COMP2140/7240 – Web/Mobile Programming

Lecture Week 6: Introduction to A2 & Getting Started

Dr Aneesha Bakharia (Senior Lecturer, EECS) Email: a.bakharia1@uq.edu.au

React Web App

Mode Product/ Artefact/ Multimedia

Category Computer Code, Project

Weight 20%

Due date 26/09/2025 5:00 pm

Learning

outcomes L01, L02, L04, L05, L06, L07, L08

Task description

In the *React Web App* assessment, students will build an interactive web app with a cross-platform JavaScript framework (React), testing across web browsers & device sizes. The web app will communicate with web & device APIs. Throughout, students will evaluate the design, implementation & architecture for delivering a complex web app.

The React Web App will be graded in the associated Project Demonstration and Code Review session. The React Web App must be submitted prior to the Project Demonstration and Code Review session.

Artificial Intelligence (AI) and Machine Translation (MT) are emerging tools that may support students in completing this assessment task. Students may appropriately use AI and/or MT in completing this assessment task. Students must clearly reference any use of AI or MT in each instance.

A failure to reference generative AI or MT use may constitute student misconduct under the Student Code of Conduct.

React Web App

Mode Product/ Artefact/ Multimedia

Computer Code, Project Category

Weight 20%

Due date 26/09/2025 5:00 pm

Learning

L01, L02, L04, L05, L06, L07, L08 outcomes

Task description

In the React Web App assessment, students will build an interactive web app with a cross-platform JavaScript framework (React), testing across web browsers & device sizes. The web app will communicate with web & device APIs. Throughout, students will evaluate the design, implementation & architecture for delivering a complex web app.

Compared to COMP2140, COMP7240 includes an additional research component in the assessment, requiring students to critically investigate current practices and challenges in accessibility and responsive design.

The React Web App will be graded in the associated Project Demonstration and Code Review session. The React Web App must be submitted prior to the Project Demonstration and Code Review session.

Artificial Intelligence (AI) and Machine Translation (MT) are emerging tools that may support students in completing this assessment task. Students may appropriately use AI and/or MT in completing this assessment task. Students must clearly reference any use of AI or MT in each instance.

A failure to reference generative AI or MT use may constitute student misconduct under the Student Code of Conduct.

COMP7240 has a different weighting and an additional task

What do you need to build?

Table 1: Available Project Options

Project Name & Description Main Feat	tures
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 Al technology, ReadySetHire allows applicants to answer naturally in spoken language, making the experience more authentic and efficient. This tool can be used for screening applicants at scale and includes - Add Invited Set Up interviews, add - Add	ers to: eate, edit, and delete Interviews d interview questions ite applicants through unique interview links clicants answer using an audio recorder, with AI transcription everting speech to text e question per page interview flow for clarity and focus ck interview status (Not Started and Completed) view Applicant question answers eative Integration of GenAI

Custom Project Guidelines

- You must submit your project idea proposal to <u>COMP2140@eecs.uq.edu.au</u> 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.

Example Screens

Some Guidelines to for viewing Sample Screens

- 1. They are sample screens.
- 2. You need to design your own layout and look and feel using a CSS Framework.
- 3. The sample screens are not designed for usability and don't include too many user instructions. You will need to include these in your app.

The Interviews Screen

ReadySetHire - Al-Powered Interview Platform Interviews **Interviews Management** + Add Interview **Job Role Questions Applicants** Actions Title **Status** ≈ 2 Applicants Frontend Developer Interview Frontend Developer ② 3 Questions Published 1 Completed 1 Pending 은 O Applicants **Backend Engineer Assessment** 0 ⑪ **Backend Engineer** ② 0 Questions Draft

