

## React Web Assessment Item

The **React Web Assessment Item** has a **20%** weighting. Important points to note:

- You will also need to do a **Code Review** on your **React Web Project** worth an additional **10%** weighting in your Week 9 Contact.
- You can choose between doing the Standard project or you can choose to do a Custom project. To do a Custom project please email the Course Coordinator, Aneesha (a.bakharia1@uq.edu.au) with your project idea by Wed 4 Sept 2024. The Teaching Team has incorporated opportunities for students to pursue Custom projects, allowing those with existing skills or ideas to utilize this course as a platform for further development. It's an excellent way to elevate your current expertise or bring your innovative concepts to life, guiding you to the next level. Both the Standard and Custom projects need to implement 1 Advanced feature (as detailed in Table 2).
- The React Web Assessment and the React Native Assessment Items are linked. This means that you develop a version for the web and a native version with complimentary functionality for mobile devices. For Custom projects you can't build the same app for Web and Mobile display. If you do a Custom project you need to consider both the Web and Mobile aspects and detail these in your email.
- The React Web project is intended to evaluate your proficiency in designing and developing client-side web applications in React.
- **Table 1** provides an overview of the Standard project and guidelines for getting approval for a Custom project.
- **Table 2** provides the detailed requirements/rubric for the Standard React Web Project. Please make sure you carefully read them and feel free to ask questions.
- **You must come up with your own UI design for your project.** Screen grabs in the Getting Started Guide serve to explain how the app functionality works. You are allowed to use CSS Frameworks such as Bootstrap, Tailwind CSS, Daisy UI and component libraries such as Mantine and Shadcn UI. The Week 7 lecture and content will cover using a CSS Framework.
- You can use open source React Components where detailed in Table 2. You can use other components as long as the components are not high level form building or API calling ones that won't allow you to learn the fundamentals of React. You are encouraged to explore and incorporate other React components but will need to read the documentation yourself.
- You are not building a full SaaS application – No login or Admin Panel is required. For the Standard Project, an API to store and retrieve your data is provided. See the Getting Started Guide slide deck for the API endpoint and example FetchAPI code.
- You can use Typescript if you know how program with it.
- What happens if you don't complete the React Web Assessment Item? How will you be able to complete the React Native project? You will still have access to the API to add records that can be used in the React Native project.

Table 1: Available Project Options

Project Name & Description	Main Features
<b>StoryPath</b>  StoryPath is a location experience platform designed to allow users to create and explore virtual museum exhibits, location-based tours, and treasure hunts with clues. The platform features a Web app built in React that enables users to author these experiences (React Web Assessment Item), and a React Native Player for deploying them (React Native Assessment Item), making it easy to bring location-driven narratives to life.	Allows users to: <ul style="list-style-type: none"> <li>- Create multiple projects and add locations to a project.</li> <li>- Author content and activities that can be triggered by location entry or by scanning a QR code.</li> <li>- Setup flexible scoring options to allow for a variety of location-based experiences (i.e. virtual museum exhibits, location-based tours, and treasure hunts with clues, etc)</li> <li>- Preview and test their location-based experiences from within a Web browser.</li> </ul>
<b>Custom Project Guidelines</b> <ul style="list-style-type: none"> <li>• You must submit your project idea proposal to the Course Coordinator via email by the <b>Wed 4 Sept 2024</b>.</li> <li>• A map must be included in the React Native Project which will complement the React Web Project.</li> <li>• You will need to provide your own API to store and retrieve your data. You can use external services or deploy an API to your own UQ Cloud Zone. If you are already familiar with React, you might like to explore Next.js and build your front-end and back-end with it. The Teaching Team have used PostgREST (<a href="https://docs.postgrest.org/en/v12/">https://docs.postgrest.org/en/v12/</a>) for the API and are happy to provide instructions and share the setup used for the Standard project to quickly make an API.</li> <li>• Example Custom project ideas include Augmented Reality experiences, Renting out your Electric Vehicle Charging station, or Projects that use the camera or other sensors, etc</li> </ul>	

## Code Submission:

You must submit a single **zip file** named 's1234567\_Firstname\_Lastname\_ReactWeb.zip' (replace with your student number) that includes your **source code folder(s)** and a Readme.md detailing where you used GenAI (e.g. ChatGPT or Claude Sonnet 3.5). If you have developed a custom project, please ensure that API keys are provided, so that the marker is able to test your code. Provide 1 example location experience to showcase your app and the advanced feature you have included.

