RMIT UNIVERSITY

Full Stack Development / Further Web Programming
COSC2758 UG /COSC2938 PG (Semester 1, 2025)
Assignment 2

Assessment Type	To be attempted in a group of 2. **If you would like to join a new group for assignment 2, please inform the lecturer at your earliest. In absence of a notification, your assignment 1 group will be rolled over to assignment 2.**
	Submit online via Canvas → Assignments → Assignment 2. Marks awarded for meeting requirements as closely as possible. Please read rubrics for details. Clarifications/updates may be made via announcements/relevant discussion forums.
Due Date	Week 13, Sunday 8th June 2025, 11:59 pm *Melbourne Time* This is a longer assessment, start in time to avoid delays and unexpected last minutes issues.
Marks	45

2.1 Overview & Learning outcomes

In this assignment, you will develop a Full Stack Web Application to complete the front-end prototype built from assignment 1. You are required to use the following stacks:

✓

- **Frontend:** React TS
- **Backend** : Node & Express with TypeORM
- **Relational Database**: Cloud MySQL

✗

- Using stacks other than the ones mentioned above will **FETCH A ZERO** for the whole assignment.
- Using object-oriented react components will **result in a ZERO** for the whole assignment.

The tasks are divided into *four* parts: PA (Pass), CR (Credit), DI (Distinction) & HD (High Distinction).

The **HD** section tasks will require self-research; you will not get straight answers in the course material. While we are happy to assist you with those tasks, you must do most of the work and research. This is done on purpose to prepare you for future work and the rigours of the IT industry.

If you find a specification open to interpretation, post a query identifying the specification in the corresponding discussion board for assignment 2. Software development in real life does not come with a definitive roadmap and flowcharts complete with instructions. More often than not, it is the job of the developer to clarify client requirements. For this assignment and course, the lecturer is considered as the client.

You are responsible for regularly attending lectorial, practical and consultation session(s).

- bring your questions to online discussion board and consultation sessions
- watch the online lectorial recordings regularly if you cannot attend the face-to-face session.
- do NOT start the work on assignment at the last minute.
- do NOT ask for last minute extensions, these are often rejected. Extensions can only be granted for personal and medical reasons, provided you can supply some documentary evidence.

This assessment relates to the following learning outcomes of the course which are:

CLO2: Demonstrate proficiency with a web application development framework

CLO3: Implement a range of techniques and procedures for developing a small to medium-scale web application

CLO4: Demonstrate knowledge of and utilise software engineering patterns in development

CLO5: Design and manage the development life cycle of a complete application.

2.2 GitHub & Submission

You will be marked on the use of GitHub and the development process. In week 1, you were emailed an invitation to join the **rmit-fsd-2025-s1** GitHub organisation. Please read the following:

- **DO NOT work on the assessment 1 repository as these assessments are being marked.** Copy the code from **the assessment 1 repository to a new assessment 2 repository.**
- Since you are working in a group, include both student IDs in the repository name as:

  <student-id>-<student-id>-a2 for example **s3123456-s3654321-a2**

- Include the URL of your GitHub repository in the README file. As an example,

  https://github.com/ rmit-fsd-2025-s1/s3123456-s3654321-a2

✓

You need to submit **one zipped archive** containing:

*your whole project folder **WITHOUT** node_modules directory + README file containing GitHub URL*

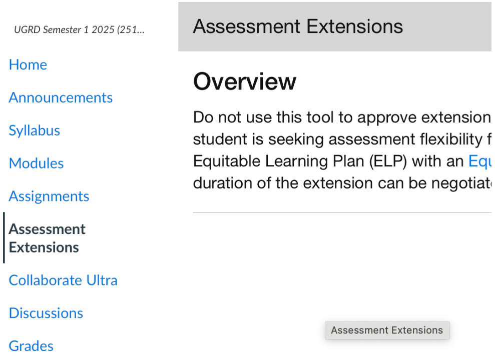
Assessment declaration: When you submit work electronically, you agree to the assessment declaration:

https://www.rmit.edu.au/students/my-course/assessment-results/assessment

2.3 Extension requests

After the due date, you will have **five** business days to submit your assignment as a late submission. Late submissions will incur a penalty of 10% per day. After these 5 days, Canvas will be closed, and you will lose ALL the assignment marks.

All extension requests must be made via the **Assessment Extensions** page of the Course canvas (*screenshot below*):



Assessment Extensions

Overview

Do not use this tool to approve extension
student is seeking assessment flexibility f
Equitable Learning Plan (ELP) with an Equ
duration of the extension can be negotiat

2.4 Academic integrity and plagiarism

Academic integrity is about the honest presentation of your academic work. It means acknowledging the work of others while developing your insights, knowledge and ideas. You should take extreme care that you have:

- acknowledged words, data, diagrams, models, frameworks and/or ideas of others you have quoted (i.e. directly copied), summarised, paraphrased, discussed or mentioned in your assessment through the appropriate referencing methods,
- provided a reference list of the online code used so your reader can locate the source if necessary. This includes material taken from Internet sites **& your own course material**.

