|  |  |  |  |
| --- | --- | --- | --- |
| **Qualification details** | | | |
| **Training Package Code and Title** | ICT - Information and Communications Technology (Version 8.1) | | |
| **Qualification National Code and Title** | ICT50220 Diploma of information Technology (Release 2) | **State code** | BGJ4 |
| **Assessment Title** *(as per DAP)* | Assessment Task One (Individual Project) | | |
| **Unit National Code & Title** | ICTWEB513 Build dynamic websites | | |
| ICTWEB514 Create dynamic web pages | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Date Due** | Week Five | | **Date Received** | |  | |
| **Student Name** | Joshua Sambajon / 30084775 | | | | | |
| **Student Declaration** | I declare that the evidence submitted is my own work: | | | | | |
| **Assessor Name** |  | | | | | |
| **Assessment Decision** | Satisfactory | | | Not Yet Satisfactory | | |
| **Assessor Signature** |  | | | **Date** | |  |
| **Is student eligible for reassessment (Re-sit)?** | No | Yes | | **Re-assessment Date:** | | Week Nineteen |

|  |  |  |  |
| --- | --- | --- | --- |
| **Feedback to student** | | | |
| *Via Blackboard (LMS) – Please check [Grade] section.* | | | |
| **Feedback from student** | | | |
| *Via Blackboard (LMS) – Please use [Comment] section during submission.* | | | |
| **Student signature** |  | **Date** |  |

|  |  |
| --- | --- |
| **Assessment Instructions** | |
| **TO THE ASSESSOR** |  |
| Type of Assessment | Individual Project |
| Duration of the assessment | 5 class sessions (Weeks 1-5) |
| Location of assessment | Classroom |
| Conditions | Assessor to ensure that the noise levels, natural interactions and time variances are maintained as it would be in the Software Development industry.  Learners are required to complete the required tasks in class and submit the required documentation electronically via Blackboard |
| Elements and Criteria | As detailed in the assessment plan  You are required to make sure that all students meet the elements, performance criteria and oral communication items as outlined in the provided solution |
| **TO THE STUDENT** |  |
| Purpose of Assessment | You are required to show you can:  ICTWEB513 Build dynamic websites   * Demonstrate your skills and knowledge by creating, coding, debugging, and testing a dynamic website, * Establish user requirements and then research and collect information about business requirements and legislative standards, * Manage time and tasks to produce a hierarchy of web pages showing navigation.   ICTWEB514 Create dynamic web pages   * Review technical requirements for client-side dynamic content, * Apply applicable languages and technologies to develop templates for web site creation, * Test and evaluate the dynamic content and present feedback.   The student must demonstrate the ability to complete the tasks outlined in this assessment and is expected to use systematic analytical processes and effect time management to meet the goals/deadlines outlined in the DAP. |

|  |  |
| --- | --- |
| Allowable Materials | Blackboard Topics, SDLC, Weekly readings (PDF), Example programs and Independent Outside of Class Activities |
| Required Resources | Web links and example code can be downloaded from the Blackboard portal.  PC with Notepad++, Turnkey Web Server, GitHub, MSOffice.  Internet Access to GitHub and www.citems.com.au/ |
| Reasonable Adjustment | In some circumstances, adjustments to assessments may be made for you. If you require support for literacy and numeracy issues; support for hearing, sight or mobility issues; change to assessment times/venues; use of special or adaptive technology; considerations relating to age, gender and cultural beliefs; format of assessment materials; or presence of a scribe you need to inform your lecturer. |
| Assessment Submission | All questions and programming activities must be attempted. All written answers must be submitted in this assessment document in the appropriate space.  Use of research tools and peers in formulating answers are acceptable – but work submitted must be your own work.  Final project documentation is to be uploaded to the appropriate area in the Blackboard course created for this unit.  If you are marked as NYS (Not Yet Satisfactory) on your first attempt, you will be provided with another opportunity to re-attempt the assessment. |
| Portfolio Description | A project of web coding tasks and written questions which should be completed in class and finished in the students’ own time on a weekly basis as per the Delivery and Assessment schedule.  Question 1 – Design Specifications  Question 2 – Web Page Content  Question 3 – Version Control  Question 4 – Design Approval  Question 5 – Website Development  Question 6 – Testing  Question 7 – Demonstration, Feedback and Signoff |

# Scenario

You have applied for the role of a Senior Web Programmer with CITE Managed Services, as part of the application process you are required to demonstrate your knowledge and skills by creating a multi-page website. The details and criteria are provided in the following paragraphs.

