Faculty of Engineering, Environment and Computing
6003CEM Web API Development

Assignment Brief - Coursework 2 of 2

Module Title **6003CEM Web API Development**	Individual / Group **Individual**	Cohort **AY2022/2023**	Module Code **6003CEM**
Coursework Title **RESTful API Full Stack Software Portfolio**			Hand out date **2023.02.13(FT)** **2022.02.01(PT)**
Lecturer **CY CHENG (Full Time)** **HC WONG (Part-Time)**			Due date and time Date: **2023.06.12 (FT)** Online: **23:59:59** **2023.05.30 (PT)** Online: **23:59:59**
Estimated Time (hrs) **85hrs**	Coursework type: **Code and Documentation Portfolio**		% of Module Mark / Credit value assessed: **75% / 15 CATS credits**

Submission arrangement: **Online via Moodle, GitHub, and YouTube/MS Stream**

File types and method of recording: **file upload (TXT, DOCX, PDF) to Moodle containing URL links to**

- **Demonstration video on YouTube (unlisted)**
- **Git repository x2**

Mark and Feedback date: **2023.07.12(FT)**;**2023.06.30(PT)**
Mark and Feedback method: **Moodle**

Module Learning Outcomes (LOs) Assessed:

	Learning Outcome	Assessed?
1	Demonstrate systematic knowledge of the current state of the art in web technologies being deployed by mainstream businesses to create Web APIs.	**Yes**
2	Demonstrate comprehensive, critical awareness of the research basis of those technologies.	**Yes**
3	Scope, design and implement a simple Web API to solve a given problem.	**Yes**
4	Describe how and why Web APIs are more important to business than building web applications from first principles each time one is required.	**Yes**
5	Understand and implement Web API security and authentication.	**Yes**

Purpose

This assessment is designed to allow you to demonstrate your knowledge and critical awareness of the various HTTP-based API techniques that are available for web application development by building a working RESTful API and a secure web client for it. A scenario is provided through which you can illustrate your technical skills and knowledge in API design and development by completing the following.

- Build a secure RESTful API using Node JS libraries.
- Build a secure web client for the API using the React JS framework.
- Design an industry-standard OpenAPI/Swagger Specification (OAS) that fully documents your API resources, representations, operations, and URI endpoints in a machine-readable format.
- Provide a suite of functional API endpoint tests containing a variety of mock HTTP requests that verify your API works according to the provided AOS specification (including HTTP request and response header and body data).
- Showcase your working full-stack application in a short demonstration video.

While the code implementations needed to create the front and back-end components in your application are the key elements of this assessment, to gain marks, you **must** also address the correct design, documentation and testing approaches that are appropriate for commercial API development.

For an explanation of how the marks are awarded, see the grading rubric at the end of this document.

Scenario

You work for a company that specialises in providing small-scale full-stack web applications to small and medium-sized enterprises (SMEs) and charities. You have been approached by a charity, "The Pet Shelter", who wish to develop a web application for matching shelter cats with new owners and arranging visits to shelter locations to meet the cats. The charity has locations in several parts of the city, and there will be a staff member at each location who uses the app to input and manage details of the current cats available at their physical site. The public should be able to browse the application and easily search and filter the cats shown. Prospective owners should be able to create a user account in the application, create a list of favourites, and directly message the relevant shelter employee to ask for details of any particular cat they are interested in.

The charity does not have an internal IT department, and they have contracted your company to provide a minimum viable product meeting with the requirements listed above.

Assessment Task

Following a brief feasibility study, your manager has tasked you with producing a **RESTful API and React SPA** prototype for the client. It should include **appropriate API and code documentation and endpoint tests** to ensure its future maintainability and robustness. They have also asked you to record a **short demonstration video** for the client to showcase the functionality available.

In order of importance, your manager has identified that the developed software project should have the following features. Details of the implementation are up to you, based on your expectation of user needs.

