**Assignment: APPD2A**

**App Development 2**

For Learner Use:

|  |  |
| --- | --- |
| Surname of Learner | Sewcharan |
| Name of Learner | Arya |
| Learner ID | 0412050140083 |
| Student Number | ASDBN012 |
| Date of Test Given |  |

For Assessors Use:

|  |  |
| --- | --- |
| Name of Facilitator |  |
| Name of Assessor |  |
| Mark Allocation | 170 |
| Mark Obtained |  |
| Competency Status (C / NYC) |  |

|  |  |
| --- | --- |
| 119458 Occupational Certificate: Software Engineer | * 251201-001-00-00-KM-02, Programming, NQF Level 6, Credits 20 * 251201-001-00-00-PM-03, Program and deploy applications, NQF Level 6, Credits 25 * 251201-001-00-00-PM-04, Test or debug source code to ensure client’s needs are met, NQF Level 3, Credits 15 |

|  |  |  |  |
| --- | --- | --- | --- |
| Candidates Signature |  | Date of Submission |  |
| Assessors Signature |  | Date Marked |  |

**Assignment set up**

This assignment is made up of four tasks:

* **Task A –** Market the exhibition to a suitable audience
* **Task B –** Plan the exhibition presentation
* **Task C –** Plan, develop and debug your software
* **Task D** – Test and deploy your software

**Grading criteria**

**Pass –** Candidates must achieve 102

**Merit –** Candidates must achieve 128

**Distinction –** Candidates must achieve 145

**Scenario**

You have recently completed all required coursework towards obtaining a Qualification in I.T Engineering.

As the final Summative Assessment, you are required to build the software of your choice to demonstrate and exhibit what you have learned and achieved over the period of study to local Software Development Companies, family, friends and other interested parties.

This assessment is your opportunity to take your newfound savvy with programming for a spin and develop your own piece of software.

So long as your project draws upon the lessons learned during this course, the nature of the project is entirely up to you.

You may implement your project in any language(s). You are welcome to utilise any tools or platforms other than what was covered in this course. We ask that you build something of interest to you – that you solve a real problem, that you impact your community, or that you change the world. Strive to create something that outlives this course.

Some potential ideas include:

* a web-based application using JavaScript, Python and SQL
* an iOS app using Swift
* a video game using Unity and C#
* an Android app using Java or Kotlin
* a Chrome extension using JavaScript
* a command-line program using C or Rust
* a hardware-based application for which you program some device

... Or anything else!

Moreover, you will be required to present your work at an exhibition that will be held on campus. Your presentation must include the following elements:

* An oral presentation
* Graphics, visuals or video to assist in demonstrating your work
* A demonstration of the working software you have built
* Motivation behind the development of the software
* How the software could be further improved
* What you have learnt from building the software

**Task A – Market the Exhibition to a suitable audience (10)**

1. Select a minimum of five companies that may find you and your work of interest as a software engineer. (5)

Take the kind of software you are building and your target audience into account when contacting people to attend the exhibition. For example, a highly technical project might not be of much interest to a non-technical individual.

Complete the table below with the information.

|  |  |  |  |
| --- | --- | --- | --- |
| **Company Name** | **Contact Person** | **Contact Number** | **Contact Email** |
| KwaZulu-Natal Department of Education (KZN DoE) | Head office | +27 33 846 5000 | headoffice@kzndoe.gov.za |
| Umlazi District Office, KZN DoE | District Manager Mr P.N. Cele | 031 360 6597 / 6210 | peac2eman.ntokozo0@gmail.com |
| Naspers | Eoin Ryan | +27 21 406 2121 | investorrelations@naspers.com |
| STEMLab |  | 063 434 9750 | info@stemlab.co.za |
| Department of Science, Technology and Innovation | Dr Mlungisi Cele | +27 12 843 6300 | webmaster@dsti.gov.za |

