# Student Version

| Section A – Course details | | | |
| --- | --- | --- | --- |
| **Qualification code:** | ICT50220 | **Qualification title:** | Diploma of Information Technology  (Front End Web Development) with Diploma of Information Technology  (Back End Web Development) |
| **Subject code:**  **Unit code:** | (D7)  ICTWEB513  ICTWEB514  ICTWEB519 | **Subject title:**  **Unit title:** | (Full stack)  Build dynamic websites  Create dynamic web pages  Develop complex web page layouts |
| **Department name:** | BDIT, Computing & Information Technology | **CRN number:** | Enter CRN number |

| Section B – Assessment task details | | | |
| --- | --- | --- | --- |
| **Assessment number:** | 2 of 2 | **Semester/Year:** | 2/2024 |
| **Due date:** | Session 16 | **Duration of assessment:** | 6 Weeks |
| **Assessment method** | Project/Report/Portfolio | **Assessment task results** | Ungraded result   (Satisfactory or Unsatisfactory) |
| Other: |

| Section C – Instructions to students |
| --- |
| **Task instructions:** |

1. This assessment task requires learners to build a server-side API that saves and stores data in a database, along with a client-side application for the server-side API. The teacher will be playing the role of the client for this assessment task. The server-side API the learner developers may be about or for any topic, company or community of the learner’s choice. As long as the website meets the requirements outlined in this project.    
     
   This assessment has been divided into 6 key parts:  
     
   Part 1 – Planning  
   Part 2 – Prototype  
   Part 3 – Development  
   Part 4 – Testing & Debugging  
   Part 5 – Handover and Sign Off

* This task is to be completed individually.
* Using this assessment document, please read all questions in this document and record your answers in the document against each question.
* To be satisfactory in this assessment task all answered questions must correctly meet the marking guide criteria.
* Discuss with your assessor if you feel you require special consideration or adjustment for this task.
* Learners can consult class learning material via Brightspace and other softcopy information, including information from the Internet. However, all answers must be in a learner’s own words. Where a quote is used the learner must cite the information source.
* You must submit all required working files, documentation, and any other assets that you feel may be required in a zipped file, including the completed and signed coversheet. The assessment must be completed and submitted electronically to Brightspace by the due date. If this is not possible, you must contact your assessor to gain written approval for an alternative arrangement for submitting the assessment.
* It is expected all documents will be completed and submitted electronically but if this is not possible, make alternative arrangements for submitting the documents with your assessor.
* Leaner must contribute to and abide by organisational standards including intellectual property, privacy laws, and plagiarism and academic honesty. Further information is detailed at: https://holmesglen.edu.au/Students/Student-Resources/

| Section D – Conditions for assessment | |
| --- | --- |
| **Conditions:**  Student to complete and attach Assessment Submission Cover Sheet to the completed Assessment Task.   * This is an individual task. However, you are required to get information, feedback and ideas from your assessor, peers or industry to help complete the assessment planning guide. * You must meet all criteria listed in the marking guide to be satisfactory in this task. * You must submit all required working files, documentation, and any other assets that you feel may be required in a zipped file, including the completed and signed coversheet. The assessment must be completed and submitted electronically to Brightspace by the due date. If this is not possible, you must contact your assessor to gain written approval for an alternative arrangement for submitting the assessment. * If not successful within the enrolment period as per Holmesglen assessment procedure, you will be requested to resubmit within 7 days of receiving feedback. You will have the opportunity to resubmit if any part of the assessment is deemed unsatisfactory (you are permitted TWO (2) resubmission per assessment task). Resubmissions must be submitted by the resubmission due date provided by your teacher. * This task is open book. You may use the internet for research purposes only. All answers must be in your own words. Where a quote is used, you must cite the information source. * If you feel you require special allowance or adjustment to this task, please discuss with your assessor within one week of commencing this assessment. Any change to assessment arrangements must be reviewed by the Education Manager and approved by the Head of Department. * You can appeal an assessment decision according to the Holmesglen Assessment Complaints and Appeals Procedure. * You are expected to dedicate time to develop this assessment task both in and out of the classroom. * Leaner must contribute to and abide by organisational standards including intellectual property, privacy laws, and plagiarism and academic honesty. Further information is detailed at: https://holmesglen.edu.au/Students/Student-Resources/ * Development tools should include but are not limited to: Visual Studio Code, Chrome or Fire Fox   (You have access to these tools in labs saved in Brightspace or they can be downloaded). | |
| **Equipment/resources students must supply:** | **Equipment/resources to be provided by the RTO:** |
| * A MAC or * PC/laptop with the following minimum specification:   + Quad-Core CPU, 16GB of RAM, 250GB of Storage, 2 GHz or faster processor   + Windows 10 OS, or higher * Headset with microphone and webcam (if learning remotely) * Internet access   Access to applications used in learning is available through Holmesglen MyHorizon or can be downloaded via the link provided: • Brightspace (Learning Management System) - https://holmesglen.brightspace.com/ • 365 Microsoft office suite - https://portal.office.com • WebEx - https://holmesglen.webex.com/ • Holmesglen OneDrive • Google Chrome – recommended web browser https://www.google.com/intl/en\_au/chrome/ • 7-Zip - https://www.7-zip.org/download.html • GitHub Desktop- https://desktop.github.com • Visual Studio Code - https://code.visualstudio.com/ • Figma - https://www.figma.com/ • Node.js- https://nodejs.org/en • Database server • Postman - https://www.postman.com/ • Compass • MySQL Client | A Mac or PC/laptop with the following minimum specification:  Quad Core CPU  • 8GB of RAM  • CPU with minimum 2ghz processor or faster • 200GB of Storage • Headset with microphone (webcam optional but preferred)  • Access to internet connection (ADSL or cable connection desirable)  Applications: • Microsoft Word - access through Holmesglen MyHorizon • WebEx - free to download • Visual Studio Code – free to download • GitHub • Figma • Visual Studio Code: https://code.visualstudio.com/ • Node.js • Database server • Postman • Compass • MySQL Client • Libraries and frameworks required for building dynamic websites • Website testing and debugging tools • Onedrive or google drive/dropbox account for storage • 7Zip or an equivalent compression utility - free to download • Google Chrome – recommended web browser (and additional Browsers including Firefox & Edge) • Access to internet through the campus network |