If you fail to cite the sources of your material properly, you risk being accused of plagiarism for presenting someone else's work and ideas as your own without proper attribution.

RMIT University treats plagiarism as a very serious offence constituting misconduct. Plagiarism covers a variety of inappropriate behaviours, including:

- Contract cheating- paying someone to do your work
- Failure to properly document a source
- Copyright material from the internet or databases
- Collusion between students
- Posting assignment tasks on technical forums (*reddit, stack exchange, etc.*) and asking for solution(s)
- Copying code with the aid of generative AI tools.

For further information on our policies and procedures, please refer to:

https://www.rmit.edu.au/students/student-essentials/assessment-and-results/academic-integrity

2.5 Use of generative AI tools

As we increasingly integrate technology into our academic and professional practices, it is imperative to address the use of generative AI tools, such as ChatGPT, in your university assessments. These tools should be used cautiously, as they function by synthesising and repackaging existing text from various sources. We will discuss the implications of using such technologies, including the academic standards and their significance in your future careers, during lectorial sessions. We encourage you to produce and submit only your original work and to seek clarification from instructors if you have any uncertainties.

Furthermore, please note that the Co-pilot feature will **not** be enabled within the official RMIT's Git organisation for this course (**rmit-fsd-2025-s1**). If you choose to use Co-pilot independently, it is crucial to use it responsibly. Remember, generative AI should not be used to replicate code directly.

✓ We are committed to academic integrity and will conduct face-to-face meetings to address any cases of suspicious code submissions.

Ensure you properly reference any code that AI tools generate to maintain transparency and uphold academic honesty. Here are RMIT's official AI-related referencing guidelines:

https://rmit.libguides.com/referencing_AI_tools#s-lg-box-22377324

2.6 Assessment scenario overview & system requirements

The senior executive committee at the university has accepted the **TT** front-end prototype and now recommends that a full-stack version of a website be developed. The committee recommends including extra features and constraints for the website- a summary is presented here; you will find more details in **Section 2.7: Tasks**.

READ the following 10 system requirements:

1) The committee wants the developer to use a Cloud MySQL database (**Server: 209.38.26.237; Port: 3306**) for backend purposes.

   The phpMyAdmin can be accessed via this URL: https://getmysql.com/

2) The TypeScript code must be compiled without errors when executing the `npm run build` command to get full marks for a requirement.

3) The full-stack version of TT must be a multi-user website that supports **three** types of users: candidates, lecturers, and admin.

4) The candidates can apply for a tutor and/or lab assistant role, which are considered separate roles for this assignment.

5) A lecturer can only see their assigned courses in the system. If you are attempting only the PA, CR & DI sections, manually assign lecturers to courses, in the backend database.

6) A lecturer may be assigned to more than 1 course. A tutor may apply for role(s) for multiple courses.

7) The TT admin dashboard must be on a separate website. This will be mentioned in the HD section of the assignment.

8) The website **must** be fully styled and look professional. The content must make sense i.e., use of lorem ipsum is **not** allowed.

9) Only well-established and documented third-party libraries/modules **should** be used. If you are in doubt about a module's legitimacy, contact your lecturer for advice.

10) The digital assets (images, icons, audio & video) must be outsourced from royalty-**free** websites. You should **not** steal someone else's assets to enhance the look and feel of your website. High-quality & free assets can be obtained from:

    https://unsplash.com/ (images)
    https://uifaces.co/ (avatars)
    https://fonts.google.com/icons?selected=Material+Icons:home (icons)

The assessment tasks are outlined in the next section→

2.7 Tasks (please ensure you have read the 10 requirements in section 2.6)

To proceed to higher parts, **you must complete** all the specifications in the lower part; you must **not** cherry-pick specifications from various parts. For example, complete all the specifications in the PA part before proceeding to the CR part, and so on. **Use of object-oriented React / React JS /technology stacks other than what is mentioned on page 1 will FETCH A ZERO for the whole assignment.**

Here is the architecture diagram for the website:

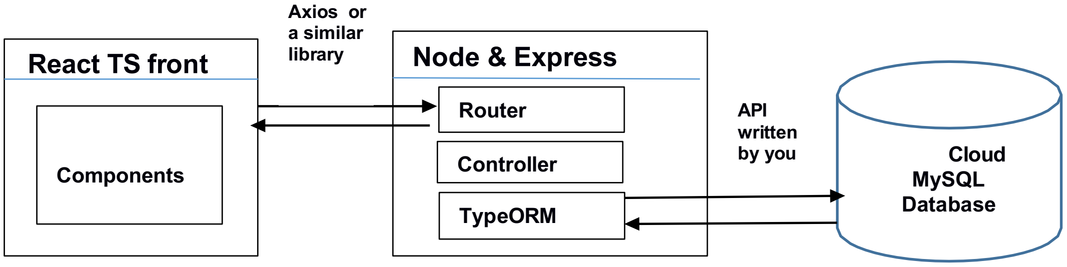


Fig. 1 Full Stack Architecture Diagram

React front-end app talks to API defined in Node + Express layer via Axios library. REST API created by you in Node + Express layer (*backend*) communicates with the database. Create separate projects – *one for React frontend app* & *one for Node + Express + TypeORM* layer.

