

React Web Assessment Item

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

- You will also need to do a **Project Demonstration and Code Review** for your **React Web Project** worth an additional **8%** weighting in Week 10.
- You can choose between doing the Standard project or a Custom project. To do a Custom project please email (COMP2140@eecs.uq.edu.au) with your project idea by Mon 8 Sept 2025. 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.
- The React Web project is intended to evaluate your proficiency in designing and developing client-side web applications in React. You are given access to a RESTful API for saving and retrieving data.
- **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.** You are allowed to use CSS Frameworks such as Bootstrap, Tailwind CSS, Daisy UI and component libraries such as Mantine and Shadcn UI.
- You can use open source React Components where detailed in the Table 2 Rubric. 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.

Table 1: Available Project Options

Project Name & Description	Main Features
<p>ReadySetHire - AI-Powered Interview Platform</p> <p>ReadySetHire is a modern web application designed to streamline the hiring process. It provides recruiters with an easy way to set up interviews, add interview questions, invite applicants, and automatically capture their spoken responses through speech-to-text transcription. By using the latest AI technology, ReadySetHire allows applicants to answer naturally in spoken language, making the experience more authentic and efficient.</p> <p>This tool can be used for screening applicants at scale and includes opportunities to extend functionality with Generative AI, such as providing feedback or auto-suggesting interview questions. In the future, ReadySetHire could also serve as a practice platform for job seekers, helping them prepare for interviews with realistic AI-driven simulations.</p>	<p>Allows users to:</p> <ul style="list-style-type: none"> - Create, edit, and delete Interviews - Add interview questions - Invite applicants through unique interview links - Applicants answer using an audio recorder, with AI transcription converting speech to text - One question per page interview flow for clarity and focus - Track interview status (Not Started and Completed) - Review Applicant question answers - Creative Integration of GenAI
<p>Custom Project Guidelines</p> <ul style="list-style-type: none"> • You must submit your project idea proposal to COMP2140@eecs.uq.edu.au by Mon 8 Sept 2024. • 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 (https://docs.postgrest.org/en/v12/) for the API and a self-paced tutorial is provided, if you also wish to take this approach. With PostgREST you only need to setup a database and then enable the RESTful API. • Custom projects can also be a Chrome or VS Code extension but must use React. 	

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 or Gemini). If you have developed a custom project, please ensure that API keys are provided, so that the marker is able to test your code.

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, Claude Code, Cursor, etc) 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.

Standard Project Detailed Specification

Table 2 contains the detailed specification for each required screen in the Standard Project. Note: Screens/Functionality are listed below however you don't need to stick to the screen structure as you have flexibility with your design (e.g. Questions could be a separate screen or embedded inline on the Interview Add/Edit form).

Table 2: Detailed Specifications

Screen/Functionality	Key Required Features
App design and Navigation	<ul style="list-style-type: none"> Consistent Header Clear application navigation and workflow Appropriate use of colour, images, icons and instructions Footer
Interviews	<ul style="list-style-type: none"> List of Interviews with main details displayed (either in a table rows or cards) Ability to Add, Edit and Delete an Interview Main interview form fields: Title, Job Role, Description, Status (i.e. Published or Draft). All fields are required. Ability to see Number of Questions attached to the Interview and link to the Questions Screen Ability to see Number of Applicants and their Interview Status (i.e. Not Started or Completed) and link to the Applicants Screen
Questions	<ul style="list-style-type: none"> List of Interview Questions with main details displayed (either in a table rows or cards) Ability to Add, Edit and Delete an Interview Question Main interview form fields: Question and Difficulty (i.e. Easy, Intermediate & Advanced). All fields are required.
Applicants	<ul style="list-style-type: none"> List of Applicants with main details displayed (either in a table rows or cards) Ability to Add and Edit an Applicant Main Applicant form fields: Title, Firstname, Surname, Phone Number, Email Address and Interview Status (i.e. Not Started or Completed). All fields are required.

	<ul style="list-style-type: none"> • Generate and Copy a link for an Applicant to take an Interview • View Answers to Interview questions for each Applicant
Take Interview UI	<ul style="list-style-type: none"> • Welcome Screen (include Details of Applicant, Interview and Instructions) • Display 1 Question per Page and include a Next Question Navigation Button. The Applicant should not be able to return to a previous question. • Include an Audio Recorder that allows for pausing. The Applicant should not be able to restart the recording. The Speech should be transcribed to text and saved as the answer to the Answer endpoint. Transformers.js (demo'ed in the Week 7 Lecture can be used for transcription) or a custom Express.js endpoint that runs a Whisper CLI can be used (demo'ed in the Week 7 Lecture). • Displays a "Thank you for completing the interview" message after Interview has been completed
Advanced GenAI Feature	<ul style="list-style-type: none"> • Integrate GenAI in a creative way in the project. • Suggested ideas include: Generating interview questions, providing applicants with feedback or providing a summary of applicant answers to the HR or Hiring Manager. • Custom projects will also need to implement a GenAI feature. • You will be provided with an API key to make Large Language Model (LLM) Calls