1. Write out the message you will use in your email when you are contacting companies to attend the exhibition. Have an assessor review the email **before** sending it. (5)

|  |
| --- |
| Good day,  I hope this email finds you well.  My name is Arya Sewcharan, and I am currently pursuing my IT Engineering diploma. I am excited to share one of my key projects, an Interactive Periodic Table Digital Science Lab, which I believe aligns with your company's innovative and solution-driven approach.  I would be honoured to demonstrate my digital portfolio and showcase this project at the Exhibition Presentation hosted by my institution. This event provides students with the opportunity to present their digital solutions to leading tech companies.  Date: 22 September 2025  Time: 10:30 am  Venue: iStudent Academy, 16 Peter Mokaba Ridge  I would greatly appreciate your presence at this exhibition and look forward to the possibility of connecting with your company.  Thank you for your time and consideration.  Kind regards,  Arya Sewcharan  079 633 6014 |

**Task B – Plan the Exhibition Presentation (10)**

You will have to present your work to everyone attending the Event. This will be in the form of an **oral presentation**. You will have access to a computer and a projector with which you can present what you have made.

Your presentation **must** include graphics – a PowerPoint, video and/or images to accompany your presentation will suffice. Your presentation **must** be **at least 3 minutes** long and may lasta maximum of **10 minutes**. Attach your graphics to this assignment and reference these files in the box below.

Write out the monologue for your presentation. Your monologue must include the following:

* Motivation behind the development of the software
* How the software could be further improved
* What you have learnt from building the software

|  |
| --- |
| Good day to each and everyone of you, my name is Arya Sewcharan, and it is my honour to present to you my digital science lab software featuring an Interactive Periodic Table.  The motivation behind this project comes from my own school experience. I studied physical science to keep my doors open after school, but as a partial learner who understood better through practice than theory, it was difficult. I attended a public school that did not have enough funds to perform experiments. This meant that many of the experiments in the syllabus were things I only read about in textbooks. This made science feel stressful instead of exciting.  From that experience, I began dreaming of a digital platform where students could perform experiments virtually, a space where lack of resources would not hold back learning. That dream inspired me to create this Interactive Periodic Table, a digital science lab that allows students to select elements, combine them, and see the results of real chemical reactions. This makes learning science more interactive, fun, and accessible.  Looking ahead, I would like to expand the software by including other branches of science, such as biology. I also want to add a built-in bot that answers student's questions, provides clarity, and offers short tutorials or alternative ways of explaining concepts. Future updates may include quizzes, 3D models of molecules, and video demonstrations. This would help different types of learners, especially those who struggle with theory-based learning.  Through this project, I have learnt just how powerful programming can be in solving real-world problems. I have discovered that with creativity and coding, you can build tools that break down barriers in education and make learning more engaging. I also gained a better understanding of JavaScript and how software development works from designing the interface, to handling data, to making sure everything runs smoothly.  In conclusion, my Interactive Periodic Table is not just a school project, it is a vision of how technology can transform science education. I hope that one day, students everywhere can use tools like this to learn, experiment, and enjoy science, regardless of their school's resources.  Thank you all for your time and consideration. |

**Task C – Plan, Develop and Debug your Software (90)**

The nature of this project is inherently open, and you have the freedom to build whatever software you would like. It is encouraged to take this as an opportunity to challenge yourself, learn to use new tools, technologies and techniques, and most importantly to solve a real problem that is important to you or your community.

However, any attempt to cheat, plagiarise or steal another’s work, **will not be tolerated**. The use of libraries and open-source software is permitted and encouraged, if its use is necessary and sufficient motivation is provided. You cannot simply portray the work of others as your own.

1. Describe the software that you will build, your motivation behind it, and who might benefit from it. (5)

|  |
| --- |
| The software I am building is an Interactive Periodic Table designed as a digital science lab. It allows users to explore all chemical elements in a periodic table layout, view their properties, and combine two elements to see the possible reaction between them. When a user selects an element, the software will fill one jar with that element and display details about the selected element such as its element name, group, state, and common uses, when the user selects another element, it will fill another jar and displays its details, the two elements then combine in the third jar and shows their chemical reaction. The periodic table is colour-coded by groups to make it more visually appealing and easier to understand.  My motivation behind creating this software is to make learning science more engaging and accessible. Many students find science overwhelming, so I wanted to design a tool that is not only educational but also interactive and fun. By bringing the table to life with visuals and simulations, it helps learners connect theory with practice.  This software will benefit students, teachers, and schools by serving as a modern learning aid that allows users to explore chemical reactions in a safe, virtual environment. Public schools in particular can use it as a cost-effective way to introduce digital science labs where access to real lab equipment may be limited. |