PA part [23 marks]

a. (5 marks) Database schema model files

Create an ER (Entity-Relationship) diagram that will represent the database schema for the website. It should display the tables with the fields, keys, constraints, and relationships between the tables. **Think of these questions**: *How many tables do I need? Which fields do I need in these tables? What data types should these fields use? What kind of relationships exist among these tables? Is the database normalised? (i.e., avoid duplicated data, do not use too few or too many tables)*.

Place this model diagram inside the Node & Express project. You may use a scanned image or a PDF document.

Using TypeORM in the Node & Express (backend) project, create model files that represent the above tables, keys, and constraints.

b. (6 marks) Sign-up & Sign-in

Implement the sign-up and sign-in features from assignment 1. This time the user details are stored in the MySQL database. The API in the backend should handle all the database operations. All the form input validations must be handled on the React end, as well as being validated in the backend / REST API.

Upon a successful log in, the user must see a welcome message in the format of *Welcome* username.
Introduce a logout link for a logged in user.

Add a new **profile feature**- this component shows a single candidate's information in the main content area. The completed Profile will display user details, and also the date of joining.

ATTENTION postgraduate students (COSC2938): candidate should be able to choose/edit an avatar on the profile page. *How and from where will you source this avatar? Where will you store it? Do you need to store it?*

c. (6 marks) Candidate's page

Implement the tutors page from assessment 1 with the following changes:

CHANGE 1: all the relevant details are stored in the MySQL database. The API in the backend should handle all the database operations. All the form input validations must be handled on the React end, as well as being validated in the backend / REST API.

CHANGE 2: implement the requirements 2.6.4 and 2.6.6 (mentioned on page 4).

d. (6 marks) Lecturer's page

Implement the lecturer's page from assessment 1 with the following changes:

CHANGE 1: all the relevant details are stored in the MySQL database. The API in the backend should handle all the database operations. All the form input validations must be handled on the React end, as well as being validated in the backend / REST API.

CHANGE 2: implement the requirements 2.6.5 and 2.6.6 (mentioned on page 4).

CR [8 marks]

e. Implement the additional features for lecturers from assessment 1 with the following changes:

CHANGE 1: all the relevant details are stored in the MySQL database. The API in the backend should handle all the database operations. All the form input validations must be handled on the React end, as well as being validated in the backend / REST API.

CHANGE 2: Lecturers should be to filter the applicants by candidate's name, type of session applied (tutorial, lab) availability, skill set.

DI [5 marks]

f. Implement the visual representation feature for lecturers from assessment 1 with the following changes:

CHANGE: all the relevant details are stored in the MySQL database. The API in the backend should handle all the database operations. All the form input validations must be handled on the React end, as well as being validated in the backend / REST API.

ATTENTION: Postgraduate students (COSC2938)

Refactor the existing React codebase to clearly separate data visualisation responsibilities from the structural and functional logic of components. *What architectural approach would you adopt to effectively separate visual representation from business logic, in a way that supports efficient future expansion of the codebase? Explain your reason via code comments in the appropriate file*.

Proceed to next page for the HD section →

HD [9 marks]

Note: *this section requires rigour, research, and effort. You will not find direct answers anywhere; you should do some reading on the Internet before devising your own implementations. DO NOT COPY/PASTE solutions from ChatGPT and/or CoPilot. REMEMBER, we will conduct a face-to-face interview to flag a suspicious code.*

Your task is to create an Admin Dashboard with the following requirements:

- **Requirement # 1: Create separate project(s) for this section i.e. admin react front & admin GraphQL. Admin Dashboard should not be linked to TT website.**
- **Requirement # 2: For the data fetching part (from the Cloud MySQL), you must use GraphQL- i.e., use of REST API is not allowed.**
- **Requirement # 3: Admin must login with credentials (admin, admin) to access the dashboard.**

g. (3 marks) An admin must be able to perform the following tasks:

- Assign lecturer to course(s) for the semester
- Add/Edit/Delete courses available in a semester
- Block/unblock login of a potential candidate

h. (3 marks) An admin should be able to generate the following reports:

- List of candidates chosen for each of the courses.
- A candidate chosen for more than 3 courses
- Candidates who have not been chosen for any of the courses

i. (3 marks) IMPLEMENT ONE OF THE FOLLOWING features:

Implement a real-time feature using **GraphQL subscriptions** that allows an admin to notify lecturers when a specific candidate becomes unavailable for hiring at the start of the semester. The candidate's name should appear greyed out with a warning in the frontend TT website. You will receive **no marks if GraphQL subscriptions have not been used.**

OR

Add **6 contextual unit tests in the backend of the project (Node & Express).** Explain the test via code comments in the relevant file.

2.8. Marking Guidelines

The marks allocated have been added to each of the tasks. **However, you will need to read the rubric for details**.