The multi-page website will utilise the Bootstrap framework for navigation and display information as requested. Ensure your development follows an Agile methodology that is recorded and maintained using your GitHub account.

You should consult with the CITE representative (your Lecturer) if you are unsure about any of the problems or questions in this assessment. Your primary research should focus on the resources on the Blackboard LMS and CITE web site, additional information can be collected from the Internet, ensure all sources are referenced in your submission. You must demonstrate your working website before uploading to Blackboard, your Lecturer (Assessor) will sign off to ensure all the criteria are satisfied.

## Minimum Client Requirements

* A single home web page (index.html) as the entry point into the website.
* A single contact web page (contact.html) with links to the CITE and SMTAFE websites.
* The user can navigate between all web pages using a suitably labelled navigation system.
* Navigation can be vertical/horizontal or tabbed using the appropriate Bootstrap framework.
* The navigation must be consistent across all web pages. All web pages must have a consistent theme (colours, fonts, etc)
* The user can select/click an item on the content web pages and the appropriate answer/definition will be displayed.
* The content display must be accordion or collapse; any variation must be approved by the Lecturer before implementation.
* The website must be compatible with all contemporary web browsers.
* The website must be compatible with all major devices (PC, Mobile).
* The website must be WCAG compliant where appropriate.
* The development must fully utilise all aspects of the Bootstrap framework version 5; visit the Bootstrap URL to review and select the appropriate components, https://getbootstrap.com.

## Suggested Interface Design

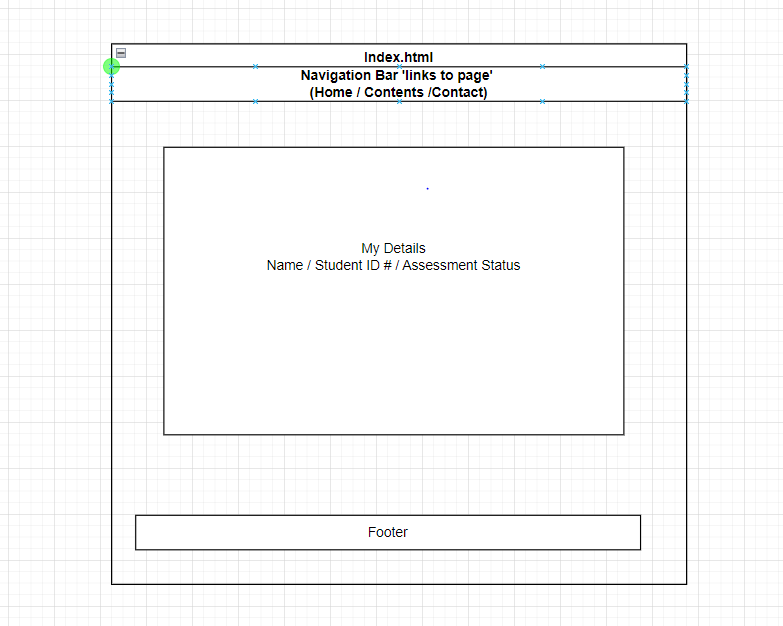
|  |  |  |
| --- | --- | --- |
| Graphical user interface, application  Description automatically generated | Graphical user interface, text, application, email  Description automatically generated | Graphical user interface, text, application, Word  Description automatically generated |
| Home Page | Content (accordion layout) | Content (collapse layout) |