**Student answer sheet / Marking sheet**

|  |  |  |  |
| --- | --- | --- | --- |
| Section E – Marking Sheet - Student Answer Sheet | | | |
| **Subject Code:**  **Unit code**: | (D7)  ICTWEB513  ICTWEB514  ICTWEB519 | **Subject Title:**  **Unit title:** | (Full stack)  Build dynamic websites  Create dynamic web pages  Develop complex web page layouts |

| **Criteria for assessment** | | **Satisfactory** | | **Comment** |
| --- | --- | --- | --- | --- |
| **Yes** | **No** |
| **Marking criteria:** Part 1 - Planning Assessment Documentation | | | | |
| 1. | The learner has Identified the all the business requirements and applicable legislative standards highlighted in the brief. |  |  |  |
| 2. | The organizational standards and procedures that need to be been confirmed |  |  |  |
| 3. | The purpose, expectations and functionality of the application have been determined and documented |  |  |  |
| 4. | All Client-side dynamic components/features has been identified |  |  |  |
| 5. | All Server-side dynamic components/features has been identified |  |  |  |
| 6. | All appropriate languages and core technology that meet the technical requirements have been identified and new digital technologies and applications have been investigated |  |  |  |
| 7. | All core features of the user interface design, target platform, user needs and design principles to be used have been documented. |  |  |  |
| 8. | All of the websites architectural requirements have been identified |  |  |  |
| 9. | The learner has discussed and negotiates key aspects with required personal, including required capabilities, efficiencies and effectiveness |  |  | **observation** |
| **Marking criteria:**  Part 2 - Prototype Assessment Documentation/Working files | | | | |
| 1. | A wireframe that demonstrates the main sections of the design & layout have been developed and it aligns with the requirements outlined in the brief. |  |  |  |
| 2. | A hierarchy/site map of website has been produced and it aligns with the requirements outlined in the brief. |  |  |  |
| 3. | A prototype of user interface has been produced |  |  |  |
| 4. | The content is logical and accessible for the end user according to website requirements |  |  |  |
| **Marking criteria:**  Part 3 - Development  Assessment Documentation/Working files | | | | |
|  | A boilerplate templates have been created according to technical requirements |  |  |  |
|  | All required React components for the client have been developed according to the website requirements |  |  |  |
| 3. | All React components have been incorporated into the website to produce the web application |  |  |  |
| 4. | All web page structure has been developed according to design specifications |  |  |  |
| 5. | Components and Layout have been position and styled for 3 web pages according to design specifications outlined in Part 1 |  |  |  |
| 6. | Required Cascading style sheet and application pages have been developed according to design specifications |  |  |  |
| 7. | The web application has been developed using the selected languages from Part 1 |  |  |  |
| 8. | A database has been integrated into the application and the data has been organized according to data storage requirements outlined in the project brief. |  |  |  |
| 9. | Client and server-side dynamic content has been incorporated into each web page |  |  |  |
| 10. | The client and server-side code is meets the requirements and design requirements outlined in the project brief |  |  |  |
| 11. | Web security features have been implemented |  |  |  |
| 12. | http and appropriate http status codes have been used to communicate between the client and server |  |  |  |
| 13. | Session management has been implemented on the client and stateless programming has been used to develop the API. |  |  |  |
| 14. | Programming control structures have been used |  |  |  |
| 15. | The application demonstrates correct syntax and uses of programming languages |  |  |  |
| **Marking criteria:**  Part 4 - Testing  Assessment Documentation/Working files | | | | |
| 1. | The website functionality and structure has been tested against the requirements and the website meets all the requirements outlined in the testing document |  |  |  |
| 2. | The learner has confirmed the website is secure and bug free according to cyber security procedures and protocols |  |  |  |
| 3. | The functionality of website has been tested in 2 browsers and devices and update as required according to organization procedures |  |  |  |
| 4. | Accessibility has been checked and confirmed for web sections, elements and pages |  |  |  |
| 5. | React components of the website have been tested according to organisational procedures |  |  |  |
| 6. | Required dynamic content functions according to task requirements |  |  |  |
| 7. | The HTML and CSS have been validated against industry standards |  |  |  |
| **Marking criteria:**  Part 5 – Hand Over and Sign Off Assessment Documentation/Working files | | | | |
| 1. | The web application has been presented to required personnel and feedback has been sought |  |  |  |
| 2. | The web application has been updated as required based on feedback |  |  |  |
| 3. | The web application has been finalized, meets all requirements, all policies procedures and standards have been adhered to and sign-off has been obtained from required personnel |  |  |  |
| 4. | The learner has Accepts responsibility for planning and sequencing complex tasks and workload |  |  |  |
| 5. | The learner has used and investigates new digital technologies and applications during the development of this project |  |  |  |
| 6. | The learner has used platforms (Eg. GITHUB & Digital Communication tools) to manage and manipulates data, and communicates with others in a secure and stable digital environment |  |  |  |
| 7. | The learner has documented the design structure according to organisational procedures |  |  |  |
| 8. | At least 2 scripts have been documented according to the requirements outlines in the project brief |  |  |  |

