language is developed by a worldwide team of volunteers, who make its source code freely available on the Web, and it may be used without payment of licensing fees or investments in expensive hardware or software. This reduces software development costs without affecting either flexibility or reliability. The open-source nature of the code further means that any developer, anywhere, can inspect the code tree, spit errors, and suggest possible fixes, this produces a stable, robust product wherein bugs, once discovered, are rapidly resolved sometimes within a few hours of discovery.

4.2.1.6.5 Third-Party Application Support

One of PHP’s Strengths has historically been its support for a wide range of different databases, including MySQL, Postgres SQL, Oracle, and Microsoft SQL Server. PHP 8.0

Supports more than fifteen different database engines, and it includes a common API for database access. XML support makes it easy to read and write XML documents though they were native PHP data structures, access XML node collections using Xpath, and transform XML into other formats with XSLT style sheets.

4.2.1.6.6 Community Support

One of the nice things about a community-supported language like PHP is the access it offers to the creativity and imagination of hundreds of developers across the world. Within the PHP community, the fruits of this creativity may be found in PEAR, the PHP Extension and Application Repository and PECL, the PHP Extension Community Library, which contains hundreds of ready-, made widgets and extensions that developers can use to painlessly add new functionality to PHP. Using these widgets is often a more time-and cost-efficient alternative to rolling your own code.

4.2.1.7 PHP Server

The PHP Community Provides Some types of Software Server solution under The GNU (General Public License).

These are the following

1. WAMP Server
2. LAMP Server
3. MAMP Server
4. XAMPP Server

All these types of software automatic configure inside operating system after installation it is having PHP, MySQL, Apache, and operating system base configuration file, it doesn't need to configure manually.

4.2.1.8 PHP MySQL Connectivity

Use the `mysql_connect()` function to establish connection to the MySQL server. To access the database functionality, we must make a connection to database using php. `mysql_connect()` function is used to establish the connection to mysql server. four arguments need to be passed to `mysql_connect()` function.

1. **hostname**: if you are working on local system, you can use **localhost** or you can also provide ip address or server name.
2. **username**: if there is an existing user, you can provide **username**. default **username** is ‘root’.
3. **password**: by default, password is blank or null.
4. **dbname**: it is an optional field. it is basically a name of the database that needs to be connected.

`mysql_connect(host, username, password, dbname);`

host (Server name) - Either a host name (server name) or an IP address

username - The MySQL username

password - The password to log in with

dbname - Optional. The database to be used when performing queries

Note: There are more available parameters, but the ones listed above are the most important.

In the following example we store the connection in a variable (\$con) for later use in the script.

```
<?php  
// Create connection  
  
$con=mysqli_connect("localhost", "root", "", "triviaclickz") or  
die(mysql_error());  
  
?>
```

Here localhost is server name. root is MySQL default username. default password is blank and database name is my_db. mysql_error() function provides mysql connectivity error message.

4.2.1.9 PHP MySQL Close Connection

```
<?php  
// Create connection  
  
$con=mysqli_connect("localhost", "root", "", "triviaclickz") or
```

```
die(mysql_error());  
//code to be executed.  
  
// Close connection  
  
mysql_close($con);  
  
?>
```

After work with the database is done we have to close the connection using `mysql_close()` function in which the connection to the database is passed.

4.2.2 MySQL Introduction

There are many database management systems currently available, some commercial and some free.

Some of them: Oracle, Microsoft Access, MySQL, and Postgres SQL.

These database systems are powerful, feature-rich software, capable of organizing and searching millions of records at very high speeds. Understanding Databases, Records, and Primary Keys. Every Database is composed of one or more tables. These Tables, which structure data into rows and columns, impose organization on the data. The records in a table(below) are not arranged in any order.

4.2.2.1 Introduction to SQL

SQL stands for Structured Query Language. It is used for storing and managing data in relational database management system (RDBMS). It is a standard language for Relational Database System. It enables a user to create, read, update, and delete relational databases and tables. All the RDBMS like MySQL, Informix, Oracle, MS Access, and SQL Server use SQL as their standard database language. SQL allows users to query the database in several ways, using English-like statements.

4.2.2.1.1 SQL follows the following rules

Structure query language is not case sensitive. Generally, keywords of SQL are written in uppercase. Statements of SQL are dependent on text lines. We can use a single SQL statement on one or multiple text line. Using the SQL statements, you can perform most of the actions in a database. SQL depends on tuple relational calculus and relational algebra.

4.2.2.1.2 SQL process

When an SQL command is executing for any RDBMS, then the system figure out the best way to carry out the request and the SQL engine determines that how to interpret the task. In the process, various components are included. These components can be optimization Engine, Query engine, Query dispatcher, classic, etc. All the non-SQL queries are handled by the classic query engine, but SQL query engine won't handle logical files.

4.2.2.1.3 What is SQL

- SQL stands for Structured Query Language
- SQL lets you access and manipulate databases
- SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987.

4.2.2.1.4 What Can SQL do

- SQL can execute queries against a database
- SQL can retrieve data from a database
- SQL can insert records in a database
- SQL can update records in a database
- SQL can delete records from a database
- SQL can create new databases
- SQL can create new tables in a database
- SQL can create stored procedures in a database
- SQL can create views in a database
- SQL can set permissions on tables, procedures, and views
- SQL is a Standard – BUT...

Although SQL is an ANSI/ISO standard, there are different versions of the SQL language.

However, to be compliant with the ANSI standard, they all support at least the major commands (such as SELECT. UPDATE. DELETE, INSERT. WHERE) in a similar manner.

Note: Most of the SQL database programs also have their own proprietary extensions in addition to the SQL standard!