## Question 1 Design Specification

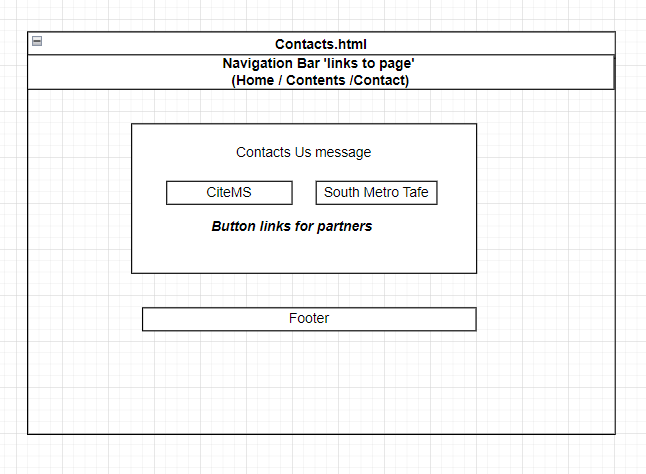
Provide a suitable description/explanation for each client requirement, and then insert your GUI design with labels that highlight all the major features. Complete the following Design Specification template to answer this question.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Design Specification | | | | |
| Developer Name | Joshua Sambajon | | Date |  |
| Turkey Web Server URL |  | | | |
| Technical Requirements | | | | |
| Requirement | | Description | | |
| 1. What is the purpose of the Website? | | The website serves as an interactive platform for CITE Managed Services, featuring a home page with general information and a contact page with links to CITE and SMTAFE. It uses Bootstrap 5 for a modern, responsive design and includes a navigation system for easy page transitions. | | |
| 2. Functionality: How will a user navigation the website and access the content? | | Each page of the website features a uniform navigation bar for easy navigation, with links to the contact page and home page. The home page includes general information and interactive content using Bootstrap's Accordion or Collapse components for detailed views. The contact page provides links to external websites. Both the content display and navigation are designed to be user-friendly and intuitive. | | |
| 3. Cross Platform: How will the website display on various OS and Device? | | The responsive design ensures the website functions well across various devices and operating systems, including tablets, smartphones, laptops, and desktops. This is achieved using Bootstrap's responsive grid system and media queries. We will test the website on multiple devices and browsers (Chrome, Firefox, Safari, Edge) to ensure consistent functionality and performance. | | |
| 4. Libraries and Frameworks: What web technologies will be used in the website? | | Bootstrap 5: For layout and responsive design, including navigation bars, accordions, and collapsible content.  HTML5: For structuring the content of the web pages.  CSS3: For styling and ensuring visual consistency.  JavaScript (Bootstrap's JS components): For interactive elements such as accordions and collapsible sections. | | |
|  | |  | | |
| Prototype Specification (GUI Design Diagram and Navigation Diagram) | | | | |

**GUI Design Diagram**

A screenshot of a computer screen

Description automatically generated



## A diagram of a navigation diagram Description automatically generatedQuestion 2 Web Page Content

Your next task is to create the content for each of the web pages on your website. You are required to research and provide suitable answers for the following groups of questions/definitions. Each group of questions/definitions must be displayed on a separate web page using a similar page layout as shown in your design.

### Content Questions (group one)

Provide a suitable answer for the following questions.

What are the principles of analysis and design?

Coding techniques are essential strategies for writing efficient, maintainable, and scalable code. Website coding techniques involve various methods and best practices to create efficient, maintainable, and scalable code. Key techniques include:

* **Modularization:**Breaking code into smaller, reusable functions or components makes it easier to manage and debug.
* **DRY (Don't Repeat Yourself):** This principle emphasizes avoiding code duplication, which simplifies maintenance and reduces errors.
* **KISS (Keep It Simple, Stupid):** Keeping code straightforward enhances readability and minimizes complexity.
* **Code Refactoring:** Regularly restructuring code improves clarity and efficiency without altering its functionality.
* **Version Control:** Using systems like Git to track changes helps manage collaboration and different versions of a codebase.
* **Error Handling:**Implementing proper error management ensures the application is robust and user-friendly.
* **Test-Driven Development (TDD):**Writing tests before coding helps ensure functionality works as intended.
* **Optimization:**Focusing on efficient algorithms and data structures improves performance and reduces resource usage.
* **Object-Oriented Programming (OOP)**This approach organizes code into classes and objects, promoting modularity and reusability.
* **Responsive Design:**Especially in web development, this technique ensures that applications adapt to various screen sizes and devices.

What are website architectural requirements?

* Website architectural requirements focus on the foundational elements necessary for a website to function effectively. These typically include:
* **Performance and Scalability:** Ensuring the website can handle the expected traffic and grow as needed without performance degradation.
* **Security:** Implementing measures to protect against threats such as data breaches and hacking attempts.
* **Accessibility:** Making sure the website is usable by people with disabilities, adhering to standards like WCAG (Web Content Accessibility Guidelines).
* **Responsiveness:** Ensuring the website performs well on various devices and screen sizes, including desktops, tablets, and smartphones.
* **Compliance:** Adhering to relevant laws and regulations, such as GDPR for data protection or CCPA for privacy.
* **Content Management:** Integrating with content management systems (CMS) for easy content updates and management.
* **SEO:** Optimizing the site for search engines to improve visibility and ranking in search results.
* **Integration:** Connecting with other systems or APIs as required (e.g., payment gateways, social media).

