GRIFFITH COLLEGE CORK

ASSIGNMENT TITLE SHEET

FACULTY OF COMPUTING SCIENCE

Course:	BSC
Stage/Year:	3
Module:	WT
Semester:	Semester 2
Assignment Number:	Assignment 1
Date of Title Issue:	X
Assignment Deadline:	X
Assignment Submission:	Upload to Moodle
Assignment Weighting:	20% of module

Assignment Title

WT – Assignment 1 - Project proposal & user interface design

Artificial Intelligence Assessment Scale (AI AS):

This assignment is subject to AI permission option: (select all that apply)

No AI Use [ ]
AI for Planning [ ]
AI for Editing [x]
AI for Support Tasks [ ]
AI for Collaboration [ ]
Full Use of AI [ ]

AI Assessment Scale Framework:
https://www.griffith.ie/sites/default/files/2025-05/f-6.5-ai-in-learner-assessment-policy.pdf

Please note:

- All AI generated content and process must be declared.
- Any task that is not listed as permitted may be assumed to be prohibited.
- If it is determined that a learner has used an AI tool in any way that is not permitted by the chosen level, then they will be subject to the disciplinary procedures listed in the Academic Integrity and Misconduct policy regarding the misuse of AI in assignments.

Assignment Specific AI Instructions:

- AI may be used in **support tasks only** – eg. using the IDE'S AI assistant to generate in-place code suggestions.

- Generative AI may **not** be used to create blocks of code or whole sections of assignments.
- Where generative AI such as ChatGPT has been used for tutorial purposes, links must be provided to the chat.
- **No AI may be used to write reports, documentation, or comments, and may not be used to create images, charts/graphs, wireframes, sitemaps, etc**.
- You may be called in for VIVA to explain your work if unauthorized/uncredited AI use is suspected.

Learning Outcomes

This assignment is assessing the following programme and related module learning outcomes:

LO1: Discuss the features of web application frameworks

LO2: Compare and contrast single-page applications with other formats

LO3: Evaluate the architecture, System Analysis, design and features of client-side web application frameworks

LO4: Evaluate the architecture, System Analysis, design and features of server-side side web application frameworks

LO6: Plan, design and build web applications using both client-side and server-side web application frameworks.

Assessment Criteria

Assessment criteria applied to this assignment:

Assignment will be assessed on strength of ideas, clarity of writing, and quality of designs.

Submission Instructions and Plagiarism

Plagiarism:

All work must be your own. Any cases of suspected plagiarism will be thoroughly investigated and may be brought in for VIVA. All parties involved in cases of plagiarism will receive a 0 and their student record marked accordingly. Any submission that is found to be plagiarised will receive a grade of 0 without the possibility of resubmission.

By submitting your assignment you accept that you understand what plagiarism is and that your assignment is not plagiarised in any way.

You may not submit work or partial work that has previously been submitted for assessment.

Instructions for submission:

Upload your report as a **PDF** to Moodle. Each team submits **only one copy**.

Submission naming convention: Save your proposal and UI design as a PDF file with the name of your project (only PDF is accepted). Be sure include all members' names in the coversheet.

Referencing: All third-party assets (images, audio, fonts, etc) must be clearly referenced in your documentation.

Penalties

Late Penalties:

Standard penalties will be applied to work that is submitted late, as per faculty guidelines. https://moodle.griffith.ie/mod/resource/view.php?id=18202

Penalties that apply to this assignment:

-20 marks for proposing technologies not specified by the brief or approved by the lecturer.

-10 marks for wrong file format (submit **only** PDF files!).

Project Overview

Over the semester, you will learn how to use both client side and back-end technologies to make a web application / website. You will work on each iteration of the project with a partner/s, and over the course of the semester, will build a dynamic web application iteratively (each assignment milestone will be an iteration of your web application). To maintain some cohesion between each assignment, and leave you with a final, portfolio-worthy deliverable, you will decide on a project concept.

You and your partner/s will define your own project based on something that is interesting to your team. You should first find a team – see Moodle for more information.

If you are looking for a partner, please email your lecturer and you will be assigned to a team.

