### COMPUTER SCIENCE

**A Project Report**

**Submitted by:**

**Aagam Damseth (2235947)**

**in partial fulfillment for the award of the degree**

**of**

### BACHELOR OF SCIENCE

### IN

**GRAPHICS AND WEB DESIGN**

****

**CHANDIGARH SCHOOL OF BUSSINESS**

**JHANJERI, MOHALI PUNJAB (INDIA) - 140406**

**(AFFILIATED TO I.K.G. PUNJAB TECHNICAL UNIVERSITY, KAPURTHALA, PUNJAB (INDIA) NOVEMBER 2024**

**CANDIDATE’S DECLARATION**

I hereby certify that the project entitled “COMPUTER SCIENCE” submitted by Aagam Damseth (2235947) in partial fulfillment of the requirement for the award of degree of the BSC submitted in I.K. Gujral Punjab Technical University, Kapurthala at Chandigarh School of Business, Jhanjeri is an authentic record of my own work carried out during a period from OCTOBER, 2024 to DECEMBER, 2024 under the guidance of Prof. Mukesh. The matter presented in this project has not formed the basis for the award of any other degree, diploma, fellowship or any other similar titles.

**Signature of the Student**

**Place:**

**Date:**

**CERTIFICATE**

This is to certify that the project titled “COMPUTER SCIENCE” is the bona fide work carried out by Aagam Damseth (2235947) in partial fulfillment of the requirement for the award of degree of the BSC (GWD) submitted in I.K. Gujral Punjab Technical University, Kapurthala at Chandigarh School of Business, Jhanjeri is an authentic record of my own work carried out during a period from October, 2024 to December, 2024 under the guidance of Prof. Mukesh. The Major Project Viva-Voce Examination has been held on 17/12/2024.

**Faculty HoD Director**

**ACKNOWLEDGEMENT**

I express my sincere gratitude to the I.K. Gujral Punjab Technical University, Kapurthala for giving me the opportunity to work on the Major Project during my final year of BSC is an important aspect in the field of Computer Application.

I also owe my sincerest gratitude towards Prof. Mukesh for his/her valuable advice and healthy criticism throughout my project which helped me immensely to complete my work successfully.

I would also like to thank everyone who has knowingly and unknowingly helped me throughout my work. Last but not least, a word of thanks for the authors of all those books and papers which I have consulted during my project work as well as for preparing the report.

**ABSTRACT**

This project explores the field of Computer Science by examining its advantages, disadvantages, and diverse applications, followed by the creation of a working quiz website. The project highlights how computer science drives innovation, efficiency, and transformation across various industries such as healthcare, finance, and education. It also addresses the challenges associated with the field, including cybersecurity threats, job displacement due to automation, and privacy concerns. To demonstrate the practical aspects of computer science, a quiz website was developed using HTML and CSS, showcasing how web development technologies can be applied to create interactive platforms. The website features a user-friendly interface with multiple-choice questions, real-time feedback, and responsive design. This project aims to provide a comprehensive understanding of the impact of computer science on modern society and the practical applications of web development technologies

**CONTENTS**

1. **CERTIFICATE**
2. **ACKNOWLEDGEMENT**
3. **ABSTRACT**
4. **CONTENTS**
5. **INTRODUCTION**
6. **SYSTEM ANALYSIS AND DESIGN**

6.1 RESOURCE REQUIRED

6.2 SOURCE CODE

6.3 WORKING

6.4 OUTPUT

1. **ADVANTAGES AND DISADVANTAGES**
2. **APPLICATONS**
3. **CONCLUSION / RECOMMENDATIONS**
4. **REFERENCES**
5. **APPENDICES**

**INTRODUCTION**

Computer Science is the study of computers, computational systems, and the theoretical foundations of information and computation. It involves understanding how computers work, programming them to perform tasks, and applying algorithms and data structures to solve complex problems efficiently.

In the modern world, computer science plays a crucial role in almost every industry, from healthcare to finance, entertainment to education, and beyond. The field is constantly evolving with innovations in areas such as artificial intelligence (AI), machine learning (ML), big data, cyber security, and cloud computing, all of which have revolutionized how we live and work.

Key Areas of Computer Science:

1. Algorithms and Data Structures: The backbone of computer science, where problem-solving methods are designed, and data is organized efficiently for storage and retrieval.

2. Programming Languages: These are the tools used to write software applications. Languages like Python, Java, C++, and JavaScript are widely used for both frontend and backend development.

3. Operating Systems: The system software that manages computer hardware and software resources and provides common services for computer programs.

4. Database Systems: Concerned with the organization, storage, retrieval, and management of large sets of data. SQL and NoSQL databases are commonly used.

5. Artificial Intelligence (AI): A rapidly growing area of computer science that involves creating machines or software that can mimic human intelligence, including learning, reasoning, and decision-making.

6. Networking and Cyber security: Focuses on the communication between computers and the protection of information from unauthorized access, damage, or theft.

7. Software Engineering: The application of engineering principles to software development, ensuring that software is reliable, scalable, and efficient.

8. Cloud computing refers to the delivery of computing services like storage, processing, and networking over the internet. It allows businesses and individuals to access computing resources on demand without investing in physical hardware. Leading cloud providers include Amazon Web Services (AWS), Microsoft Azure, and Google Cloud.

Here are some of the key hardware specifications relevant to Computer Science:

1. **Central Processing Unit (CPU)**

The CPU is often called the “brain” of the computer. It executes instructions from software applications and performs computations. The CPU plays a key role in determining the overall performance of a computer. Key specifications include:

Clock Speed: Measured in GHz (Gigahertz), it determines how fast the CPU can process instructions. Higher clock speeds result in faster processing.

Cores: Modern CPUs have multiple cores (dual-core, quad-core, octa-core, etc.), allowing multiple tasks to be executed in parallel.

Threads: Each core can support multiple threads (the smallest unit of a task), which improves multitasking and processing efficiency.

Cache: L1, L2, and L3 cache are small, high-speed memory areas that store frequently used instructions and data to reduce the time it takes for the CPU to access information.

**2. Memory (RAM)**

Random Access Memory (RAM) temporarily stores data and instructions that the CPU is currently using or processing. More RAM allows for faster access to information, improving system performance, especially for multitasking and running complex applications.