What are website design structures, including hierarchy and navigation design?

* Website Design structures define how the website’s layout and content are arranged. Common structures include:
* **Single Page Structure:** All content is displayed on one long, scrollable page. It is ideal for small websites, portfolios, or landing pages.
* **Multi-page Structure:** Multiple interconnected pages with a navigation menu that guides users between different sections. This is common for larger websites like e-commerce or corporate sites.
* **Grid-based Structure:** A layout that follows a grid to align text, images, and other elements, creating a sense of order and balance.
* **Footer Navigation:** Additional navigation links placed in the footer of the page. This can include links to less prominent pages, such as FAQs, blog, or terms of service.
* **F-Pattern Structure:** Designed based on how users naturally scan content in an "F" shape (across the top and down the left side). This is commonly used for text-heavy websites.
* **Z-Pattern Structure:** Follows a Z-like reading pattern, typically used for websites with less text and more visual elements, like landing pages.
* Website hierarchy refers to the arrangement of content in a way that communicates its relative importance, guiding users from most important to least important. It involves several aspects:
* **Typography:** Using different font sizes, styles, and weights to differentiate headings, subheadings, and body text. Larger and bolder text tends to attract more attention.
* **Colors and Contrast:** Highlighting important elements (like calls-to-action) with distinct colors or using contrasting shades to draw the user’s eye.
* **Whitespace:** The intentional use of empty space around elements to create focus and organization. Whitespace helps prevent clutter and makes content easier to digest.
* **Images and Icons:** Using images and icons to visually break up text and guide users toward important information or actions.

Content hierarchy organizes the website’s pages and guides users through the structure:

* **Home Page:** Often the most visited page, it typically provides a summary of key sections of the website. Its purpose is to direct users to the information they’re looking for.
* **Primary Navigation:** Usually located at the top or side of a page, it helps users find core pages like "About Us," "Services," "Products," and "Contact."
* **Sub-navigation (Dropdowns/Mega Menus):** Provides a secondary layer of navigation, helping users dive deeper into specific categories or content.
* **Call-to-Action (CTA):** Key buttons or links that guide users to take action, like signing up for a newsletter, purchasing a product, or requesting more information.
* **Footer:** Located at the bottom of the page, the footer typically contains links to additional resources like privacy policies, contact information, or social media.
* Navigation design ensures users can easily find and access the information they need. Key components include:
* **Global Navigation:** The primary navigation menu, typically located at the top of the page. It provides links to the main sections of the website, such as Home, About Us, Services, and Contact.
* **Local Navigation:** Menus or links specific to a section of the website, often found in sidebars or within the content area. It helps users navigate within a particular section or category.
* **Breadcrumbs:** A navigation aid showing users their current location within the site hierarchy. For example: Home > Products > Electronics > Smartphones. This helps users understand where they are and allows them to easily backtrack.
* **Search Functionality:** A search bar that allows users to quickly find specific content or products. It’s especially useful for sites with large amounts of content.
* **Dropdown Menus:** Menus that expand when clicked or hovered over to reveal additional options. These helps keep the main navigation clean and organized while providing access to subcategories.
* **Hamburger Menu:** An icon (usually three horizontal lines) that reveals a side menu when clicked or tapped. Commonly used in mobile or minimalist designs to save space.
* **Sticky Navigation:** A menu that remains visible and fixed at the top of the page as users scroll down. This ensures that navigation options are always accessible.
* **Footer Navigation:** Additional navigation links placed in the footer of the page. This can include links to less prominent pages, such as FAQs, blog, or terms of service.

What are user-interface design requirements and production processes?

* User-interface (UI) design requirements and production processes are crucial aspects of creating effective and user-friendly digital products. Here’s an overview of each:
* **Usability:** The interface should be intuitive and easy to use. This involves:
  + **Clear Navigation:** Easy access to different sections of the application or website.
  + **Consistent Layout:** Uniform design elements and behaviors throughout the application.
  + **Accessibility:** Design that is usable for people with various disabilities (e.g., color contrast, screen reader support).
* **Visual Design:**
  + **Aesthetics:** Use of colors, fonts, and images that align with the brand and enhance user experience.
  + **Responsiveness:** Design should work well on various devices and screen sizes (e.g., mobile, tablet, desktop).
  + **Hierarchy and Layout:** Effective use of space to guide users’ attention to important elements.
* **Functionality:**
  + **Interactive Elements:** Buttons, links, forms, and other elements should work as expected.
  + **Feedback Mechanisms:** Providing feedback (e.g., loading indicators, success/error messages) in response to user actions.
* **Performance:**
  + **Speed:** The UI should be responsive and fast to ensure smooth user interactions.
  + **Efficiency:** Minimal resource usage to prevent lag or crashes.
* **Branding:**
  + **Alignment with Brand Identity:** Consistency with the company’s branding guidelines (e.g., logo usage, color schemes).
  + **Tone and Voice:** The interface should reflect the brand’s personality and communicate appropriately with users.
* **Scalability:** The UI design should accommodate future updates and additional features without major redesigns.
* **Error Handling:** Clear and helpful error messages and options for users to recover from errors.