- Using SQL in Your Web Site
- To build a web site that shows data from a database, you will need:
 - An RDBMS database program (i.e. MS Access, SQL Server, MySQL)
 - To use a server-side scripting language, like PHP or ASP
 - To use SQL to get the data you want
 - To use HTML/CSS to style the page

4.2.2.1.5 RDBMS

RDBMS stands for Relational Database Management System. RDBMS is the basis for SQL, and for all modern database systems such as MS SQL Server, IBM DB2, Oracle, MySQL, and Microsoft Access.

The data in RDBMS is stored in database objects called tables. A table is a collection of related data entries, and it consists of columns and rows.

Example

```
SELECT * FROM Customers;
```

Every table is broken up into smaller entities called fields. The fields in the Customers table consist of CustomerID, CustomerName, ContactName, Address, City, PostalCode and Country. A field is a column in a table that is designed to maintain specific information about every record in the table.

A record, also called a row, is each individual entry that exists in a table. For example, there are 91 records in the above Customers table. A record is a horizontal entity in a table. A column is a vertical entity in a table that contains all information associated with a specific field in a table.

4.2.2.2 Types of SQL

4.2.2.2.1 Data Definition Language (DDL)

DDL Consists of statements that define the structure and relationships of a database and its table. These Statements are used to Create, drop, and modify databases and tables.

4.2.2.2.2 Data Manipulation Language (DML)

DML statements are related to altering and extracting data from a database. These statements are used to add records to, update records in, and delete records from, a database; perform queries; retrieve table records matching one or more user specified criteria; and join tables together using their common fields.

4.2.2.2.3 Data Control Language (DCL)

DCL statements are used to define access levels and security privileges for a database.

You would use these statements to grant or deny user privileges; assign roles; change passwords; view permissions; and create rule sets to protect access to data.

The Syntax of SQL is quite intuitive. every SQL statement begins with an “action word”, like DELETE, INSERT, ALTER etc.

it ends with a semicolon. whitespace, tabs, carriage returns are ignored.

Example

```
CREATE DATABASE employee;
```

```
SELECT name FROM users where email = "anuj.lpu1@gmail.com";
```

```
DELETE FROM cars WHERE year_of_manufacture < 1980;
```