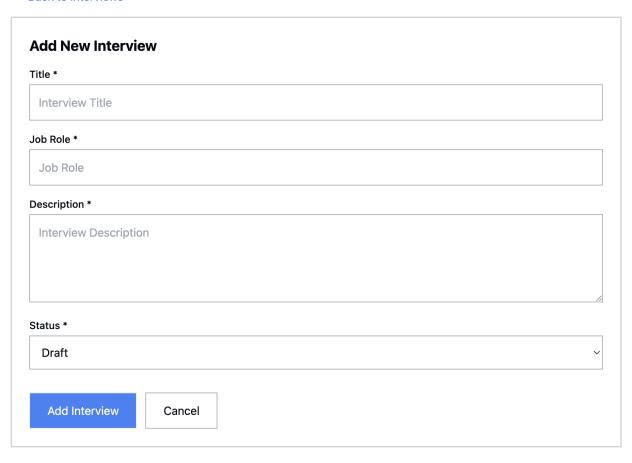
© 2025 ReadySetHire - Streamlining the hiring process with AI

The Add Interview Form

ReadySetHire - Al-Powered Interview Platform

Interviews

← Back to Interviews





The Questions Screen

ReadySetHire - AI-Powered Interview Platform

Interviews

← Back to Interviews

Questions Management

Interview: Frontend Developer Interview

+ Add Question

Question	Interview	Difficulty	Actions
Tell me about your experience with React	Frontend Developer Interview	Easy	
How do you handle state management?	Frontend Developer Interview	Intermediate	
Explain the virtual DOM	Frontend Developer Interview	Advanced	1

The Applicants Screen

ReadySetHire - AI-Powered Interview Platform

Interviews

← Back to Interviews

Applicants Management

Interview: Frontend Developer Interview

+ Add Applicant

Name	Email	Phone	Interview	Status	Actions
Mr John Smith	john@email.com	555-0123	Frontend Developer Interview	Completed	⊘ Copy Link Take Interview ☑ 🖔
Ms Sarah Johnson	sarah@email.com	555-0456	Frontend Developer Interview	Not Started	⊘ Copy Link Take Interview ☑ 🛚

© 2025 ReadySetHire - Streamlining the hiring process with Al

The Take Interview UI

ReadySetHire - AI-Powered Interview Platform

Interviews

Welcome to Your Interview

Applicant Details:

Ms Sarah Johnson sarah@email.com

Interview:

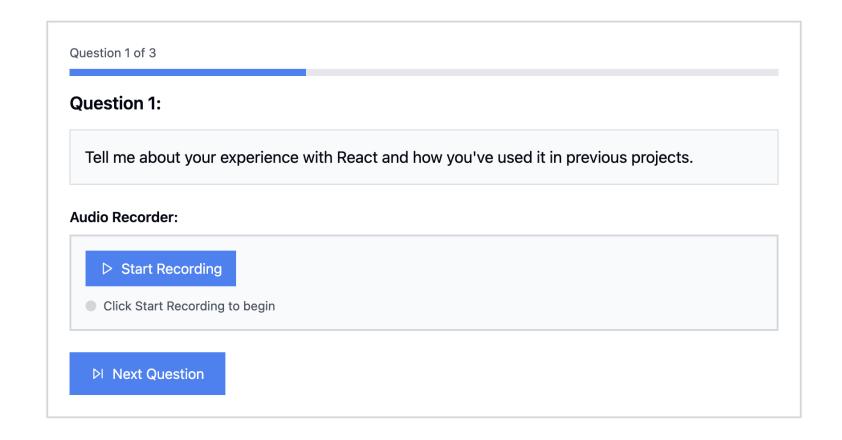
Frontend Developer Interview Interview for React/JavaScript frontend position

Start Interview

The Take Interview UI - Question

ReadySetHire - Al-Powered Interview Platform

Interviews



The Take Interview UI – Thank You

ReadySetHire - Al-Powered Interview Platform

Interviews



Thank You for Completing the Interview!

Your responses have been successfully recorded.