Production Processes:

* **Research and Planning:**
  + **User Research:** Understanding user needs, preferences, and pain points through surveys, interviews, and usability testing.
  + **Competitive Analysis:** Studying competitors’ interfaces to identify best practices and opportunities for differentiation.
  + **Requirement Gathering:** Defining functional and non-functional requirements for the UI.
* **Design:**
  + **Wireframing:** Creating low-fidelity wireframes to outline the basic structure and layout of the UI.
  + **Prototyping:** Developing interactive prototypes to test and validate design concepts.
  + **Visual Design:** Designing high-fidelity mockups with detailed visual elements.
* **Development:**
  + **Front-End Development:** Translating the design into code using HTML, CSS, and JavaScript (or frameworks/libraries like React, Angular).
  + **Integration:** Integrating the UI with back-end systems and APIs to ensure functionality.
* **Testing:**
  + **Usability Testing:** Conducting tests with real users to identify usability issues and gather feedback.
  + **Cross-Browser Testing:** Ensuring compatibility across different web browsers.
  + **Performance Testing:** Checking for speed and efficiency under various conditions.
* **Deployment:**
  + **Staging Environment:** Deploying the UI to a staging environment for final testing and review.
  + **Production Release:** Launching the UI to the live environment for end users.
* **Maintenance and Iteration:**
  + **Monitoring:** Tracking user feedback and performance metrics to identify areas for improvement.
  + **Updates:** Implementing updates and enhancements based on user feedback and evolving requirements.
* **Documentation:**
  + **Design Documentation:** Documenting design decisions, patterns, and guidelines for future reference.
  + **User Guides:** Creating guides or help resources for users to understand and utilize the UI effectively.

### Content Questions (group two)

Provide a suitable answer for the following questions.

What are programming controls and design structures?

* Programming Controls:
* **Conditional Statements:** Control the flow of execution based on conditions. Examples include if, else, switch, and case statements, which allow different code paths to be executed based on boolean conditions or expressions.
* **Loops:** Enable repetitive execution of code. Common loops include for, while, and do-while, which execute a block of code multiple times based on specific conditions.
* **Functions/Methods:** Encapsulate reusable code into named blocks. Functions take input (parameters), perform operations, and return a result. They help modularize and organize code.
* **Exceptions:** Handle errors and unexpected situations using try, catch, and finally blocks. Exception handling ensures that a program can gracefully handle runtime errors and maintain stability.
* Design Structures:
* **Modularity:** Breaks down a program into smaller, self-contained modules or components. Each module addresses a specific aspect of functionality, making the code easier to manage and maintain.
* **Object-Oriented Design (OOD):** Uses classes and objects to model real-world entities. Key principles include:
  + **Encapsulation:** Bundling data and methods into a single unit (class).
  + **Inheritance:** Reusing existing classes to create new classes.
  + **Polymorphism:** Using a unified interface to interact with different classes.
* **Design Patterns:** Proven solutions to common design problems. Examples include:
  + **Singleton:** Ensures a class has only one instance.
  + **Factory:** Creates objects without specifying the exact class.
  + **Observer:** Notifies multiple objects about state changes.
* **Data Structures:** Organize and store data efficiently. Common structures include:
  + **Arrays:** Fixed-size collections of elements.
  + **Linked Lists:** Dynamic collections of elements where each element points to the next.
  + **Stacks:** LIFO (Last In, First Out) data structure.
  + **Queues:** FIFO (First In, First Out) data structure.

What are website testing procedures?