Size: Typically measured in GB (Gigabytes) or TB (Terabytes), depending on the system. For intensive tasks, like running simulations or data analysis, higher amounts of RAM (16GB, 32GB, or more) are beneficial.

Speed: RAM speed is measured in MHz and higher speeds can significantly enhance the performance of memory-intensive tasks.

Type: The most common types of RAM include DDR4 and DDR5, with DDR5 being the latest and faster version.

**3. Storage Devices**

Storage devices are used to save data, files, and applications permanently (until deleted). The speed and size of storage impact the overall performance and usability of a system.

Hard Disk Drive (HDD): Traditional storage, typically cheaper but slower than solid-state drives. HDDs are usually used for mass storage with larger capacities.

Solid State Drive (SSD): SSDs are much faster than HDDs because they store data on flash memory. They have no moving parts and are more durable. SSDs significantly reduce boot time and file transfer speed.

NVMe (Non-Volatile Memory Express): A type of SSD that connects directly to the motherboard via the PCIe interface, offering even faster read/write speeds compared to standard SSDs.

**4. Graphics Processing Unit (GPU)**

The GPU is responsible for rendering graphics and processing visual information. While it’s most commonly associated with gaming and multimedia, modern GPUs are also critical for parallel computing, which is heavily used in machine learning, data analysis, and AI.

CUDA Cores (NVIDIA): A term used for processing units in NVIDIA GPUs that allow parallel processing. GPUs with more cores can handle more tasks simultaneously.

VRAM (Video RAM): Specialized memory used by the GPU to store graphical data like textures, frame buffers, and shaders. Higher VRAM is essential for running graphic-intensive applications such as 3D rendering or gaming.

**5. Motherboard**

The motherboard is the main circuit board that connects all hardware components. It ensures communication between the CPU, RAM, storage, and other peripherals. Key features to consider include:

Chipset: The chipset determines the motherboard’s capabilities, such as the number of supported cores, RAM size, and connectivity options.

Ports and Slots: A motherboard comes with various ports (USB, HDMI, Ethernet) and slots (PCIe) for expanding the system with additional devices like graphics cards or network cards.

**6. Power Supply Unit (PSU)**

The PSU provides power to all components of the computer. It converts AC power from a wall outlet into DC power that the internal components require. Important specifications to consider:

Wattage: The PSU must provide enough power to support all connected components. High-performance systems, such as gaming rigs or workstations, require higher wattages.

Efficiency: Modern PSUs have efficiency ratings like 80 Plus Bronze, Silver, Gold, etc. Higher ratings indicate better energy efficiency, leading to lower electricity consumption.

**7. Input/Output Devices**

These devices allow users to interact with the computer and enable the computer to communicate with the outside world.

Keyboard: Used for input, there are different types (mechanical, membrane) that vary in terms of durability and key response.

Mouse: Another input device, available in optical, laser, or wireless varieties.

Onitor: Displays visual output from the computer. Key specifications include resolution (HD, 4K), refresh rate (60Hz, 144Hz), and response time (for gaming monitors).

Speakers/Headphones: Output devices used for audio output.

**8. Networking Components**

Networking components are essential for connecting the computer to local networks (LAN) or the internet (WAN).

Ethernet Card: A network interface card (NIC) used for wired internet connections.

Wi-Fi Adapter: A wireless card used to connect the computer to Wi-Fi networks.

Router/Switch: Devices used to manage network traffic and direct data between multiple devices.

In addition to hardware, software plays an equally important role in the field of Computer Science. Software defines how hardware is utilized and provides the tools and applications needed to perform tasks, process data, and run algorithms. Here are the key software specifications relevant to Computer Science:

1. Operating System (OS)

The Operating System (OS) is the software that manages the hardware and software resources of the computer. It provides an environment where programs can run and coordinates the activities of the computer. Some important aspects of the OS include:

Type: The OS can be a desktop OS (e.g., Windows, macOS, Linux) or a mobile OS (e.g., Android, Ios).

Kernel: The core part of the OS, managing memory, CPU, input/output devices, and system processes.

File System: Manages how data is stored and retrieved on storage devices (e.g., NTFS, FAT32, EXT4).

Security: Features like user authentication, encryption, and system integrity checks to protect data and ensure safe operation.

2. Integrated Development Environment (IDE)

An IDE is software that provides comprehensive facilities for software development, typically including a code editor, compiler, debugger, and tools to build and deploy applications. Key specifications include:

Languages Supported: Popular IDEs support multiple programming languages such as Python, Java, C++, JavaScript, and more.

Code Completion and Syntax Highlighting: Helps developers write code faster and reduces errors.

Debugging Tools: IDEs offer integrated debuggers to step through the code and troubleshoot errors.

Examples of IDEs:

Visual Studio Code (supports multiple languages like JavaScript, Python, etc.)

Eclipse (primarily for Java)

PyCharm (focused on Python)

IntelliJ IDEA (supports Java, Kotlin, etc.)

3. Programming Languages

Programming languages provide the syntax and structure for writing software. The language you choose impacts the performance, usability, and suitability of an application. Key software specifications for programming languages include:

Type: High-level languages (e.g., Python, Java, C++) or low-level languages (e.g., Assembly or C).

Paradigms: Some languages follow different programming paradigms, such as:

Object-Oriented Programming (OOP): e.g., Java, Python, C++

Libraries and Frameworks: Languages often come with libraries or frameworks that make it easier to develop applications. For example, Django (for Python), Spring (for Java), or React (for JavaScript).

4. Database Management System (DBMS)

A DBMS is a software system used to manage databases and allow for easy access, manipulation, and organization of data. Key specifications for DBMS include:

Type:

Relational DBMS (RDBMS): e.g., MySQL, PostgreSQL, Oracle. They store data in tables with predefined schemas and relationships.

NoSQL DBMS: e.g., MongoDB, Cassandra, CouchDB. These are used for unstructured or semi-structured data and are more flexible.

Query Language: The most common query language is SQL (Structured Query Language) for RDBMS, but NoSQL databases use various query mechanisms (e.g., MongoDB Query Language).

Data Integrity and Security: Ensures that data remains consistent, secure, and protected from unauthorized access.

5. Web Development Frameworks

For building web applications, developers use web frameworks that streamline the development process by providing prebuilt functions, components, and tools. Specifications include:

Frontend Frameworks:

React.js, Vue.js, Angular (JavaScript-based) for building user interfaces.

