TOMORROW'S WEB

ASSESSMENT 2: Proof of Concept

Tutor/Marker: Abdul Basit

Contribution towards overall module mark	70%
Date set	3 March 2021
Marked work returned by	12 June 2021
DEADLINE DATE	25th May 2021 - 11.59am

Assessment 2: Proof of Concept

The Brief

You are tasked with building a proof-of-concept web application using one or more of the available web browser APIs listed here:

https://developer.mozilla.org/en-US/docs/Web/API

Because the focus of the application will be using these APIs, it is expected that the majority of the code will be running in the user's web browser, i.e. using HTML, Javascript and CSS.

There may or may not be some server side functionality required, depending on your choice of application (for example, you may need a server to save user details or log in). You may use any language or framework you choose for the server-side component (if any).

The specifics of the application functionality are left for you to decide, however, there should be a fully functional UI present to interact with the APIs.

Please ensure that you discuss your idea with your module tutor to ensure that your approach meets the requirements of the brief. For more abstract/flexible briefs such as 'Proof of Concept' regular interaction to help steer your project is essential.

Deliverables

- A project folder / zip archive that contains all required code files and assets. This should be made available on Google Drive.

- A GitHub URL to the project source code. The code in the GitHub repository should match the code in the project folder / zip archive delivery.
- A publicly accessible URL to the deployed application (if possible, but preferred).
- A 1000-word development document, which should explain your research, application, as well as detail which APIs were used and how.

The Development Document

Your application must be accompanied by a development document of a minimum of 1000 words (there is no maximum word count).

Please include the following elements (note the suggested word counts):

- *Concept:* A short description of your core idea (~50 words).
- *Specification:* A clear outline of the specification of your application (~100 words).
- *Research, Development and Technical Description:* Details your research into creating the application, as well as details on how the application was built, and how it functions. For example, What APIs were used? Why were they chosen? Were any alternatives tried? How were they utilised? (~550 words).
- *Critical Reflection:* A critical evaluation of the successes and limitations of your proof of concept. This should also describe what personal learning you need to pursue in future to refine your skills (~300 words).

Submission

Please follow the submission instructions below. Work that is submitted incorrectly may not be accepted or could incur a points penalty.

Before submitting, have you…

- Checked that any digital work is functioning as expected?
- Spell-checked and grammar-checked any written work that accompanies your digital work? Please make an appointment with the Writing and Learning Centre or speak to your tutor if you are experiencing challenges in this area.
- Formatted your written work to the specification below?
- Referenced all sources of information accurately? Please refer to www.citethemrightonline.com (Harvard) for guidance.

Your work must be submitted via Turnitin. Please adhere to the following method:

- Log into your university Google Drive account.
- Create a folder for your project. Call it something meaningful.
- Upload your work to your new folder.
- Right click your folder, select ‘Get Shareable Link’, turn on ‘Link Sharing’ and then copy the URL provided.
- Paste your Google Drive URL and your GitHub repository URL into your development document.
- Log into Minerva, go to the Assessment tab and submit your Word document via the appropriate Turnitin link.

Marking Criteria

Assessment 2: Proof of Concept will be marked against the following criteria:

1. *Concept:* Selection of appropriate APIs and web technologies, originality of core idea.
2. *User Experience:* Quality of user interface design and visual design, intuitiveness of application, clarity of instructions/user prompts.

3. *Implementation:* Technical implementation, selection and application of techniques, and code readability.
4. *Documentation*. How well the development document communicates concept, design implementation. Includes quality of critical reflection.

Criteria	Weighting	Description	Marks
Concept Selection of appropriate APIs and web technologies, originality of core idea	15%	A highly limited concept that may not require web APIs. Technologies used are likely long established and difficult to argue as emergent. Idea overall is highly unoriginal and/or confused.	0 - 19 (Low Fail)
		A poor concept that is largely unoriginal or limited in scope. APIs or other web tools scoped are likely well established and arguably not emergent.	20 - 39 (Fail)
		A basic concept that does deploy a web technology or API that can be considered emergent. The idea built around such technologies however likely does not engage them beyond a simple demonstration of core functionality.	40 - 49 (Third)
		A fair concept that demonstrates some original thinking, yet is likely underdeveloped in terms of creative exploration. APIs and web technologies selected are arguably emergent.	50 - 59 (2:2)

		A good concept that targets APIs and web technologies that can be considered emergent. Core idea is largely original and reliant on an appropriate level of creative exploration.	60 - 69 (2:1)
		A very good concept that is original and demonstrates an ability to explore several appropriate emergent APIs and web technologies in a creative context.	70 - 79 (First)
		An excellent concept that skillfully consolidates several emergent web technologies and APIs to advance a highly original and compelling creative idea.	80 - 89 (High First)
		Beyond expectations for this level of study.	90 - 100 (Outstanding)
User Experience Quality of user interface design and visual design, intuitiveness of application, clarity of instructions/user prompts	15%	A highly limited design that pays little to no attention to the quality user experience. Visual design is likely confusing, and the application overall is difficult to operate.	0 - 19 (Low Fail)
		User experience is considered but with major limitations. Visual design is overall poor, and instructions may be missing or difficult to understand.	20 - 39 (Fail)
		A basic attempt at establishing a coherent user experience. Interface design allows for relatively successful navigation, yet overall the application may be unintuitive.	40 - 49 (Third)