1. (Essential)
   - Charity workers can register, log in, then add, remove and update details of the cats available at their centre. Hint: consider requiring a known *'sign up code'* during registration to confirm that the user is an employee.
   - The public can browse, search and filter the current list of cats to help them find a suitable pet.
   - Authentication and authorisation prevent the API being used by non-registered users, except to do safe GET requests on cat listings.
   - All URI endpoints, HTTP verbs, and JSON data representations handled by the API are documented using the OpenAPI Specification (OAS) standard, version ≥2.0.
   - API endpoint functionality is thoroughly tested via a mock HTTP request library called by an appropriate automated testing framework.
   - Code and project documentation for both the front and back-end components are provided.
   - A video demonstrating the core functionality of the developed API/web application is provided.

2. (Important)
   - The API allows charity workers to upload photos to be associated with each listing.
   - Members of the public can sign up for a user account to let them create a list of 'favourites'.
   - The public can use the app to send a direct message to the charity expressing interest in any cat on the site. The charity can respond to the message through the app and can delete any message sent or received.

3. (Useful)
   - When a new cat listing is made 'live' the API automatically posts a tweet to the charity's social media (e.g. Xiaohongshu) feed, with basic details of the new item.
   - Members of the public can register their accounts using an external authentication service. For example, their Google credentials can be used to log in via Google's OAuth2-based sign-in process.

4. (Nice to have)
   - Since some shelter staff are not very good at identifying cat breeds, but a cat's breed is an important property of its listing, the cat listing and editing interface offers a 'breed selection helper'. This uses the famous ‘Cats as a Service (CaaS)’ API to access cat images from the Cat API Dataset: https://thecatapi.com/
   - Any other useful features that you come up with yourself.

Portfolio Submission Process

The following steps must be completed before the deadline, in order to submit a portfolio of work that provides evidence you have developed an application that meets the requirements.

1. **Video - MS Stream / YouTube submission**
   - Upload a maximum 5-minute video demonstration of your API/application to YouTube.
   - The video should include text overlays or a slide deck to explain to the viewer what is being demonstrated. *No extra audio should be added to the video.*
   - Copy the video sharing link for submission to Moodle.

2. **Backend API, OpenAPI specification, and API endpoint tests**
   - Commit and push these to a repository in GitHub. You may share directly from https://repl.it to GitHub. Please check the username of your module leader and share the repo to him.

3. **Frontend React JS single page application (SPA) code**
   - Commit and push the SPA to a single repository in GitHub.
     - **o This must be different from your API repository above.**

4. **Portfolio Links - Moodle submission**
   - The following URLs must link to evidence that you have completed the tasks above.
   - Create a new text file and add the **URL links** to each of the above portfolio items to it, **with a title to say what each link is for**:
     1. Sharing link for the uploaded demo video on YouTube
     2. Backend repository link on GitHub
     3. Frontend repository link on GitHub
   - So, in total, your Moodle submission should be a file containing **THREE URL links**.
   - Upload the file to the assessment link on Moodle before the deadline.

Note: Any file updates after the timestamp on Moodle submission will not be marked.

Constraints

Backend

All the backend services you develop should provide a **Node JS-based, Typescript-based or Deno-based RESTful API.** This must communicate with your application using the **JSON data format** by default, but you can provide other representations if you wish. The frameworks and modules you choose to build the backend are up to you: options include Koa, Express, Restify, and others.

Database

Similarly, the choice of database integration on the backend is up to you: options include MySQL, SQLite, MongoDB, and others.

Frontend

All the frontend components you develop should be **React (JS or TS)** components, **other frontend libraries (Vue, Angular, etc.) are not permitted for this assessment**. In terms of visual design the CSS framework you choose, if you decide to use one, is up to you: options include React Bootstrap, Ant Design, Material UI, and others.

If in doubt as to the suitability of a particular framework, Node module, or DBMS please ask the Module Leader first.

Notes:1. You are expected to use the Coventry University APA style for referencing.
2. Please notify your registry course support team and module leader for disability support.
3. Any student requiring an extension or deferral should follow the university process outlined here.

