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

Assessment Overview

In this **individual** assessment, you will develop a three frontend applications. In addition, marks will be allocated for code quality and best practices, documentation and **Git** usage.

Learning Outcomes

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

1. Apply design patterns and programming principles using software development best practices.

2. Design and implement full-stack applications using industry relevant programming languages.

Assessments

Assessment	Weighting	Due Date	Learning Outcome
Practical	20%	13-11-2024 (Wednesday at 4.59 PM)	1
Project	80%	13-11-2024 (Wednesday 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 and your progress on this assessment. This assessment will need to be completed by **Wednesday, 13 November 2024** 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 using an **AI generative tool**. You need to demonstrate to the course lecturer that you can meet the learning outcome(s) for this assessment.

However, if you get stuck, you can use an **AI generative tool** to help you get unstuck, permitting you to acknowledge that you have used it. In the assessment's repository **README.md** file, please include what prompt(s) you provided to the **AI generative tool** and how you used the response(s) to help you with your work. It also applies to code snippets retrieved from **StackOverflow** and **GitHub**.

Failure to do this may result in a mark of **zero** for this assessment.

Policy on Submissions, Extensions, Resubmissions and Resits

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

Submission

You **must** submit all application files via **GitHub Classroom**. Here is the URL to the repository you will use for your submission – https://classroom.github.com/a/eGYqq7-g. If you do not have not one, create a **.gitignore** and add the ignored files in this resource - https://raw.githubusercontent.com/github/gitignore/main/Node.gitignore. Create a branch called **project**. The latest application files in the **project** 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. Extensions will **only** be granted if you are unable to complete the assessment by the due date because of **unforeseen circumstances outside your control**. The length of the extension granted will depend on the circumstances and **must** be negotiated with the course lecturer before the assessment due date. A medical certificate or support letter may be needed. Extensions will not be granted on the due date and for poor time management or pressure of other assessments.

Resits

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

Instructions

Functionality - Learning Outcomes 1 and 2 (50%)

Milestone One - Due Wednesday, 11 September 2024 (Week 8) at 4.59 PM.
Code Defence - Thursday, 12 September 2024 (Week 8) at 10.00 AM.

- **Movie Listings Application (10%):**

  – Create a new directory called **movie-listings-application** and create a new **React** application using **Vite**.
  – Can run in development without modification.
  – Use **React Query** and the **Fetch API** to fetch data from the **The Movie DB API**.

- Navigation bar using **Tailwind CSS** and **Shadcn UI** with the following movie type options:
  * Trending. This is the default option.
  * Top Rated
  * Action
  * Animation
  * Comedy

  Note: You will use **React Router** to navigate between the options.

- The endpoints for the movie types are as follows:
  * Trending - https://api.themoviedb.org/3/trending/all/week?api_key=<API KEY>&language=en-US
  * Top Rated - https://api.themoviedb.org/3/movie/top_rated?api_key=<API KEY>&language=en-US
  * Action - https://api.themoviedb.org/3/discover/movie?api_key=<API KEY>&with_genres=28
  * Animation - https://api.themoviedb.org/3/discover/movie?api_key=<API KEY>&with_genres=16
  * Comedy - https://api.themoviedb.org/3/discover/movie?api_key=<API KEY>&with_genres=35

  Note: Replace <API KEY> with a **The Movie DB API** key given by your course lecturer.

- When an option is selected, display the title, overview, poster path and release date of the first 10 movies.
  * Style each movie to look like a card using **Tailwind CSS** and **Shadcn UI**.
  * Display five movies per row.

Milestone Two - Due Wednesday, 16 October 2024 (Week 11) at 4.59 PM.
Code Defence - Thursday, 17 October 2024 (Week 11) at 10.00 AM.

- **Hacker News Application (20%):**

  - Create a new directory called **hacker-news-application** and create a new **React** application using **Vite**.
  - Can run in development without modification.
  - Use **React Query** and **GraphQL** to fetch data from the **Hacker News API**.
  - Navigation bar using **Tailwind CSS** and **Shadcn UI** with the following story options:
    * Ask Stories. This is the default option.
    * Best Stories
    * Job Stories
    * New Stories
    * Show Stories
    * Top Stories

    Note: You will use **React Router** to navigate between the options.

  - 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 25 stories.
    * Style each story to look like a card using **Tailwind CSS** and **Shadcn UI**.
    * 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=pretty.

- – On a page, display the top 15 leaders. You can get this information from the following URL:
  https://news.ycombinator.com/leaders. **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 5 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=pretty.

Milestone Three - Due Wednesday, 18 November 2024 (Week 15) at 4.59 PM.
Code Defence - Thursday, 19 November 2024 (Week 15) at 10.00 AM.

- **Trivia Application (20%):**

  - – Create a new directory called **trivia-application** and create a new **Next.js** application.

  - – Can run in development without modification.

  - – Use **React Query** and **GraphQL** to fetch data from the **OpenTDB API**.

  - – Create a form with the following inputs:
    - ∗ Amount. This is the number of questions to retrieve. The default value is 10.
    - ∗ Category. This is a dropdown list. The options can be retrieved from the following endpoint:
      https://opentdb.com/api_category.php. The default option is **Any Category**.
    - ∗ Difficulty. This is a dropdown list with the following options:
      - · Any Difficulty. This is the default option.
      - · Easy.
      - · Medium
      - · Hard
    - ∗ Type. This is a dropdown list with the following options:
      - · Any Type. This is the default option.
      - · Multiple Choice
      - · True/False

  - – When the form is submitted, perform a request to the https://opentdb.com/api_config.php.

  - – Use the response to display the questions and answers in a quiz format.

  - – Create a form that allows the user to submit their answers. After the user submits their answers, display
    their score and the correct answers.

Code Quality and Best Practices - Learning Outcome 1 (45%)

- A **Node.js .gitignore** file is used.

- Environment variables' key is stored in the **.env.example** file.

- Appropriate naming of files, variables, functions and components.

- Idiomatic use of **TypeScript**, control flow, data structures and in-built functions.

- Efficient algorithmic approach.

- Sufficient modularity.

- Each file has a **JSDoc** header comment located at the top of the file.

- Code is formatted.

- No dead or unused code.

Documentation and Git Usage - Learning Outcomes 1 (5%)

- A **GitHub** project board or issues to help you organise and prioritise your development work. The course lecturer needs to see consistent use of the **GitHub** project board or issues for the duration of the assessment.

- Provide the following in your repository **README.md** file:

  - How do you setup the environments, i.e., after the repository is cloned?

  - How do you run your **frontend applications** locally?

  - How do you check your code?

  - How do you format your code?

- Use of **Markdown**, i.e., headings, bold text, code blocks, etc.

- Correct spelling and grammar.

- Your **Git commit messages** should:

  - Reflect the context of each functional requirement change.

  - Be formatted using an appropriate naming convention style.

Additional Information

- If you do not attend a milestone code defence, you will receive a mark of **zero** for that milestone.

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

- You need to show the course lecturer the initial **GitHub** project board or issues before you start your development work. Following this, you need to show the course lecturer your **GitHub** project board or issues at the end of each week.