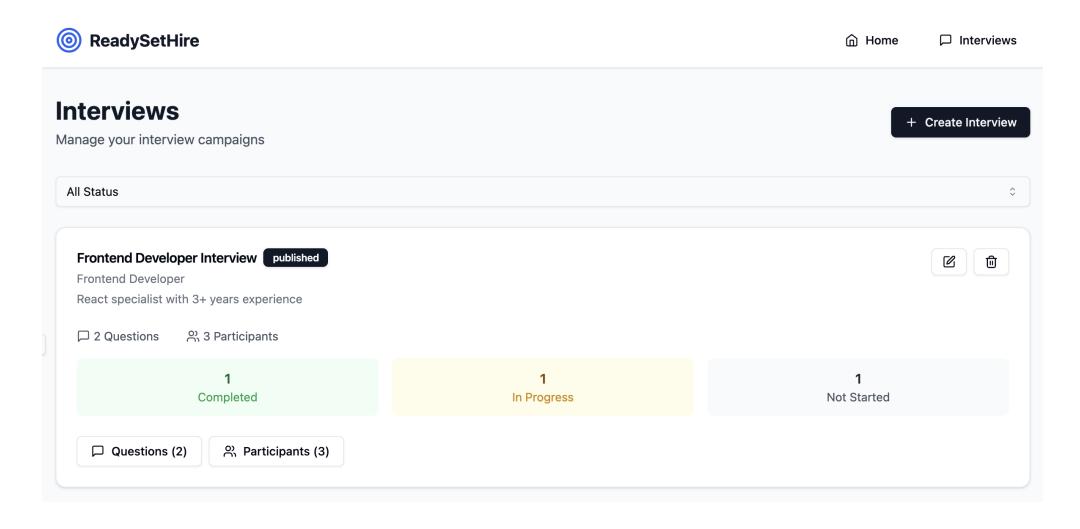
Our team will review your interview and get back to you soon.

You may now close this window.

Interview: completed for: Sarah Johnson Interview: Frontend Developer Interview

You have design flexibility... (Alternate Design)

Marks Allocated in the Rubric



How do I get started?

How do I get started? (part 1)

- 1. Read the Assessment Brief & Rubric
- 2. Use Vite to create the starting React Project e.g.

 npm create vite@latest interview-app -- --template react cd interview-app

 npm install

 npm run dev
- 3. Review and understand the file structure of the generated app.
- 4. Design your UI using a CSS Framework. Bootstrap, Tailwind CSS, Daisy UI, Mantine, Shdcn UI are all allowed. It is a good idea to create HTML mockups of each screen or prototype using React components
- 5. Install and setup React Router (Lecture 5 slides & Week 6 Applied Class)

 npm install react-router-dom
- 6. Split up the HTML and make components. Make the layout.

 Move your HTML into the components. Setup routes for each page/screen.

How do I get started? (part 2)

- 7. Learn to use a RESTful API (overview in next slide)
 This can be challenging so we provide some sample code to get you started! Get your token from BB My Grades
- 8. Build the Interview, Question and Applicant List Screens
- 9. Build the Add/Edit Forms for each
- 10. Build the Take Interview UI (with the Audio to Text transcription)
 The Week 7 Lecture will cover Audio to Text transcription!
- 11. Build your Advanced Feature most importantly have fun!

4	Advanced GenAl Feature	10 marks	10 marks
4.1	Creative integration of GenAl into the Interview Application	10	10

12. COMP7240 – Extra Research Task

5	Responsible AI (For COMP7240 Students – Research Question)	10 marks
5.1	Write a short bullet-point outline highlighting the responsible AI and accessibility considerations involved in using speech-to-text translation for hiring interviews at an organisation.	10

RESTFul API's

What is a RESTful API?

- Representational State Transfer
- Architectural style for distributed systems

Stateless Communication

Each request from a client contains all the information needed for processing

Resource-Based

- Resources are identified by URLs
- Once you know the resource name the process of listing, creating, editing and deleting is the same

HTTP Methods

GET, POST, PATCH/PUT, DELETE

Interview App RESTful API

https://comp2140a2.uqcloud.net/api

React Web Project Assessment – RESTFul API

This document provides a summary of the available RESTful API endpoints .