4. The University cannot take responsibility for any coursework lost or corrupted on disks, laptops or personal computers. Students should, therefore, regularly back-up any work and are advised to save it on the University system.
5. Assignments that are more than 10% over the word limit will result in a deduction of 10% of the mark i.e. a mark of 60% will lead to a reduction of 6% to 54%. The word limit includes quotations but excludes the bibliography, reference list and tables.
6. You are encouraged to check the originality of your work by using the draft Turnitin links on Moodle.
7. Collusion between students (where sections of your work are similar to those submitted by other students in this or previous module cohorts) is taken extremely seriously. It will be reported to the academic conduct panel. This applies to both coursework and exam answers.
8. A marked difference between your writing style, knowledge and skill level demonstrated in class discussion, test conditions, and that demonstrated in a coursework assignment may result in you having to undertake a Viva Voce to prove the assignment is entirely your work.
9. If you use the services of a proof-reader in your work, you must keep your original version and make it available as a demonstration of your written efforts. Also, please read the university Proof Reading Policy.
10. You must not submit work for assessment that you have already submitted (partially or in full), either for your current course or for another qualification of this university, except resits, where for the coursework, you may be asked to rework and improve a previous attempt. This requirement will be specifically detailed in your assignment brief or specific course or module information. Where earlier work by you is citable, i.e. it has already been published/submitted, you must reference it. Identical pieces of work submitted concurrently may also be self-plagiarism.

Mark allocation guidelines for: RESTful API Full Stack Software Portfolio.

Part	Topic	Description / Breakdown	Marks
Backend API	Codebase	**Architecture, Code Quality, Security, Data Validation, External Integrations** The client's essential requirements are met (for pass) and important/useful/nice-to-have requirements are met (for higher grades) • Clear separation between routing, business, and persistence logic • Pluggable modular structure. • Wide range of appropriate language constructs including clear modular structure and use of abstract design patterns or paradigms such as functional programming. • Concise easily maintainable code. • Full or close-to-full exception handling including for asynchronous code. • Fully working authentication system using Basic auth (pass) plus JWT or more advanced techniques (for higher grades) • Correct configuration of Cross-Origin Resource Sharing (CORS) to limit API access to known hosts. • Includes effective user permissions and/or roles such as RBAC to provide appropriate protection to data. • Includes filtering and sorting capability for requested resources. • Data integrity is managed through effective validation using the JSON Schema standard or similar. • Includes implementation logic for conditional HTTP requests based on appropriate conditional request/response headers. • Implemented route behaviour corresponds to the OpenAPI Specification provided. • Includes working external API integration(s) for access to external sources of application data.	20
	OpenAPI	**API Design and** The client's essential requirements are met (for pass) and important/useful/nice-to-have requirements are met (for higher grades) • The API design demonstrates a good understanding of REST principles (resources, collections, methods and headers). • Fully REST-compliant API including implementation of HATEOAS principles. • Overall API design captured in a machine-readable format such as an OpenAPI Specification document. • Effective JSON representations of resources recorded in appropriate machine-readable format such as JSON Schema standard or OpenAPI Specification. • OAS and JSON Schema(s) provide full or close-to-full coverage of the resources and methods actually implemented in the codebase. • API is designed to provide feedback for invalid requests through appropriate response codes and messages. • Includes provision for conditional HTTP requests based on appropriate conditional request/response headers. • Specification is accessible in human-readable HTML format via SwaggerHub or similar OAS service.	10
	API Testing	**API Resource Endpoint** The client's essential requirements are met (for pass) and important/useful/nice-to-have requirements are met (for higher grades) • Many or most of the API endpoints have been tested with a standard test framework. • Tests are provided for both valid and invalid HTTP requests to the API. • Tests include situations requiring authentication by the API for successful responses. • The test framework or NPM project uses setup and teardown correctly to normalise each test. • A dedicated test database is used, and the production database is not touched during testing.	10
	Maintenance	**Versioning, Code and Project Documentation, Unit Tests** The client's essential requirements are met (for pass) and important/useful/nice-to-have requirements are met (for higher grades) • Evidence of regular Git commits over an extended period of time. • Effective use of branching and merging. • Code is well documented using a standard method such as JSDoc for documentation generation. • Complex functions have tests defined. • Clear instructional project documentation is provided alongside code files: setup, configuration, and running the project are explained to other developers.	5
		TOTAL	**45**