HTML, CSS, and JavaScript as core building blocks of the frontend.

Backend Frameworks:

Node.js (JavaScript-based), Django (Python), Spring Boot (Java), Ruby on Rails (Ruby) for building server-side applications.

Full Stack Development: Some frameworks offer full-stack solutions, combining both frontend and backend development, like Meteor.js.

**SYSTEM ANALYSIS AND DESIGN**

In the realm of computer science, web development plays an essential role in creating interactive and engaging user experiences. One such example is the development of a quiz website, which can be built using HTML (HyperText Markup Language) and CSS (Cascading Style Sheets). These two foundational technologies are crucial for creating functional and visually appealing websites. A quiz website can be an excellent demonstration of how these technologies come together to form an interactive web application. The combination of HTML for structuring the content and CSS for styling the layout is an essential aspect of web development, forming the backbone of the user interface (UI) and user experience (UX).

### ****HTML and CSS in Web Development****

Before diving into the specific application of HTML and CSS in a quiz website, it’s important to understand their roles in web development.

* **HTML (HyperText Markup Language)**: HTML is the standard language used to create the structure of a web page. It defines the layout, content, and semantics of the page by using tags such as <div>, <form>, <input>, <button>, and others. HTML provides the skeleton of the website, organizing the content in a meaningful and accessible way for browsers to render and for users to interact with.
* **CSS (Cascading Style Sheets)**: CSS, on the other hand, is responsible for styling the HTML content. It controls the look and feel of the website, including fonts, colors, spacing, layout, and responsiveness. CSS ensures that the user interface is visually appealing and easy to navigate. It allows developers to apply styles to individual HTML elements or groups of elements, creating a cohesive and user-friendly design.

In the context of computer science, HTML and CSS are often used together to build the frontend of a website—what users see and interact with. By combining the structural power of HTML with the visual design capabilities of CSS, web developers can create websites that are both functional and aesthetically pleasing.

### ****Creating the Quiz Website****

For this example, I will describe the development of a simple quiz website with 10 questions. This quiz website is constructed using both HTML and CSS to create a smooth and intuitive user experience. Here’s how these technologies come together to form the complete system:

The HTML code for the quiz website is structured to create the layout of the quiz, define the questions, and organize the input elements for user interaction. The quiz consists of 10 multiple-choice questions, each with four answer options.

CSS enhances the quiz’s usability and appearance by applying styles to the HTML elements. The styling helps to create a visually appealing interface, making the quiz more engaging and easier to navi

**Hardware Requirements**

**Computer (Desktop/Laptop)**:

**Processor**: A modern processor such as Intel i3/i5 or AMD Ryzen will suffice for a quiz website. While a basic website does not require heavy computational power, having a decent processor helps when working with multiple applications or larger projects.

**RAM**: A minimum of 4 GB of RAM is necessary for running your text editor/IDE, browsers, and other tools simultaneously. However, 8 GB of RAM is recommended, especially if you're working with larger files or using additional programs like graphic design tools or local servers.

**Storage**: For a simple quiz website, around 50 GB of free storage is sufficient for coding, file storage, and testing. If your project involves large images or assets, more storage may be necessary.

**Operating System**: The operating system could be Windows, macOS, or Linux-based OS (Ubuntu, Fedora, etc.). The key requirement is compatibility with the development tools you intend to use (code editors, browsers, and version control systems).

**Internet Connection**:

A stable internet connection is essential for multiple tasks such as downloading resources, accessing online tutorials, testing the website in various browsers, and uploading the website for deployment. For a local website, you might only need internet to fetch libraries or plugins, but for testing and hosting, fast internet speeds are beneficial.

**Testing Devices**:

To ensure your quiz website is fully functional, testing it on multiple devices with different screen sizes is essential. You'll need access to laptops, smartphones, and tablets to verify the responsiveness of the site and its compatibility with different browsers. Modern browsers come with developer tools for simulating different screen sizes, but real device testing helps ensure more accurate results.

**Web Hosting Server (For Deployment)**:

Once the quiz website is ready, you’ll need a hosting service to make it available online. You can opt for shared hosting, VPS, or cloud servers, depending on your needs. Shared hosting should suffice for a small-scale quiz website. Hosting services should provide FTP or SSH access to upload your files and support HTML, CSS, and JavaScript files. Backup storage is also a good practice to ensure that your website's files are safe.

**Software Requirements**

**Text Editor/IDE**:

You need a text editor or an Integrated Development Environment (IDE) to write your HTML, CSS, and potentially JavaScript. Popular text editors and IDEs include:

**Visual Studio Code (VS Code)**: A versatile and popular code editor that supports extensions, syntax highlighting, and debugging features.

**Sublime Text**: A lightweight and fast text editor with a user-friendly interface.

**Atom**: An open-source text editor with many customization options.

**Notepad++**: A basic text editor, particularly useful for Windows users. These tools are necessary for writing clean and efficient code for the quiz website.

**Web Browsers**:

Browsers such as **Google Chrome**, **Mozilla Firefox**, and **Safari** are essential for testing your website. These browsers offer developer tools that allow you to inspect HTML, debug JavaScript, and adjust CSS styles. You should test the website on multiple browsers to ensure it displays and functions correctly for all users.

**Google Chrome DevTools** or **Firefox Developer Tools** can be used to simulate different screen sizes, inspect elements, and troubleshoot any issues related to responsiveness or functionality.

**Version Control** (Optional but Recommended):

**Git**: A version control system (VCS) is a must if you're working on a larger project or collaborating with others. Git tracks changes in your code and helps manage different versions of the project. This allows you to revert changes if needed and work in teams without conflicts.

**GitHub** or **GitLab**: Online platforms that allow you to host Git repositories. These platforms make it easy to share your code, track issues, and collaborate with others. They also serve as backup storage for your project.

**Graphics Software** (Optional):

If you plan to use images or custom graphics in your quiz website, graphic design tools may be needed. Tools like **Adobe Photoshop** or **GIMP** (open-source alternative) are ideal for creating and editing images. **Figma** is another great tool for designing user interfaces, and it can also be used for prototyping the website's layout before development.

