**Course Title: Fundamentals in Javascript   
  
ASSESSMENT PLAN**

| **Assessment Tools** | **Descriptions** | **Total Weightage** | **Type** | **Assessment Schedule** |
| --- | --- | --- | --- | --- |
| Quiz or Assignment 1 | Home work for Day 1, covering:   * Getting started with HTML/CSS/JS * Conditional Statements and Loops * Function and Operators | **50%** | Individual | End of Lesson 1 |
| Quiz or Assignment 2 | Homework for Day 2, covering:   * Array and data manipulation * Closure * Callback function | Individual | End of Lesson 2 |
| Quiz or Assignment 3 | Take-home Project for Day 3, covering:   * OOP in JS * HTML element related DOM APIs | Individual | End of Lesson 3 |
| Quiz or Assignment 4 | Take-home Project for Day 4, covering:   * Asynchronous programming * HTTP related DOM APIs | Individual | End of Lesson 4 |
| End of Module Project | Students will deliver a frontend application in vanilla JS | **50%** | Group | End of Lesson 4 and 5 |
|  | **Total** | **100%** |  |  |

**END OF MODULE PROJECT (CAPSTONE PROJECT)**

| **Project Title / Description** | : | Implement a xkcd comic reader |
| --- | --- | --- |
| **Project Objective(s)** | : | xkcd.com is a webcomic site that provides api to retrieve its comics details, as documented at <https://xkcd.com/json.html>  However, due to CORS not being enabled on the official xkcd API, we created a proxy with CORS enabled for you under https://intro-to-js-playground.vercel.app/api/xkcd-comics/<comic -id> that would return similar data  You can get the specific comics using the endpoints like below   * <https://intro-to-js-playground.vercel.app/api/xkcd-comics/1> * <https://intro-to-js-playground.vercel.app/api/xkcd-comics/600> * <https://intro-to-js-playground.vercel.app/api/xkcd-comics/2475>   Alternatively, you can use <https://xkcd.vercel.app/>  Create a custom website to display 3 comics with comic title, comic image, and 3 buttons above it (prev, random, next) to control what comics to show.    Something like the image above, with the difference that you need to show the comic before it and after it. For example if the current image is image 200, then you should show image 199 on the left and image 201 on the right. When the current image is the last image (2475), then show image 2474 on the left and image 1 on the right.  When the user clicks next/prev, the comics should change to next/prev 3 comics. For instance, if the current comics are comics 1, 2, and 3, comics 4, 5, 6 should be displayed.  To provide a good user experience, show a loading indicator while you make the API calls.  A text input and a “Go” button next to it should be displayed to allow the user to go to a specific comic number. If a user enters invalid value, an error message should be shown to indicate that the number is invalid.  A select element should be displayed to allow users to select showing 1, 3, or 5 comics at a time.  You are not allowed to use any third-party script/library for this project. Only plain JavaScript and DOM API is allowed. |
| **Project Duration** | : | **121 hours** |
| **Mode of Delivery** | : | Group Consultation  +  Group Presentation |

**MARKING RUBRICS FOR THE** **END OF MODULE PROJECT (CAPSTONE PROJECT)**

|  | **Score** | | |
| --- | --- | --- | --- |
| **Category** | **0-1** | **2-3** | **4-5** |
| **Problem and Solution**  **(x2)** | * **0-1** - No to very minimal attempt made * **2-3** – The app is unusable. | * **2-4** Students implement only 1-2 of the requirements. * **5-6 -** Students implement half of the requirements. | * **7-8** Students implement 80% of the requirements. * **9-10** Students implement the complete set of requirements. |
| **Documentation** | * **0** - No code / attempt documentation written. * **1** –The code is undocumented. | * **2 -** There are some   comments explaining  the code, however, provides no extra context to the code readers.   * **3 -** There are some comments explaining   the code that the readers can’t understand without. | * **4 -** The code is self-documented and easy to understand **or** there’s external README file that’s really helpful * **5** – The code is written   well, easy to understand. The app is documented in a README that someone can get up and running with |
| **Presentation and**  **Communication** | * **0** - Student do not   present clearly on the  work and/or unable to  answer questions related to the work.   * **1** – Student could briefly   present the  work and/or unable to  answer questions related to the work. | * **2 -** Student present the   work and are able to  answer some  questions related to the work.   * **3 -** Student present the   work and are able to  answer most questions related to  the work. | * **4 -** Student   communicates well  during presentation  and able to answer most of the questions related to the work.   * **5** – Student   communicates well  during presentation  and able to answer all the questions related to the work. |