* Website Testing Procedures:
* **Functional Testing:** Ensures that all features of the website work as intended. This includes verifying that all buttons, links, forms, and other interactive elements function correctly.
* **Usability Testing:** Evaluates the website's ease of use and overall user experience. This involves testing navigation, readability, and user interactions to ensure they are intuitive and user-friendly.
* **Compatibility Testing:** Checks that the website performs well across different browsers, devices, and operating systems. This includes testing on various versions of browsers like Chrome, Firefox, Safari, and Edge, as well as different screen sizes and resolutions.
* **Performance Testing:** Assesses the website's speed and responsiveness under various conditions. This includes testing load times, responsiveness, and how the site performs under heavy traffic or stress.
* **Security Testing:** Identifies potential vulnerabilities and ensures that the website is protected against security threats. This involves testing for issues like data breaches, SQL injection, cross-site scripting (XSS), and other security risks.
* **Accessibility Testing:** Ensures that the website is accessible to users with disabilities. This includes checking compliance with accessibility standards like WCAG (Web Content Accessibility Guidelines) and ensuring compatibility with screen readers and other assistive technologies.
* **Regression Testing:** Verifies that new updates or changes to the website do not negatively impact existing functionality. This involves retesting previously tested features to ensure they still work correctly after updates.
* **Integration Testing:** Tests how well different components of the website work together. This includes verifying that different modules or third-party integrations function correctly when combined.

What are website debugging methods?

* Website Debugging Methods:
* **Browser Developer Tools:** Most modern browsers provide built-in developer tools (e.g., Chrome DevTools, Firefox Developer Tools) that allow you to inspect and modify HTML/CSS, debug JavaScript, and analyze network requests.
* **Console Logging:** Use console.log() statements in JavaScript to output values and trace code execution. This helps to understand the flow of the application and identify issues.
* **Breakpoints:** Set breakpoints in your code to pause execution at specific points, allowing you to inspect the state of variables and the call stack to understand where and why issues occur.
* **Error Messages:** Pay attention to error messages and stack traces provided by the browser or server. These can give clues about the location and nature of the problem.
* **Code Validators:** Use code validation tools (e.g., HTML validators, CSS linters) to check for syntax errors and adherence to coding standards, which can help identify and fix issues early.
* **Network Analysis:** Use network analysis tools to monitor and debug HTTP requests and responses. This helps diagnose issues related to data fetching, API calls, and server responses.
* **Performance Profiling:** Analyze performance using profiling tools to identify bottlenecks and optimize the performance of your website.
* **Cross-Browser Testing:** Test your website across different browsers and devices to ensure consistent behavior and identify browser-specific issues.

What are website coding techniques?

* Website coding techniques involve various methods and best practices to create efficient, maintainable, and scalable code. Key techniques include:
* **Semantic HTML:** Use HTML elements according to their intended purpose (e.g., <header>, <footer>, <article>) to improve readability, accessibility, and SEO.
* **CSS Preprocessing:** Use CSS preprocessors like Sass or LESS to write more maintainable and modular CSS. These tools support variables, nested rules, and mixins.
* **Responsive Design:** Apply responsive design techniques to ensure the website adjusts gracefully to different screen sizes and devices. Techniques include flexible grids, fluid images, and media queries.
* **JavaScript Frameworks:** Use frameworks like React, Angular, or Vue.js to build dynamic and interactive web applications. These frameworks provide structures and tools to streamline development.
* **Code Minification:** Minify HTML, CSS, and JavaScript files to reduce file sizes and improve load times. This involves removing unnecessary characters, whitespace, and comments from the code.
* **Version Control:** Use version control systems like Git to track changes, collaborate with others, and manage different versions of the codebase.
* **Progressive Enhancement:** Build a basic, functional version of the website that works for all users, and then enhance the experience for users with more advanced browsers or devices.
* **Accessibility Best Practices:** Implement accessibility features such as ARIA (Accessible Rich Internet Applications) roles and properties, keyboard navigation, and screen reader compatibility to make the website usable for people with disabilities.

### Content Definitions (group three)

Provide a definition for the following programming concepts.

Authentication and web security.

* **Authentication:**is the process of verifying the identity of a user or system. It ensures that users are who they claim to be before granting access to resources or services. This often involves credentials like usernames and passwords, biometric data, or authentication tokens.
* **Web Security:**refers to the measures and protocols put in place to protect web applications and services from cyber threats and unauthorized access. This includes protecting data in transit (using HTTPS), implementing secure coding practices, and safeguarding against common vulnerabilities (such as SQL injection or cross-site scripting).

Hypertext transfer protocol (HTTP).