CHAPTER 5

SYSTEM AND DATABASE DESIGN

System Design

5.1 Flow Chart Diagram

5.1.1.1 User Registration

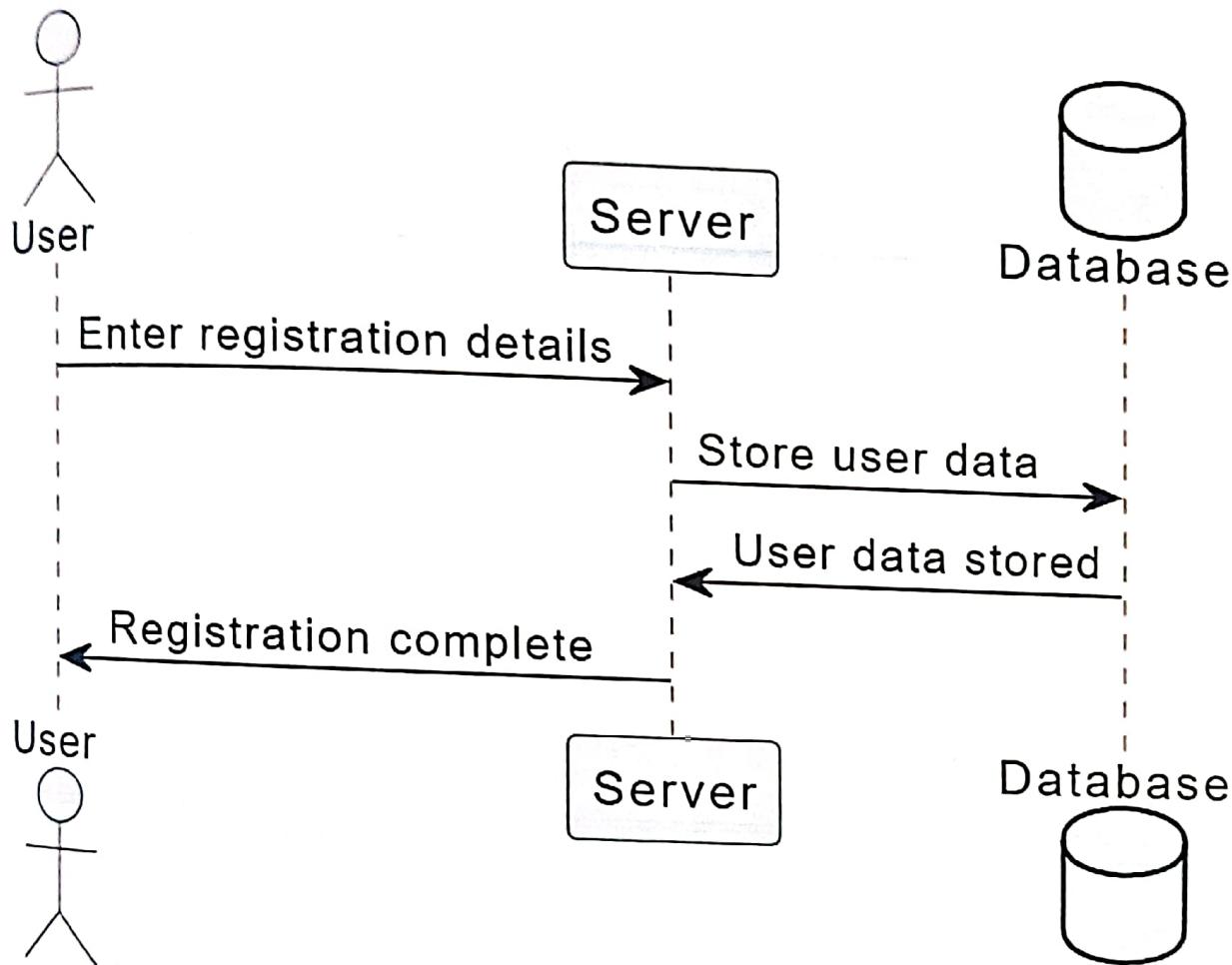


Figure 5.1 User Registration

5.1.1.2 User Login

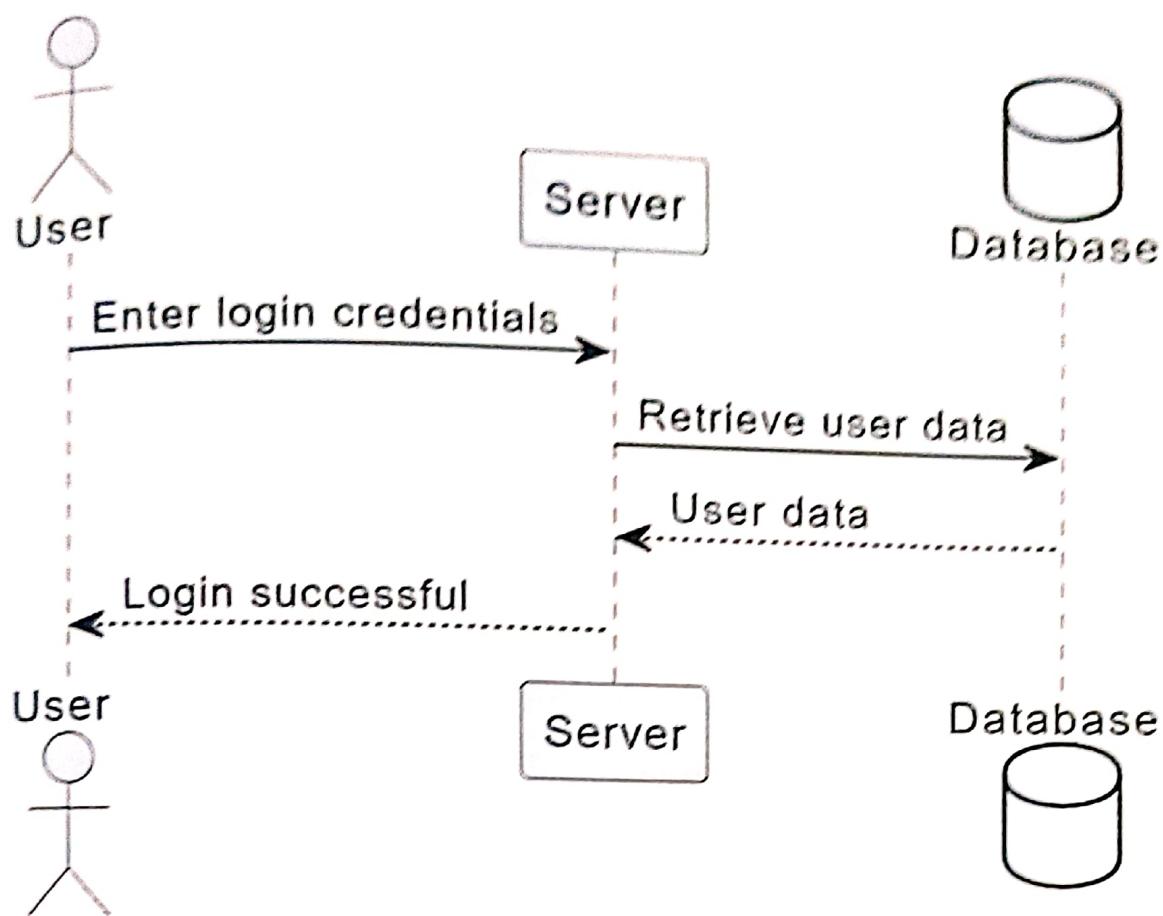


Figure 5.2 User Login

5.1.1.3 User Interaction

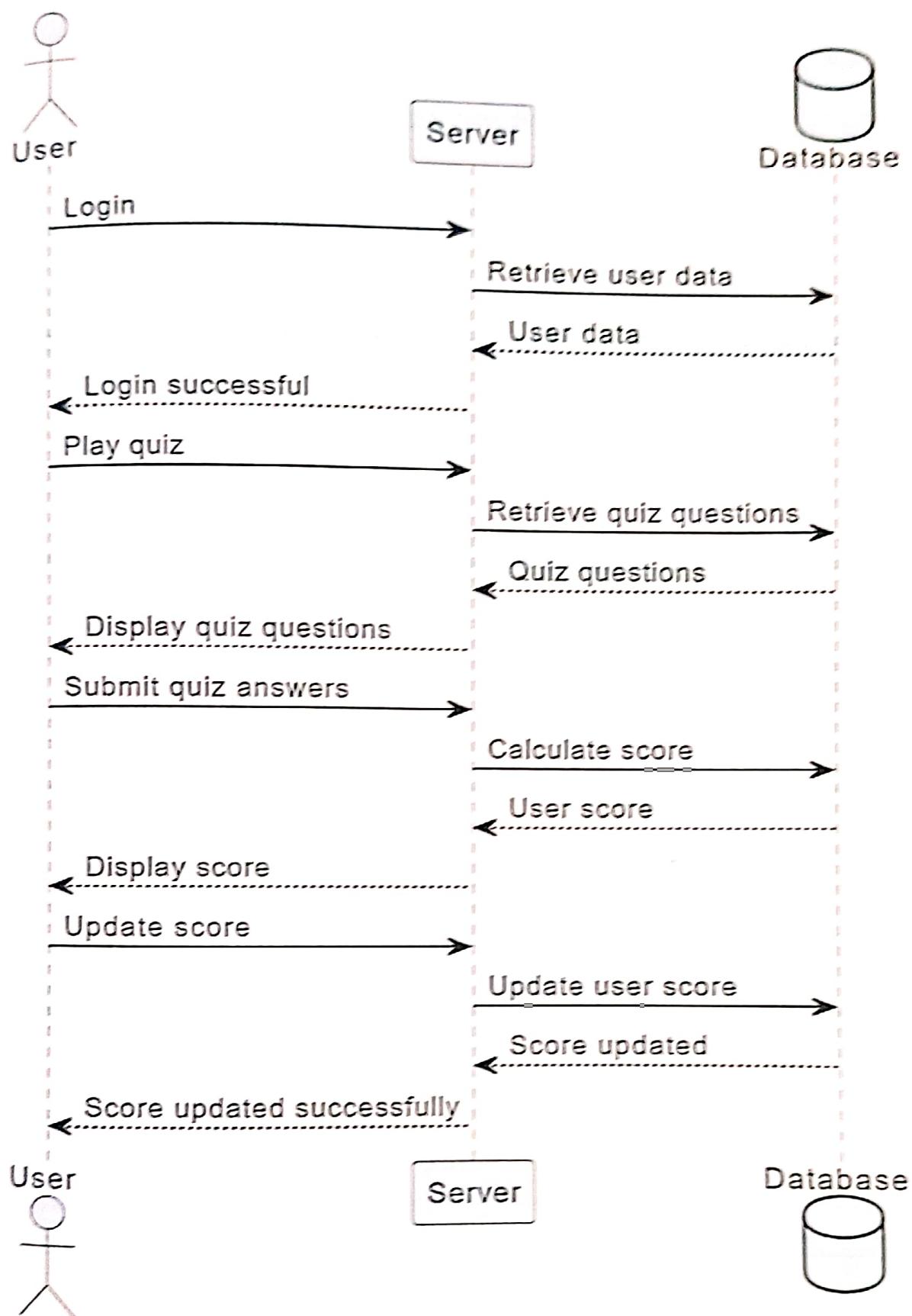


Figure 5.3 User Interaction

5.2 Database Design

5.2

5.2.1 Data Dictionary

The Data Dictionary is very important part of the project through which we can know that where data are stored in the project.

The Data Dictionary in this project is as below -

This project contains one database and four (4) tables.

5.2.2 Table Structure of Database

| Table | Action | Browse | Structure | Search | Insert | Empty | Drop | Rows | Type | Collation | Size |
|-------------|--------------------------|--------|-----------|--------|--------|-------|------|------|--------|--------------------|----------|
| contact_tbl | <input type="checkbox"/> | | | | | | | 1 | InnoDB | utf8mb4_general_ci | 16.0 Kib |
| fmail_tbl | <input type="checkbox"/> | | | | | | | 2 | InnoDB | utf8mb4_general_ci | 16.0 Kib |
| scores_tbl | <input type="checkbox"/> | | | | | | | 8 | InnoDB | utf8mb4_general_ci | 16.0 Kib |
| signup_tbl | <input type="checkbox"/> | | | | | | | 5 | InnoDB | utf8mb4_general_ci | 16.0 Kib |
| 4 tables | | Sum | | | | | | 26 | InnoDB | utf8mb4_general_ci | 54.0 Kib |

Check all With selected:

Print Data dictionary

Figure 5.4 Table Structure of Database

TriviaClickz – Online Quiz contains 4 MySQL tables

- contact_tbl
- fmail_tbl
- scores_tbl
- signup_tbl

5.2.2.1 contact_tbl MySQL Table

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|------------|--------------|--------------------|------------|------|---------|----------|-------|--------------------|