# Assessment Submission Cover Sheet (VET)

|  |  |  |
| --- | --- | --- |
| **Student declaration** | |  |
| By submitting this assessment task and signing the below, I acknowledge and agree that:   1. This completed assessment task is my own work. 2. I understand the serious nature of plagiarism and I am aware of the penalties that exist for breaching this. 3. I have kept a copy of this assessment task. 4. The assessor may provide a copy of this assessment task to another member of the Institute for validation and/or benchmarking purposes. | | |
| **Student ID:** | **100683785** | |
| **Student name:** | **Vorakorn Taweetawon** | |
| **Submission or observation date:** |  | |
| **Student signature**  For electronic submissions: By typing your name in the student signature field, you are accepting the above declaration. | **Vorakorn Taweetawon** | |

| Section F – Feedback to Student | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **Has the student successfully completed this assessment task?** | | | | | **Yes** | **No** |
| **☐** | **☐** |
| **Additional Assessor comments (as appropriate):** | | | | | | |
|  | | | | | | |
| **Resubmission allowed:** | **Yes ☐** | **No ☐** | **Resubmission due date:** |  | | |
| **Assessor name:** |  | | | | | |
| **Assessor signature:** |  | | | | | |
| **Date assessed:** |  | | | | | |

**Supporting document**

# Portfolio Instructions

|  |  |  |  |
| --- | --- | --- | --- |
| **Subject Code:**  **Unit code**: | (D7)  ICTWEB513  ICTWEB514  ICTWEB519 | **Subject Title:**  **Unit title:** | (Full stack)  Build dynamic websites  Create dynamic web pages  Develop complex web page layouts |