* **Hypertext Transfer Protocol (HTTP) :** is a protocol used for transferring hypertext requests and information on the web. It defines how messages are formatted and transmitted between a web server and a web client (usually a browser). HTTP operates over TCP/IP and is the foundation of data communication on the World Wide Web.

Session management.

* **Session Management** involves handling the process of maintaining state and user information across multiple interactions or requests in a web application. It typically includes creating, tracking, and terminating sessions. Sessions are used to keep users logged in, manage their preferences, and store temporary data across multiple pages. Techniques for session management include cookies, session tokens, and server-side storage.

Stateless programming.

* **Stateless Programming:** refers to a design approach where each request from a client to a server is treated as an independent transaction, with no knowledge of previous requests. In a stateless system, the server does not retain any state or context between requests. Each request must contain all the information necessary to process it, as the server does not keep any memory of past interactions. This approach simplifies scaling and can improve reliability, but often requires additional mechanisms for managing state, such as using tokens or cookies to preserve session information.

### Content Definitions (group four)

Provide a definition for the following types of technologies.

The three major programming control structures.

* The three major programming control structures are fundamental constructs used to dictate the flow of a program's execution:
* **Sequential:** This is the default control structure where statements are executed one after another in the order they appear. Each statement is executed exactly once.
* **Selection:** This control structure allows the program to make decisions and execute different code blocks based on certain conditions. It includes constructs like if, else if, and else statements, as well as switch statements.
* **Iteration:** Also known as looping, this structure enables the repeated execution of a block of code as long as a specified condition holds true. Common iteration constructs include for, while, and do-while loops.

Hypertext markup language (HTML) and markup languages.

* **Hypertext Markup Language (HTML):** is the standard language used to create and design web pages and web applications. HTML uses a system of tags and attributes to structure content on the web, defining elements like headings, paragraphs, links, images, and more.
* **Markup Languages:** are a family of languages used to annotate text and other content in a way that is syntactically distinguishable from the text itself. These annotations provide information about the structure, presentation, and semantics of the content. HTML is a type of markup language, as are XML, Markdown, and LaTeX.

Cascading style sheets (CSS).

* **Cascading Style Sheets (CSS):** is a stylesheet language used to describe the presentation and layout of a document written in HTML or XML. CSS allows developers to control the look and feel of web pages, including elements like fonts, colors, margins, spacing, and positioning. The "cascading" aspect refers to the order of precedence when multiple styles are applied, allowing for hierarchical overrides and specificity

Syntax and the uses of programming languages.

* **Syntax:** Refers to the set of rules and conventions that define the structure of statements in a programming language. It dictates how code must be written and organized to be correctly interpreted by the compiler or interpreter. Syntax includes rules for things like keywords, operators, delimiters, and the organization of code blocks.
* **Uses of Programming Languages:** Refers to the various applications and domains where different programming languages are applied. Different languages are designed with specific purposes in mind. For example:
  + **JavaScript** is commonly used for web development to create interactive and dynamic web pages.
  + **Python** is popular for data analysis, machine learning, and scripting.
  + **Java** is widely used for enterprise applications and Android development.
  + **C** is often used for system programming and applications requiring high performance

## Question 3 Version Control

CITE would like you to use GitHub as the primary source control, setup an appropriate structure in your GitHub account to manage the Assessment One website development. Add a project to your repository which reflects the basic Agile development process you intend to pursue. Complete the following GitHub Version Control template to answer this question.

|  |  |  |  |
| --- | --- | --- | --- |
| GitHub Version Control | | | |
| Repository Name: | ICTWEB513\_AT1 | | |
| URL | https://github.com/jsambajon92/ICTWEB513\_AT1 | Date | 21/08/2024 |
| Screen Shot(s) | Version 2    Version 3 | | |

## Question 4 Design Approval

Once you have complete questions 1,2 & 3 arrange for your document to be reviewed by the Lecturer/Assessor for approval, sign off and feedback before completing the development and testing.

