For: ATHE Level 4 Extended Diploma in Computing

Unit 14 The Principles of Full-Stack Development Assignment

Note to Learners

There is no need to implement this in a Production environment, i.e. on a "live" external, public-facing website; a development server and environment are acceptable for creating, documenting, and testing your work. You tutor will help you set this up if necessary.

Assignment Scenario

You have been working as a developer for a medium-sized software development studio that specialises in full-stack development for commercial web solutions in the large municipality of Centrala.

The Centrala Environmental Agency (CEA) has recently approached the studio to help with the development of a web-based API and associated website for tracking the rainfall measured across the city and its immediate suburban areas.

Many regional universities and academics involved in studying climate change and its environmental effects have requested access to CEA weather data and it has been suggested that a user-friendly website portal with documented web-based API endpoints may be the easiest way to access the data.

As part of the proposals, the CEA has outlined the following functionality requirements and provided sample data for testing your solution.

Centrala rainfall data is taken from 18 different geographical locations scattered across the city:

Erean	Brunad	Bylyn	Docia	Marend	Pryn
Zord	Yaean	Frestin	Stonyam	Ryall	Ruril
Keivia	Tallan	Adohad	Obelyn	Holmer	Vertwall

All rainfall is measured in millimetres (mm), rounded to one decimal place.

Each location listed has a rainfall sensor station that uploads measurements to a central data source every 6 hours, so a single day will have 4 different readings.

After discussion with stakeholders, it is agreed that the following API requests are required:

- Rainfall for a specified day (all locations).
- Rainfall for a single specified location (all days).
- Rainfall for a specified day for a single specified location.
- Rainfall for all days, for all locations.

Although the service and data is not intended to be charged for, i.e. it is open source in nature, the CEA are interested in its usage and access so have asked that:

- All data users are registered, resulting in the issue of a free API key.
- All API data requests include the use of the API key.
- All data requests are logged with remote IP access, client user agent string, date, and time.
- A special “admin”-only API key permits access to non-documented API requests that allow addition, amendment and deletion of all 4 data entries specified by day, location.

In addition, the CEA has sought your professional judgement on the creation of the various API endpoints, preferred request and response formats, and the underlying technologies and tooling being used to create the solution.

The CEA has requested that as part of your solution, you should include full documentation on the design process, highlighting the various full-stack options available, your studio’s final decisions and selection justification.

Your solution must then be built to meet the client specification and fully tested. This includes hardening it against potential threats, with a final evaluation completing the project that evaluates it against the CEA brief and industry best practice.

Sample data – Comma Separated Value (CSV) format

```
"area","date","am1","am2","pm1","pm2"
"Erean","01/10/2023",2.3,3.0,1.2,1.0
"Erean","02/10/2023",4.5,6.0,4.1,4.0
"Erean","03/10/2023",2.1,1.0,5.0,4.0
"Erean","04/10/2023",0.0,3.0,0.0,2.0
"Erean","05/10/2023",1.0,6.0,1.2,1.3
```

Task 1

Before starting the development for the CEA, Luzo Okake, CEO of the studio has asked you to prepare a briefing document covering the various full-stack development options that must be considered before design and development work can formally start.

This process will help you to review and select the appropriate full-stack of technologies to use for the project.

This document should cover:

- A description of the common components used in a full-stack solution.
- An explanation of the key differences between commercial-grade full-stack solutions.
- An explanation of the functional operation of a web-based API.

Extension activities:

To achieve a **Merit**, Luko wants you to provide additional content that:

- Describes the advantages that cloud-based solutions using SOA have over monolithic solutions.

Learning outcomes and assessment criteria

LO1 1.1, 1.2, 1.3,1M1

Task 2

Now that the stack has been selected, design and implementation of the solution is ready to begin.

Design and implement the following elements:

- Endpoint URLs for each required element of the API (both open access and Admin only).
- Validation of the API keys.
- Storage of the rainfall data in an appropriate backend database.
- Design (and samples) for each Endpoint's request and associated response.
- Design and implementation of the website portal where stakeholders may freely register for an API key.
- Logging of user requests for archive and monitoring purposes.

Remember: There is no need to implement this in a Production environment, i.e. on a "live" external, public-facing website; a development server and environment are acceptable for creating, documenting, and testing your work.

Extension activities:

To achieve a **Merit**, Luko wants you to provide additional content that:

- Justifies the use of the selected full-stack technologies used in your solution.

Learning outcomes and assessment criteria

LO2 2.1, 2.2, 2M1

Task 3

The final task is to identify and describe common vulnerabilities and threats to web-based applications such as CEA's rainfall website portal and API.

Once these vulnerabilities and threats have been suitably documented, you should protect the backend code and data from these using industry recommended best practice for configuration and coding techniques.

After code and backend services have been protected, it is necessary to test both the website and API thoroughly to ensure common threats are suitably neutralised by your hardening process.

Extension activities:

To achieve a **Merit**, Luko wants you to provide additional content that:

- Justifies the decisions made to harden the web-based application.