Part	Topic	Description / Breakdown	Marks
Frontend Client	Codebase	**Architecture, Code Quality, Security, Data Validation, API Access** The client's essential requirements are met (for pass) and important/useful/nice-to-have requirements are met (for higher grades) • Clear architecture and component hierarchy enabling effective flow of data/communication between all interacting components (for example raising state or context objects). • Wide range of appropriate language constructs including clear modular structure and use of abstract design patterns or paradigms such as functional programming. • Concise easily maintainable code. • Full or close-to-full exception handling including for asynchronous code. • Effective API credential management using contexts, browser storage capabilities, or other methods to allow 'login'. • Data integrity is managed through validation before sending or displaying data. • All pages connect and retrieve data from the API • Appropriate error handling in place for unusual situations such as timeouts etc. • Conditional HTTP requests are made when appropriate.	10
	Design	**User Interface Design and User Experience (UI/UX)** The client's essential requirements are met (for pass) and important/useful/nice-to-have requirements are met (for higher grades) • Sophisticated layout and design reflecting a well-thought-out application. • Good UX including intuitive guidance to users through layout and/or text/colour changes • Effective navigation structure between 'pages' in the single page application • Uses conditional rendering effectively to show/hide appropriate UI components depending on the current context • Provides user feedback through flash messages or modal popups when needed and appropriate. • Smooth navigation with quick feedback. • Accessible design that would work on mobile with no additional changes.	10
	Maintenance	**Versioning, Code and Project Documentation, Unit Tests** The client's essential requirements are met (for pass) and important/useful/nice-to-have requirements are met (for higher grades) • Evidence of regular Git commits over an extended period of time. • Effective use of branching and merging. • Code is well documented using a standard method such as JSDoc for documentation generation. • Complex functions have tests defined. • Clear instructional project documentation is provided alongside code files: setup, configuration, and running the project are explained to other developers.	5
		TOTAL	**25**

Part	Topic	Description / Breakdown	Marks
Demonstration Video	Video	**Quality Video to Clearly Demonstrate Functionality** The client's essential requirements are met (for pass) and important/useful/nice-to-have requirements are met (for higher grades) • Professional level demonstration giving a clear idea of how the application works and what it is capable of. • Application is demonstrated smoothly. • Appropriate text narrative takes the user on a journey through the application (text overlays or slide deck). • No sound is added to the video.	5
		TOTAL	**5**