|  |
| --- |
| **Model answers for the above questions and assessment criteria** Please refer to the project exemplar for working file examples |
| **Project Brief** Read through the brief below take note of the needs and requirements outlined in the brief. |
| **Introduction**   1. This assessment task requires learners to build a server-side API that saves and stores data in a database, along with a client-side application for the server-side API. The teacher will be playing the role of the client for this assessment task. The server-side API the learner developers may be about or for any topic, company or community of the learner’s choice. As long as the website meets the requirements outlined in this project.    This assessment has been divided into 6 key parts:  Part 1 – Planning Part 2 – Prototype Part 3 – Development Part 4 – Testing & Debugging Part 5 – Handover and Sign Off  * This task is to be completed individually. * Using this assessment document, please read all questions in this document and record your answers in the document against each question. * To be satisfactory in this assessment task all answered questions must correctly meet the marking guide criteria. * Discuss with your assessor if you feel you require special consideration or adjustment for this task. * Learners can consult class learning material via Brightspace and other softcopy information, including information from the Internet. However, all answers must be in a learner’s own words. Where a quote is used the learner must cite the information source. * You must submit all required working files, documentation, and any other assets that you feel may be required in a zipped file, including the completed and signed coversheet. The assessment must be completed and submitted electronically to Brightspace by the due date. If this is not possible, you must contact your assessor to gain written approval for an alternative arrangement for submitting the assessment. * It is expected all documents will be completed and submitted electronically but if this is not possible, make alternative arrangements for submitting the documents with your assessor. * Leaner must contribute to and abide by organisational standards including intellectual property, privacy laws, and plagiarism and academic honesty. Further information is detailed at: https://holmesglen.edu.au/Students/Student-Resources/     **Website User Requirements Client**   * Home Page * Collections of items page A page that displays a collection of items (things/topics/people/etc.) A title, image and URL should be displayed as a minimum for each item in the collection * Details Page/Item Page A page that allows the user to view an individual item (thing/topic/person/etc.) from the collection. * Pages that allow the user to perform CRUD operations * Use of a framework such as React.js or Svelte * Routes must be created for all pages * Error pages/alerts such as 404 errors must be created and implemented * User Input Form * Any sensitive data must be encrypted in the database * Data may be store temporarily on the client in state or local storage but any data that is required for persistence must be save to the database. * Ensure that the UI and UX for the application is logical and accessible * Display data from the API in appropriate controls, such as text boxes, drop downs, radio buttons etc. * Format data received from the API so that it can be easily understood and digested by the user. Ensure this formatting aligns with your overall UI design. * Ensure your client-side app keeps track of data between browser requests. * e.g. Your application keeps the current state of the application between requests. * Save user interaction with the website. * The website must be responsive * Data must be validated   **Website User Requirements Server**   * MongoDB or MySQL must be used as the Database Technology * Data must be organised into at least 2 tables/collections * All database rows/documents must contain an ID * The API must return JSON data * The API data must be saved and retrieved from a database * All 4 core CRUD operation or Queries and Mutations must be used in your application * All data must be validated using a module such as JOI * A schema should be created that validates the data before sending it to the database * A GUI for the APIs endpoints must be created using a client-side Library * The API must be stateless * This data must be validated and sanitized * Highly sensitive data must be encrypted   **General Requirements**   * You will also need to investigate and document at least 2 new web technologies. One of these technologies will need to be integrated into the final product.   **Organisation Coding Standards and Procedures**   * A config file must be used to store common data that is used across the site that is subject to change. E.g. The URL of a database or data file * A GIT repository should be used Any major changes should be commented and committed to the GIT Any experimental features should be created on a separate branch * All code should be commented clearly Classes and Scripts A descriptive overview should be provided for each class and script as a comment at the top of the file. Details about any parent classes should be documented at the top of the file. Members/Variables The purpose of each member should be documented as a comment. Methods/Functions The purpose of each method should be documented as a comment. Parameters The purpose of each parameter should be documented as a comment.   **Organisation Design Principles**   * Clarity Clarity is the first and most important job of any interface. To be effective using an interface you've designed, people must be able to recognize what it is, care about why they would use it, understand what the interface is helping them interact with, predict what will happen when they use it, and then successfully interact with it. While there is room for mystery and delayed gratification in interfaces, there is no room for confusion. Clarity inspires confidence and leads to further use. One hundred clear screens is preferable to a single cluttered one. * One primary action per screen Every screen we design should support a single action of real value to the person using it. This makes it easier to learn, easier to use, and easier to add to or build on when necessary. Screens that support two or more primary actions become confusing quickly. Like a written article should have a single, strong thesis, every screen we design should support a single, strong action that is its raison d'etre. * Provide a natural next step Very few interactions are meant to be the last, so thoughtfully design a next step for each interaction a person has with your interface. Anticipate what the next interaction should be and design to support it. Just as we like in human conversation, provide an opening for further interaction. Don't leave a person hanging because they've done what you want them to do…give them a natural next step that helps them further achieve their goals. * Consistency matters Following on the previous principle, screen elements should not appear consistent with each other unless they behave consistently with each other. Elements that behave the same should look the same. But it is just as important for unlike elements to appear unlike (be inconsistent) as it is for like elements to appear consistent. In an effort to be consistent novice designers often obscure important differences by using the same visual treatment (often to re-use code) when different visual treatment is appropriate. * Immediate Feedback   In ideal interfaces, help is not necessary because the interface is learnable and usable. The step below this, reality, is one in which help is inline and contextual feedback, available only when and where it is needed, hidden from view at all other times.  **Technical Requirements**   * HTML5 & CSS3 * Bootstrap or alternative * JavaScript should be used for client-side development * A local server is required for testing * Code editor such as Visual Studio Code * A responsive framework such as Bootstrap * React.js, Svelte or Vue.js * JSX if React.js * The application UI must be divided into logical reusable components * Server Node.js * Database MongoDB or SQL     **legislative Standards** Your project will need to adhere to the following legislation standards.   * Australia Privacy Principles (NPP) <https://www.oaic.gov.au/privacy/australian-privacy-principles/> * Copyright - Website <https://www.copyright.org.au/ACC_Prod/ACC/Information_Sheets/Websites___Copyright.aspx?WebsiteKey=8a471e74-3f78-4994-9023-316f0ecef4ef> * Accessibility - Level A of the Web content accessibility guidelines   <https://www.w3.org/TR/WCAG20/>  **Cyber Security Standards and Policies**   * All client user input must be validated before being submitted to the API * All data must be validated before being saved to the database * Schemes must be created with datatypes and restrictions * BCrypt must be used to encrypt highly sensitive data such as passwords (if required) * All passwords must be Hashed and Slated (if required) * JWT should be used to generate tokens (if required) * Tokens must be saved to state for future use once a user has logged in (if required) * Sensitive data must not be sent via params or query strings * Application Secrets must be stored in environmental variables * It is recommended that a module such as helmet be used to further protect the application and database * CORs Policies must be implemented * Sensitive data like secrets must not be saved to GITHUB * GITHUB transfers must use secure protocols * Deployment to cloud must be done using secure protocols * Detailed error messages and stack traces must not be returned as responses from the API |