* Question 1 Design Specification
* Question 2 Web Page Content
* Question 3 Version Control

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Design Approval (Lecturer/Assessor use only) | | | | |
| Approver Name | Title | Signature | Date | Approved? |
| Rhys Gillam |  |  | 4/09/2024 | Not approved. |
| Rhys Gillam |  |  | 25/09/2024 | Not approved. |
| Rhys Gillam |  |  | 2/10/2024 | Approved. |
| Lecturer Feedback | | | | |
| 4/9/2024  - Contents to be broken down into subgroup 1 to 4.  25/09/2024   * Navigation diagram is incorrect * Missing answers in the documentation * Should be able to navigate to any of the pages from the navbar (its the point of a navbar) from the home page * CSS present but not used (no 'hero', among others). * Group 1.1; should be in the context of web design * Group 1.3; missing other required answers * Group 2.4; review coding techniques * Group 4; fix naming of page   3/10/2024   * Navigation Diagram: Corrected * Missing Answers in document: Completed * Navbar functionality: Changed as discussed * CSS Utilization: Improved * Group 1.1 context: Adjusted with web design context * Group 1.3 Missing Answers: Missing answers provided. * Group 2.4: Coding techniques reviewed * Group 4: Page naming fixed. | | | | |

## Question 5 Website Development

Develop the software components to create a website based on your prototype and design specifications. Add the content from Question 2 and enhance the fonts and background colours to satisfy contemporary web page standards. Upload your code to the Turnkey Server. Update your GitHub with the completed website code and associated files. Your code must adhere to the CITEMS software development standards. (refer http://www.citems.com.au/)

## Question 6 Testing

Ensure your code is error free and functions correctly, then test the website on several different browsers. During these tests check the web pages scale correctly and conforms to responsive web design. Secondly, test the website on several different digital devices and record any errors. Your Test Report must include appropriate evidence that your code functions as expected (references to screen captures). Complete the following Test Report template below to answer this question.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Report | | | | | |
| Developer Name | | Joshua Sambajon | **Date** | 12/09/2024 | |
| Browser compatibility test | | | | | |
| Browser name | Description | | | | Pass / Fail |
| Google Chrome | Tested on the latest version of **Chrome *Version 128.0.6613.121 (Official Build) (64-bit)***Layout, responsiveness, and JavaScript functions work perfectly.    **Note:** The screenshot displays index.html, while contents.html demonstrates the responsiveness of the navigation bar. | | | | PASS |

|  |  |  |
| --- | --- | --- |
| Mozilla Firefox | Tested on the latest version of **Mozilla Firefox Version 130.0 (64-bit)**. Layout, responsiveness, and JavaScript functions work perfectly. | PASS |
| Microsoft Edge | Tested on the latest version of **Microsoft Edge Version 128.0.2739.67 (Official build) (64-bit)** Layout, responsiveness, and JavaScript functions work perfectly. | Pass |

|  |  |  |
| --- | --- | --- |
| Device compatibility test | | |
| Device Name | Description | Pass / Fail |
| Iphone SE | Test performed on iPhone SE with dimensions 375 x 667 at 100% zoom. | Pass |
| Iphone XR | Test performed on iPhone XR with dimensions 414 x 896 at 100% zoom. | Pass |
| Iphone 14 Pro Max | Test performed on iPhone 14 Pro Max with dimensions 430 x 932 at 100% zoom. | Pass |

|  |  |  |
| --- | --- | --- |
| Samsung S20 Ultra | Test performed on Samsung S20 Ultra with dimensions 419 x 915 at 100% zoom. | Pass |
| iPad Mini | Test performed on iPad Mini with dimension 768 x 1024 at 100% zoom. | Pass |
| iPad Pro | Test performed on iPad Pro with dimension 1024 x 1366 at 100% zoom. | Pass |

## Question 7 Demonstration, Feedback and Signoff

Ensure your code is fully commented with your Name, ID, and Date placed above the main code body of each file. Check all the above documentation has been completed and is ready for inspection. Contact your Lecturer (Assessor) and arrange to demonstrate your working website on your Turnkey Web Server, use the following Checklist to ensure you have completed all the assessment criteria.

|  |  |  |  |
| --- | --- | --- | --- |
| Checklist | | Completed | |
| **Questions** | | YES NO | |
| Q1 | Design Specifications: All fields of the Design Specification are filled in. |  |  |
| Q2 | Web Page Content: All the questions and definitions have been grouped and formatted onto separate pages. |  |  |
| Q3 | Version Control: All fields of the template are filled in. |  |  |
| Q4 | Website Development. |  |  |
| Q5 | Testing: All the fields in the Testing Report have been filled in. |  |  |
| Q6 | Demonstration: The website functions as required, and all web components work correctly. |  |  |

**Note:** All documentation must use the supplied templates/forms.

**Submit the zipped solution folder with relevant documents to Blackboard**

End of Assessment Task One