| 1 | fldName | varchar(25) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| 2 | fldEmail | varchar(50) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| 3 | fldMessage | varchar(500) | utf8mb4_general_ci | | No | None | | | Change Drop More |

Check all With selected: Browse Drop Primary Unique Index

Figure 5.5 Structure of contact_tbl Table

5.2.2.2 fmail_tbl MySQL Table

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|--------|-------------|--------------------|------------|------|---------|----------|-------|--------------------|
| 1 | fEmail | varchar(50) | utf8mb4_general_ci | | No | None | | | Change Drop More |

Check all With selected Browse Change Drop Primary Unique Index

Figure 5.6 Structure of fmail_tbl Table

5.2.2.3 scores_tbl MySQL Table

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|---------------|-------------|--------------------|------------|------|---------|----------|-------|--------------------|
| 1 | fldUser | varchar(25) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| 2 | fldDifficulty | varchar(10) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| 3 | fldScore | int(5) | | | No | None | | | Change Drop More |

Check all With selected Browse Change Drop Primary Unique Index

Figure 5.7 Structure of scores_tbl Table

5.2.2.4 signup_tbl MySQL Table

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|----------|-------------|--------------------|------------|------|---------|----------|-------|--------------------|
| 1 | fldUser | varchar(14) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| 2 | fldEmail | varchar(50) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| 3 | fldPwd | varchar(14) | utf8mb4_general_ci | | No | None | | | Change Drop More |