|  |  |
| --- | --- |
| **Part 1 - Identify and Confirm Design Requirements** Based on the information you have gathered from the brief complete the following:Assessment Documentation | |
| 1. Identify the business requirements, standards, and applicable legislative standards the project must adhere to. | **GraphQL API Requirements:**   * A GraphQL API that returns JSON at various endpoints. * The API data must be saved and retrieved from a MongoDB database. * All core CRUD operations must be implemented via GraphQL queries and mutations. * Data must be validated using a module such as JOI. * A schema will be created to validate the data before sending it to the database. * Authentication and Authorization will be implemented using JWT. * Sensitive data, like passwords, will be encrypted using bcrypt. * Error responses such as 404 and 500 must be handled gracefully and implemented**.**   **Client-Side Requirements:**   * A GUI for the GraphQL API’s endpoints will be created using React.js. * A responsive framework such as Bootstrap will be used. * HTML5 & CSS3 will be utilized to ensure a responsive and accessible UI. * UI and UX design must be logical, accessible, and consistent with the application’s purpose. * Data from the API will be displayed using appropriate controls (e.g., text boxes, drop-downs, radio buttons). * Data received from the API must be formatted to be easily understood by the user, aligned with the overall UI design. * The client-side app will maintain the application state between browser requests, such as saving the token supplied by the API for future use.   **Legislative standards**   * Australia Privacy Principles (NPP) * Copyright - Website * Accessibility - Level A of the Web content accessibility guidelines |
| 1. List the organizational standards and procedures that the project needs to adhere to. | **Security Standards**   * All data must be validated using JOI to be validated before being saved to the MongoDB database to ensure accuracy and integrity. * Authentication (via JWT) and Authorization must be implemented to ensure that only authorized users can create, update, or delete data. * Application secrets, such as database connection strings and API keys, must be stored in environmental variables and not hard-coded in the application. * Use a security module like Helmet to set various HTTP headers for protection of the application and database.   **Coding Standards and Maintainability**   * Classes and Modules: Provide a descriptive overview at the top of each class and module, including details about any parent classes. * Members: Document the purpose of each member (variable) with comments. * Methods: Document the purpose and functionality of each method, including details on what each method does and its return values. * Parameters: Explain the purpose of each parameter in method comments to ensure clarity on how methods are used.   **Other Standards**   * HTML Standards: Adhere to HTML5 standards. * CSS Standards: Follow CSS3 standards for styling. * HTML Accessibility: Ensure that HTML accessibility standards are met. |
| 1. Determine and document the purpose, expectations and functionality of the application | ***Purpose***  *The purpose of the Secondhand Fashion Finder project is to develop a platform that includes both a GraphQL API and a client-side application. The API will manage data securely and handle authentication and authorization to protect endpoints. The client-side application will provide users with an intuitive interface to interact with the API, browse, search, and manage secondhand fashion items.*  ***Expectations***   * *The API must be built using GraphQL,It should be secure, handle errors gracefully, and ensure data integrity.* * *Client-Side Application: The client-side application should adhere to industry standards, including: Mobile-First Design, HTML and CSS Standards, and Accessibility Standards*   ***Functionality***   * *Authentication and Authorization* * *Store data in a MongoDB database* * *Implement CRUD operations* * *Develop a GraphQL API to handle server-side operations, including user management and item listings.* * *Create a user-friendly that allows users to interact with the API, including searching, filtering, and viewing item listings.* * *Implement error handling to manage and display 400 (Bad Request) and 500 (Server Error) errors appropriately.* * *Secure sensitive endpoints and ensure that all data is validated before being processed or stored in the database.* |
| 1. Identify Client-Side dynamic components/features | **Client-side Features**   * Build a dynamic and interactive GUI using React.js * Implement Bootstrap or a similar responsive framework to ensure the application adapts to various screen sizes and devices, providing a consistent user experience on mobile and desktop. * Using HTML5 and CSS3 * Design an intuitive UI that guides users through the application, making it easy to browse, search, and manage secondhand fashion items. * Follow accessibility standards like WCAG to ensure that all users, can interact with the application effectively. * Format and present data from the API in a clear and user-friendly manner. * Implementing client-side state management * Using local storage or session storage to save user preferences, search history, or session tokens. * Show user-friendly error messages for common issues |
| 1. Identify Server-Side dynamic components/features | **GraphQL API Features**   * Integrating the GraphQL API with a database (MongoDB or SQL) for storing and retrieving user profiles and item listings. * Implementing GraphQL queries and mutations to handle Create, Read, Update, and Delete operations for users and items. * Using a validation library by JOI to validate incoming data within resolvers before processing and saving it to the database. * Defining validation rules within the GraphQL schema to ensure that only valid data is processed and stored. * Authentication and Authorization must be implemented * Protect sensitive queries and mutations by ensuring they are accessible only to authorized users, such as admins or verified users. * Error pages or responses such as 404 & 500 errors must be created and implemented |