Project requirements:

Your project must include dynamic behaviour, where the front end responds to user input events or web service and updates the interface accordingly.

Technologies we will be using: This project must be based on the MERN stack: utilising MongoDB, Express.js, React.js, and Node.js. Penalties will apply for deviating from this.

It must include at least **6 functionalities**, providing services to the users. For each functionality, be sure to describe its purpose, what it does, how it is used, and what the users can expect from using it.

It must include logic that will execute both:

1. Client side in a web browser, and
2. Back-end component on a web server.

Examples: logic that executes on a client side

- Validate the form data (client-side input validation)
- Auto complete some information
- Auto correct typos
- Reformat the form data entries (textual, tabular, and graphical formats)
- Filter the search results
- Sort the search results

Examples: logic that executes on a server side

- Validate the form data (server-side input validation)
- Add / update / delete / retrieve data (files or database, CRUD)
- Produce reports or confirmation pages
- Perform business logic; for example, calculate tax, compute grade or GPA
- Handle HTTP requests
- Server-side input validation is necessary (sanitizing data etc.).

It must support **multiple users** (i.e., maintain states of the application such that multiple users can access the application simultaneously and their sessions do not overlap or interfere with each other).

It must support **multiple sessions**, allowing returning users to access / retrieve / manipulate their existing information or configuration (i.e., must include data persistence using a database).

Note: It may seem unclear at this point how the front-end and the back-end components interact. That's ok! This module will help you see how to properly implement and connect the front-end and the back-end components.

Optional:

In addition to the above requirements, you may make use of any CSS framework of your choice in the final iteration of your project.

Part 1: Project proposal [10%]

Required outputs: A PDF explaining your project (use the labelled template on Moodle).

You need to outline a clear project description, a description of 6 appropriate functionalities, explain how it will support multiple users and multiple sessions, user/competitor research, a planned timeline, and table listing the division of labour.

Division of Labour
All team members are expected to contribute equally to the project. At each iteration the team will fill out a table as below and state how the work was divided amongst the team.
E.g. If the work was split evenly per team member, then that is fine and simply say “Work was evenly divided”. If this was not the case, then state with a summary sentence or percentage.

Task	**Student Name 1**	**Student Name 2**	**Student Name 3**
User research	work was evenly divided	work was evenly divided	work was evenly divided
GUI design	50%	40%	10%
Etc.			

Part 2: User interface design [10%]

Required outputs: A sitemap of your overall site architecture, low-fidelity wireframes to illustrate your different screens, a proposed colour palette and font choices.

Design the wireframes for the different screens or views (with respect to the 6 functionalities or scenarios above) of your project. Note that wireframes are **not** intended to be high-fidelity – their purpose is to illustrate the features that will be present in the final UI and how they may interact with each other. Choices in fonts and colour-palettes should be made and discussed, but not necessarily applied to the wireframe.

Each screen must adequately demonstrate its purpose and functionality. You should include all necessary elements and justify your design decisions. All all points, design decisions must be explained. You may use user research or Nielsen’s 10 Heuristics to assist in explaining this.

Note: you may revisit and update your UI design while implementing the components.

You may use any software to draw your screens. There are several good, easy to use web-based softwares (some are free, some offer free trials) that you can use to draw user interfaces, like Draw.io or Figma.

If you draw your screens/wireframes by hand, please be sure that your drawings are legible.

Include these wireframe designs inside your proposal document under the appropriate headings.

Grading

	Components	Total Marks
Clear project description	Clearly explain your project idea, with reference to how you feel it fulfils the project requirements.	10
Description of functionalities	Explain the intended 6 planned functionalities, and how it will support multiple users and multiple sessions.	25
Research	There should be a degree of user/competitor research present to support your idea.	10
Timeline	There should be a reasonably considered project timeline that shows thought for what stages the project will encompass.	5
Design specifications	Site maps, colour choices, and font choices should be clearly included and explained. Write clearly under all headings, explain all design choices no matter how basic.	20
Wireframe legibility	You may hand-draw or use a software to design your wireframes, but they must be neat and clearly explained, either in the body of your document or with image annotation, or a combination of both.	30
Total		100