Check all With selected Browse Change Drop Primary Unique Index

Figure 5.8 Structure of signup_tbl Table

CHAPTER 6

PROJECT OUTPUT SCREEN

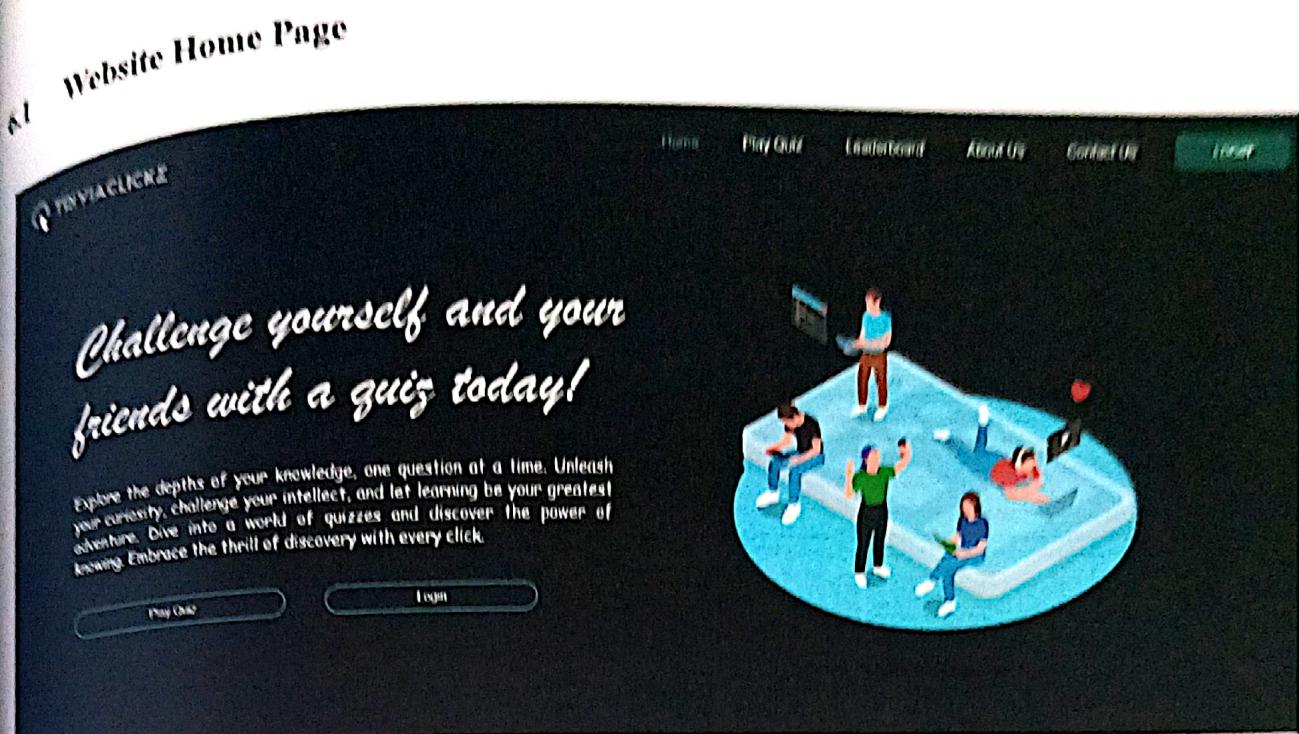


Figure 6.1 Website Home Page

62 User Registration Page

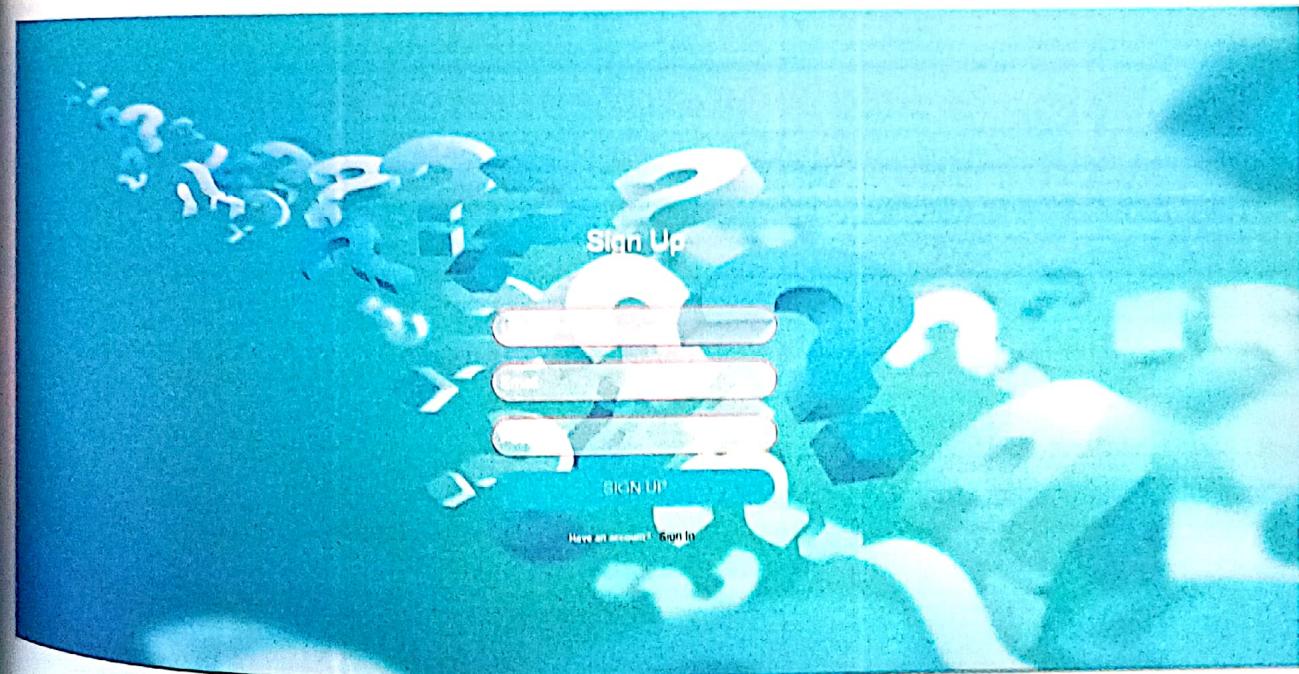


Figure 6.2 User Registration Page

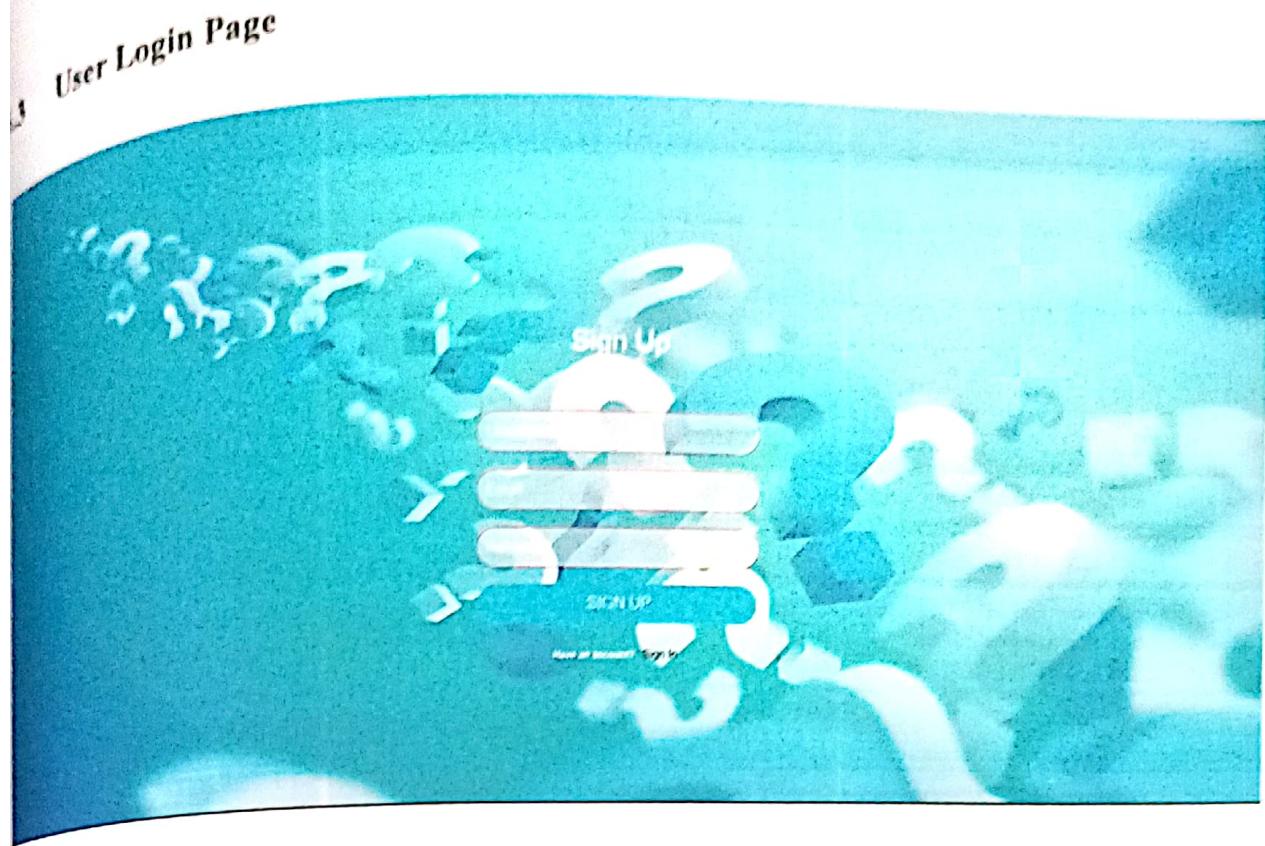


Figure 6.3 User Login Page

4 Quiz Customize Page

A screenshot of a quiz customization page for TriviaClickz. It features a dark header with the logo and navigation links: Home, Play Quiz, Leaderboard, About Us, Contact Us, and LOGOUT. The main section has a teal header "Welcome To TriviaClickz" and "Customize Your Own Quiz". It includes three dropdown menus: "Number of Questions" set to 10, "Select Category" set to General Knowledge, and "Select Difficulty" set to Medium. A large teal "Start Quiz" button is at the bottom.