| 1. A. List all languages and technology needed to build this application ensure that they meet the technical requirements   B. Investigate and document two new technology that you have identified that could be used to enhance the application (minimum 1 paragraph) You will need to select and integrate at least one of these technologies. | ***A - Languages and*** ***Technology*** *JavaScript React.js Apollo bCrypt*  *Node.js MongoDB GIT  Cloud distribution – Heroku vsCode Postman Compass  Docker*  ***B - New Technology 1*** *react-toastify https://www.npmjs.com/package/react-toastify React-Toastify allows you to add notifications to your app with ease. No more nonsense! It works perfectly with AXIOS and can be triggered to display errors that have been intercepted by AXIOS. They module has a large amount of community support. 283,677 Weekly downloads. It can be heavily customised and the documentation provides various integration examples. This module will also dramatically reduce the amount of error handling code that need to be developed.*  ***B - New Technology 2*** *npm i react-bootstrap-icons https://www.npmjs.com/package/react-bootstrap-icons The brand new Bootstrap Icons library to use as React components. It works perfectly with React. Each icon is a separate component and can be easily integrated into a react project. They module has a large amount of community support. 10,000 Weekly downloads. It can be heavily customised and the documentation provides various integration examples. This module will also dramatically reduce the amount of time it would take to build individual components for each icon required and allows for future expansion of icons.* |
| 1. List the core features of the user interface design, target platform, user needs and design principles to be used. Refer to the project brief as a reference. | **Core Features**   * Search Box: A primary search bar to allow users to quickly find specific items. * Category Drop-down Filter * Pagination * Results Area: A visually appealing area to display search results as cards. * Login Screen: A screen for existing users to log into their accounts. * Register Screen: A screen for new users to create accounts. * Add/Edit Screen: A screen for users to create new items or modify existing ones. Includes a preview of the final item as a card.   **User Needs**   * *Search and filter items* * *Register as a user* * *Login as a user* * *Create new items* * *Customise the colour and icon of the items card to suit their needs* * *Delete their cards* * *Edit their cards*   **Design Principles**   * Clarity * One Primary Action per Screen * Natural Next Step * Consistency * Immediate Feedback |
| 1. Identify all architectural requirements for this project  This would include the stack you have selected such as MERN and any application and tools need for the application and deployment. | *MARN Stack (Mongo, Apollo, Vue, Node) Node.js MVC React.js Document Database MongoDB Data will be sent as JSON between client and server applications. GIT – for distribution to cloud service Cloud distribution – Heroku* |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Client Review** Assessment Documentation | | | | | | | | | |
| You will need to organise a time with your facilitator to review and negotiate your final plan documented above. You will need to present your chooses outlined in Part 1 of your documentation and negotiate changes to the plan based on feedback. During the presentation you will be asked to discuss the capabilities, efficiencies and effectiveness of the technologies you have selected. | | | | | | | | | |
| **Skills to be observed during this task to the required standard.** Checklist (To be completed by the learner’s facilitator)The following tasks are to be completed in relation to Part 1 – Planning. Each of the skills must be observed on at least one occasion. | | | | | **Date 1** | | | **Date 2** | |
| 4/9/2024 | | |  | |
| **Satisfactory** | | | **Satisfactory** | |
| **Yes** | **No** | | **Yes** | **No** |
| 1. The learner has discussed and negotiates key aspects with required personal, including required capabilities, efficiencies and effectiveness | | | | | *Yes* |  | |  |  |
| **Assessor Name** | *Daniel Fitzsimmons* | **Assessor Signature** | *Daniel Fitzsimmons* | **Date** | | | 4/9/2024 | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Part 2 - Prototype** Based on the information you have gathered during the planning stage complete the following:Assessment Documentation & Prototype | | | | | | | | |
| **BACK-END API PROTOTYPE – DUE SESSION 10** | | | | | | | | |
| **Build a prototype Back-End API.**  The API must include:   1. *At least 2 database models* 2. *A Config file* 3. *If a RESTful API – 2 Routes and 2 Controllers  The Controllers must container appropriate CRUD operations for the endpoint* 4. *If a GRAPHQL API – 2 Resolvers and Types The Resolver must container appropriate Queries and Mutations.*   ***You will need to organise a time with your facilitator to review and sign off your prototype.*** | | | | | | | | |
|  | | | **Date 1** | | | **Date 2** | | |
| 30/1/2024 | | |  | | |
| **Satisfactory** | | | **Satisfactory** | | |
| **Yes** | **No** | | **Yes** | | **No** |
| The learner has developed a prototype that meets the requirements listed above. | | |  |  | |  | |  |
| **Assessor Name** | *Daniel Fitzsimmons* | **Assessor Signature** | *Signature* | | **Date** | | *30/1/2024* | |
| **Part 2 - Prototype** Based on the information you have gathered during the planning stage complete the following:Assessment Documentation & Prototype | | | | | | | | |
| 1. Develop at least one wireframe layout design for the website UI that aligns with the project brief and your plan. Ensure that the wireframe is annotated. | | | | | | | | |
| *Insert wireframe here* | | | | | | | | |
| 1. Produce hierarchy/site map of website and ensure it aligns with the requirements outlined in the brief. | | | | | | | | |
|  | | | | | | | | |
| 1. Produce a prototype of the user interface using HTML/JSX, CSS, Bootstrap, React.js. You may use an alternative to Brightspace if you wish. Recorded screenshots of the prototype below. Ensure that the prototype aligns with the wireframe. | | | | | | | | |
| *Insert a screenshot of the final design* | | | | | | | | |
| 1. Check to ensure that the prototype is logical and accessible for the end user according to website requirements outlined in the project brief.  *(Mark as complete once you have checked your prototype)* | | | **Complete** | | | **Not Complete** | | |
|  | | |  | | |

