Interactive Development Diploma in Immersive Media, Diploma in Information Technology Year 1 (2020/21) Semester 1.2	Version 1 Semester **Oct 2020**
Assignment Project 2 (30%) **Front-End Development**	Week 8 - 12

OVERVIEW

You are to design and develop an interactive front-end web application. In this project, you'll build an interactive front-end site. The site should respond to the users' actions, allowing users to actively engage with data, alter the way the site displays the information to achieve their preferred goals.

REQUIREMENTS

- Design, develop and implement a dynamic front-end web application using HTML, CSS, JavaScript, jQuery
- Implement front-end interactivity, using core JavaScript, JavaScript libraries and/or Application Programming Interfaces (APIs)
- Utilize **appropriate APIs or external API** to interface with the application
- Design a front-end web application based on sound design principles, accessibility
- Use version control software to maintain, upload code
- Test and deploy an interactive front-end web application to Github
- Demonstrate and document the development process through version control (Github)
- This is an **INDIVIDUAL** project

AIM

In the process of developing the project, you should delve into questions such as:

- Does the website address the needs of the intended audience
- Is the purpose of the website evident to the audience
- Has it been appropriately tested for cross browser compatibility, platforms and devices
- Are the aesthetics appealing and attractive for the intended audience
- Building skills in web development and design
- Does the site use an external API?
- Build up your responsive web design skills using **CSS Frameworks (Bootstrap, etc)**

You are to inform your tutor on the progress, purpose of your site.

1. INTRODUCTION (CREATE A PRODUCT. YOUR OWN IDEA)

Here are some ideas to get you started. Think of yourself as a young upstart web developer who's trying to build a product. Utilizing APIs or various APIs, mashup your ideas and build a comprehensive application using APIs.

You are free to ideate on a theme, do discuss and clear your idea with your tutor.

Suggested Project Example Idea 1

Create a site that calls on a Maps API (Mapbox, Google Maps API) and combine with other APIs such as parking API

External user's goal:
Find the closest Point of Interest near me, or maybe find me a parking slot within a certain distance radius

Site owner's goal:
Get users to find a parking slot near their area

Potential features to include:

- Connect to an API that suits your needs,
- Display information about POIs or maybe useful information such as parking slots
- Allow users to search for areas or add additional parking location
- Provide results in a manner that is visually appealing and user friendly

Suggested Project Example Idea 2

Build a front-end website that allows customisable data

External user's goal:
Be able to store information and retrieve information

Site owner's goal:
Get users to store their daily expenses and track their spending

Potential features to include:

- Use localstorage to store spending information
- Using charts to display information
- Possible customisation in terms of UI such as dark mode
- Provide users with clear explanations on how to customise and with clear feedback on what changes are made
- Can further use Web Databases to store information online for longer periods

Other ideas such as Thematic Ideas (Open topics)

Theme: Smart City, Smart Campus

Create a project to help staff, or students or people cut down manual work. How can we better educate technology to the people?

Eg. Facial recognition attendance, efficient peer grading, student profiling, intelligent booths, dynamic crowd management, urban transportation solutions, sustainable logistics or businesses, voice recognition and speech interfaces, green city, student portfolio management

Theme: Addiction

With technology, an alarming number are getting digitally addicted. How can we educate on the better usage of technology?
Food is everywhere but obesity is on the rise. How can we curb this? Eg. Educational games, community games, casual games that enhances experiences

Theme: Health & Community

Encourage healthy living and effective health care services. Singapore has an ageing concern, smart-city innovations enable the ageing population to achieve higher levers of social cohesion and accessibility. As Singapore is moving towards a Smart Nation objective, how can people be technology-enabled. Bring people in the community or sharing common interest together.

Use gamification and game mechanics to encourage people working towards a common goal. Create a sense of kampong spirit for the future. It has been proven that habits and systems are better at long lasting changes than setting year resolution and relying on sheer willpower alone. Study the methodologies for producing winning habits, and build an app around its methods. How can we connect students in the school? Connect mums and kids in the neighbourhood? How can we bring people together to exercise?

Eg. Care system for the elderly, preventive health, tech education, inclusive society, fighting diabetes or healthcare issues, “soccer” mum club

2. GUIDING FACTORS

The full design is implemented providing a good solution to the users' demands and expectations.

Thinking points

- Is the web site sound and fits the audience intent?
- Have you tested across browsers?
- Is the website developed as close to what was proposed in the wireframe and documentation?
- Is the navigation structure sound?
- Is an appropriate API chosen?

Real world application

- Clearly understandable site-specific content is used rather than Lorem Ipsum placeholder text
- All links to external pages open in a separate tab when clicked
- The final application is aligned to the user stories presented at the start of the project
- Testing procedures are comprehensive, with a good level of coverage, and have clearly been followed. All noticeable errors have been corrected or documented.
- How does creating the application help in terms of usage. What benefits does it bring about?
- Are the APIs easy to use?

