University of Westminster
School of Computer Science and Engineering

6COSC022W Coursework 1 (2024/25)	
Module leader	Tendai Mhlanga
Unit	Coursework 1
Weighting:	40%
Qualifying mark	30%
Description	Design of web application and essay
Learning Outcomes Covered in this Assignment:	L02. LO3, LO4
Handed Out:	16/02/2025
Due Date	**1pm 10th April 2025**
Expected deliverables	Electronic document and web application deployed on university servers
Method of Submission:	online via Blackboard
Type of Feedback and Due Date:	Feedback will be given during viva in tutorials following deadline submission and via Blackboard rubric. **All marks will remain provisional until formally agreed by an Assessment Board.**

Assessment regulations

Refer to section 4 of the “How you study” guide for undergraduate students for a clarification of how you are assessed, penalties and late submissions, what constitutes plagiarism etc.

Penalty for Late Submission

If you submit your coursework late but within 24 hours or one working day of the specified deadline, 10 marks will be deducted from the final mark, as a penalty for late submission, except for work which obtains a mark in the range 40 – 49%, in which case the mark will be capped at the pass mark (40%). If you submit your coursework more than 24 hours or more than one working day after the specified deadline you will be given a mark of zero for the work in question unless a claim of Mitigating Circumstances has been submitted and accepted as valid.

It is recognized that on occasion, illness or a personal crisis can mean that you fail to submit a piece of work on time. In such cases you must inform the Campus Office in writing on a mitigating circumstances form, giving the reason for your late or non-submission. You must provide relevant documentary evidence with the form. This information will be reported to the relevant Assessment Board that will decide whether the mark of zero shall stand. For more detailed information regarding University Assessment Regulations, please refer to the following website:http://www.westminster.ac.uk/study/current-students/resources/academic-regulations

Coursework Description

Introduction

In this project, you will develop a secure API middleware service that interfaces with RestCountries.com, a comprehensive RESTful service providing detailed information about countries worldwide. Your application will serve as an intermediary layer, processing and filtering country data while implementing robust security measures.

Project Scope

Your service will retrieve data from https://restcountries.com and provide a streamlined response containing only essential country information:

- Country name
- Currency information
- Capital city
- Spoken languages
- National flag

The application will feature a complete authentication system with user registration, login capabilities, and API key management. Users will be able to generate and manage their API keys through a dedicated web interface, with all sensitive data securely stored in an SQLite database using proper security practices including password hashing and session management.

Technical Framework

While you have flexibility in choosing your programming language (PHP, Flask, Java, or others), we encourage you to use a polyglot architecture when integrated with Coursework 2.

For optimal grading, consider containerizing your application using Docker for local deployment.

Marking Scheme

Core API Features (15 points)

REST API Integration (6 points)

- Excellent (5-6): Flawless integration with https://restcountries.com API, robust error handling, efficient data retrieval
- Good (3-4): Working integration with minor issues, basic error handling
- Fair (1-2): Basic integration with significant issues
- Poor (0): Failed or non-functional integration

Endpoint Implementation (6 points)

- Excellent (5-6): All endpoints properly implemented, well-documented, following REST principles
- Good (3-4): Most endpoints working with minor issues
- Fair (1-2): Basic endpoint implementation with problems
- Poor (0): Missing or non-functional endpoints

JSON Implementation (3 points)

- Excellent (3): Perfect JSON formatting, proper error handling, consistent structure.
- Good (2): Minor JSON formatting issues.
- Poor (0-1): Major JSON formatting problems or inconsistencies.

Security Implementation (25 points)

API Key Authentication (10 points)

- Excellent (8-10): Robust key validation, secure storage, proper error handling.
- Good (5-7): Functional authentication with minor security gaps.
- Fair (3-4): Basic authentication with security concerns.
- Poor (0-2): Inadequate or insecure authentication.

Security Protections (15 points)

- Excellent (12-15): Comprehensive security measures including password hashing, session management, input validation.
- Good (8-11): Good security implementation with some gaps.
- Fair (4-7): Basic security measures with vulnerabilities.
- Poor (0-3): Major security flaws or missing protections.

Data Management (10 points)

Advanced Implementation (7-10 points)

- Database in 3NF
- Appropriate security fields
- API key usage tracking
- Well-structured relationships

Basic Implementation (1-6 points)

- Minimum 2 tables
- No API key tracking
- Basic structure

Admin Features (12 points)

Authentication System (6 points)

- Excellent (5-6): Complete user management system
- Good (3-4): Functional authentication with minor issues
- Fair (1-2): Basic authentication with problems
- Poor (0): Non-functional authentication

API Key Management (6 points)

- Excellent (5-6): Full key management functionality
- Good (3-4): Basic key management with limitations
- Fair (1-2): Minimal key management features
- Poor (0): Missing key management

Technical Implementation (8 points)

Docker Implementation (4 points)

- Excellent (4): Fully containerized, well-configured
- Good (2-3): Working container with minor issues
- Poor (0-1): Incomplete or problematic containerization

Architecture & Documentation (4 points)

- Excellent (4): Comprehensive documentation, clear architecture
- Good (2-3): Adequate documentation with some gaps
- Poor (0-1): Insufficient or unclear documentation

Vodcast Presentation (30 points)

Application Demonstration (10 points)

- Excellent (9-10): Clear demonstration of all features, smooth execution, well-organized presentation
- Good (7-8): Good demonstration of most features with minor issues
- Fair (5-6): Basic demonstration with some technical problems
- Poor (0-4): Incomplete or unclear demonstration

Code Explanation (10 points)

- Excellent (9-10): Clear explanation of code structure, design patterns, and implementation details
- Good (7-8): Good explanation of main code components
- Fair (5-6): Basic code explanation with some gaps
- Poor (0-4): Poor or incomplete code explanation

Security & Design Justification (10 points)

- Excellent (9-10): Comprehensive explanation of:
    - Security implementation choices
    - Architectural decisions
    - Performance considerations
    - Testing approach
- Good (7-8): Good explanation of most major decisions
- Fair (5-6): Basic explanation of some decisions
- Poor (0-4): Poor or missing justification

Deliverables:

1. Zipped project with source code
   Submitted on Blackboard.

2. Video demonstration Submitted
   on Blackboard.

Please NOTE: The Module Leader and members of the Module Team reserve the right to viva the student. An email invite will sent inviting the student to a viva. Failure to show up to defend your work will result in a grade of Zero.

Use of Generative AI is forbidden in this coursework.

If you need any clarification about any of these issues, please contact me by email (at mhlangt@westminster.ac.uk) or in the tutorials or make an appointment to see me in my office.

END DOCUMENT