API Overview

• URL: https://comp2140a2.uqcloud.net/api

Authentication

Anonymous access is disabled. Authentication via an JSON Web Token (JWT) is required for all endpoints. The JWT is provided via a column in Blackboard My Grades. Example API calling code is provided to get you started.

Interview Endpoint

The /interview endpoint allows management of interview resources.

Interview Fields

Field Name	Type	Size	Required	Description	Example Value
id	Integer	-	Yes (PK)	Auto-generated identifier	1
title	String	255	Yes	Title of the interview	"Front-end Developer Interview"
job role	String	255	Yes	Role being interviewed for	"Senior Front-end Developer"
description	Text	-	No	Detailed description	"Interview for candidates with React experience"
status	String	50	Yes	Publication status	"Published", "Draft", or "Archived"
username	String	255	Yes	Owner of the interview	"s10000" (This is your student id)

- Implemented using PostgREST which maps a Database to an API.
- https://docs.postgrest.org/en/v12/

Sample – Create, Read, Update Delete (CRUD)

- You can perform CRUD (Create, Read, Update, Delete) operations on a resource (eg Project) using various HTTP methods.
- Getting all Interviews (GET)

https://comp2140a2.uqcloud.net/api/interview

- Reading an Interview by id (GET)
 https://comp2140a2.uqcloud.net/api/interview?id=eq.1
 Note this is slightly different from usual RESTFul but provides more querying capability.
 See Operators section: https://docs.postgrest.org/en/v12/references/api/tables-views.html
- Create a new Interview (POST)
- Edit an Interview (PUT/PATCH)
 Note PostgREST handles this differently. Read the docs.

Example API Code

Don't forget to include/change to your JWT from Course GradeCentre

```
import fetch from 'node-fetch';
     // Base URL for the Interview App RESTful API
     const API_BASE_URL = 'https://comp2140a2.uqcloud.net/api';
    // JWT token for authorization, replace with your actual token from My Grades in Blackboard
     // From the A2 JSON Web Token column, view Feedback to show your JWT
     const JWT_TOKEN = 'YOUR TOKEN HERE';
     // Your UQ student username, used for row-level security to retrieve your records
     const USERNAME = 's1000000';
13
      * Helper function to handle API requests.
      * It sets the Authorization token and optionally includes the request body.
      * @param {string} endpoint - The API endpoint to call.
      * Oparam {string} [method='GET'] - The HTTP method to use (GET, POST, PATCH).
      * @param {object} [body=null] - The request body to send, typically for POST or PATCH.
      * @returns {Promise<object>} - The JSON response from the API.
      * @throws Will throw an error if the HTTP response is not OK.
     async function apiRequest(endpoint, method = 'GET', body = null) {
       const options = {
         method, // Set the HTTP method (GET, POST, PATCH)
         headers: {
           'Content-Type': 'application/json', // Indicate that we are sending JSON data
           'Authorization': `Bearer ${JWT_TOKEN}` // Include the JWT token for authentication
       // If the method is POST or PATCH, we want the response to include the full representation
       if (method === 'POST' || method === 'PATCH') {
         options.headers['Prefer'] = 'return=representation';
       // If a body is provided, add it to the request and include the username
         options.body = JSON.stringify({ ...body, username: USERNAME });
```

Sample code in interviewapp_sample_api_access.zip

Tips

How can I approach building the all the screen in an efficient manner?

Centralise your API handling code.

Build both List screens and then both Add/Edit forms. Re-use the form for both adding and editing.

What components can I use?

The Rubric lists example open source React components. You can use other components as long as the components are not high level form building or API calling so that they don't allow you to learn the fundamentals of React. It is up to you to review the documentation.

You'll learn about Forms and RESTful API's in the Week 7 Lecture.



A2 Q&A



Thank you



