College of Engineering, Construction & Living Sciences
Bachelor of Information Technology
ID608001: Intermediate Application Development Concepts
Level 6, Credits 15

Project 1: Next.js Hacker News App

Assessment Overview

In this **individual** assessment, you will develop an application using **Next.js** & the **Hacker News API**. The main purpose of this assessment is to demonstrate your ability to develop an application using various taught concepts. In addition, marks will be allocated for code elegance, documentation & **Git** usage.

Learning Outcome

At the successful completion of this course, learners will be able to:

1. Apply design patterns & programming principles using software development best practices.
2. Design & implement full-stack applications using industry relevant programming languages.

Assessments

Assessment	Weighting	Due Date	Learning Outcomes
Practical: Skills-Based	20%	31-03-2023 (Fri at 10.00 AM)	1
Project 1: Next.js Hacker News App	30%	27-04-2023 (Thur at 4.59 PM)	1 & 2
Project 2: Node.js & Express Pub Quiz App	50%	15-06-2023 (Thur at 4.59 PM)	1 & 2

Conditions of Assessment

You will complete this assessment during your learner-managed time. However, there will be time during class to discuss the requirements & your progress on this assessment. This assessment will need to be completed by **Thursday, 27 April 2023** at **4.59 PM**.

Pass Criteria

This assessment is criterion-referenced (CRA) with a cumulative pass mark of **50%** over all assessments in **ID608001: Intermediate Application Development Concepts**.

Authenticity

All parts of your submitted assessment **must** be completely your work. Do your best to complete this assessment without **ChatGPT**. You need to demonstrate to the course lecturer that you can meet the learning outcome for this assessment.

However, if you get stuck, you can use **ChatGPT** to help you get unstuck, permitting you acknowledge that you have used **ChatGPT**. In the assessment's repository **README.md** file, please include what prompt(s) you provided to **ChatGPT** & how you used the response(s) to help you with your work. It also applies to code snippets retrieved from **StackOverflow** & **GitHub**. Failure to do this will result in a mark of **zero** for this assessment.

Policy on Submissions, Extensions, Resubmissions & Resits

The school's process concerning submissions, extensions, resubmissions & resits complies with **Otago Polytechnic** policies. Learners can view policies on the **Otago Polytechnic** website located at https://www.op.ac.nz/about-us/governance-and-management/policies.

Submission

You **must** submit all project files via **GitHub Classroom**. Here is the URL to the repository you will use for your submission – https://classroom.github.com/a/T9vHopU9. Create a **.gitignore** & add the ignored files in this resource - https://raw.githubusercontent.com/github/gitignore/main/Node.gitignore. The latest project files in the **master** or **main** branch will be used to mark against the **Functionality** criterion. Please test before you submit. Partial marks **will not** be given for incomplete functionality. Late submissions will incur a **10% penalty per day**, rolling over at **5:00 PM**.

Extensions

Familiarise yourself with the assessment due date. If you need an extension, contact the course lecturer before the due date. If you require more than a week's extension, a medical certificate or support letter from your manager may be needed.

Resubmissions

Learners may be requested to resubmit an assessment following a rework of part/s of the original assessment. Resubmissions are to be completed within a negotiable short time frame & usually **must** be completed within the timing of the course to which the assessment relates. Resubmissions will be available to learners who have made a genuine attempt at the first assessment opportunity & achieved a **D grade (40-49%)**. The maximum grade awarded for resubmission will be **C-**.

Resits

Resits & reassessments **are not** applicable in **ID608001: Intermediate Application Development Concepts**.

Instructions

You will need to submit an application & documentation that meet the following requirements:

Functionality - Learning Outcomes 1, 2, 3 (40%)

- **App:**
  - Display a drop down with the following options:
    * Ask Stories

    - Best Stories
    - Job Stories
    - New Stories
    - Show Stories
    - Top Stories
  - The endpoints for the stories are as follows:
    - Ask Stories - https://hacker-news.firebaseio.com/v0/askstories.json?print=pretty
    - Best Stories - https://hacker-news.firebaseio.com/v0/beststories.json?print=pretty
    - Job Stories - https://hacker-news.firebaseio.com/v0/jobstories.json?print=pretty
    - New Stories - https://hacker-news.firebaseio.com/v0/newstories.json?print=pretty
    - Show Stories - https://hacker-news.firebaseio.com/v0/showstories.json?print=pretty
    - Top Stories - https://hacker-news.firebaseio.com/v0/topstories.json?print=pretty
  - When an option is selected, display the title of the first 50 stories.
    - Style each story to look like a card.
    - Display five stories per row.
  - When a story is clicked, the user will be navigate to a page that displays the following information:
    - By
    - Kids. **Note:** This is an array of ids. If the array is empty, display **N/A**. Display the first five ids as URLs in this format: https://hacker-news.firebaseio.com/v0/item/¡Id¿.json?print=pretty. When clicked, these URLs will open in a new tab.
    - Score
    - Time. **Note:** Convert the time to a readable format.
    - Title
    - Type
    - URL. **Note:** When clicked, this URL will open in a new tab.

    This information is fetched from the following endpoint: https://hacker-news.firebaseio.com/v0/item/¡Id¿.json?print=pr
  - On a page, display the top 20 leaders. You can get this information from the following URL: https://news.ycombinator.co **Note:** You need to manually retrieve the information from the URL.
  - When a leader is clicked, the user will be navigate to a page that displays the following information:
    - About
    - Created. **Note:** Convert the time to a readable format.
    - Id
    - Karma
    - Submitted. **Note:** This is an array of ids. Display the first 10 ids as URLs in this format: https://hacker-news.firebaseio.com/v0/item/¡Id¿.json?print=pretty.

    This information is fetched from the following endpoint: https://hacker-news.firebaseio.com/v0/user/¡Id¿.json?print=pr
  - Deployed to **Vercel**. **Note:** Use your **GitHub** account to login to **Vercel**.

- **Testing:**
  - **Component tests** are written using **React Testing Library** & **Jest**.
  - At least 15 **component tests** verifying the app functionality.

- **NPM Scripts:**
  - Linting & fixing your code using **ESLint**.
  - Formatting your code using **Prettier**.
  - Running **component tests** using **React Testing Library** & **Jest**.

Code Elegance - Learning Outcome 1 (45%)

- Environment variables' key is stored in the **env.example** file.
- Variables, functions & components are named appropriately.
- Idiomatic use of control flow, data structures & in-built functions.
- Sufficient modularity.
- Each **component** & **page** file **must** have a header comment located immediately before the **import** statements.
- In-line comments where required.
- Code is linted & formatted using **ESLint** & **Prettier**.
- No dead or unused code.
- **React Testing Library**, **Jest**, **ESLint**, **Prettier** & **Commitizen** are installed as **development dependencies**.

Documentation & Git/GitHub Usage - Learning Outcomes 2, 3 (15%)

- **GitHub** project board to help you organise & prioritise your work.
- Provide the following in your repository **README.md** file:
  - URL to your application on **Vercel**.
  - How do you setup the development environment, i.e., after the repository is cloned, what do you need to do before you run the application?
  - How do you lint & fix your code?
  - How do you format your code?
  - How do you run your **component tests**?
- Use of **Markdown**, i.e., headings, bold text, code blocks, etc.
- Correct spelling & grammar.
- Your **Git commit messages** should:
  - Reflect the context of each functional requirement change.
  - Be formatted using an appropriate naming convention style using **Commitizen**.

Additional Information

- **Do not** rewrite your **Git** history. It is important that the course lecturer can see how you worked on your assessment over time.