Figure 6.4 Quiz Customize Page

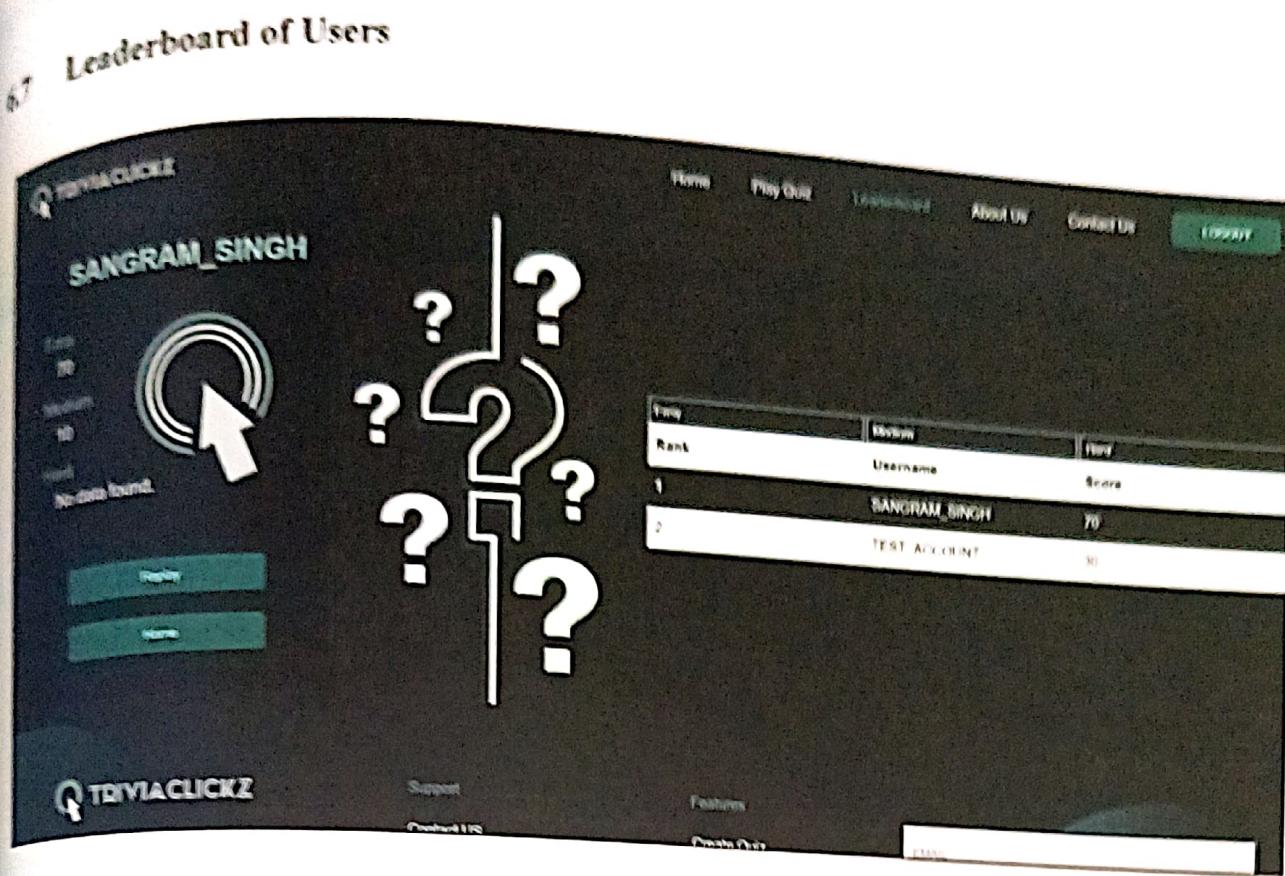


Figure 6.7 Leaderboard of Users

6.8 About Us Page

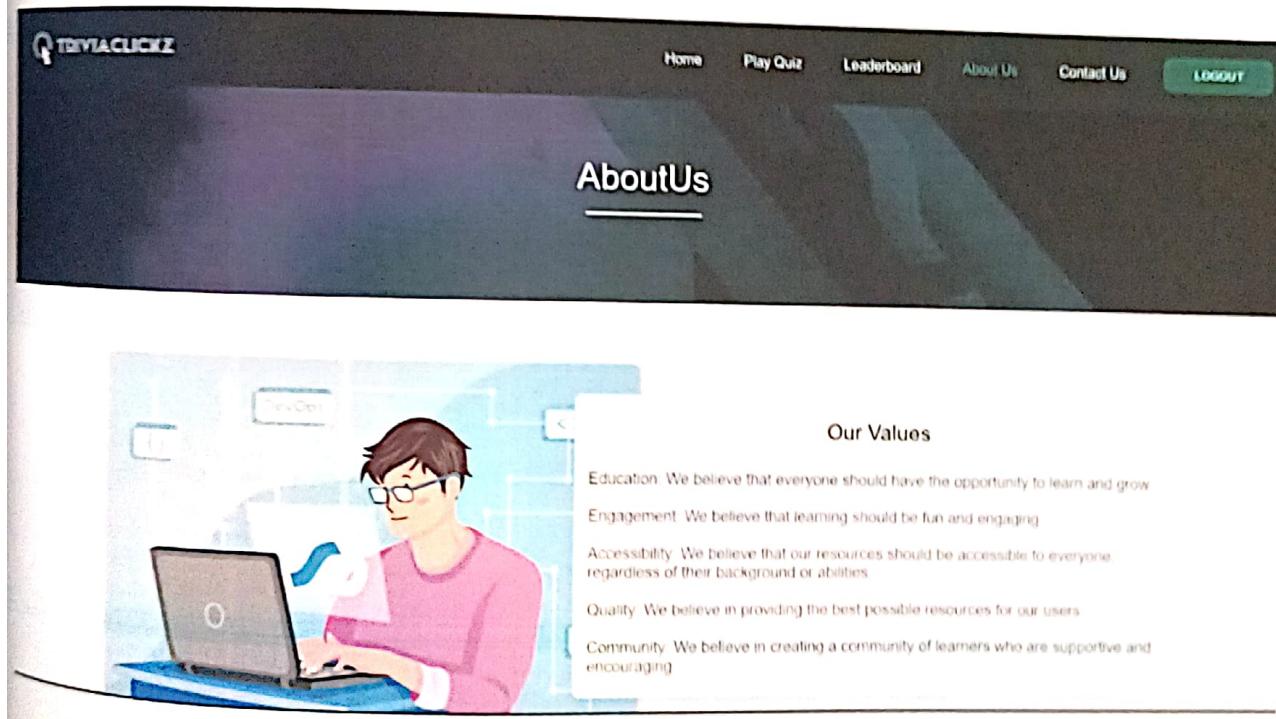


Figure 6.8 About Us Page

Contact Us Page

9

TRIVIACLICKZ

Get in touch

Contact us anytime! We're always happy to help.

