TASK3: MVP

Graded Task ; Due date: *11:59pm (Melbourne Time) Sunday Week6*

Task description

This task is one of the most important tasks in the unit. It is your time to showcase your skills and expertise you developed so far in the unit on your project of choice! This is a graded task which means the quality of your work is very important. You will need to demonstrate a working technical solution using at least **all** the fundamental technologies covered in the unit. You are expected to demonstrate an end-to-end user story of your project. This should be one of the main (critical) user stories of your projects. User login/logout or a simple list of items are not considered the main user stories - think of yourself showing a demo to an investor as to what you want to show them.

Rubric

PASS: You have implemented, reported and demonstrated the core feature(s) of the application demonstrating all the fundamental skills in the unit.

CREDIT: You have implemented, reported and demonstrated the core user story and extra more user stories relevant to your project.

DISTINCTION: You have achieved PASS/CREDIT levels but you have also included extra features that provide a much richer story for the user, going back to our uber, you have implemented timers, routes and visualizations for this - you have implemented a complete "experience". Your implementation should be of good quality, well structured, maintainable, and well-documented.

HIGH DISTINCTION: You have achieved DISTINCTION level, and your MVP is polished, and of quality internal quality and external quality (user experience) that you are ready to sign up users to start using your product.

Further explanations

I wrote this to try and help you build your MVP and clearly understand how to move forward.

To complete the assessment, you will need to provide the following information on the forms provided, depending on your unit (or units). Do multiple if you belong to different units. Make sure you only submit once. Also, make sure you don't make changes to your application once you submit it.

You will loose marks if you have any commits made beyond the submission time.

https://forms.office.com/r/uR0YEJDBvS	SIT780
https://forms.office.com/r/mX8QX4vj8C	SIT725
https://forms.office.com/r/iWfD1bZ4gG	SIT737

Please upload a document on Ontrack that contains the same information you have submitted in the form.

Submission details

- An outline of each fundamental technology that you must cover in the unit, description of how you implemented these in your project (text and code snippets).

  No more than 100 words in total .

- A link to GitHub Repo of your project including project source code.
  - This is very important because I will be cloning your code. Also, make sure that your code runs off npm start. This is in the form similar to this https://github.com/alexbonti/greenhouse-poc
  - If you provide the wrong URL, I will not be able to view your code.
  - If you don't make your project public, I will not be able to see your code.
- A link to a 3 minutes video of you explaining your project code
  - Kindly explain how this your code works.

- A link to a 2 minutes video demo of your project running.
  - Show us what your application is meant to do.

To better understand what it is you need to provide technically, I have made the following.
For SIT737 and 780 you need to look at two rubric. The SIT725 and your corresponding one. This is because both units are a continuation of 725 (Software engineering), therefore, you must make use of the basics that you have learnt there (Eg wireframe and code style) .

From code perspective, the same rules apply from SIT725 , meaning that the programming language is NodeJS, and you are not to use a framework (React or Angular or similar), unless you have prior authorization. Also, the only UI libraries allowed are Bootstrap and Materialize.

Technological Rubric

SIT725 (Prereq)

Item	Pass	Credit	Distinction	High D.
Code structure	You have not implemented the MVC Model. Your code somehow works, it is all over the place, but it works.	You have not implemented the MVC Model. Your code is tightly coupled but it is readable and well documented.	You have applied the MVC model.	Your code is pleasant to read. Comments are made well, and the structure is well organized. Also, best decisions were made to optimize the code.
Paper Prototype	None	You have created a simple wireframe that resembles the application structure	Your wireframe is interactive. You don’t need dynamic data, but it should resemble a real-life interaction.	
GIT	Git repository exists		Proper Development and Feature branches are created	
User experience	Your application has no form, thins are put together casually.	The ui is created through basic Bootstrap/Materialize components		It is evident that some research and effort have been put into both ui and user

				experience

Specific Units

SIT 737 Cloud development

Item	Pass	Credit	Distinction	High D.
Service-Oriented You need to showcase the decoupling of services.	You make use of node red to serve the service.	You showcase at least a 3 layers architecture (Web Server, Data Service and DB). Your page requests data from another server.	As per credit, but you also have an external API (that you may have created)	You have made use of a mix of PaaS and cloud functions
Consume the UI	Consume data by visualizing it			
Deployment (you cannot use a VM), it has to be a PaaS or similar.	Deployed On NodeRed	Deployed on IBM Cloud	Showcase Continous delivery	
Architecture	Provide a diagram of the application structure (what talks to what)			

SIT780 Enterprise Applications

Item	Pass	Credit	Distinction	High D.
Dockerized	You have successfully dockerized your system			
Continuous Delivery		Showcase your ability to		

		continuously build your application through Git Actions		
Testing (either one with Authentication)			You have implemented Testing in your application	
Authentication			The system uses passport for authentication	

Features - for all Units

(I have taken my own project as an example of what needs to be provided)

Item	Pass	Credit	Distinction	High D.
Core Features Implemented, reported and demonstrated core features. These are the minimum to make your project worthwhile	The ability to share my idea with others.			
Secondary Features These are supporting features that create a complete experience.		Allow others to comment or vote.		
Rich Story Features			Sign in Feature - users can only post if they are logged in. Email notification: owner gets notified when a person	

			makes a comment.	
User Experience Don’t just make your application just work, make it for useful.				I have researched various crowdfunding services and tried to apply similar UX, I have also tried to make the application aesthetically pleasing.