| 304CEM Grading Rubric for Front-end (35%) | | | | | | |
|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 |
| Responsive 5% | No responsive design | An attempt for a responsive design but it is not working properly | Around half of the pages are responsive, or there are all responsive with major issues in each page | Most of the pages are responsive, with very few major issue with design such as wrong placement of large components | Fully responsive design, with minor issues, such as wrong placement of some small elements or extra shifts in some components | Fully responsive design UI |
| CSS 5% | No CSS used at all | Most of the CSS are working from the template. No real modification or use of its customised CSS | Half of the website is based on the original and customised CSS which heavily dependent on the template CSS | Most of the pages used original CSS, with major components throughout the pages used the template CSS | Well designed and original CSS, with some elements used from the template | Well designed, original CSS creation, completely different from the standard template |
| Front-end code Architecture 5% | No architecture | An attempt for a front-end code architecture was made, but it is not working properly, or it is mostly incomplete | Code architecture is not complete, with major flaws in some pages | Generally good architecture that needs improving in some places. | Good architecture with some minor flow in the design of the code | Clear architecture, clear separation of JS, CSS and contents |
| Front-end Coding (HTML & JS ) 10% | No code supplied through Github | An attempt has been made to write code to implement some of the basic functionality although this may not be successful. No attempt at documentation. | Working code base showing the application of basic programming principles. Code may contain linting errors and warnings. | Demonstration of the usage of modularity to organise the code. Code contains linting warnings but no errors. | The code is modular. Code contains no linting errors or warnings with minor errors in the code | Code contains no linting errors or warnings. |
| Completeness 5% | Zero achievement | At attempt has been made to create front-end for the requested API, however they still need major work and improvements | Have of the back-end services have been utilised in the front-end | Most of the requested backend APIs were used in the front-end, with some having minor or major issue | Most of the requested back-end APIs have front-end pages ready to serve them | All requested backend APIs have been utilised in the front-end |
| Connecting to APIs 5% | No connecting to APIs from the front-end | An attempt for connecting and retrieving data from the APIs was done but it is mostly having issues in connecting | half of the pages connecting and retrieving data from the APIs with few having minor or major issues in connecting | Most of the pages connecting and retrieving data from the APIs with few having minor or major issues in connecting | All pages connecting and retrieving data from the API with few having minor or major issues in connecting | All pages connecting and retrieving data from the API |

| 304CEM Grading Rubric for Backend API (35%) | | | | | | |
|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 |
| API Design 10% | No API demonstrated in the screencast | An attempt has been made to implement a basic API however this does not work as expected | Simple functional API demonstrating a basic understanding of REST principles (resources, collections, methods and headers) | The API is fully functional and includes an authentication mechanism. The API demonstrates a good understanding of REST principles. | The API demonstrates user registration and authentication. It provides feedback for invalid requests through appropriate response codes and messages. | Fully REST-compliant API that includes filtering and sorting and conditional GET requests. It makes use of the full range of request and response headers. |
| Architecture 5% | No code supplied through Github | An attempt to write the API however it fails to work correctly. | All code for routing and business logic maintained in a single file | Code split into several files but overlap between routing and business logic. | Clear separation between routing and business logic code. | Clear separation between routing and business logic code with no code duplication. |
| Coding 10% | No code supplied through Github | An attempt has been made to write code to implement some of the basic functionality although this may not be successful. No attempt at documentation. | Working code base showing the application of basic programming principles. Code may contain linting errors and warnings. No attempt at code documentation | Demonstration of the usage of modularity to organise the code. Code documentation is incomplete. Code contains linting warnings but no errors. | The code is modular and includes full exceptionhandling. Code is fully annotated and explained. Code contains no linting errors or warnings and is fully documented. | The API demonstrates a wide range of appropriate language constructs including clear modular structure. Code contains no linting errors or warnings and is fully documented. |
| Completeness 5% | No APIs were developed at all, or all of the created APIs are not working | Only few of the requested APIs were created, the exiting APIs may lack functionality | Half of the requested APIs were created and working properly, the existing APIs may not work or lack functionality | Most of the requested APIs were created, the existing APIs may not work or lack functionality | All requested API were created, with some of them may not working or lack functionality | All requested APIs were created and all working correctly |
| 3rd party API 5% | No 3rd party API call | An attempt to call other API but fail to display the result | Success to call one to two 3rd party API and cannot display the result properly | Success to call one to two 3rd party API and can display the result properly | Rich (more than two) 3rd party API call and related to the web content but the results cannot be displayed properly | Rich 3rd party API call related to web content and all results properly display |

| 304CEM Grading Rubric for Persistent Storage (20%) | | | | | | |
|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 |
| Persistent Storage Design 10% | No Persistent storage at all | An attempt has been made a persistent storage however it is not working or lacking most of the functionality | Persistent storage working, however there is major work needed to improve the design. | A working persistent storage design with some major issues | Well-designed persistent storage with few minor improvements needed to improved efficiency | Well design and fully functional persistent storage |
| CURD operations 10% | Hardcode data / no SQL operations | An attempt to use SQL statement but cannot produce correct results | Operations are correct but cannot interact with front end | Proper SQL statements to interact with database however the results cannot properly display in front end | Proper SQL statements to interact with database with inappropriate data type and can display partial data. | Proficiency using proper SQL statement with supporting data type to carry data e.g. JSON/Array and display correctly in front end |

| 304CEM Grading Rubric for Video or Live Application (10%) | | | |
|---|---|---|---|
| | 0 | 1 | 2 |
| Video 10% | No video submitted | A video was submitted in youtube or multimedia platform. However, the video has no narration or not present by the students. Or cannot demonstrate all the functionality of the website | A video was submitted in youtube or multimedia platform with student clear narration to present the works with fully demonstrating the functionality of the web site |