**SOURCE CODE**

**HOME PAGE**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Quiz Page</title>

<style>

body {

font-family: Arial, sans-serif;

background-color: #f4f4f9;

margin: 0;

padding: 0;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

}

.quiz-container, .results-container {

background: #fff;

padding: 20px 30px;

border-radius: 10px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

width: 100%;

max-width: 600px;

text-align: center;

}

h1 {

margin-bottom: 20px;

color: #333;

}

.question {

margin: 20px 0;

text-align: left;

}

button {

padding: 10px 20px;

background: #007BFF;

color: #fff;

border: none;

border-radius: 5px;

cursor: pointer;

font-size: 16px;

}

button:hover {

background: #0056b3;

}

#score {

font-size: 18px;

color: #333;

margin-bottom: 20px;

}

a {

text-decoration: none;

color: #007BFF;

font-weight: bold;

}

a:hover {

text-decoration: underline;

}

.question label {

display: block;

}

form {

text-align: left;

}

</style>

</head>

<body>

<div class="quiz-container">

<h1>Simple Quiz</h1>

<form action="results.html" method="POST">

<!-- Questions -->

<div class="question">

<p>1. What is the capital of India?</p>

<label><input type="radio" name="q1" value="A"> A) Mumbai</label><br>

<label><input type="radio" name="q1" value="B"> B) New Delhi</label><br>

<label><input type="radio" name="q1" value="C"> C) Shimla</label><br>

<label><input type="radio" name="q1" value="D"> D) bihar</label>

</div>

<div class="question">

<p>2. Which programming language is best for web development?</p>

<label><input type="radio" name="q2" value="A"> A) Python</label><br>

<label><input type="radio" name="q2" value="B"> B) JavaScript</label><br>

<label><input type="radio" name="q2" value="C"> C) C++</label><br>

<label><input type="radio" name="q2" value="D"> D) Java</label>

</div>

<div class="question">

<p>3. What is 5 + 3?</p>

<label><input type="radio" name="q3" value="A"> A) 6</label><br>

<label><input type="radio" name="q3" value="B"> B) 7</label><br>

<label><input type="radio" name="q3" value="C"> C) 8</label><br>

<label><input type="radio" name="q3" value="D"> D) 9</label>

</div>

<div class="question">

<p>4. What is the largest planet in the solar system?</p>

<label><input type="radio" name="q4" value="A"> A) Earth</label><br>

<label><input type="radio" name="q4" value="B"> B) Jupiter</label><br>

<label><input type="radio" name="q4" value="C"> C) Saturn</label><br>

<label><input type="radio" name="q4" value="D"> D) Mars</label>

</div>

<div class="question">

<p>5. What is the boiling point of water?</p>

<label><input type="radio" name="q5" value="A"> A) 50°C</label><br>

<label><input type="radio" name="q5" value="B"> B) 100°C</label><br>

<label><input type="radio" name="q5" value="C"> C) 150°C</label><br>

<label><input type="radio" name="q5" value="D"> D) 200°C</label>

</div>

<div class="question">

<p>6. Which of the following is not a programming language?</p>

<label><input type="radio" name="q5" value="A"> A) PythonC</label><br>

<label><input type="radio" name="q5" value="B"> B) Java</label><br>

<label><input type="radio" name="q5" value="C"> C) HTML</label><br>

<label><input type="radio" name="q5" value="D"> D) C++</label>

</div>

<div class="question">

<p>7. Which data structure uses Last In First Out method?</p>

<label><input type="radio" name="q5" value="A"> A) Queue</label><br>

<label><input type="radio" name="q5" value="B"> B) Stack</label><br>

<label><input type="radio" name="q5" value="C"> C) Array</label><br>

<label><input type="radio" name="q5" value="D"> D) Linked ListC</label>

</div>

<div class="question">

<p>8. Which of the following is an example of an operating system?</p>

<label><input type="radio" name="q5" value="A"> A) MS Word</label><br>

<label><input type="radio" name="q5" value="B"> B) Chrome</label><br>

<label><input type="radio" name="q5" value="C"> C) Windows 11</label><br>

<label><input type="radio" name="q5" value="D"> D) Photoshop</label>

</div>

<div class="question">

<p>9. Which of these is a high level programming language?</p>

<label><input type="radio" name="q5" value="A"> A) Python</label><br>

<label><input type="radio" name="q5" value="B"> B) Machine Code</label><br>

<label><input type="radio" name="q5" value="C"> C) Assembl</label><br>

<label><input type="radio" name="q5" value="D"> D) Binary</label>

</div>

<div class="question">

<p>10. Which of the following is an example of an open source operating system?</p>

<label><input type="radio" name="q5" value="A"> A) Windows</label><br>

<label><input type="radio" name="q5" value="B"> B) iOS</label><br>

<label><input type="radio" name="q5" value="C"> C) macOS</label><br>

<label><input type="radio" name="q5" value="D"> D) Linux</label>

</div>

<button type="submit">Submit</button>

</form>

</div>

</body>

</html>

**RESULT PAGE**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Quiz Results</title>

<style>

body {

font-family: Arial, sans-serif;

background-color: #f4f4f9;

margin: 0;

padding: 0;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

}

.results-container {

background: #fff;

padding: 20px 30px;

border-radius: 10px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

width: 100%;

max-width: 600px;

text-align: center;

}

h1 {

margin-bottom: 20px;

color: #333;

}

p {

font-size: 18px;

color: #333;

}

a {

text-decoration: none;

color: #007BFF;

font-weight: bold;

}

a:hover {

text-decoration: underline;

}

</style>

</head>

<body>

<div class="results-container">

<h1>Your Quiz Answers</h1>

<p>1. B)<br>2. B)<br> 3. C)<br> 4. B)<br>5. B)<br>6. C)<br>7. B)<br>8. C)<br>9. A)<br>10. D)</p>

<a href="quiz.html">Back to Quiz</a>

</div>

</body>

</html>

**WORKING OF HOME PAGE**

**HTML**

I begin by declaring the document type as <!DOCTYPE html> to specify that this is an HTML5 document.

The <html lang="en"> element starts the HTML document, and I set the language to English with lang="en".

Inside the <head>, I include metadata like:

<meta charset="UTF-8"> to define the character encoding for the page.

<meta name="viewport" content="width=device-width, initial-scale=1.0"> to ensure the page scales properly on different devices (like phones and tablets).

I set the title of the page using <title>Quiz Page</title>.

I also define internal CSS styles for page formatting within the <style> tags.

Inside the <body>, I create a div with the class quiz-container to contain all the quiz content.

