RMIT UNIVERSITY

Full Stack Programming / Further Web Programming
COSC2758/COSC2938 (Semester 1, 2024)
Assignment 1

Assessment Type	To be attempted in a group of 2. • **NO INDIVUAL SUBMISSIONS ALLOWED** • **Undergraduate + Postgraduate groups are NOT ALLOWED** • **You need to register the group via the following URL:** http://tiny.cc/zxcjxz ○ **Failing to submit the group registration will stop you from submitting assessment files via Canvas**
	Submit online via Canvas → Assignments → Assignment 1. 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 5, Sunday 14th April 2024, 11:59 pm *AEST*
Marks	25

1. Overview (you must read this first)

You will use React with JS **OR** React with TS to create a *client-side prototype of the web application*. **No database** is to be used at this stage. **Use of a database will fetch ZERO for assignment 1**.

The server-side implementation and full-stack development of the web application from assignment 1 will be completed in assignment 2.

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

> The DI & HD section tasks will require self-research, you will not get straight answers in the course material. While we are happy to assist you on those tasks, most of the work and research must be done by you. This is done on purpose to prepare for your future work and the rigors 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 1. Software development in real life does not come with a definitive roadmap and flowcharts complete with instructions. More often than so, it is the job of the developer to clarify requirements from the client. For this assignment and course, the lecturer (*Matt Hayward)* is considered as the client.

2. Learning Outcomes

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

- demonstrate proficiency with a web application development framework
- implement a range of techniques and procedures for developing a small to medium-scale web application
- design and manage the development life cycle of a complete application.

3. Assessment details

The COVID-19 pandemic left its devastating mark on many of the businesses, many of which have struggled to survive. One such example is boutique retail food outlets. These small-medium scale businesses have found it very hard to revive back and turn profits. With the ever-growing adoption of online services that deliver food, the operators are forever looking at ways to improve the ways in which they can engage with their customers.

SOIL is a long-term organic food grocer with several store locations around Melbourne. They focus on bringing premium, organic fresh food to the community. In addition to being food grocers, they also offer face to face seminars on diet, nutrition, and small-scale organic farming. In other words, **they are a lot more than an online shop**.

One challenge *SOIL* is facing is that whilst they do a very good job delivering above experience, they have not kept up with the times and have very little technology and online capabilities as part of their business. Their website is obsolete at best and on top of that, they are forced to close some of their stores due to ongoing stiff competition from online food businesses. Their competitors have features like online shopping, forums for discussion, reviews, and search. *SOIL* does not have these features and as a result they are being left behind, with customers preferring to go to other online options because they provide a better user experience including home delivery.

For all the above reasons, you have been approached by a representative of *SOIL* who have secured funding to build a new website experience for their business.

At this stage, the **front-end React website** will be used by the *SOIL* business representatives-only. The details are as follows:

3.1
The website should support the following features:

a. Frontend features of the prototype such as complete UI (user interface) with clearly distinguishable areas: *header, footer, main areas* and *navigation bar*
b. Landing home page displaying information regarding organic foods and nutritional advice
c. Sign up & Sign in pages
d. Special deals page
e. For logged in users
    i. user profile page, edit and delete profile details
    ii. the ability to leave reviews (both number of stars and a written review) for a particular food product
    iii. Feature that enables logged in users to create, edit, delete reviews

NOT all features are to be implemented for assessment 1. Kindy read the tasks on the next page.

3.2
All the data will be stored in HTML5's **localStorage**.
Use of databases (MongoDB, Relational databases and/or Firebase) **is not allowed**.

3.3
The website must be fully styled and look professional. The content must make sense i.e., use of *lorem ipsum* **is not allowed**.

3.4
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)
https://www.flaticon.com/ (more icons)

4. Tasks

To proceed to higher parts, **you must complete** all the specifications in the lower part, you must not cherry pick specifications from various parts. As an example, complete all the specifications in PA part before proceeding to CR part and so on.

Create a client-*side* prototype for the SOIL website with FUNCTIONAL React using JS or TS *with the latest version,* using hooks.

Use of object-oriented or old React or databases will FETCH A ZERO for the whole assignment.

The tasks are shown below:

PA part [13 marks]

a. (3 marks) **Styling and content of pages**

You should **not** use readymade online templates; you MUST create your own template using CSS or any styling library.

> CSS libraries such as Tailwind are allowed, as are libraries such as Chakra UI. Component libraries which provide pre-built navbars and other non-primitive components are not allowed.
>
> The core purpose of this is so that you know how to create these layouts yourself. If you are unsure if a particular library is allowed, please contact your lecturer Matt - matt.hayward@rmit.edu.au

Your web application must have clearly distinguishable areas such as- *header, footer, main content area* and/or other *sections*. These areas must be broken into individual components.

The landing page should display information about organic products sold online by *SOIL* and a navigation bar with appropriate links.

You will be marked on the styling, layout and how components are defined for this part. Writing bloated components, ungainly page design, not following general web design principles would lead to loss of marks.

b. (4 marks) **Sign-up** & **Sign-in**

Signup:

The Signup component will present a form with name, email, and password fields to the user for sign-up.