General guidance on marking bands

Mark band	Outcome	Guideline
90-100% 1st	Meets learning outcomes	1st - Exceptional work with very high degree of understanding, creativity and critical/analytic skills. Evidence of exceptional research well beyond minimum recommended using a range of methodologies. Exceptional understanding of knowledge and subject-specific theories. Demonstrates creative flair, a high degree of originality and autonomy. Exceptional ability to apply learning resources. Demonstrates well-developed problem-solving skills. Work completed with very high degree of accuracy and proficiency and autonomy. Exceptional communication and expression, significant evidence of professional skill set. Student evidences deployment of a full range of exceptional technical and/or artistic skills.
80-89% 1st		1st - Outstanding work with high degree of understanding, creativity and critical/analytical skills. Outstanding understanding of knowledge and subject-specific theories. Evidence of outstanding research well beyond minimum recommended using a range of methodologies. Demonstrates creative flair, originality and autonomy. Outstanding ability to apply learning resources. Demonstrates clear problem-solving skills. Assessment completed with high degree of accuracy and proficiency and high-level of autonomy. Outstanding communication and expression, evidence of professional skill set. Student evidences deployment of a full range of technical and/or artistic skills.
70-79% 1st		1st - Excellent work with clear evidence of understanding, creativity and critical/analytical skills. Thorough research well beyond the minimum recommended using methodologies beyond the usual range. Excellent understanding of knowledge and subject- specific theories with evidence of considerable originality and autonomy. Excellent ability to apply learning resources. Demonstrates consistent, coherent substantiated argument and interpretation. Demonstrates considerable creativity and clear problem-solving skills. Assessment completed with accuracy, proficiency, and considerable autonomy. Excellent communication and expression, some evidence of professional skill set. Student evidences deployment of a highly developed range of technical and/or artistic skills.
60-69% 2:1		2:1 - Very good work demonstrating strong understanding of theories, concepts and issues with clear critical analysis. Thorough research, using established methodologies accurately, beyond the recommended minimum with little, if any, irrelevant material present. Very good understanding, evidencing breadth and depth, of knowledge and subject-specific theories with some originality and autonomy. Very good ability to apply learning resources. Demonstrates coherent substantiated argument and interpretation. Demonstrates some originality, creativity and problem-solving skills. Work completed with accuracy, proficiency, and autonomy. Very good communication and expression with evidence of professional skill set. Student has a thorough command of a good range of technical and/or artistic skills.
50-59% 2:2		2:2 - Good understanding of relevant theories, concepts and issues with some critical analysis. Research undertaken accurately using established methodologies, enquiry beyond that recommended may be present. Some errors may be present and some inclusion of irrelevant material. Good understanding, with evidence of breadth and depth, of knowledge and subject-specific theories with indications of originality and autonomy. Good ability to apply learning resources. Demonstrates logical argument and interpretation with supporting evidence. Demonstrates some originality, creativity and problem-solving skills but with inconsistencies. Expression and presentation mostly accurate, proficient, and conducted with some autonomy. Good communication and expression with appropriate professional skill set. Student consistently demonstrates a well-developed range of technical and/or artistic skills.

40-49% 3rd Class		3rd - Meet the learning outcomes with a basic understanding of relevant theories, concepts and issues. Demonstrates an understanding of knowledge and subject-specific theories sufficient to deal with concepts. Assessment may be incomplete and with some errors. Research scope sufficient to evidence use of some established methodologies. Some irrelevant material likely to be present. Basic ability to apply learning resources. Demonstrates ability to devise and sustain an argument. Demonstrates some originality, creativity and problem-solving skills but with inconsistencies. Expression and presentation sufficient for accuracy and proficiency. Sufficient communication and expression with basic professional skill set. Student demonstrates technical and/or artistic skills.
30-39% Fail	Fails to achieve learning outcomes	Fail – Very limited understanding of relevant theories, concepts and issues. Little evidence of research and use of established methodologies. Some relevant material will be present. Deficiencies evident in analysis. Fundamental errors and some misunderstanding likely to be present. Limited ability to apply learning resources. Student's arguments are weak and poorly constructed. Very limited originality, creativity, and struggles with problem-solving skills. Expression and presentation insufficient for accuracy and proficiency. Insufficient communication and expression and with deficiencies in professional skill set. Student demonstrates some deficiencies in technical and/or artistic skills.
20-29% Fail		Fail - Clear failure demonstrating little understanding of relevant theories, concepts and issues. Minimal evidence of research and use of established methodologies and incomplete knowledge of the area. Serious and fundamental errors and aspects missing Little evidence of ability to apply learning resources. Students arguments are very weak and with no evidence of alternative views. Little evidence of originality, creativity, and problem-solving skills. Expression and presentation deficient for accuracy and proficiency. Insufficient communication and expression and with deficiencies in professional skill set. Student demonstrates a lack of technical and/or artistic skills.
0-19% Fail		Fail - Inadequate understanding of relevant theories, concepts and issues. Complete failure, virtually no understanding of requirements of the assignment. Material may be entirely irrelevant. Assessment may be fundamentally wrong, or with major elements missing. Not a serious attempt. No evidence of research. Inadequate evidence of ability to apply learning resources. Very weak or no evidence of originality, creativity, and problem- solving skills. Students presents no evidence of logical argument and no evidence of alternative views. Expression and presentation extremely weak for accuracy and proficiency. Communication and expression very weak and with significant deficiencies in professional skill set. Student evidences few or no technical and/or artistic skills