## Additional Questions:

If you have any questions about this assessment brief, you're welcome to post them on the course Ed Discussion and we'll get back to you soon.

## A Message About Plagiarism:

**⚠ Plagiarism is considered a serious offence at UQ.** Failure to declare the distinction between your work and the work of others will result in academic misconduct proceedings.

- The use of Generative AI (i.e. ChatGPT, Claude Sonnet, Google Gemini, Microsoft Bing Chat and Github Copilot) is allowed for this assessment item to assist you in designing your web application and learning new concepts. However, treat what you're producing here as a "trade secret" and don't share your code with other students. Also include details of where Generative AI has been used in a Readme.md file or within your code comments.
- If you're inspired by design or code from online tutorials or any other external source, ensure you reference any inspirations for academic purposes (using APA/IEEE referencing styles) in an Readme.md file.

## The Projects and Location API endpoints for the StoryPath Standard Project:

A RESTful API is provided to provide Create, Read, Update and Delete access to the Project and Location API endpoints. The Getting Started Guide explains the concepts behind a RESTful API and also gives example FetchAPI code. You will need to use the JWT provided via a Blackboard GradeCentre column: "A2 React Web JWT".

**Project:** <https://0b5ff8b0.uqcloud.net/api/project>

Field Name	Data Type	Recommended Form Element	Description
id	INTEGER	None (Auto-generated)	A unique identifier for the project, automatically generated by the database.
title	VARCHAR(255)	Text Input (<input type="text">)	The name of the project, required for submission.
description	TEXT	Textarea (<textarea>)	A brief description of the project, not displayed to participants.
is_published	BOOLEAN	Checkbox (<input type="checkbox">)	Indicates whether the project is published. A checked box means the project is published.
participant_scoring	VARCHAR(100)	Dropdown (<select>)	Select how participants will be scored. Options include: "Not Scored", "Number of Scanned QR Codes", "Number of Locations Entered".
username	VARCHAR(255)	None (Auto-populated)	The username of the project creator. All projects you add need your student username. You don't need to show this on the form. It is used to provide row level security to only allow you to retrieve what you submit.
instructions	TEXT	Textarea (<textarea>)	Instructions for participants, explaining how to engage with the project.
initial_clue	TEXT	Textarea (<textarea>)	The first clue to start the project. This field is optional.
homescreen_display	TEXT	Dropdown (<select>)	Choose what to display on the homescreen of the project. Options include: "Display initial clue", "Display all locations".

**Location:** <https://0b5ff8b0.uqcloud.net/api/location>

Field Name	Data Type	Form Element	Description
location_name	VARCHAR(255)	Text Input (<input type="text">)	The name of the location, required for submission.
location_trigger	VARCHAR(50)	Dropdown (<select>)	Select how this location content will be triggered. Options include: "Location Entry" "QR Code Scan" "Both Location Entry and QR Code Scan".
location_position	VARCHAR(100)	Text Input (<input type="text">)	Enter the latitude and longitude for this location. This is the format: (27.4975,153.013276)
score_points	INTEGER	Number Input (<input type="number">)	Specify the number of points participants earn by reaching this location, required for submission.
clue	TEXT	Textarea (<textarea>)	Enter the clue that leads to the next location. Entry is optional.
location_content	TEXT	Rich Text Editor (<ReactQuill>)	Provide additional content that will be displayed when participants reach this location. User's can format text and add images. Images must be stored as base64 when the HTML is stored. You need to ensure the images are small in file size and in dimensions. You don't need to provide an image editor, but when building sample content, you should use another editor to resize.

## The Preview Functionality:

The Preview feature allows users to test and simulate the experience of going to locations or scanning QR codes within the app. This helps users visualize how the app will behave in real-world scenarios, including the accumulation of points and tracking of locations visited.

The Preview uses the `project.instructions`, `project.participant_scoring`, `project.initial_clue`, `project.homescreen_display`, `location.score_points` and `location.clue` fields to work out what is displayed on the Homescreen and the Location pages.

- The Preview is displayed within a container sized to resemble a mobile device, allowing users to test and simulate the experience of going to locations or scanning QR codes. It includes a dropdown menu that lets users easily switch between different locations, helping them visualize how the app will behave in real-world scenarios, including the accumulation of points and tracking of locations visited.
- The Homescreen needs to display a header with the Project title, `project.instructions`, the `project.initial_clue` if `project.homescreen_display` is set to "Display initial clue" or display all locations if "Display all locations" is selected.
- Changing the dropdown must load the content at the location. If a `location.clue` has been entered it should also be displayed. Use `project.participant_scoring` to work out how to score changing between Locations.
- All Locations including the Homescreen must display a Score count (depending upon `project.participant_scoring`) and Locations visited count.
- Visits and scores should be tracked within React and will be lost when the user leaves the Preview. You don't need to track the data yet – in your A3 React Native project you will work on tracking.