To achieve a **Distinction**, you must:

- Evaluate your full-stack solution against the original client brief and the industry best practice you have researched.

Learning outcomes and assessment criteria

LO3 3.1, 3.2, 3.3, 3M1, 3D1

Guidelines for assessors

The assignments submitted by learners must achieve the learning outcomes at the standards specified by the assessment criteria for the unit. To achieve a merit or distinction grade, the learners must demonstrate that they have achieved all the criteria set for these grades. Where work for the pass standard is marginal, assessors can take into account any extension work completed by the learners. The suggested evidence listed below is how learners can demonstrate that they have met the required standards. The command verbs contained within the AC are highlighted in bold in the suggested evidence boxes.

Task number	LOs and AC	Suggested evidence PASS	Suggested additional evidence MERIT	Suggested additional evidence DISTINCTION
1.	LO1 1.1, 1.2, 1.3, 1M1	Task 1 focuses on the theoretical outcomes of the unit such as the principles of full-stack development, its common components, commercial examples, competing architectures (SOA vs monolithic) and the creation of web-based APIs. Learners should evidence this Learning Outcome by producing a document. **LO1 AC 1.1:** Learners should be able to introduce and **describe** the common components of a full-stack solution. They should be able to differentiate between front-end and back-end technologies, detail different components used, e.g. DBMS, Scripting Languages, OS, Web Server, Backend and frontend frameworks etc. **LO1 AC 1.2:** Learners should be able to **explain** the different types of commercially available stack, the use of SOA in modern solutions, different hosting options. Their coverage should help them determine	**LO1 1M1** To meet this criterion, learners should be able to state the advantage of cloud-based solutions using SOA over traditionally hosted monolithic solutions. This typically includes: • Greater scalability • Reuse of services between different projects, e.g. a federated login system etc. • Separation of concerns between different systems • Improved security • Faster development time • More efficient processing	

		different options they could consider for design and implementation. **LO1 AC 1.3:** Learners should **explain** the common features of a web-based API. Being able to describe endpoints, requests, responses, status codes and data formats should support their thinking and preparation for creating their own API to meet the stakeholder's requirements.		
2.	LO2 2.1, 2.2, 2M1	Task 2 focuses on the practical application of design, and the use of tools and techniques to document the design of the proposed website portal and API (including both functionality and the user interface) and its chosen full-stack implementation. **LO2 AC 2.1, 2.2:** Learners produce **designs** for their solution using appropriate design and interface design tools. They are asked to **implement** their design and document all aspects including the visual design of the website portal's interface, the API endpoints (both open access and non-documented ones used only by admin). It must also include all requests and responses, details of the data format(s) used in the requests and responses, and how the backend database will be constructed, i.e. to store rainfall data, user account data (from registration) and activity logs for the API.	**LO2 2M1:** To meet this criterion, learners should **justify** their full-stack selection for their website and API solution. This should include a rationale for picking the various components, e.g. ease of development, portability, industry support, industry reputation etc.	

3	LO3 3.1, 3.2, 3.3, 3M1, 3D1	Task 3 focuses on the practical hardening of the solution based on the learner's ability to identify the common vulnerabilities and threats the website and API may face. Learners should evidence this Learning Outcome by first producing a document, then practically addressing the issues through configuration and coding practices, and then testing these thoroughly. **LO3 AC 3.1:** Learners must identify and **describe** the common vulnerabilities and threats to web-based applications. Ideally this should be linked to industry guidance, e.g. NCSC (for UK) or OWASP or localised equivalent. **LO3 AC 3.2:** Learners must attempt to practically address vulnerabilities and threats which they have identified could threaten the website portal and API, e.g. **apply** protection against SQL injection in the portal's registration process or prevent a website's directory being listed by a simple GET request. **LO3 AC 3.3:** The testing criterion should be evidenced through practical testing activities. Learners should identify which aspects of the solution need to be **tested** – but this must include vulnerabilities they have previously identified and whether the website and API is consequently functionally sound and secure. To do this, it is recommended they create a	**LO3 3M1:** Learners produce a document which **justifies** the decisions they have made to harden their website portal and API. This should include the relevance of the threat, the appropriateness of the protection applied, for every identified vulnerability. Referencing industry "best practice" is an ideal addition, but specific links are essential.	**LO3 3D1:** The document produced for LO3 3M1 could be expanded to include an **evaluation** of the full-stack solution produced (as compared to the original stakeholder requirements) and incorporate its adherence to recommended best practice, particularly in terms of design, implementation and hardening against threats and vulnerabilities. Again, linking decisions to industry "best practice" is an ideal addition.

		test plan, and carry out sufficient testing across the range specified in the indicative content to ensure that the solution is fit for purpose. They will need to use their test plan, create test values, expected outcomes, together with evidence of testing results and any action taken to correct errors. In terms of user testing, learners could formally test other's solutions and provide feedback (to simulate stakeholders).		