Front-Loaded - End Point Assessment: Project Activity

|  |  |
| --- | --- |
| **Summary** | |
| **Project Activity:**  In preparation for a learner’s End Point Assessment (EPA), learners will put their newfound knowledge into action via a case study project, progressing their learning in the workplace.  **Workplace application submission:**  Complete a project by fulfilling all deliverables stated, with explanations as to how this project has been completed.  The DLC will assist apprentices with this activity in the workplace. | |
| **Title** | **Detailed Definition and Instructions** |
| **EPA Module Activity**  **EPA Project: Case Study Introduction** | **Introduction:**  This submission is all about bringing what you have learned during your apprenticeship, both in training and in the workplace, to life, exploring and demonstrating both theoretical and practical elements of your learning.  Now that you have covered an introduction to software development, building responsive web applications, automation and software quality, design, DevOps and software craft, it’s time to put this all together!  For this task you will be required to complete a project scenario requiring the building of software, and the subsequent automated testing. You will be required to target your writing to address the mapped knowledge, skills, and behaviours. To help with this we have included advice on what to include and tips to the instructions below.  You are to produce a project that fulfils a given case study. This is a demonstration of how you have applied the skills relevant to the subject area. Include your work products with supporting narrative or annotation to describe the steps you have taken to complete the brief via deliverables. Gather feedback and analyse your findings alongside workplace evidence demonstrating your ability to manage and lead projects. |

|  |
| --- |
| **EPA Project** |
| [need some support to fill this section] |
| **Project Introduction** |
| You are a Software Engineer that has been asked to design a software solution for a business. For this, you are required to develop an application, write tests and create a CI/CD pipeline to test the software with each push to the repository.  This project will involve concepts from all of your training modules; more specifically, this will involve:   * Introduction to Software Development * Building Responsive Web Applications * Automation and Software Quality * Design and DevOps * Software Craft |
| **Example Case Study Brief** |
| Below is an example Case Study Brief which has a supporting GitHub repo demonstrating all of the required tasks to complete.  You have been hired as a Software Engineer to create a social media application for book lovers. A graphical user interface is required to allow users to write book reviews, as well as create and manage their personal profiles.  In order for the application to be effective it must be able to do the following:   * Create new book reviews * Read existing book reviews * Update existing book reviews * Delete book reviews * Create user profiles * Update user profiles * Delete user profiles   The owners are requesting the data be stored within some form of database and are open to more functionalities being added.  The team is also looking into the benefit of adopting a DevOps mindset throughout this project and would like to explore the possibility of setting up a Jenkins pipeline linked to the GitHub repository to automatically build and test the software with each push to the repository. |

|  |
| --- |
| **Objectives** |
| The overall objective of the project is the following:  **To plan, design and implement a solution for a given application, with utilisation of supporting tools, methodologies and technologies that encapsulate all modules covered during training.**  Specifically, you have been given a project scenario to adhere to during this project. By minimum requirements, the application must be developed in a general-purpose language and be linked to a database for CRUD-based operations. The application must have a front-end GUI and be unit tested, with tests automated via a pipeline.  The exact approach you take to meeting the requirements can be different to the approach taken in the example provided to you. You can use any tools and tech you are familiar with, or that you think would be appropriate in this context. |
| **Project Steps** |
| This project has been broken down into smaller sections reflective of your learning within your apprenticeship.  Whilst this is not a strict requirement of the order of work, the below is the minimum required in order to complete the project. You must complete every learning point listed below as to meet the Knowledge, Skill and Behaviour points stated in your assessment plan. |
| **Project Steps: Step 1 – Requirements and Planning** |
| **Below are the tasks required to complete this step:**   * Begin by writing a small brief on the requirements for this application, including the following:   + Justification of why this is a suitable app for the domain   + Tech requirements for development   + 5 or more user stories for features that this app will complete   + Brief discussion of a solution, including some starting solution features * Following this, ask your work colleagues to review and have them suggest at least 2 additional features. Once suggested, create a user story for these features. * Add acceptance criteria, functional/non-functional requirements and time estimates to your user stories and any resulting sub-tasks.   **KSBs met in Step 1:**   * **K2:** roles and responsibilities within the software development lifecycle (who is responsible for what). * **K11:** software designs and functional/technical specifications. * **B2:** Applies logical thinking. For example, uses clear and valid reasoning when making decisions related to undertaking work instructions. |