Write your name:

Write your email:

Write your message:

Submit

Home Play Quiz Leaderboard About Us Contact Us (Logout)

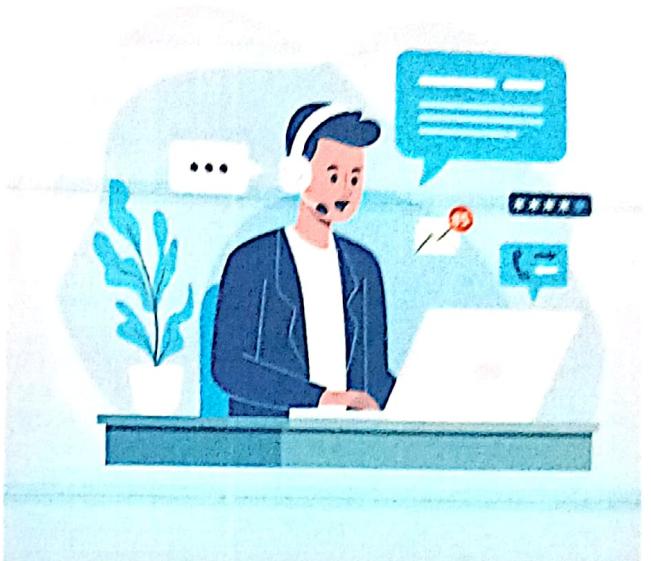


Figure 6.9 Contact Us Page

CONCLUSION

In conclusion, the culmination of the online quiz website project represents a significant achievement in our pursuit of creating an interactive and engaging platform for users to explore and test their knowledge. Central to the success of this endeavour is the incorporation of the OpenTDB API, a pivotal component that has enriched the website's content and elevated the overall user experience.

The OpenTDB API serves as a comprehensive repository of diverse and dynamic questions, spanning various subjects and difficulty levels. Through seamless integration, our website gains access to this vast question bank, providing users with an extensive array of quizzes that cater to different interests and proficiency levels. The versatility of OpenTDB ensures that the content remains not only varied but also up-to-date, fostering continuous engagement and interest among our user base.

The user-friendly interface and interactive features, complemented by the wealth of questions from OpenTDB, contribute to a multifaceted and enjoyable quizzing experience. The strategic implementation of this API has allowed us to strike a balance between educational substance and entertainment value, making the website a valuable resource for both learning and leisure.

Throughout the development process, meticulous attention was dedicated to ensuring a seamless and efficient integration between the website and the OpenTDB API. Rigorous testing procedures and user feedback mechanisms were established to identify and address any potential challenges, resulting in a robust system that delivers a smooth and reliable question-fetching mechanism.

As we reflect on the project's completion, it is imperative to acknowledge the instrumental role played by the OpenTDB API in achieving our objectives. The collaborative synergy between our team and this external resource has not only realized the envisioned goals but has also positioned the website for sustained success in the dynamic landscape of online learning and quizzing.

Looking ahead, the integration with OpenTDB not only ensures the current viability of our platform but also sets the stage for future enhancements and expansions. This strategic approach to leveraging external tools underscores our commitment to staying at the forefront of innovation and providing a rich and evolving experience for our users. In essence, the online quiz website stands as a testament to effective collaboration, technological ingenuity, and the seamless integration of external APIs to create a sophisticated and feature-rich platform that resonates with a diverse and growing audience.

There are exciting possibilities for the online quiz website. Here are some ideas for how we can make the platform even better in the future.

More Interesting Quizzes

We can add a wider range of quiz topics to appeal to different interests. This might include new subjects, quizzes related to specific industries, and quizzes that reflect different cultures.

Fun Challenges and Rewards

We could make learning more fun by adding game-like elements. This might include earning badges, progressing through levels, and getting rewards for achievements, all of which can encourage users to participate more.

Personalized Learning

Imagine if the website could suggest quizzes based on your performance and interests. This personalized approach could make the learning experience more tailored to each user.

Community Interaction

It would be great to let users create and share their quizzes. This feature could foster a sense of community, allowing users to challenge their friends, participate in group quizzes, and even engage in friendly competitions.

Insights into User Behaviour

By tracking how users perform on quizzes, we could gain valuable insights. This data could help us understand popular topics, improve engagement strategies, and continually enhance the overall platform.

Interactive Content

We might consider adding more interactive elements to quizzes, such as videos, images, and audio. This could make the learning experience more dynamic and engaging.

Mobile App for Convenience

Creating a mobile app could make it easier for users to access quizzes on the go. This would expand our reach and make the platform more accessible to a broader audience.

Multilingual Support

To welcome users from different parts of the world, we could offer quizzes in multiple languages. This would ensure that language is not a barrier to learning and enjoyment.

Adaptable Learning Experience

Imagine if the difficulty of questions adjusted based on each user's performance. This adaptive learning approach would provide a personalized and challenging experience for everyone.

Building Partnerships

Collaborating with educational institutions, organizations, or sponsors could bring in more resources and opportunities for growth. It's a way to make the platform even more impactful.

Keeping Everything Up to Date

To ensure the platform runs smoothly and securely, we should stay updated with the latest technology. Regular updates will guarantee that users have a hassle-free experience.

By exploring these ideas, we can transform the online quiz website into a dynamic and user-friendly learning platform. This future aim to keep users engaged, foster a sense of community, and ensure the platform stays relevant and enjoyable for a diverse audience.

REFERENCES

API: <https://opentdb.com/>

Fetch Function: JavaScript Web API for making HTTP requests to servers. From basic JS.