SIT725 7.5P

PROJECT PLAN (GROUP TASK)

Introduction

This task is intended to help you achieve your target grade by making it clear what you need to do in the group project.

By now, you would have had your first sprint planning (for sprint 1) and have a good idea about the app you are building and the features planned.
Now, you need to think about how you will contribute individually, and how that will help you get to your target grade.
Note that there is **no** separate grade for the project or group grade. Your **grades are individual** and will depend on 1) the completion of your OnTrack tasks **and** 2) your individual contributions to the group project.

In this task, you need to meet with your team and discuss your project and your target grade expectations, and plan how each member will contribute. All the members need to be present in the planning sessions.

Please review the rubric below.

Rubric for Individual Contributions for the Group Project

Criteria	**Pass -> *"can do allocated work with guidance"***	**Credit -> *"performs their role in the Project"***	**Distinction -> *"can see themselves in the team and assist others"***	**High Distinction -> *"can demonstrate excellence"***
Core Development Note: students may contribute either to backend or frontend or both.	Writes basic code that contributes to the project. General Development: No significant customization, mostly a default	Writes code that contributes to the project. The code is tidy and commented properly. Your work reflects	Writes code that contributes to the project in substantial ways. The code reflects the MVC model (route/controller etc) and classes are used where appropriate Your work is tidy and neat, show casing research.	Writes code that contributes to the project in exceptional ways. The code reflects the MVC model (route/controller etc) and classes are used where appropriate Whether you contributed to backend or frontend,

	setup. E.g. - If you contributed to frontend, your code would have a basic UI	some basic research and effort.		everything is of quality. Code is neat, well documented. Your work shows signs of research, going beyond what was discussed in the unit and the application is well constructed and follows best practices. E.g. For frontend contributions - A HI-FI (or close to) prototype is provided – (detailed, interactive UI prototype that closely resembles the final product). For backend contributions - well-structured, optimized[1], backend code with clear separation of concerns.
CRUD		Contributed to basic CRUD operations (Create, Read, Update, and Delete)	Contributed to advanced CRUD operations. These could be ✅ Data Validation & Error Handling. E.g. Prevent incorrect or malicious input (e.g., empty fields, incorrect formats, or duplicates). ✅ Role-Based Access Control (RBAC) for CRUD Operations (restrict CRUD based on user roles). ✅ Bulk insert, update, or delete ✅ Soft delete (instead of permanent deletion).	
Design		For frontend contributions, a low fi wireframe OR a basic static mockup has been created indicating the major components of the UI they contributed to. For backend contributions, a basic system architecture diagram has been created, outlining	**[D & HD]** For frontend contributions, A clickable wireframe or interactive prototype (e.g., in Figma, Adobe XD) is provided that simulates user interactions and mimics key functionalities of the UI they contributed to. For backend contributions, a detailed system architecture diagram is provided, showing how components interact in the system.	

[1] goes beyond just "working". Eg. designed to handle increased load/ minimise response times.