1. Write a detailed design specification detailing all the highlights and features of your software and how the flow of interaction between the end user and the software should work. (20)

|  |
| --- |
| The Interactive Periodic Table is a web-based digital science lab that allows users to explore the periodic table, learn about each element, and virtually preform chemical reactions. It is designed to be visually appealing, user-friendly, and accessible for students and teachers.  **Key Features**   * **Periodic Table Layout:**   + All 118 elements are displayed in the standard periodic table layout.   + Each element is represented by its symbol and atomic number.   + Groups are represented by distinct colours for easy identification (e.g., Alkali earth metals = red, Alkali Metal = green, Noble gases = yellow).   + A legend is included as a key reference to explain which colour refers to which group. * **Element Details Panel:**   + When a user clicks on an element, the software displays its details:     - Element name     - Symbol and atomic number     - Group and period     - State of matter (solid, liquid, gas)     - Common uses and applications * **Reaction Jars:**   + Three jars are displayed in the side panel:     - **Jar 1:** Fills with the first selected element.     - **Jar 2:** Fills with the second selected element.     - **Jar 3:** Shows the result of combining the two elements.   + If a valid chemical reaction exists, the product and its properties are shown (e.g., H + O → H₂O).   + If no known reaction exists, the software displays “No reaction.” * **Interactive Design:**   + Hovering over an element highlights it and shows a quick preview of its name.   + Clicking selects it and fills the jars.   + A “Reset” button clears the jars and allows users to start again. * **Accessibility:**   + Clean, minimal interface suitable for all age groups.   + Large text and colour contrast for readability.   + Works on computers, tablets, and projectors in classrooms.   **Flow of Interaction Between User and Software**   1. **Start Screen:** When the user opens the software they see the full periodic table with colour coding, a legend and a side panel. 2. **Selecting First Element:** When theuser clicks an element Jar 1 fills with the element. Element details (name, group, state, uses) appear in the side panel. 3. **Selecting Second Element:** When theuser clicks an element Jar 2 fills with the element. Element details (name, group, state, uses) appear in the side panel. 4. **Reaction Display:** The software will automatically check if a known reaction exists between the two chosen elements.    * If yes, Jar 3 shows the resulting product and description with colour representation.    * If no, Jar 3 displays “No reaction.” 5. **Exploration & Reset:** The user will then click the “clear selection’ button to clear all jars and details to test other reactions.   **Design Highlights**   * Educational: Combines theoretical knowledge with interactive practice. * Visual: Bright colours, clear layout, and animations make learning engaging. * Safe Simulation: Students can experiment with reactions virtually without real lab risks. * Cost-Effective: Schools can use it as a substitute or supplement to physical labs. * Scalable: Future updates may include quizzes, 3D models of molecules, video demonstrations and a bot. |

1. List out all the libraries, tools, engines etc. that you will be using to build your software. Provide links to each, e.g. a GitHub repository, website or package manager page. Provide your reasoning for the use of each. (5)

Return to this question to add/remove libraries and tools you end up using during the development of your software. If you stopped using a certain library or tool, make a note of why you decided to stop using it and what your alternative was.

|  |
| --- |
| Notepad ++: used to write my code.  JavaScript (language): The interactive behaviour such as selecting elements, filling jars, running reaction logic is implemented through the use of JavaScript.  HTML (structure): used for page markup and periodic table element information that will be displayed.  CSS (styling): used for styling my software, colour-coding, responsive layout and simple transitions.  GitHub: (remote repo & hosting): I used GitHub to host my code and publish my project.  <https://github.com/>  GeeksforGeeks: Used for guidance, examples, and explanations for HTML, CSS, and JavaScript.  Website: <https://www.geeksforgeeks.org/>  Stack Overflow: used for debugging issues and finding coding solutions.  Website: <https://stackoverflow.com/> |

