| **Written Questions** | | **S / US** |
| --- | --- | --- |
| **Q1** | Did you find the requirements outlined for the application to be built and consult with your client (assessor) to confirm that the requirements are correct.  (check correct response)  **Yes  No** |  |
| **Q2** | Did you confirm that there are no applicable legislative or organisation standards or procedures that need to be applied to the application being built in this assessment.  (check correct response)  **Yes  No** |  |
| **Q3** | Determine the best web technology / protocol with GET, POST, PUT, DELETE and other related functionality with which to build your web based Restful API.  Which technology / protocol did you decide on? |  |
| **Q4** | Provide 2 advantages and 2 limitations of using the HTTP protocol when developing web applications. |  |
| **Q5** | Briefly describe the process you used to review and debug your API code as outlined in item 9 of the Task Summary above. |  |
| **Q6** | Did you Seek feedback on your work from your assessor.  The feedback demonstration will primarily be done in postman, however at least one API call must be demonstrated in each of 2 browsers and on at least 2 machines, (discuss with your assessor to clarify) Make updates based on the feedback your assessor provides.  (check correct response)  **Yes  No** |  |
| **Q7** | Briefly outline 3 principles of web analysis & design |  |
| **Q8** | Briefly discuss the features and functionalities of 2 different web applications |  |
| **Q9** | Briefly discuss programming control structures and applicable languages |  |
| **Q10** | Briefly discuss 2 code debugging techniques |  |
| **Q11** | Briefly discuss the web application development process |  |
| **Q12** | Briefly discuss the legislative and organisational requirements applicable to creating web applications |  |
| **Q13** | Briefly Discuss web programming concepts including:  o authentication and web security  o hypertext transfer protocol (HTTP)  o session management  o stateless programming |  |
| **Q14** | Briefly describe the different roles of the following web front end and backend languages, including:  o hypertext markup language (HTML)  o cascading style sheets (CSS)  o JavaScript |  |
| **Q15** | Briefly describe the work performed in web application with reference to its management of statelessness |  |
| **Q16** | Briefly describe organisational procedures applicable to creating web applications. |  |
| **Q17** | Which IDE (Integrated Development Environment) did you use to develop your API?  List 2 things that were good about the environment and 2 things that were bad about it. |  |
| **Q18** | Briefly describe benefits and functions of noSQL database and schema free data persistence, as well as traditional relational data models |  |
| **Q19** | Briefly describe methods and different features and functions between scaling out and scaling up (horizontal and vertical) |  |
| **Q20** | Briefly describe language used in required programming language for noSQL applications |  |
| **Q21** | Briefly describe partitioning in a noSQL environment and its related terms |  |
| **Q22** | Briefly describe functions and features for time-to-live (TTL) requirements |  |
| **Q23** | Briefly describe authorisation and authentications procedures and levels of responsibility according to client access requirements |  |
| **Q24** | Briefly describe distribution of data storage across partitions |  |
| **Q25** | Briefly describe debugging and testing methodologies and techniques |  |
| **Q26** | Briefly describe functions and features of sort keys in noSQL storage |  |
| **Q27** | Briefly describe features of transport encryptions, authentication and authorisation |  |
| **Q28** | Briefly describe different noSQL data store formats, including:  o key value  o document based  o column based  o graph based |  |
| **Q29** | Briefly describe different noSQL data types, including:  o numeric  o string  o boolean  o complex  o date time. |  |
| **Q30** | Briefly describe language used in object-oriented (OO) programming |  |
| **Q31** | Briefly describe HTTP protocol |  |
| **Q32** | Briefly describe functions and features of debugging and testing tools |  |
| **Q33** | Briefly describe principles of model view controller design pattern, conventions and architecture, including:  o scalability  o maintainability  o reusability |  |
| **Q34** | Briefly describe features, structures, logic and modes of interactions between models, controllers and views, including:  o HTTP Request/Response and redirects  o HTTP request handlers, routes and parameters  o Query strings and key/vale pairs  o model binding  o convention over configuration  o HTML language, templates and dynamic rendering  o view models and data models. |  |
| **Q35** | Briefly describe principles of database management systems applicable to deploying applications to production environments |  |
| **Q36** | Briefly describe software development life cycle (SDLC) that may be used in deploying applications to production environments |  |
| **Q37** | Briefly describe programming language used to create deployment applications |  |
| **Q38** | Briefly describe Information and Communications Technology (ICT) hardware, software, security protocols and standards and organisational policies relevant to deployment of applications. |  |
| **Q39** | Briefly describe design and build an advance UI design.  o In the course of the above the candidate must:  o determine an organisation’s technology, development tools, and UI platform  o apply advanced techniques in order to create a complex user interface (UI). |  |
| **Q40** | Briefly describe UI prototyping techniques and purpose. |  |
| **Q41** | Briefly describe data structures applicable to applying intermediate object-oriented language skills |  |
| **Q42** | Briefly describe object-oriented programming concepts and programming language required to apply intermediate object-oriented language skills |  |
| **Q43** | Briefly describe process and techniques related to use of a graphical user interface (GUI), to interact with an operator |  |
| **Q44** | Briefly describe documenting applications required to apply intermediate object-oriented language skills. |  |
| **Q45** | Briefly describe features and different applications that applies to the HTTP network protocol |  |
| **Q46** | Briefly describe features and anatomy of REST API HTTP request and response, including HTTP headers and body |  |
| **Q47** | Briefly describe language used in programming language |  |
| **Q48** | Briefly describe HTTP GET, POST, PUT and OPTIONS methods and features of each |  |
| **Q49** | Briefly describe CORS. |  |