		the major components of the backend they contributed to.		
SRS	Made meaningful contributions to the SRS document by outlining key system functionalities, ensuring major features were documented.	Contributed significantly to a well-structured SRS document, helping define both functional and non-functional requirements with clear descriptions.	**[D & HD]** Played a significant role in developing a well-structured SRS document, helping define both functional and non-functional requirements with clear descriptions and logical organization. Supported the document with relevant research, such as industry best practices, standards, case studies, or comparisons with similar systems.	
Testing	Developer testing: You have tested your own code during development. Evidence of testing is recorded somewhere (e.g., a Trello card/comments in a PR etc).	Contributed to manual testing: Test cases are written down and document how you manually tested specific functionalities. Test cases describe the input, expected output, and test steps.	Contributed by writing unit tests for specific functionalities of the application, ensuring core features were covered and tested.	Played a key role in implementing end-to-end testing, contributing test scripts and ensuring critical workflows were tested. Contributed test scripts for automation testing, specifically for end-to-end, ensuring that full user workflows were covered. (This does not refer to automated unit tests.) Assisted in preparing a report documenting the test coverage and results.
Real Time			Contributed to implementing real-time functionality by integrating broadcasting sockets, enabling updates to be sent to multiple users simultaneously.	Developed and refined real-time features by implementing per-user notifications, ensuring targeted updates are delivered only to specific users based on defined conditions.
Discuss with tutor during workshops			Discuss with your tutor and showcase your work during the workshops in at least 1 occasions, outside of scheduled demos in order to get feedback, and integrate that feedback into your contributions. Tutor will note down names and pass on to the markers.	Discuss with your tutor and showcase your work during the workshops in at least 2 occasions, outside of scheduled demos, in order to get feedback, and integrate that feedback into your contributions. Tutor will note down names and pass on to the markers.
Leadership *Leadership does not mean doing all the work—it also does not mean just being "the boss" and delegating work. Leadership is about helping the team			You have taken on leadership responsibilities at **specific points** in the project. This could include (but is not limited to): ✅ Acting as the Scrum Master ✅ Active participation in sprint planning, helping define scope, refine requirements,	You have **consistently** demonstrated leadership **throughout** the project. This could include (but is not limited to): ✅ Acting as the Scrum Master ✅ Active participation in sprint planning, helping define scope, refine

succeed, supporting your teammates, and creating an environment where everyone can contribute effectively.			prioritizing and planning tasks, etc. ✅ Providing technical leadership on specific activities (e.g., guiding architecture decisions, setting coding standards, etc). ✅ Mentoring or assisting other team members in debugging, Git etc.	requirements, prioritizing and planning tasks, etc. ✅ Providing technical leadership on specific activities (e.g., guiding architecture decisions, setting coding standards, etc). ✅ Mentoring or assisting other team members in debugging, Git etc.
Tools	**Pass**	**Credit**	**Distinction**	**High Distinction**
GIT	You make use of git to the minimum, you treat almost as a remote drive rather than a version control system.	You use GIT using basic versioning, such making commits and pushing code regularly. You rarely use branches.	You make good use of branches to keep track of your work. You follow a structured workflow, such as using feature branches for tasks instead of working directly in main.	You have releases, feature branches and overall an absolute champion in keeping track of versioning. All the branches are 1 to 1 with Trello tasks.
Trello	You use Trello at a basic level, but your tasks are vague or unclear, making it difficult for others to understand what you worked on. Minimum requirement: You move your tasks across stages (e.g., "To Do" → "Doing" → "Done"), even if little additional detail is provided.	Your tasks have clear names that describe the work at a high level, making it easy to understand what you're working on.	Your tasks are well-documented and easy to follow: • Tasks have clear names and descriptions, so others understand what was done. • You regularly update tasks to reflect progress. • Uses Trello features like checklists, comments, labels & due dates where appropriate. .	
README file	You have contributed to the README in some way, ensuring the project has instructions for markers to run it. The README must contain: ✅ Project name ✅ Basic setup instructions (e.g., how to install dependencies and run the project) ✅ Team member names	**[C, D & HD]** You have significantly contributed to the README by improving clarity, structure, or completeness. E.g. Improved the structure and formatting/ detailed setup instructions/ example usage or expected output/ basic troubleshooting tips (e.g., common installation issues) etc.		

HD: all of the blue + at least ONE of the orange
D: all of the blue + at least ONE of the orange
C: all of the blue + at least ONE of the orange
P: all of the blue

Submission details and Delivery

Your submission must have:

1. Overview of team contributions & target grades: Create table with Name, Main Role/s and Target Grades as columns and fill in with your group member details (See given template in task resources)
2. Individual Plans: For each member in the group, copy and paste the Individual Contributions rubric, tick the criteria boxes that apply and briefly mention the specific tasks that you plan to do under each criteria (See given template in task resources).

Note: You must use the given template in task resources for your submission. This will make it easier for markers to understand your plan and give feedback in a timely manner.