**Softwarica College of IT & E-Commerece**

##### STW304CEM Web API Development

**Assignment Brief 2019/20**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Module Title:  **STW304CEM Web API Development** | Ind/Group  **Indvidual** | | Cohort  **Sept 2019** | Module Code:  **STW304CEM** |
| Coursework Title (e.g. CWK1)  Developing a full stack web application | | | | Hand out date:  6 December 2019 |
| Lecturer  Achyut Timsina | | | | Due date:  12 February 2020 |
| Estimated Time (hrs):  Word Limit\*: n/a | | Coursework type: Individual / Practical | | % of Module Mark  100% |
| Submission arrangement online via Softwarica Moodle: Upload through Assignment links  File types and method of recording: URLs (source code repositories, live application or screencast)  Mark and Feedback date: Within 3 weeks of assignment submission  Mark and Feedback method: Rubric marks and comments via Softwarica LMS | | | | |

|  |
| --- |
| Module Learning Outcomes Assessed:   1. Develop a secure, open-standards-based API to support server-client communication. 2. Create modern web content involving asynchronous data retrieval, client-side DOM manipulation, standards adherence and user-user interaction. 3. Manage data persistence cross both server and client web-based solutions. 4. Design and implement an API and client based on given, non-trivial requirements using a range of appropriate developer tools. |
| Task and Mark distribution:   1. Persistent Storage: 10 % 2. Public Web API (Back End): 40 % 3. Front End Web Application: 45 % 4. Video recording: 5 % |
| Notes:   1. You are expected to use the [CUHarvard](https://curve.coventry.ac.uk/open/file/bdfb947c-9d43-48d3-8ec8-f511682e1dd1/1/The%20CU%20Guide%20to%20Referencing%20in%20Harvard%20Style.pdf) referencing format. For support and advice on how this students can contact [Centre for Academic Writing (CAW)](http://www.coventry.ac.uk/study-at-coventry/student-support/academic-support/centre-for-academic-writing/?theme=main). 2. Please notify your registry course support team and module leader for disability support. 3. The University cannot take responsibility for any coursework lost or corrupted on disks, laptops or personal computer. Students should therefore regularly back-up any work and are advised to save it on the University system. 4. If there are technical or performance issues that prevent students submitting coursework through the online coursework submission system on the day of a coursework deadline, an appropriate extension to the coursework submission deadline will be agreed. This extension will normally be 24 hours or the next working day if the deadline falls on a Friday or over the weekend period. This will be communicated via email and as a Softwarica Moodle announcement. |

Assignment Brief

You need to develop a full stack, RESTful web application for this assignment. You can decide yourself the exact requirements of your assignment. We strongly encourage you to find real client(s) to gather real requirements for this project. You will need to do following four tasks for this assignment:

1. **Persistent Storage (10%):**

Create a persistent data storage for SQL or NoSQL databases and write short report about its structure or design. You should explain and justify your database design and chosen technology in this short report (not more than 1000 words) and submit it through Softwarica LMS.

1. **Backend System (40%):**

A backend system in a form of full RESTful public web API using NodeJS and Express. Full source code of your project should be hosted on a git repository and a link to your repository should be submitted on Softwarica LMS. It should have following features:

* User registration and login
* At least two create, update, delete and retrieve methods on the backend system in addition to login and registration (at least 8 different API endpoints)
* All source code should be fully unit tested using appropriate unit testing framework
* All source code should be suitably documented

1. **Front-end System (45%):**

A web front end using HTML, CSS, Bootstrap and jQuery. You may also use ReactJS or AngularJS or Vue.JS in the front end for more extra credits and challenge to yourself.

Your front end should communicate with your backend to read and write data. This application should at least allow the user to register, login and perform insertion, update, deletion and retrieval of relevant entities in the application. Full source of your front end should be hosted on a git repository.

A link to your git repository should be submitted on Softwarica LMS.

1. **Video Screencast (5%):**

You must show appropriate evidence of the project by submitting a link to your video screencast of the finished application. The video should be in 720p, mp4 format and not longer than 10 minutes. It should be hosted on YouTube and published as “unlisted”, and the link should be submitted on Softwarica LMS.