|  |
| --- |
| **Project Steps: Step 2 – Version Control and Development** |
| **Below are the tasks required to complete this step:**   * Create a repository for this application and ensure to adhere to the feature-branch model. The minimum requirement is a main branch, a dev branch, and some feature branches **(K6)**.   **Back-end:**   * Take the user stories from the previous step and begin writing the code to complete the story and feature via a general-purpose language, such as JavaScript, Java or Python **(K9, K11, S1, S7, S11, S12, S16)**. * For at least 1 of these features, write your code in a TDD format i.e., write the tests for a feature first, then write the code to complete it **(S6, K11)**. * Design the database schema and set up tables if necessary **(B2, S12, K11)**. * Link the database to your application for persistence **(K11)**.   **Front-end:**   * Create a Graphical User Interface (GUI) to interface with the back-end and display book reviews and user profiles. * This could be a simple Frontend using HTML, CSS & JS, or could include libraries such as React, and frameworks like Bootstrap or MaterialUI **(K9, K11, S1, S7, S11, S12, S16)**. * Ensure that the front-end allows the users to perform all of the CRUD based operations defined by your back-end.   **KSBs met in Step 2:**   * **K6:** How teams work effectively to produce software and how to contribute appropriately. * **K9:** Principles of algorithms, logic and data structures relevant to software development for example: Arrays, Stacks, Queues, Linked Lists, Trees, Graphs, Hash Tables, Sorting Algorithms, Searching Algorithms, Critical sections and race conditions. * **K11:** Software designs and functional/technical specifications. * **S1:** Creates logical and maintainable code. * **S6:** Identifies and creates test scenarios. * **S7:** Applies structured techniques to problem solving, can debug code and can understand the structure of programmes to identify and resolve issues. * **S11:** Applies an appropriate software development approach according to the relevant paradigm (for example object oriented, event driven or procedural). * **S12:** Follows software designs and functional/technical specifications. * **S16:** Applies algorithms, logic and data structures. * **B2:** Applies logical thinking. For example, uses clear and valid reasoning when making decisions related to undertaking work instructions. |

|  |
| --- |
| **Project Steps: Step 3 - Testing** |
| **Below are the tasks required to complete this step:**   * Begin by selecting an appropriate unit testing framework for your back-end and front-end. Here are some examples:   + JUnit & Mockito (Java)   + Pytest (Python)   + Jest/Mocha (JS) * Identify the units of code that need to be tested, such as classes or functions that handle specific functionality **(B2)**. * Use your chosen testing framework to write unit tests for each unit of code and verify that each unit of code works correctly **(K9, K11, S6, S7, S12, S16)**.   + Mock dependencies, such as API responses or databases, to isolate the code under test and ensure consistent behaviour in tests.   + Run tests frequently to catch bugs early and refactor to prevent them from causing larger issues **(S4)**.   **KSBs met in Step 2:**   * **K9:** Principles of algorithms, logic and data structures relevant to software development for example: Arrays, Stacks, Queues, Linked Lists, Trees, Graphs, Hash Tables, Sorting Algorithms, Searching Algorithms, Critical sections and race conditions. * **K11:** Software designs and functional/technical specifications. * **S4:** Tests code and analyses results to correct errors found using unit testing. * **S6:** Identifies and creates test scenarios. * **S7:** Applies structured techniques to problem solving, can debug code and can understand the structure of programmes to id]entify and resolve issues. * **S12:** Follows software designs and functional/technical specifications. * **S16:** Applies algorithms, logic and data structures. * **B2:** Applies logical thinking. For example, uses clear and valid reasoning when making decisions related to undertaking work instructions. |
| **Project Steps: Step 4 – CI** |
| **Below are the tasks required to complete this step:**   * Install Jenkins on a server or cloud platform, such as AWS or Azure. * Set up and configure a pipeline to build and test the application. * Run the pipeline frequently, such as on every push to the repository using a webhook, to catch issues early and ensure consistent builds.   **KSBs met in Step 2:**   * **K11:** Software designs and functional/technical specifications. * **S10:** Build, manage and deploy code into the relevant environment. * **S12:** Follows software designs and functional/technical specifications. * **B2:** Applies logical thinking. For example, uses clear and valid reasoning when making decisions related to undertaking work instructions. |

|  |
| --- |
| **Resources and Environment** |
| **For this project, you will need access to the following tooling & environments:**   * A computer with a browser, Word processor & spreadsheet software installed. * Git Bash & access to GitHub. * General purpose programming language with an appropriate IDE. * Unit Testing and Mocking packages. * An appropriate DBMS (such as MySQL), either cloud hosted or local. * Access to an appropriate cloud environment, such as AWS, if needed. |
| **Deliverables** |
| The final deliverable for this project is the completed solution with full documentation around utilisation of supporting tools, processes followed, and pipelines ran. This will require a fully functional application in a version control system, commented and described via inline comments and a README contained in the repository.  You will be required to utilise the Feature-Branch Model, and to push a working copy of your code to the main(master) branch at the end, and ensure you are merging your feature branches with your dev branch regularly. It is recommended to use the feature-name naming strategy for your feature branches.  Alongside this, documentation and demonstration of the pipeline used should be present, alongside justifications of tool usage. Any user stories with their accompanying information (i.e., acceptance criteria, functional/non-functional requirements) should also be included, as well as feedback from colleagues and how this feedback was actioned.  In order to keep a record of how you have achieved all of the above, you are to also individually produce a portfolio that evidences each learning point described in the criteria.  You will be required to include all supporting documentation for your project within your remote repository, as well as the aforementioned documentation, to your DLC for your BUD submission. |
| **Next Steps** |
| [need some support to fill this section] |