1. Create a detailed implementation plan for your software and how you will go about building it according to your design specification. Provide the expected time each step will take, in which language the solution will be implemented, and any challenges you might face. (10)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Implementation Plan**  **Languages & Tools used:**  Frontend: HTML, CSS, JavaScript  Testing: Manual browser testing and console debugging  Version Control: GitHub  **Step 1: Setup & Project Structure**  Tasks:  Create folder “My Software”.  Create a html file, JavaScript file and CSS file within the folder.  Link files in periodic table.html (style.css, script.js).  Code periodic table data (periodic table grid, element JSON).  Time: 2 days  Challenges: Making the structure scalable for later features.  **Step 2: Build Periodic Table Layout**  Tasks:  Use CSS grid and flexbox to display all 118 elements in proper layout.  Ensure responsiveness, no horizontal scrolling and resize for display.  Language: HTML and CSS  Time: 1–2 days  Challenges: Positioning lanthanides & actinides without breaking grid layout.  **Step 3: Element Data & Info Panel**  Tasks:  Create elements.json with name, symbol, group, state, and uses.  When a user clicks on an element display details in side panel (symbol, group, state, use).  Language: JavaScript + JSON  Time: 2 days  Challenges: Handling click events efficiently without performance lag.  **Step 4: Interactive Reactions (Jar Filling)**  Tasks:  Create “Reaction Jars” in the side panel.  Allow selection of 2 elements which displays elements in the jars and their chemical reaction.  Use reactions.json for known reactions.  Language: JavaScript  Time: 3 days  Challenges:  Mapping all valid element combinations.  Handling unknown or invalid reactions gracefully (show “No reaction”).  **Step 5: Styling & UX**  Tasks:  Add hover effects and color-coded groups (metals, nonmetals, noble gases, etc.).  Improve layout - side panel for element info & reactions.  Use smooth animations for jar filling (CSS transitions).  Language: CSS + JavaScript  Time: 2 days  Challenges: Making UI clean without overwhelming users with too much data.  **Step 6: Testing & Debugging**  Tasks:  Test all elements are clickable.  Test reactions from the reaction library.  Check performance across browsers (Chrome, Edge).  Time: 2 days  Challenges: Browser inconsistencies, fixing bugs in event listeners.  **Step 7: Presentation Preparation**  Tasks:  Create a PowerPoint to explain my software.  Practice monologue.  Time: 2 days  Challenges: Making sure the PowerPoint runs smoothly on projector hardware.  Estimated Total Timeline: 15 days   | **Step** | **Task** | **Time** | | --- | --- | --- | | 1 | Setup | 2 days | | 2 | Layout | 1–2 days | | 3 | Element Data | 2 days | | 4 | Reactions | 3 days | | 5 | Styling | 2 days | | 6 | Testing | 2 days | | 7 | Presentation | 2 days |   **Challenges to Anticipate:**  Large Data Handling – Managing 118 elements and reaction combinations without slowing UI.  Accuracy – Ensuring scientific correctness in reactions.  Cross-Browser Issues – CSS grid/flex may behave differently.  Time Management – Staying on schedule with coding and presentation prep. |

By this point in the assignment, you have a solid idea in mind of what you would like to build and have begun working on the implementation of your software project.

1. Create a GitHub repository for your software project. (35)

Your GitHub repository must have an appropriate name and be made **public** so that the assessor may access it and view all your commits. Your repository must also contain a README.md and an appropriate licence.

By the end of your project, you must have made **at least 5 commits, excluding the initial commit**, each detailing what was accomplished in that commit.

Your GitHub repository and commit history will be assessed.

Provide a link to your repository below for examination:

|  |
| --- |
|  |

In software development, nothing is perfect. There will always be bugs to fix and issues to resolve. Any issues you encounter while building your project, log them here.