Version control systems are used effectively:

- all code is managed in git with well-described commit messages
- there is a separate, well-defined commit for each individual feature/fit
- there are no very large commits which make it harder to understand the development process and could lead the assessor to suspect plagiarism

The full application development process is documented:

- the purpose of the application is clearly described in the **README**
- the project's documentation describes the design work undertaken for this project and the reasoning behind it
- wireframes, mockups, diagrams, etc., created as part of the design process are included in the project

SUGGESTED API LIST *(You may use others. You are NOT limited to this list of APIs. Find ones with good documentation. Having an API listed here doesn't mean it is easy to implement.* ***Do check properly and test first before actually going in-depth into an API selection*** *)*

https://any-api.com
https://dev.twitter.com/
http://www.dex.sg/collections/
https://developers.data.gov.sg
https://www.nea.gov.sg/api/
https://github.com/public-apis/public-apis
https://rapidapi.com/collection/list-of-free-apis
https://rapidapi.com/
https://apilist.fun/
https://public-apis.xyz/
https://www.mapbox.com/
https://www.mas.gov.sg/development/fintech/financial-industry-api-register
https://docs.onemap.sg/#onemap-rest-apis

https://github.com/TonnyL/Awesome_APIs
https://dev.to/camerenisonfire/10-intriguing-public-rest-apis-for-your-next-project-2gbd
https://public-apis.io/
http://www.omdbapi.com/
https://the-one-api.dev/
https://www.reddit.com/dev/api/
https://spoonacular.com/food-api

3. SUBMISSION CRITERIA

All files are to be submitted to **Network drive & deployed to Github**

Note: Always backup your files using other mediums (eg. Google Drive, Dropbox, External HDD, ictspace server) throughout the assignment.

No.	Deliverable	Naming Convention	File Format
1.	Wireframe	ID_StudentID_StudentName_Assg2_wireframe	.xd
2.	Website *Includes item no.3*	ID_StudentID_StudentName_Assg2_website	All pages to have .html extension homepage to named as **index.html** Ensure all files are in appropriate folders
3.	Documentation. README.md	**README.md**	This file should be placed together with your website source codes
4.	Demo Recording	ID_StudentID_StudentName_Assg2_pitch.mov or .mp4 ID_StudentID_StudentName_Assg2_pitchdeck.pdf (if needed)	An application pitch using a movie clip with narration demonstrating the website and design process. You should demo at least the site. You may use slides to further explain your application **.mp4/.mov** .pdf (pdf of the slides used)

Include all original artwork when possible. (.psd, .ai, .docx, etc.)
*Note: Files should be properly named and structured. Failure to do so will result **in heavy penalization** of marks. Marks are given for proper file organization*

DELIVERABLE DETAILS

1. Wireframe

- Set of wireframes of any fidelity with appropriately named artboards
- Wireframes are to include both desktop and mobile views

2. Website (Read Guiding Factors and Rubrics Guidelines)

- Fully functional responsive website that is user-friendly
- All pages are to be have .html extension
- Organise HTML and CSS into well defined and commented sections
- Ensure that there are no broken links
- Is the site validated?
- Have the APIs been validated

It is no longer about the number of pages but how wholesome the entire application is. Utilize AJAX techniques to update your site asynchronously and APIs to make your application more meaningful.

3. Readme.md
The readme is to contain all documentation and research done.
It should also detail down:

- the user audience intent and purpose.
- Who the website is catering for? Value that it is providing to users
- What is the website catering for?
- Attribute all external source code used

See example **README.md**
https://guides.github.com/features/mastering-markdown/

There is a clear seperation between code written by you and code from external sources (e.g libraries or tutorials). **All code from external sources is attributed to its source via comments above the code AND in the readme.** The readme is to contain all documentation and research done.

- The purpose of the application is clearly describe in the README
- Readme is well-structed and easy to follow
- Readme is written in markdown and uses markdown formatting consistently and effectively

4. Pitch Recording
Recording of your application pitch. Assume you are presenting to your audience and deliver your best pitch on the application usage, design and technical implementation.