I add a heading <h1>Simple Quiz</h1> to give the page a title, which is displayed prominently.

The quiz form starts with <form action="results.html" method="POST">. This means when the form is submitted, the data will be sent to results.html using the POST method.

For each question, I wrap the content inside a <div> with a class question for styling.

I use a <p> tag to display the question text.

I provide four options for each question using radio buttons (<input type="radio">). The nme="q1" ensures that only one radio button can be selected per question (since they share the same name).

Each radio button has a value that identifies the option when the form is submitted.

Each radio button option is wrapped in a <label> element so that clicking the label also selects the option, making it user-friendly.

After all the questions, I place a <button> with type="submit". This button triggers the submission of the form. When clicked, the answers selected by the user are sent to the action URL (results.html), which can then process the answers.

**CSS**

In the <style> tag, I define the styles for the page.

For the body, I set a background color (#f4f4f9), set the font to Arial, and remove default margins and padding.

I use Flexbox (display: flex) to center the content both horizontally and vertically. The height: 100vh ensures the page fills the entire height of the viewport.

The .quiz-container class styles the main box around the quiz. I give it a white background, some padding, rounded corners (border-radius), and a soft shadow (box-shadow) for a card-like appearance.

I set the width to 100%, but limit the maximum width to 600px so the quiz doesn't get too wide on larger screens.

The text-align: center ensures that the text inside is centered.

The button has padding for spacing, a blue background (#007BFF), and white text.

I remove the default border and round the corners with border-radius.

The cursor: pointer changes the cursor to a hand when hovering over the button. I also define a hover effect, changing the background to a darker blue (#0056b3) when the button is hovered over.

### Summary:

* **HTML Structure**: I created a quiz with 10 multiple-choice questions using radio buttons.
* **Form Submission**: The form uses the POST method to send the answers to a "results.html" page when the user clicks "Submit".
* **Styling**: I styled the page to be centered on the screen with a clean, modern look using Flexbox and a card-like container for the quiz.

**WORKING OF PAGE 2**

### <head> ****Section****

In the <head> part, I define some settings and styles for the webpage:

**Character Encoding**:

<meta charset="UTF-8">: This ensures the webpage can display any special characters correctly.

**Responsive Design**:

<meta name="viewport" content="width=device-width, initial-scale=1.0">: This makes the page adjust to different screen sizes, especially for mobile devices.

**Title**:

<title>Quiz Results</title>: Sets the title that appears in the browser tab to "Quiz Results".

**CSS Styling**:

Inside the <style> tag, I added some basic styles to make the page look nice, such as background color, fonts, and spacing.

### <body> ****Section****

This is where the content of the page is shown to users:

**Results Container**:

<div class="results-container">: This section contains everything the user will see on the results page. It is centered on the screen using Flexbox and has a white background with padding for space.

**Quiz Title**:

<h1>Your Quiz Answers</h1>: This heading introduces the results section.

**Displaying Results**:

<p>1. B)<br>2. B)<br>3. C)<br>... displays the answers to each quiz question. Each answer is written as "question number. answer choice", like "1. B)".

**Back to Quiz Link**:

<a href="quiz.html">Back to Quiz</a>: This link lets the user go back to the quiz page. It is styled with blue text and will underline when hovered.

### ****3. CSS Styles****

I used CSS to control how the page looks:

**Body Styles**:

Set the background color to light gray and font to Arial for readability.

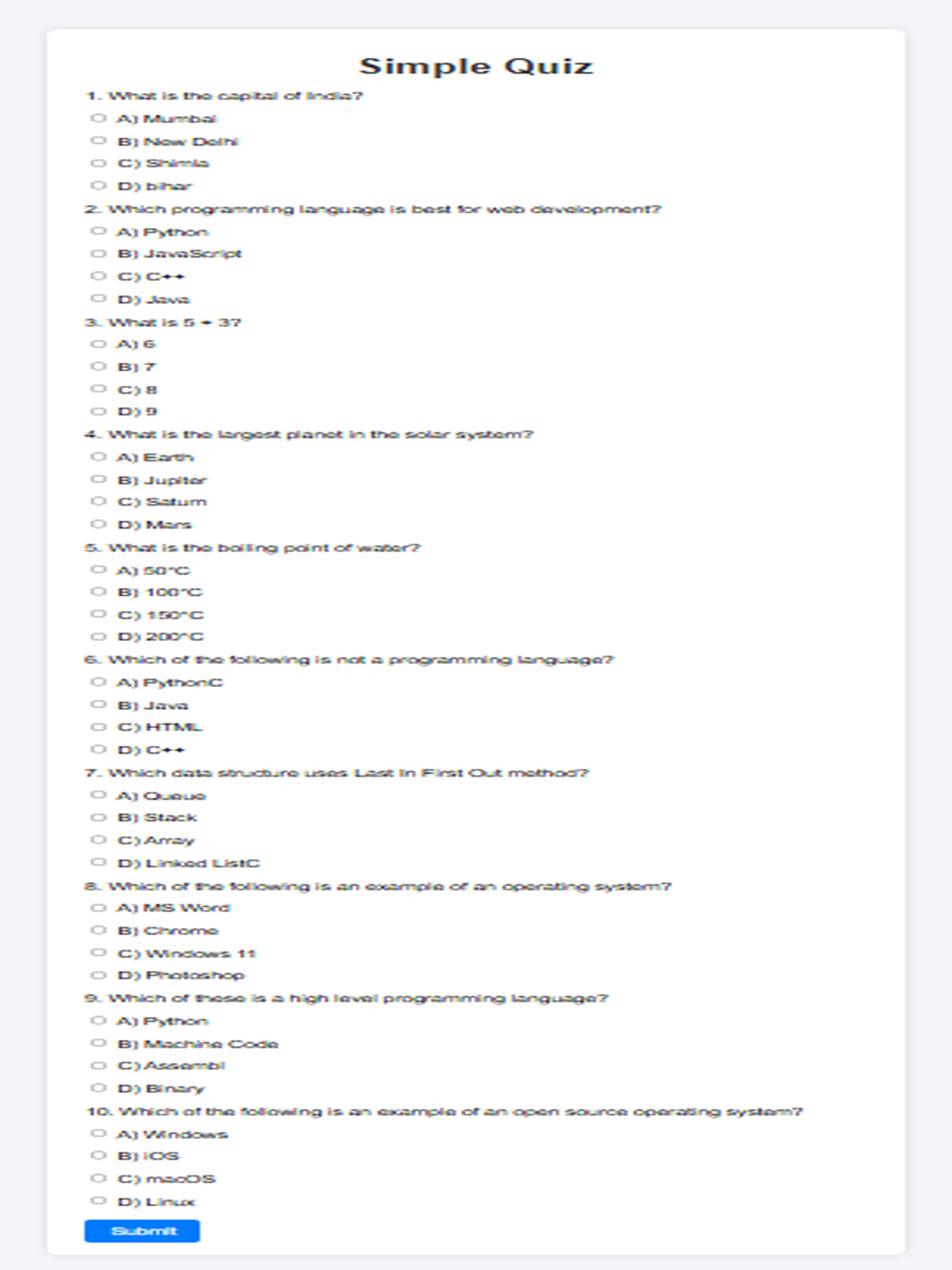
**Results Container**:

White background with rounded corners and shadow to make it stand out.

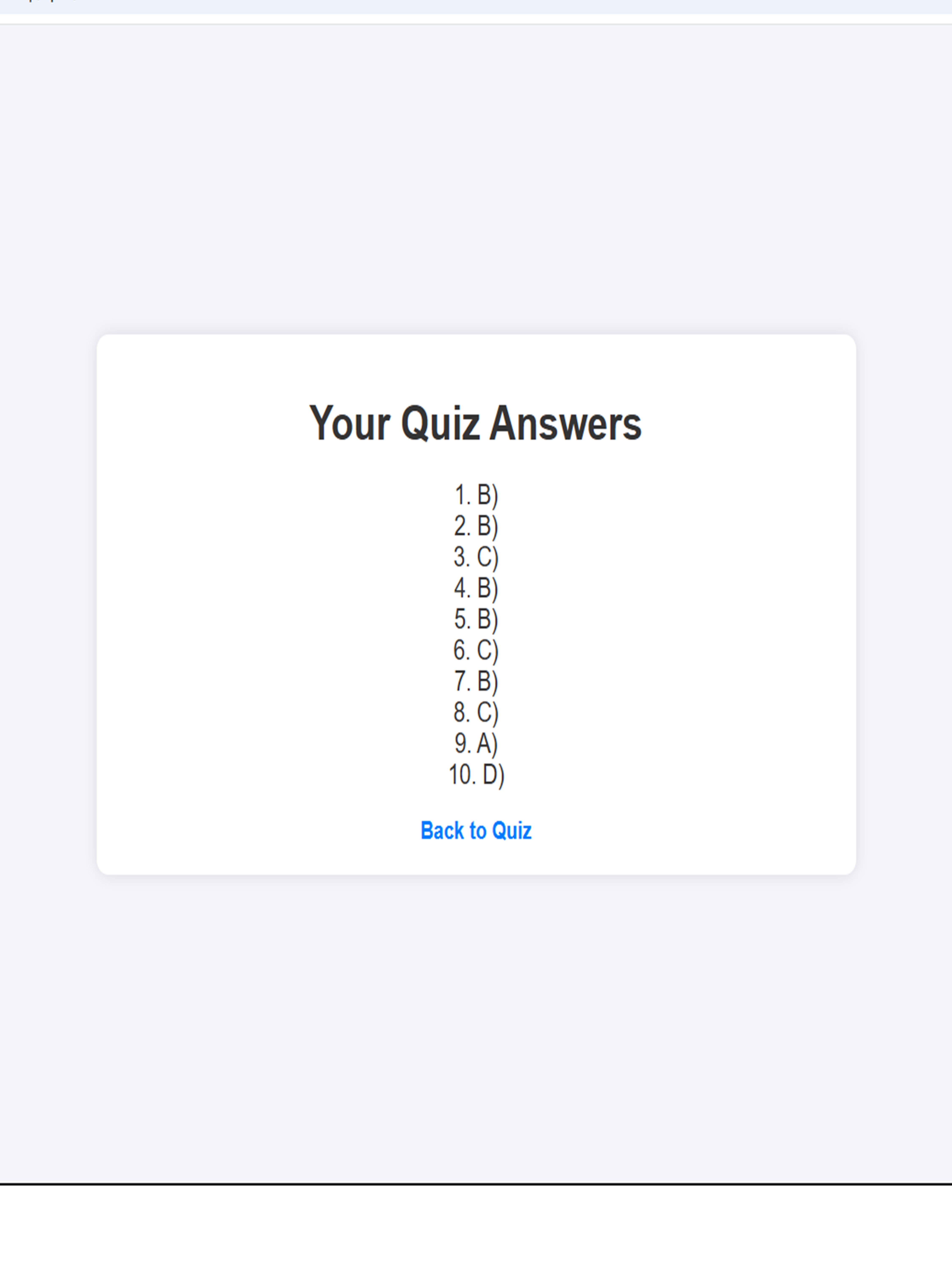
**Text and Links**:

The heading and text have a dark gray color, and the link is blue with an underline effect when hovered.

**OUTPUT PAGE 1**



**PAGE 2**



**ADVANTAGES OF COMPUTER SCIENCE**

### 1. ****Problem-Solving Skills****

* Computer Science helps develop strong problem-solving abilities. Learning to write code and design algorithms requires logical thinking, creativity, and critical analysis, which can be applied in many other aspects of life and work.

### 2. ****Innovation and Creativity****

* It enables the creation of new technologies and tools that can transform industries. Innovations such as Artificial Intelligence (AI), machine learning, virtual reality, and blockchain all stem from computer science.

### 3. ****Improved Efficiency****

* Computers can process large amounts of data quickly and accurately. Computer science allows for the automation of repetitive tasks, which boosts productivity in various sectors like healthcare, finance, manufacturing, and more.

### 4. ****Career Opportunities****

* With the increasing reliance on technology, there is a high demand for computer science professionals. Careers in software development, cybersecurity, data science, and IT support are growing rapidly and offer competitive salaries.

### 5. ****Global Connectivity****

* Computer science has revolutionized communication. The internet, social media, video conferencing, and instant messaging are all made possible through advancements in computer science, helping people stay connected around the world.

### 6. ****Data Analysis and Decision Making****

* Computer science facilitates the collection and analysis of data. This allows businesses, governments, and individuals to make informed decisions based on accurate, real-time information.