		Visual design is acceptable yet would likely benefit from significant revision.	
		A fair design that demonstrates a sound approach to user experience. Application is navigable with limited user prompts, yet there are likely some features that are difficult to locate or use. Visual design is sound but may be limited in sophistication.	50 - 59 (2:2)
		A good attempt to establish a coherent user experience. Interface design is clear, and in-built user prompts may be established to aid understanding of more complex features. Visual design is well considered for a proof of concept.	60 - 69 (2:1)
		Very good attention to user experience. The application is easy to use on the first attempt, and appropriate instructions/user prompts are likely provided to aid comprehension. Visual design is balanced and clean, and likely approaching the standards of a prototype.	70 - 79 (First)
		Excellent attention to detail across all aspects of user experience. Interface design is intuitive, and significant effort has been put into ensuring that the user is guided through the features of the application. Visual design is likely very well considered and	80 - 89 (High First)

		approaching the requirements of a deployable product.	
		Beyond expectations for this level of study.	90 - 100 (Outstanding)
Implementation Technical implementation, selection and application of techniques, and code readability.	50%	A very limited proof of concept that does not meet the requirements of the brief. The final result will likely be non-functional in several critical areas. Code readability may require significant overhaul with code commenting poor or missing.	0 - 19 (Low Fail)
		A poor implementation with major errors that may render the final result non-functional. Development is likely substandard, with the supplied codebase underdeveloped and inefficient. Code readability may be poor with commenting failing to adequately annotate key techniques deployed.	20 - 39 (Fail)
		A basic implementation that demonstrates limited knowledge of essential techniques for API deployment. There will likely be modest to occasionally major errors in development techniques leading to a piece that is only moderately functional. Code readability may not be satisfactory, and code commenting is likely only adequate. The application is likely not to be deployed to the web.	40 - 49 (Third)

		A fair implementation that demonstrates a modest understanding of API deployment. Implementation likely has several minor errors, yet these do not overly affect functionality. A range of appropriate web development techniques are demonstrated, yet there is likely room for significant optimisation in several areas. Code readability and commenting is sound but may require additional attention. The application may not be deployed to the web.	50 - 59 (2:2)
		A functional proof of concept that may deploy several APIs and web technologies to meet the specifications of the app. Implementation will likely benefit from further optimisation, and there may be minor errors related to code readability and commenting. The application is likely deployed to the web.	60 - 69 (2:1)
		A very good implementation that demonstrates an ability to successfully consolidate a range of APIs and other web technologies. The proof of concept is likely entirely functional and efficient with only minor errors in optimisation. Code is well commented and easy to interpret. The application is highly likely to be deployed to the web.	70 - 79 (First)

		An excellent proof of concept that demonstrates an high level of ability to combine several APIs and web technologies. Development goes beyond the requirements of functionality, and more towards shaping a compelling user experience. There are likely no obvious flaws in the application, and supporting codebase is optimised and commented with precision. The application is highly likely to be deployed to the web.	80 - 89 (High First)
		Beyond expectations for this level of study.	90 - 100 (Outstanding)
Documentation How well the development document communicates concept, design implementation. Includes quality of critical reflection.	20%	Very limited documentation that demonstrates little or no understanding of the design and build process. Critical reflection is missing or inadequate. Structure is unacceptable.	0 - 19 (Low Fail)
		A poor attempt that does not meet the requirements of the development document. May be missing key sections or badly structured. Inadequate technical description and/or critical reflection.	20 - 39 (Fail)
		Basic documentation that offers only minimal information about the design direction and technical implementation. Correct use of technical terminology is lacking and	40 - 49 (Third)

		critical reflection is limited in depth. Structure is acceptable.	
		A fair attempt that provides only key information about the design and technical implementation. May be limited in the clarity of the technical description and depth of the critical reflection. Document structure is acceptable.	50 - 59 (2:2)
		A good to very good development document that provides a clear overview of the design direction and a sound description of the technical implementation. Critical reflection is adequate, but may be limited in depth. Document structure has only minor issues.	60 - 69 (2:1)
		Very good overview of the project that provides detailed description of the concept design and technical implementation. Structure is without error. Critical reflection is well considered.	70 - 79 (First)
		An excellent overview that provides a high level of insight into the design and technical implementation. Reflection is highly critical and detailed. Structure is without error.	80 - 89 (High First)
		Beyond expectations for this level of study.	90 - 100 (Outstanding)

Intended Learning Outcomes (ILOs)

ILO	Assessed
Critical review of existing deployments of emerging web technologies and ideation on new commercial and creative applications.	✓
The conception and implementation of proof of concept applications that exploit emerging web technologies.	✓
Synthesis of API and framework documentation to include their key features into web development work.	✓
Precise communication of the context, opportunities and limitations of emerging web technologies to a specialist audience.	

Mark penalties may be applied to late submissions without prior approval of an extension. Please ensure that you prepare and submit your work in good time to allow for any issues that may arise.