IMPORTANT TASKS & DELIVERABLES

- Read the requirements for the application carefully. Some requirements are **necessary** for a Pass grade.
- For submission, you should
    - **For Network Drive submission:** Create a folder ID_studentid_studentname_Assg02 and include all deliverables in that folder.
    - **For Github submission:** Ensure all deliverables are named appropriately. Your

-
-
- should start with an index.html and contain a README.md
- **Cross check** all your files are working and can be opened

- All documents submitted must be in **their intended formats**

DUE DATE

Assignment is due on **Week 12 Sunday 2359h 10th Jan 2021**

4. ASSIGNMENT WEIGHTAGE

ITEMS	WEIGHTAGE
Technical *HTML & CSS Implementation? Proper version control? Website is coded in a responsive format? Follow guiding factors and rubrics guidelines? Code is well validated?* Apply **Object Oriented Programming** (OOP) to your data *API Usage? Deployed to Github? Appropriate Git usage* *Does the code pass through a linter (e.g JSHint) with no major issue* *Code organisation such as JS code is linked at the bottom of the body and handled properly before the HTML loads* *Easy to read code? Semantic markup?Code handles internal errors.*	**50%**
Wireframe *Wireframe explains the design and caters to responsive design? Layered files with proper naming conventions?* *Number of wireframes should match the number of webpages and include both desktop and mobile iterations accordingly* *Insert screenshots of the finished projects to the README. Appropriately documented design process?*	**15%**
Visual Appeal & Consistency Properly used typography? Attractive graphics? Properly selected images and content? Easy to read? Aesthetic? User is kept informed of the various parts of the application through progress or feedback actions Consistent design is applied across pages and interactions	**15%**
Checkpoint & Demo & Real World Application Practised? Clearly explained and demo shown?Demo video clearly explains the application? Clear README?	**20%**
GRAND TOTAL	**100%**

5. ASSIGNMENT RUBRICS GUIDEBOOK

Development & Implementation

Code demonstrates characteristics of 'clean code':
Consistent and appropriate naming conventions within code and in file naming, e.g.

- file names and class names, are descriptive and consistent
- for cross-platform compatibility, file and directory names will not have spaces in them and will be lower-case only
- all HTML attributes and CSS rules, are consistent in format, follow standards for the language and are appropriate and meaningful

File structure

- whenever relevant, files are grouped in directories by file type (e.g. an assets directory will contain all static files and code may be organised into sub-directories such as css, etc)
- there is a clear separation between custom code and any external files (for example, library files are all inside a directory named 'libraries')
- files are named consistently and descriptively, without spaces or capitalisation to allow for cross-platform compatibility.

Readability

- id/class/attribute names clearly indicate their purpose
- code is indented in a consistent manner to ease readability and there are no unnecessary repeated blank lines (and never more than 2)
- CSS code is split into well-defined and commented sections
- Semantic markup is used to structure HTML code
- HTML and CSS are kept in separate, linked files
- CSS files are linked to in the HTML file's head element
- errors are handled gracefully and users are notified of the problem where appropriate.
- non-travial **JavaScript code files are linked to the bottom of the body element**

Comments

- all custom code files include clear and relevant comments explaining the purpose of code segments

Compliant code

- HTML code passes through the official W3C validator with no issues
- CSS code passes through the official (Jigsaw) validator with no issues
- JavaScript code passes through a linter (e.g. jshint.com) with no major issues

Robust code

- no logic errors are found when running code
- errors caused by user actions are handled
- inputs are validated when necessary.
- where used, API calls that fail to execute or return data will be handled gracefully, with the site users notified in an obvious way
- navigating between pages via the back/forward buttons can never break the site, there are no broken links

- user actions do not cause internal errors on the page or in the console

6. SUBMISSION DEADLINE

Please note that there is a mid-project milestone submission on **Week 10.** Making it a **total of 2 significant deadlines** for this assignment.

INTERIM SUBMISSION

Deadline: Sun 27th Dec 2020 2359 [WEEK 10]

Marks are awarded based on progress at current checkpoint

All deliverables to be properly named and submitted on **Network Drive and GitHub** as instructed by your lecturer.

FINAL SUBMISSION

Deadline: Sun 10th Jan 2021 2359 [WEEK 12]

All deliverables to be properly named and submitted on **Network Drive and GitHub** as instructed by your lecturer.

Your website must be fully functional on your Github Pages account. Your source code submissions on Network Drive will be checked to ensure that your submission is timely and valid.

WORK REVIEW

As part of your assessment, you may be required to give a brief explanation of selected pieces of content, writeup done by you or your team.

PLAGIARISM AND COPYRIGHT

Plagiarism means, "*copying any part of a source, and then submitting it, claiming that it is your own work*."

Please ensure that all the works submitted by you are not copied from other sources. Any attempt to plagiarize will be dealt with severely, and it may result in your failing the module.

- This is an **INDIVIDUAL** assignment. All work must be attributed and credited
- If you are found committing plagiarism, you will score 0 for your assignment

LATE SUBMISSION

Late submission will be **penalised** (10% of the marks for each day late after 12 noon). Submission will **not be accepted after 5 days** (including weekends and public holidays) from the date of submission.

DUE DATE

Assignment is due on **Week 12 10th Jan 2021 Sunday 2359h**

Assignment is worth 30% of your total module grade.

End of Project Brief (Total: 9 pages)