STW304CEM Grading Rubric for Backend API (40%)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | 0 |  |  | 1 |  |  | 2 |  |  | 3 |  |  | 4 |  |  | 5 |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | **Version** | | | |  | No access given to | |  | Code access provided but | |  | Access provided to GitLab | |  | Access has been given to | |  | Evidence of regular commits | |  | Commits over an extended | |
|  | **Control** | | | |  | the remote | |  | does not match the code | |  | however there is no | |  | the code on GitLab with | |  | over an extended period of | |  | period of time, demonstrating | |
|  | **5%** |  |  |  |  | repository on | |  | demonstrated. | |  | evidence of regular | |  | evidence of regular | |  | time. | |  | the use of branching and | |
|  |  |  |  |  |  | GitLab. | |  |  |  |  | commits. | |  | commits. | |  |  |  |  | merging. | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | **Automated** | | | |  | No tests have | |  | Limited attempt at flawed | |  | An attempt has been made | |  | Evidence of a limited | |  | A range of tests showing how | |  | A full suite of automated tests | |
|  | **Testing** | | | |  | been written or | |  | tests. | |  | to write simple tests | |  | number of tests written | |  | they contribute to code | |  | ensuring full code coverage | |
|  | **5%** |  |  |  |  | run. | |  |  |  |  |  |  |  | and run. | |  | quality. | |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | **API Design** | | | |  | No API | |  | An attempt has been made | |  | Simple functional API | |  | The API is fully functional | |  | The API demonstrates user | |  | Fully REST-compliant API that | |
|  | **10%** | |  |  |  | demonstrated in | |  | to implement a basic API | |  | demonstrating a basic | |  | and includes an | |  | registration and | |  | includes filtering and sorting | |
|  |  |  |  |  |  | the screencast | |  | however this does not work | |  | understanding of REST | |  | authentication | |  | authentication. It provides | |  | and conditional GET requests. | |
|  |  |  |  |  |  |  |  |  | as expected | |  | principles (resources, | |  | mechanism. The API | |  | feedback for invalid requests | |  | It makes use of the full range of | |
|  |  |  |  |  |  |  |  |  |  |  |  | collections, methods and | |  | demonstrates a good | |  | through appropriate response | |  | request and response headers. | |
|  |  |  |  |  |  |  |  |  |  |  |  | headers) | |  | understanding of REST | |  | codes and messages. Limited | |  | Full HATEOAS implementation. | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | principles. | |  | HATEOAS support. | |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | **Architecture** | | | |  | No code supplied | |  | An attempt to write the API | |  | All code for routing and | |  | Code split into several | |  | Clear separation between | |  | Clear separation between | |
|  | **5%** |  |  |  |  | through GitLab | |  | however it fails to work | |  | business logic maintained | |  | files but overlap between | |  | routing and business logic | |  | routing and business logic | |
|  |  |  |  |  |  |  |  |  | correctly. | |  | in a single file | |  | routing and business | |  | code. | |  | code with no code duplication. | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | logic. | |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | **Coding** | | | |  | No code supplied | |  | An attempt has been made | |  | Working code base | |  | Demonstration of the | |  | The code is modular and | |  | The API demonstrates a wide | |
|  | **10%** | |  |  |  | through GitLab | |  | to write code to implement | |  | showing the application of | |  | usage of modularity to | |  | includes full exception- | |  | range of appropriate language | |
|  |  |  |  |  |  |  |  |  | some of the basic | |  | basic programming | |  | organise the code. Code | |  | handling. Code is fully | |  | constructs including clear | |
|  |  |  |  |  |  |  |  |  | functionality although this | |  | principles. Code may | |  | documentation is | |  | annotated and explained. | |  | modular structure. Code | |
|  |  |  |  |  |  |  |  |  | may not be successful. No | |  | contain linting errors and | |  | incomplete. Code contains | |  | Code contains no linting | |  | contains no linting errors or | |
|  |  |  |  |  |  |  |  |  | attempt at documentation. | |  | warnings. No attempt at | |  | linting warnings but no | |  | errors or warnings and is fully | |  | warnings and is fully | |
|  |  |  |  |  |  |  |  |  |  |  |  | code documentation | |  | errors. | |  | documented. | |  | documented. | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | **Completeness** | | | |  | No APIs were | |  | Only few of the requested | |  | Half of the requested APIs | |  | Most of the requested | |  | All requested API were | |  | All requested APIs were | |
|  | **5%** |  |  |  |  | developed at all, or | |  | APIs were created, the | |  | were created and working | |  | APIs were created, the | |  | created, with some of them | |  | created and all working | |
|  |  |  |  |  |  | all of the created | |  | exiting APIs may lack | |  | properly, the existing APIs | |  | existing APIs may not | |  | may not working or lack | |  | correctly | |
|  |  |  |  |  |  | APIs are not | |  | functionality | |  | may not work or lack | |  | work or lack functionality | |  | functionality | |  |  |  |
|  |  |  |  |  |  | working | |  |  |  |  | functionality | |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

STW304CEM Grading Rubric for Front-end (45%)

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

STW304CEM Grading Rubric for Persistent Storage (10%)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  | 0 |  |  | 1 |  |  | 2 |  |  | 3 |  |  | 4 |  |  | 5 |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |
|  | **Persistent** | | | |  |  | No Persistent | |  | An attempt has been made a persistent | |  | Persistent storage working, | |  | A working persistent | |  | Well-designed persistent storage with | |  | Well design and fully | |
|  | **Storage** | | |  | |  | storage at all | |  | storage however it is not working or | |  | however there is major work | |  | storage design with some | |  | few minor improvements needed to | |  | functional persistent | |
|  | **Design** | | | | |  |  |  |  | lacking most of the functionality | |  | needed to improve the design. | |  | major issues | |  | improved efficiency | |  | storage | |
|  | **10%** | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

STW304CEM Grading Rubric for Video Screencast (5%)

|  |  |  |
| --- | --- | --- |
|  | 0 | 1 |
| **Video (5%)** | No video submitted | A video was submitted |