Professional Diploma in Cross-platform Applications Development / Professional Certificate in Stack Web Development / Professional Diploma in Web3 Applications Development

Project Work (Part I) – ITP4511Q Modern APIs Development

Deadline

The deadline for this assignment is **5 April 2024 (23:59)**.

Purpose

This assessment will demonstrate your knowledge and critical awareness of the various HTTP-based API techniques available for web application development by building a working RESTful API. 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 NodeJS/TypeScript libraries.
- 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 in a short demonstration video.

While the code implementations needed to create the back-end components in your application is 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.

Scenario

Wanderlust Travel is a new travel agency that aims to provide customers with a seamless platform to explore hotels and/or air tickets for their travel plans. The agency wants to launch a modern travel query and reservation system with a React-based Single Page Application (SPA) for users and a RESTful API backend to manage data and integrate with public APIs for hotels and air ticket information.

Free APIs suggested:
1. https://developer.hotelbeds.com/ - Hotelbeds
2. https://rapidapi.com/Travelpayouts/api/flight-data - Flight Data by RapidAPI

Assessment Task

Following a brief feasibility study, your manager has tasked you with producing a **TypeScript RESTful API and React TS 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 available functionality.

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

expectations of user needs.

1. (Essential)
   - The travel agency operator can register, log in, and then add, remove and update hotel availability details. 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 hotels to help them find a suitable pet.
   - Authentication and authorisation prevent the API from being used by non-registered users, except to do safe GET requests on dog 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 the front-end 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 users to upload profile photos.
   - Members of the public can sign up for a user account to let them create a list of 'favourites'.
   - The registered can use the app to send a direct message to the travel agency operator expressing interest in any hotel on the site. The operator can respond to messages through the app and can delete any message sent or received.

3. (Useful)
   - When a new listing is made 'live', the API automatically posts a message to the operator's social media feed (such as Facebook / Tweeter), with basic details of the new item.
   - Coding style and Git repository can made the highest possible to avoid as much as security issues.
   - 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. **(Note: Travel Agency operator cannot register their account as *operators* with any external authentication service. If implemented, it will be claimed as a security issue, and NO mark will be given for "useful" functions)**

4. (Nice to have)
   - Using external API to get the hotels' information.
   - Provide a Hotel and Flight package service
   - Any other useful features that you come up with yourself.

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 6-minute video demonstration of your API/application to YouTube.
   - The video should **include a voice description with text overlays or a slide deck** to explain what is being demonstrated to the viewer. *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**s

3. **Portfolio Links - Moodle submission**
   - The following URLs must link to evidence that you have completed the tasks above.
   - ZIP only with the structure of the TypeScript Project without any code not developed by you. Add the demonstration **URL links** to the readme file of the project.
   - Upload the ZIP file to Moodle before the deadline.

Note:

1. **Any file updates after the timestamp on Moodle submission will not be marked.**
2. **No Full marks policy: Completing a function does not mean you can get full marks. Marking criteria include coding style, project coding structure, etc which may affecting the marking.**

Constraints

Back-end

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

Database

You are required to connect the MongoDB or local JSON.