### 7. ****Automation and AI****

* Automation powered by computer science can reduce human labor, increase precision, and allow tasks to be performed more efficiently. Artificial intelligence is also revolutionizing industries by enabling machines to perform tasks that were once considered only possible for humans.

### 8. ****Better Access to Information****

* With the development of the internet and search engines, information is more accessible than ever before. Computer science helps organize, store, and retrieve vast amounts of data efficiently.

### 9. ****Enhances Everyday Life****

* Computer science impacts daily activities through smartphones, apps, navigation systems, e-commerce platforms, and entertainment. It makes life more convenient, comfortable.

### 10. ****Improved Healthcare****

* Computer science is essential in the development of medical technologies such as diagnostic tools, robotic surgeries, and telemedicine. It enables better management of patient data and facilitates faster research in pharmaceuticals and treatment methods.

### 11. ****Support for Other Disciplines****

* Computer science supports numerous fields such as biology, chemistry, economics, physics, and social sciences. Tools like simulations, data analytics, and computational models have expanded the boundaries of research in these areas.

### 12. ****Sustainability and Environmental Impact****

* Computer science is crucial for developing technologies that monitor and improve environmental sustainability. For example, it helps in energy optimization, climate modeling, and efficient waste management systems.

### 13. ****Security and Privacy****

* With increasing threats to data and privacy, computer science plays a crucial role in cybersecurity. It allows for the creation of encryption, firewalls, and other security measures to protect sensitive information.

### 14. ****Global Collaboration and Open Source****

* Through collaborative platforms and open-source software, computer science enables developers from around the world to work together on projects, advancing innovation and sharing solutions for common problems.

### 15. ****Economic Growth****

* The technology industry is a significant driver of economic growth, with companies and startups that focus on software development, digital products, and IT solutions providing jobs, generating revenue, and fostering new industries.

**DISADVANTAGES**

### 1. ****Job Displacement****

* As automation and artificial intelligence (AI) technologies advance, there is a risk of job displacement in various industries. Tasks traditionally performed by humans are increasingly being automated, leading to reduced employment opportunities for some workers.

### 2. ****Cybersecurity Threats****

* The growing reliance on digital technologies has made systems more vulnerable to cyberattacks, hacking, and data breaches. Security vulnerabilities can expose sensitive personal, financial, or governmental data to malicious actors, causing financial, reputational, and personal harm.

### 3. ****Privacy Concerns****

* The widespread collection and use of personal data by companies, governments, and other entities raises serious concerns about privacy. People’s personal information can be tracked, monitored, and misused, leading to breaches of privacy and potential identity theft.

### 4. ****Digital Addiction and Dependency****

* With the increasing integration of computers and technology into everyday life, there is a risk of digital addiction. Excessive use of screens, video games, social media, and other digital platforms can negatively affect mental health, relationships, and productivity.

### 5. ****Social Isolation****

* While technology can help connect people globally, it can also lead to social isolation. Over-reliance on digital communication may reduce face-to-face interactions, leading to feelings of loneliness and decreased social skills.

### 6. ****Digital Divide****

* Access to computer science and technology is not evenly distributed, creating a "digital divide." People in rural or underdeveloped areas may lack access to the internet or modern technology, hindering their ability to participate in the digital economy, education, and other opportunities.

### 7. ****Environmental Impact****

* The production, use, and disposal of electronic devices contribute to environmental pollution. Manufacturing computers and smartphones consumes vast resources, and e-waste, when not properly managed, can lead to hazardous environmental effects.

### 8. ****Complexity and High Cost of Development****

* The development of advanced computer systems, software, and technologies often involves high costs and complex infrastructure. This can be a barrier for smaller companies or individuals, preventing them from accessing or creating the latest technological solutions.

### 9. ****Over-reliance on Technology****

* As society becomes more dependent on computers and technology, there is a risk of losing essential skills. People may struggle to perform tasks without the help of technology, and over-reliance can lead to problems when systems fail or are unavailable.

### 10. ****Ethical Issues****

* Many aspects of computer science raise ethical dilemmas. For example, AI-driven decision-making systems can perpetuate bias, surveillance technologies can infringe on privacy, and autonomous machines (like self-driving cars or robots) may have moral implications regarding responsibility and accountability.

### 11. ****Obsolescence of Skills****

* As technology evolves rapidly, workers in the computer science field may need constant retraining to keep up with the latest tools, languages, and techniques. This creates a cycle of obsolescence, where skills and knowledge quickly become outdated.

### 12. ****Security Risks in the Internet of Things (IoT)****

* The increasing number of interconnected devices in the Internet of Things (IoT) creates new entry points for cyberattacks. These devices, such as smart home gadgets and wearable health devices, can be hacked or misused, leading to security and privacy concerns.

### 13. ****Social Impact of Automation****

* Automation and AI can have significant social impacts, including the reduction of jobs in sectors like manufacturing, retail, and customer service. This can lead to economic inequality and social unrest if not properly managed.

### 14. ****Over-information and Misinformation****

* The internet and digital platforms offer vast amounts of information, but much of it can be inaccurate or misleading. This contributes to the spread of misinformation, fake news, and conspiracy theories, which can harm individuals and societies.

**APPLICATIONS**

### 1. ****Software Development****

* **Applications**: Creating software programs, apps, and operating systems for various platforms such as desktop computers, mobile devices, and web applications. Popular examples include Microsoft Office, iOS/Android apps, and web services like Google and Facebook.
* **Impact**: Facilitates communication, entertainment, work, and education through digital tools.

### 2. ****Artificial Intelligence (AI)****

* **Applications**: Machine learning, natural language processing, computer vision, and robotics. Examples include speech recognition systems (e.g., Siri, Alexa), recommendation systems (e.g., Netflix, Amazon), and self-driving cars.
* **Impact**: Automates tasks, improves decision-making, and enhances user experiences across industries like healthcare, finance, and entertainment.

### 3. ****Data Science and Big Data****

* **Applications**: Analyzing large datasets to extract meaningful insights. This includes data mining, predictive analytics, and business intelligence. Examples are customer behavior analysis, fraud detection, and health data analysis.
* **Impact**: Helps organizations make data-driven decisions, improve services, and optimize processes.

### 4. ****Cybersecurity****

* **Applications**: Protecting systems, networks, and data from digital attacks. Key technologies include encryption, firewalls, antivirus software, and penetration testing.
* **Impact**: Ensures the safety and privacy of personal, corporate, and government data in the digital world.

