MCAST

ASSESSMENT AND INTERNAL VERIFICATION FRONT SHEET (Individual Criteria)

Course Title	Bachelor of Science (Honours) in Applied Data Sciences, Bachelor of Science (Honours) in Creative Computing		Lecturer Name & Surname	Ranier Bonnici	
Unit Number & Title	ITMSD-606-2312| Full Stack Development				
Assignment Number, Title / Type	2, Creating an MVC API (Home)				
Date Set	15-12-2025	Deadline Date	18-01-2026		
Student Name		ID Number		Class / Group	IT-CCD-6.3A, IT-ADS-6.3A

Assessment Criteria	Maximum Mark	Mark Achieved
R&U2 Explain RESTful design principles	5	
R&U4 Describe fundamental security concepts	5	
R&U6 Illustrate how to manage state for different users	5	
E&C2 Create RESTful endpoints to manage application data	10	
A&A3 Apply basic validation and error handling in backend	7	
R&U7 Illustrate the use of authentication mechanisms	5	
A&A4 Deploy a working full stack solution to a live or test environment	7	
A&A5 Assess the correctness of the application	7	
Total	**51**	

Notes to Students:

- This assignment brief has been approved and released by the Internal Verifier through Classter.
- Assessment marks and feedback by the lecturer will be available online via Classter (Http://mcast.classter.com) following release by the Internal Verifier
- Students submitting their assignment on VLE/Turnitin will be requested to confirm online the following statements:

Student's declaration prior to handing-in of assignment

- ❖ I certify that the work submitted for this assignment is my own and that I have read and understood the respective Plagiarism Policy

Student's declaration on assessment special arrangements

- ❖ I certify that adequate support was given to me during the assignment through the Institute and/or the Inclusive Education Unit.
- ❖ I declare that I refused the special support offered by the Institute.

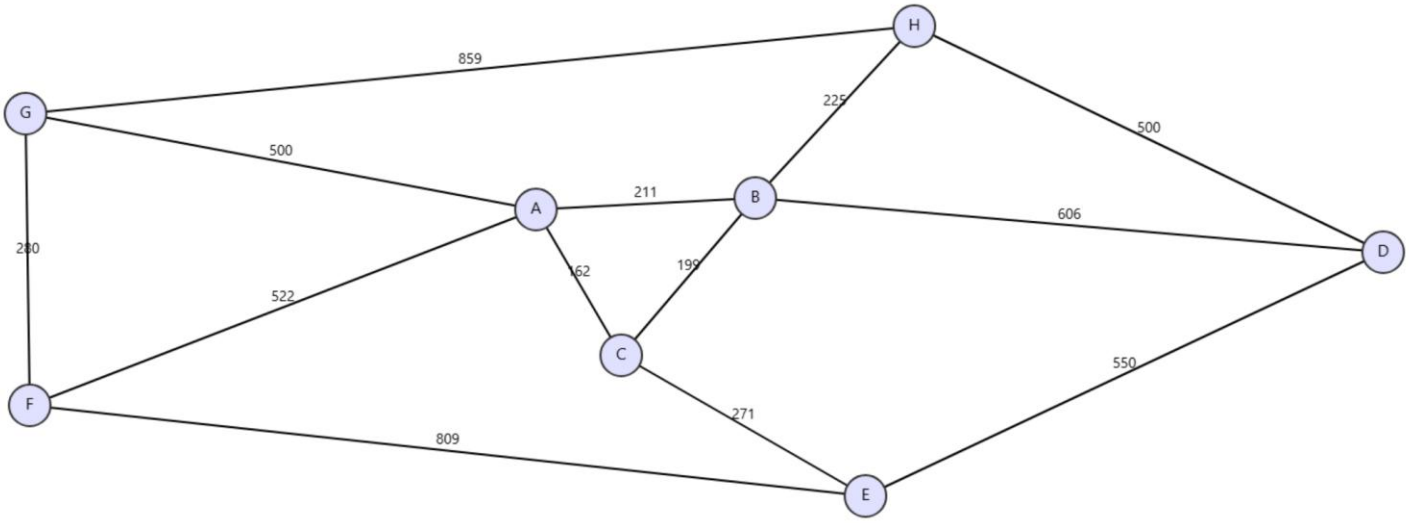
Instructions

- The assignment deadline is on 19th January.
- You are to submit the assignment on the VLE, under the section Assignment 2.

Section A (Artefact, 34 marks)

You are required to develop a RESTful ASP.NET Core MVC API that calculates the shortest route between two points on a map.

A map is represented as a graph, where each bi-directional edge contains the distance between two nodes.



In the map provided, the shortest route between nodes `G` and `E` is `GACE`, and the shortest distance is `933`.

Required Endpoints

1. `/api/map/SetMap`

   Purpose: Stores the map (graph) used for all later requests.
   Required API Key (X-Api-Key): `FS_ReadWrite`
   Input: Graph JSON (refer to sample)
   Response: `200 OK` on success

2. `/api/map/GetMap`

   Purpose: Returns the currently stored map.
   Required API Key (X-Api-Key): `FS_Read`
   Input: None
   Output: Graph JSON (refer to sample)

3. `/api/map/ShortestRoute`

   Purpose: Returns the node sequence for the fastest route.
   Required API Key (X-Api-Key): `FS_Read`
   Inputs:
   - from (string) – start node, e.g., `G`
   - to (string) – destination node, e.g., `E`

   Output: Path string (e.g., "`GACE`")

4. `/api/map/ShortestDistance`

   Purpose: Returns the travel time of the fastest route.
   Required API Key (X-Api-Key): `FS_Read`
   Inputs: same as `ShortestRoute`
   Output: Travel distance as an `int` (e.g., `933`)

Error Handling

5. You must return the following responses in the correct situations:

   `200 OK`
   Valid request processed successfully.

   `400 Bad Request`
   Use 400 when the client's request is invalid, including:
   - Missing parameters (`from`, `to`, or map data)
   - Unknown node names (e.g., "`Z`" if Z does not exist)
   - Map has not been set

   `401 Unauthorized`
   Used when:
   - No API key is provided
   - The API key is incorrect
   - The API key does not grant permission (e.g., using `FS_Read` on `SetMap`)

Deployment Requirement

6. Deploy your API using Render.com (free tier) or another alternative host. Provide the public API URL.

Section B (Theory, 17 marks)

1. Provide screenshots that show testing of the `ShortestDistance` endpoint in Postman (inputs and outputs).

Answer questions 2 and 3 in 100 words each

2. The API in the brief contains a REST anti-pattern. Discuss this anti-pattern and suggest a better design.
3. Identify and discuss two weaknesses of the authentication method used.

Criteria

Criterion	Question	Task	Marks
RU7	Section A, Q1-4	`SetMap` requires key `FS_ReadWrite` `GetMap`, `ShortestRoute`, `ShortestDistance` require key `FS_Read`	2 3
RU6	Section A, Q1 Section A, Q2	`SetMap` endpoint works correctly. `GetMap` endpoint works correctly.	2.5 2.5
EC2	Section A, Q3 Section A, Q4	`ShortestRoute` endpoint works correctly. `ShortestDistance` endpoint works correctly.	5 5
AA3	Section A, Q5	Responses are provided as indicated.	7
AA4	Section A, Q6	Deploy the API on Render.com (or alternative).	7
AA5	Section B, Q1	Postman screenshots show testing of `ShortestDistance` (inputs & outputs).	7
RU2	Section B, Q2	Anti-Pattern identified. Solution provided.	2.5 2.5
RU4	Section B, Q3	Two weaknesses listed.	5
Total			**51**