## React Web Rubric

Table 2 includes the grade breakdown for each required feature. Select 1 advanced feature worth 10 grade points. Note the assessment will be marked out of 100 but then scaled to be out of 20. "Custom" is used in the rubric to allow the rubric to also apply to custom projects.

**Table 2: Project functionality and grade breakdown.**

ID	Feature Description	Max Grade = 100
<b>1</b>	<b>Core Functionality</b>	<b>55 marks</b>
1.1	Package.json includes all dependencies, React code runs without errors, follows required folder naming convention and includes an example Project to illustrate the chosen Advanced Feature.	5
1.2	Includes the ability to list Projects, add, edit and delete Projects   Custom. No paging is required. Add/Edit forms should include all mandatory fields and dropdowns provided by the Projects API endpoint.	10
1.3	Includes the ability to list Locations, add, edit and delete Locations   Custom. No paging or re-ordering is required. Add/Edit forms should include all mandatory fields and dropdowns provided by the Locations API endpoint.	15
1.4	Includes a WYSIWYG editor on the Locations form that support a variety of formatting options (Bold, Italic, changing fonts, colours and background colours), headings (e.g. H1), links, images (inserted via Base64), etc. You can use a React WYSIWYG editor that supports these features such as Quill or Lexical. Your chosen editor should save content to HTML – this will make it easier when you use this content in your React Native project.	5
1.5	Allows for QR codes to be generated for all Locations. Any React QR Code library can be used such as react-qr-code.	5
1.6	Includes a Preview which allows users to test their location-based project and simulates the movement/scanning of QR code to view content at a location or scanned via QR code. The Preview should implement all scoring and display options.	15
<b>2.</b>	<b>User Interface Design</b>	<b>10 marks</b>
2.1	Uses a CSS Frameworks and Custom CSS where appropriate. Includes a landing page with a header, footer, hero image and links to the Projects   Custom Section. No Login is required.	5



2.2	UI is intuitive with all implemented features easy to use and user instructions provided.	5
<b>3</b>	<b>Code Style and Quality</b> (Note: Max Grade will be capped at 12.5 grade points if only 50% of functionality is implemented)	<b>25 marks</b>
3.1	JSX and Javascript/Typescript adheres to Functional paradigm (i.e. avoids embedded imperative logic).	5
3.2	Breaks down functionality into well-defined, manageable components.	5
3.3	Centralised Fetch API access, reducing duplicate code in each component that uses the API.	5
3.4	Implements appropriate error handling for each component, ensuring robust error detection and management while displaying clear, user-friendly error messages that enhances user experience and facilitates troubleshooting.	6
3.5	Well formatted and documented code with inline comments and function doc strings included.	4
<b>4</b>	<b>Advanced Features (Select 1 advanced feature to equal to 10 grade points. Multiple advanced feature items will not be graded.)</b>	<b>10 marks</b>
4.1	Add a Location Picker to the Location form that allows the selection of Long and Lat coordinates from a map. You can use an open source React component e.g. React Leaflet <a href="https://react-leaflet.js.org/">https://react-leaflet.js.org/</a>	10
4.2	Add a map option for Homescreen display (i.e. add a map option in the project.homescreen dropdown). This will display the location markers on a map on the Homescreen in the Preview. You can use React Leaflet <a href="https://react-leaflet.js.org/">https://react-leaflet.js.org/</a>	10
4.3	Add a Multiple Choice Quiz Question that can be added as a question on the Locations screen. This question should be included in scoring with an additional scoring option for the project. You can store the question as json in the location.extra field.	10
4.4	Allow other media formats to be embedded in the WYSIWYG editor or via another form field that saves to location.extra. Ideas here could include 3D models or slideshows. As most WYSIWYG editors support embedding YouTube videos, this would not be considered an advanced feature.	10
4.5	Use GenAI to help build a location experience e.g. help the user brainstorm a whole location experience and generate clues or review the whole experience and provide feedback. Simple prompts to just add a single item of content would be considered to easy given the capabilities of GenAI models. API keys or ChatGPT 4o mini and Claude Haiku are available as long as you won't publish your client-side code that includes the API keys (i.e. not secure as its client-side).	10