1. Report any logical errors or bugs you encounter during the development process and the techniques you employed to solve them. Provide **at least 2-3 screenshots in total**, not every issue requires a screenshot. (10)

|  |
| --- |
| Issue 1: Elements not displaying correctly  Description: When loading the table, some element tiles were not aligned in their correct positions (e.g., Hydrogen appearing outside the grid).  Cause: Incorrect grid row and column assignment in the CSS and HTML structure.  Solution: Rechecked the periodic table layout and corrected the grid-template areas in CSS so each element matched its group and period.  Screenshot:  Issue 2: When clicking on an element, the jar did not update with the selected element.  Cause: The event listener in the JavaScript file was incorrectly targeting the wrong element ID.  Solution: I checked the HTML and ensured that each element in the periodic table had the correct id. Then, I updated the JavaScript to correctly map the selected element to the jar.  Screenshot: (Insert screenshot of console error message like *“Cannot read property ‘innerText’ of null”* and then a screenshot after fixing, showing the jar filling correctly.)  Issue 3: After selecting two elements, no result displayed in the reaction jar, even when a valid reaction was expected.  Cause: The reaction lookup in my reactionLibrary used mismatched keys (e.g., "H+O" instead of "O+H").  Solution: I fixed this by normalizing the selection order in JavaScript, sorting elements alphabetically before checking the reaction library. This ensured both "H+O" and "O+H" matched correctly.  Screenshot: (Insert screenshot of code showing the corrected reaction lookup logic, and browser output showing H + O → H₂O working.)  Issue 4: The element details and reaction jars required unnecessary vertical scrolling on the side panel.  Cause: The CSS overflow-y property was set incorrectly.  Solution: I adjusted the CSS by setting overflow-y: auto; only when needed and reorganized the layout with flexbox so the panel fits properly on screen.  Screenshot: (Optional — show before/after of the side panel layout.)  **Techniques Used for Debugging**  Console Logging: Used console.log() to trace element selections and check reaction keys.  Trial and Error: Made small adjustments and tested frequently to isolate which change fixed the problem.  Documentation & Forums: Referred to GeeksforGeeks and Stack Overflow when encountering persistent issues. |

“I call it my billion-dollar mistake. It was the invention of the null reference in 1965...” – Tony Hoare

When dealing with the Internet, the Operating System, or whenever there are users involved, there is always the possibility of something going wrong. Proper error handling is vital in ensuring a positive user experience.

1. Report and explain in detail **at least 2 instances** in your software’s code that might cause an error and how you handle the possibility of these errors. (5)

|  |
| --- |
|  |

**Task D – Test and Deploy your Software (60)**

Testing is vitally important to ensure your software operates correctly and handles all edge cases.

1. Write **at least 2 unit tests** to test functionality of individual components of your software. Ensure that each test handles all known edge cases. Explain the purpose of each test and provide screenshots. (10)

|  |
| --- |
|  |

1. When you believe you are finished with your software, **perform full system and acceptance testing** on your software. Write down all the results you get from the test. Whether or not you are fully satisfied with the result of your testing, write down your conclusion. (10)

**This is an iterative process, work on your software and repeat until you are satisfied with the outcome.**

|  |
| --- |
|  |

If you’ve made it this far, you’ve finished the development of your software project, and it is ready to serve its purpose. Now the time has come to deploy your software. The steps you take in this section is largely dependent on what kind of software you’ve built.

1. Deploy your project as a final release. (40)

Should you have any queries on how best to complete this section depending on the kind of software you have built, speak to your assessor.

Many programming languages and tools have support for a release or production flag that can be set before compilation or building, intended for the final product. If your target language/tool supports this, enable this option now for best optimisation and performance.

* 1. Provide the final release build and all your source code along with this assignment document on submission and reference these files in the box below.

Both your release build and your source code will be assessed.

Code documentation, comments and organisation are of vital importance to ensure maintainability.

Ensure that your code is **sufficiently documented** **and explained** where necessary, and that your code is organised and modularised as best as possible.

* 1. Provide screenshots and photos of your final software in action.
  2. Did anything change from the original design compared to what it is now? Describe these changes here.

|  |
| --- |
|  |

**Read through the entire assignment carefully to ensure you have answered everything correctly and that your cover page is filled out.**

**If you have completed all the above, you have completed the assignment and should be ready to submit. Congratulations!**