Team Project Marking Scheme

Your API must include **at least three controllers**, and each controller must implement all **six HTTP methods**:

- **GET (GetAll)** – Retrieve all records
- **GET (GetById)** – Retrieve a specific record by its ID
- **POST** – Create a new record
- **PUT** – Replace an existing record entirely
- **PATCH** – Update part of an existing record
- **DELETE** – Remove a record

In real-world scenarios, certain methods (such as `POST`, `PUT`, or `DELETE`) might only be available to internal or authorized users. However, for this project, your focus should be on ensuring that all six methods are properly implemented and functional for each controller.

The due date for source code and project report is **Midnight of Week#11' Sunday**. One group member submits the package on behalf of the whole group. The project needs to be demonstrated in person or online, and **pre-record video is not acceptable**.

	Mark	Weighted Mark
1. Proposal	**10**	5
2. Implementation	**60**	15
Prepare data		
2.1 Data is in the cloud (RDS or DynamoDB) ○ For RDS, each table needs to have at least 10 rows ○ For DynamoDB, the table needs to include at least 10 items	3	
API Implementation		
2.2 Use the repository pattern	2	
2.3 Implement DTO classes ○ At least 3 DTO classes for each controller ○ Use AutoMapper Mapping profile	3*3*0.5+1=5.5	
2.4 Http Get method, GetAll & GetByID	3*0.5*2=3	
2.5 Http Post method	3*1	
2.6 Http Put method	3*1	
2.7 Http Patch method	3*1	
2.8 Http Delete method	3*0.5=1.5	
Publish API		
2.9 Containerize your Web API and push the image to ECR	1+1	

2.10 Publish containerized API to AWS ECS using AWS Fargate 2.10.1. Task definition 2.10.2. Run **one task rather than a service** 2.10.2. Running task works as expected	1+1+2	
Manage API using Google APIgee		
2.11 Create an APIgee proxy	1	
2.12 Create a Product	1	
2.13 Add two developers	1	
2.12 Your API proxy policy should include “Verify API key” 1.12.1 The “Verify API key” works properly	1+2	
2.13 Generate portal to support your potential API client(s)	1	
Consume API		
2.14 Implement a client to consume the published API, the client can be Web app or mobile app		
2.14.1 Instantiate HttpClient object, pay attention to API key	1	
2.14.2 invoke all 6 methods provided in your API	6*3=18	
Non-functional requirements		
2.15 Overall (performance, readability, maintainability, usability, etc.)	5	
3 Presentation & Report	**20**	5
3.1 Presentation 3.1.1 present the functionalities 3.1.2 answer questions	5+5	
3.3 The architecture of the project	4	
3.3 Data modeling document (e.g., ER diagram, sample data)	2	
3.4 Experience(s) gained & lessons learnt by doing the project	2	