|  |  |  |
| --- | --- | --- |
| **Part 3 - Development**Working Files | | |
| It is now time to build the frontend of your web application using front-end framework  **Ensure that you develop all of the features outlined in the brief.**  ***Below is a list of specific criteria you will need to ensure your application includes.*** *Mark each as yes once you have ensured you have included these features.* | | |
|  | **Complete** | **Not Complete** |
| 1. Boilerplate templates have been created according to technical requirements |  |  |
| 1. All required React components for the client have been developed according to the website requirements |  |  |
| 1. All React components have been incorporated into the website to produce the web application |  |  |
| 1. Web page structure has been developed according to design specifications |  |  |
| 1. Components and Layout have been position and styled for 3 web pages according to design specifications outlined in Part 1 |  |  |
| 1. Cascading style sheet and application pages have been developed according to design specifications |  |  |
| 1. The web application has been developed using the selected languages from Part 1 |  |  |
| 1. A database has been integrated into the application and the data has been organized according to data storage requirements outlined in the project brief. |  |  |
| 1. Client and server-side dynamic content has been incorporated into each web page |  |  |
| 1. The client and server side code is meets the requirements and design requirements outlined in the project brief |  |  |
| 1. Web security features have been implemented |  |  |
| 1. http and appropriate http status codes have been used to communicate between the client and server |  |  |
| 1. Session management has been implemented on the client and stateless programming has been used to develop the API. |  |  |
| 1. Programming control structures have been used |  |  |
| 1. The application demonstrate correct syntax and uses of programming languages |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Part 4 - Testing and Debugging**Assessment Documentation/Working FIles | | | | | | | | | | | | |
| 1. Test the database and application security and record the results. Below you will need to design and record tests for your server-side API and Database. The result must be recorded. | | | | | | | | | | | | |
| **Testing Requirements** | | | | **Comment if required** | | | **Functions as Expected** | | | | | |
| **Yes** | | | | **No** | |
| Test your client side application in browsers and ensure the layout behaves as expected. | | | |  | | |  | | | |  | |
| Ensure the dynamic features of your client side application behave as expected.   (Your application loads and send data to and from your API as intended) | | | |  | | |  | | | |  | |
| Test all routes and CRUD operations using postman. | | | | E.g. All CRUD operation and routes behave as expected | | |  | | | |  | |
| Tested that the 404 error page is displayed when a user inputs an undefined route. Input several undefined routes and check response. | | | | E.g. The 404 error page was displayed for each undefined route | | |  | | | |  | |
| Ensure that the UI and UX for the application is logical and accessible. Have 2-3 people test the layout and design while observing them and gathering feedback. | | | | E.g. Users were intuitively able to navigate the UI and UX with no issues discerning how to perform key operations. | | |  | | | |  | |
| 1. **Debug the code** You will need to organise with your facilitator to observe you debug your code and demonstrate the following skills on two separate occasions. You will also need confirmed the website is secure and bug free according to cyber security procedures and protocols. | | | | | | | | | | | | |
| **Skills to be observed during this task to the required standard.** Checklist (To be completed by the learner’s facilitator)The following tasks are to be completed in relation to the brief for this project. Each of the skills must be observed on two separate occasions. These may occur on the same day. | | | | | | **Date 1** | | | | **Date 2** | | |
| --/--/-- | | | | --/--/-- | | |
| **Satisfactory** | | | | **Satisfactory** | | |
| **Yes** | | **No** | | **Yes** | | **No** |
| The learner has used the language debugging facilities of an integrated development environment (IDE) to examine and output results and variables. | | | | | |  | |  | |  | |  |
| The learner has interpreted the compiler or interpreter messages to resolve syntax errors, and use debugging techniques to resolve logic errors | | | | | |  | |  | |  | |  |
