A scenario for the project

- The IMAGINary Inc., a software development company, has grown its staff members dramatically during the last quarter due to their high quality deliverables. The board of directors has decided to develop a task based tracking system for the employees.
- Each developer is assigned multiple projects and a project can have multiple developers and a manager.
- Employees usually work eight hours per day excluding weekends. If they work any hours other than the standard working hours, it will be computed as overtime.

Deliverable 1

Develop an API web application to address the above scenario. You may alter the path examples provided below but all the functional requirements must be satisfied.

Functional Requirements:

- POST /tasks

  Developers are able to report their daily tasks to the proposed system. Task consists of a project, date, number of hours (whole hours only, minutes not permitted), standard/overtime, and a description of the task.

- GET /projects

  A project manager should be able to track his/her projects, the list of developers, and each developer's contribution in hours and as a percentage from total hours (refer to Table 1).

Manager A				
Project	**Members**	**Hours**	**Overtime**	**Contribution**
Project A	Developer A	25	0	25%
	Developer B	50	20	50%
	Developer C	25	10	25%
Project B	…	…	…	…

Table 1: A manger's projects, members in projects, and their contribution.

- GET /developers/:id

  Project manager should also be able to track overtime per developer.

- PUT /tasks/:id

  Developers should be able to update their tasks (date and no of hours).
- DELETE /tasks/:id

  Developers should be able to delete mistakenly added tasks.

It is very important that your APIs work, the public interfaces you create should conform to industry good practice for creating web APIs. The web API should be developed using **Java**, **C#**, **PHP,** or **Ruby on Rails**.

The interface must use the **REST** architecture and deliver the data in **JSON** format. There must be supporting web pages displaying **HTML** and you must provide an example client application that consumes your own API.

Deliverable 2

Write a report containing a critical evaluation of three third party API's. Your report must discuss how the evaluation will be carried out and give details and screenshots of the tools, techniques or approaches chosen.

One of the API's to evaluate will be a peers API randomly allocated to you during the week after the API submission. The other two APIs to be evaluated are a free choice.

Your report should be well written, well structured not exceeding 4,000 words. You must include a list of references (including URLs) using the Harvard.

Detailed Assessment Criteria

Category and Weighting	Criteria	Marks	LOs
API Implementation **60%**	• API implemented and presented with appropriate documentation. • Design clearly illustrated within the website documentation and clearly evident in architecture. • Developer resources for API conform to industry good practice. • Source code runs with test data. • Data is inserted, edited and retrieved from the MySQL database. • API conforms to RESTful principles	40 20 10 05 15 10	LO2

Evaluation Report **40%**	• Style of report is clear, professional and has a logically developed thread of argument throughout.	10	
	• Evaluation approach is relevant.	10	LO1
	• Appropriate academic and industry literature is used to support discussion.	30	LO3
	• Clear evidence of understanding topic area is demonstrated. Evidence of critique.	30	
	• Client application, tools & technique used for evaluation clearly described and illustrated.	10	
	• APIs selected suitable	10	

Submission Type

The report should be a PDF (preferred) or Word document. A zip file containing project work and the report should be uploaded to the Plymouth Digital Learning Environment (DLE). You will be required to perform a 10-minute defense of your solution at a mutually convenient time (which may or may not be in your timetabled slot).