Password input should **not** be visible in a cleartext format. You may want to have a confirm password field, its implementation is optional.

Sign up form will perform all the necessary validations and then save the user details in localStorage. Validations to be performed-

- *name, email and password are necessary fields.*
- *email must be in a proper format*
- *the password must be a strong password (*look at the definition of a strong password)

Provide a visual cue upon successful registration. The visual cue could be a text or a pop-up message. Upon successful registration, the user should be automatically logged in.

Signin:

The Signin component is also a form with only email and password fields for signing in:

After the proper validations (email and password) and checks have been performed, the user will be redirected to the *profile page.* Provide a visual cue upon successful login. The visual cue could be a text or a pop-up message.

ATTENTION: Postgraduate students (COSC2938)

You must also hash encode the password. Please use a suitable hashing algorithm that is considered secure by the contemporary developer community . *This will require some research on your part.*

c. (2 marks) **Profile**

The Profile component shows a single user's information in the main content area. The completed Profile will display user details, and also the *date of joining*.

d. (4 marks) **Organic product specials & small-scale farming**

Add a component to display all the specials for a week. The information pertaining to specials must be stored in localStorage.

Include a component that displays useful information for the users who want to grow smaller vegetables in their backyard.

You will be marked on the user interface, content, and accuracy of the above information. *Use of lorem ipsum will fetch a zero.*

CR part [4 marks]

e. (4 marks) **Profile management feature**

Modify the profile component to add edit and delete features.

When the user is signed in viewing their own profile, they will be able to see edit and delete options in the Profile component.

The edit allows the authorised user to edit their own profile information in a form similar to the signup form.

The delete feature deletes the user.

Provide a visual cue upon successful edit, delete operations.

Note: what do you think a user should be allowed to edit?

Note: both of the following parts require rigour, research, and effort. You will not find direct answers anywhere; you are to do some reading on the web before coming up with your own implementations.

DI part [3 marks]

f. (3 marks) **Shopping cart**

Add the ability for logged in users to **add items**, **delete items** in a shopping cart, thereby subsequently allowing them to complete the purchase.

Use a simple implementation for credit card validation and its expiry date. There are many ways to do this, you will need to do some reading on your own.

A summary page must be presented to the user after a successful purchase.

NOTE: the product details (quantity, price, etc.) must be stored in localStorage

> **ATTENTION: Postgraduate students (COSC2938)**
>
> Make use of a custom hook, ensuring that it is responsible for all actions relating to the shopping cart.

HD part [5 marks]

Add the following diet related feature to *SOIL* website:

g. The logged in users should be able to create a diet plan. Since this is an HD part, you will need to make sure of the following:

- users can create personalised profiles where they input information such as age, weight, height, activity level, dietary preferences, and health goals.
- goal Setting: Users can set specific goals such as weight loss, muscle gain, or overall health improvement. The system then generates a customized diet plan tailored to their objectives.
- meal Planning: The website offers tools for creating daily or weekly meal plans based on the user's preferences, dietary restrictions, and calorie requirements. Users can choose from a database of recipes or food items.

NOTE: simplistic implementations which do not implement above will receive very low marks. You are allowed to make use of an external REST API (please do not implement your own for Assignment 1). You will also be marked on the user interface.

For submission details, please go to the next page →

5. GitHub & Submission

You will be marked on the use of GitHub and development process. In week 2, you will be emailed an invitation to join the **rmit-fsd-2024-s1** GitHub organisation, you must accept the invitation to join the organisation. Please read the following:

- You must accept the invitation using a GitHub account that has been registered with your RMIT student email address.
- Using a personal GitHub repository and not being a part of the **rmit-fsd-2024-s1** GitHub organisation will lead to ZERO for the GitHub component in the rubric.
- Since you are working in a group include both student IDs in the repository name as:

  <student-id>-<student-id>-a1 for example **s3123456-s3654321-a1**

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

  https://github.com/ rmit-fsd-2024-s1/s3123456-s3654321-a1

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

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

After the due date, you will have 5 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.

Assessment declaration:

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

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

Proceed to next page to read about the use of ChatGPT →

6. Academic integrity and plagiarism (standard warning)

Academic integrity is about honest presentation of your academic work. It means acknowledging the work of others while developing your own 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 publication details so your reader can locate the source if necessary. This includes material taken from Internet sites.

If you do not acknowledge the sources of your material, you may be accused of plagiarism because you have passed off the work and ideas of another person without appropriate referencing, as if they were your own.

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)

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

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

7. Use of ChatGPT

Generative AI tools such as ChatGPT should be approached with caution. The AI is basically re-packaging text that somebody else has written. (We will discuss the issue in class, both in terms of academic standards and because the technology will be important in your careers.) Keep it simple and only submit work which you did yourself. Please ask if you have any questions or are not sure about anything.

For more details concerning referencing anything found via ChatGPT, kindly read:
https://rmit.libguides.com/referencing_AI_tools

8. Marking Guidelines

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