| Use the code stack trace and to detect and correct errors | | | | | |  | |  | |  | |  |
| The learner has resolved all bugs and confirmed the website is secure and bug free according to cyber security procedures and protocols | | | | | |  | |  | |  | |  |
| **Assessor Name** | *Daniel Fitzsimmons* | **Assessor Signature** | *Signature* | | **Date** | | | | --/--/-- | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Test and confirm you have completed all of the following. | | | |
| **Testing Requirements** | **Comment if required** | **Functions as Expected** | |
| **Yes** | **No** |
| 1. The functionality of website has been tested in 2 browsers browsers and devices and required updates have been made and saved to your GIT Repo. **Provided Screenshots of the application in the 2 browsers below.** |  |  |  |
| **Screenshot – Browser 1** | **Screenshot – Browser 1** | | |
| *Insert screenshot from browser 1 here* | *Insert screenshot from browser 2 here* | | |
| **Testing Requirements** | **Comment if required** | **Functions as Expected** | |
| **Yes** | **No** |
| 1. Accessibility has been checked and confirmed for web sections, elements and pages |  |  |  |
| 1. React components of the website have been tested according to organisational procedures |  |  |  |
| 1. Required dynamic content functions according to task requirements |  |  |  |
| 1. The HTML and CSS have been validated against industry standards. **Provided screenshots of the validation results below** |  |  |  |
| **Screenshot – HTML Validation Result** | **Screenshot – CSS Validation Result** | | |
| *Insert screenshot here* | *Insert screenshot here* | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Part 5 – Hand Over and Sign Off** Assessment Documentation/ Working Files | | | | | | | | | |
| Your application is almost finished but before we present the project and hand it over we need to ensure we have meet all of the clients and user’s requirements. You will need to organise a time with your facilitator to observe you referring to appropriate documentation and review your final solution for the project. You will need to present and explain your code and program structure. Be prepared for the following.   * Present your final application and accept feedback * Update your application based on the feedback * Ensure the web application has been finalized, meets all requirements, all policies procedures and standards * Demonstrate and explain the new technology you have selected and implemented * Used GITHUB and Teams throughout the project | | | | | | | | | |
| **Skills to be observed during this task to the required standard.** Checklist (To be completed by the learner’s facilitator)The following tasks are to be completed in relation to the brief for this project. Each of the skills must be observed on at least one occasion. | | | | | **Date 1** | | | **Date 2** | |
| 30/1/2024 | | |  | |
| **Satisfactory** | | | **Satisfactory** | |
| **Yes** | **No** | | **Yes** | **No** |
| 1. The web application has been presented to required personnel and feedback has been sought | | | | |  |  | |  |  |
| 1. The web application has been updated as required based on feedback | | | | |  |  | |  |  |
| 1. The web application has been finalized, meets all requirements, all policies procedures and standards have been adheadered to and sign-off has been obtained from required personnel | | | | |  |  | |  |  |
| 1. The learner has accepted responsibility for planning and sequencing complex tasks and workload | | | | |  |  | |  |  |
| 1. The learner has used and investigates new digital technologies and applications during the development of this project | | | | |  |  | |  |  |
| 1. The learner has used platforms(Eg. GITHUB & Digital Communication tools) to manage and manipulates data, and communicates with others in a secure and stable digital environment | | | | |  |  | |  |  |
| **Assessor Name** | *Daniel Fitzsimmons* | **Assessor Signature** | *Signature* | **Date** | | | *30/1/2024* | | |
| 1. Documented the design structure of your project as a flow chart. | | | | | | | | | |
| *Insert flow chart here* | | | | | | | | | |
| 1. Document at least 2 scripts according to the requirements outlines in the project brief. Recorded the path to the files.   *Ensure custom JavaScript code has been documented, comments should be added within each custom JavaScript file to explain the following:*   * *The purpose of the Scripts at the top of the script files* * *The purpose of all global variables* * *The purpose and use of functions and any returned values* * *The purpose of all parameters* | | | *EG. Client/src/utils/forms.js*  *EG. Client/src/utils/helpers.js* | | | | | | |