### 5. ****Networking and Communication****

* **Applications**: The design and management of communication systems and networks, including the internet, mobile networks, and cloud computing infrastructure.
* **Impact**: Facilitates global communication, business operations, and remote work through technologies like the internet, Wi-Fi, and 5G.

### 6. ****Healthcare and Medical Informatics****

* **Applications**: Electronic health records (EHRs), telemedicine, medical imaging, and health data analytics. AI is used in diagnostics, drug discovery, and robotic surgeries.
* **Impact**: Improves patient care, streamlines medical processes, and enables faster medical research and personalized treatment.

### 7. ****Robotics****

* **Applications**: Designing and programming robots to perform tasks in manufacturing, healthcare (e.g., surgical robots), and consumer products (e.g., robotic vacuum cleaners).
* **Impact**: Increases efficiency and accuracy in industrial processes, healthcare treatments, and everyday life.

### 8. ****Cloud Computing****

* **Applications**: The use of remote servers hosted on the internet to store, manage, and process data. Popular cloud services include Amazon Web Services (AWS), Google Cloud, and Microsoft Azure.
* **Impact**: Reduces the need for on-premise hardware, increases flexibility, and enables remote access to applications and data.

### 9. ****Virtual Reality (VR) and Augmented Reality (AR)****

* **Applications**: VR immerses users in a virtual world (e.g., gaming, training simulations), while AR overlays digital information onto the real world (e.g., Pokémon Go, AR in medical training).
* **Impact**: Enhances user experiences in gaming, education, healthcare, and retail by providing more interactive and engaging environments.

### 10. ****E-commerce****

* **Applications**: Online platforms for buying and selling goods and services, including websites like Amazon, eBay, and Alibaba. Payment systems, inventory management, and customer service are all part of this application.
* **Impact**: Transforms how people shop, conduct business, and engage with products and services, contributing to the growth of the global digital economy.

### 11. ****Game Development****

* **Applications**: Creating interactive video games, from mobile games to large-scale console and PC games. This involves game design, graphics programming, AI in games, and multiplayer networks.
* **Impact**: Provides entertainment and has become a multi-billion-dollar industry, fostering creativity and innovation in interactive storytelling and design.

### 12. ****Education Technology (EdTech)****

* **Applications**: Online learning platforms, virtual classrooms, educational apps, and e-books. Examples include platforms like Coursera, Khan Academy, and Moodle.
* **Impact**: Enhances access to education, provides personalized learning experiences, and enables remote learning worldwide.

**CONLUSION / RECOMMENDATIONS**

the study and application of Computer Science have greatly influenced the world, enabling innovations that drive the modern technological landscape. The development of a quiz website using HTML and CSS has demonstrated the practical side of computer science, showcasing the importance of web development in creating interactive platforms. Through the system analysis and website development, it is clear that computer science is essential in simplifying complex tasks, enhancing efficiency, and providing solutions in various fields, from education to business and entertainment.

Computer science has proven to be a pivotal driver of progress, from developing user-friendly applications to advancing artificial intelligence and machine learning technologies. However, like any powerful tool, it comes with challenges such as security risks, ethical concerns, and the potential for job displacement due to automation.

Recommendations

1. Continuous Learning and Skill Development: As technology evolves, it’s important to continue learning and staying updated with the latest programming languages, frameworks, and best practices. Developers and computer scientists should pursue continuous education through courses, tutorials, and projects to keep their skills relevant.

2. Focus on Security: Given the rapid advancement of technology and the increasing threats to cybersecurity, it's vital to prioritize robust security measures in all software applications. Implementing encryption, secure coding practices, and constant security audits will help mitigate risks.

3. Ethical Considerations in Development: Computer scientists should be mindful of the ethical implications of their work, ensuring that technologies are developed in ways that respect privacy, promote fairness, and do not contribute to discrimination or harm.

4. Collaboration Across Disciplines: The future of computer science lies in interdisciplinary collaboration. By working together with professionals from other fields (e.g., healthcare, engineering, and social sciences), computer scientists can create more impactful solutions to global challenges.

5. Improved Accessibility: There should be a continued focus on making technology accessible to all, including those with disabilities. Designing inclusive applications and systems will help bridge the digital divide and allow more people to benefit from technological advancements.

6. Encourage Innovation in Education: Institutions and educators should leverage the potential of computer science to foster creativity and innovation in students. Introducing coding, problem-solving, and algorithmic thinking from a young age will equip the next generation with the necessary skills to tackle future challenges.

By following these recommendations, the field of computer science can continue to thrive and provide solutions that improve the quality of life for individuals and society as a whole.

**REFERENCES**

For this project, a variety of resources were consulted to aid in the development and understanding of computer science concepts, web development techniques, and relevant technologies. Tutorials from YouTube channels such as freeCodeCamp provided valuable insights into the basics of HTML and CSS, specifically in building interactive websites like the quiz application used in this project. Additionally, ChatGPT offered assistance with HTML, CSS, and JavaScript code generation, guiding the creation and troubleshooting of the quiz functionality. Information from Google was also crucial in gathering general knowledge on computer science principles, including system analysis and the applications of web development. Insights from previous students who had prior experience with similar projects were extremely helpful, as they shared practical knowledge on building quiz websites and overcoming common challenges. Reports and academic articles on computer science, such as those available in Journal of Computer Science Advances, provided deeper insights into the theoretical and practical aspects of computer science, ensuring that the project was grounded in both practical application and academic understanding.

**APPENDICES**

Under the System Analysis and Design section of this project, the appendices are provided in four distinct subparts to offer a comprehensive understanding of the quiz website's development and functionality. The first subpart, Resources Required, outlines the hardware and software specifications necessary to run the website, including web browsers and operating systems. The second subpart, Source Code, includes the complete HTML, CSS, and JavaScript code used to create the quiz application, providing detailed insight into the structure and functionality of the website. The third subpart, Working, describes the workflow of the website, including the logic for processing user input, calculating results, and transitioning between pages. Finally, the Output (Screenshots) subpart presents visual evidence of the website's operation, including key stages like the quiz page and results page, allowing readers to understand the user interface and experience. These appendices collectively support the system analysis and design of the project, ensuring that all technical aspects are thoroughly documented and easily accessible.