Rubric

Final JS App: Attention to Detail **(20%)**

No Marks		Half Marks		Full Marks	
The submission to Gradescope **does not include** a project folder containing JavaScript file(s), **missing** both the static data files and cached real-time data.	**0.00 %**	The submission to Gradescope **includes** a project folder containing JavaScript file(s), but is **missing** either the static data files or cached real-time data.	**2.50 %**	The submission to Gradescope **includes** a project folder containing JavaScript file(s), the static data files and cached real-time data.	**5.00 %**
The project folder **does not feature** an appropriate file/folder naming/structure, and the starting file is **difficult** to locate.	**0.00 %**	The project folder features a **mostly appropriate** file/folder naming/structure.	**2.50 %**	The project folder features an **appropriate** file/folder naming/structure, with a starting file that is **easy** to locate.	**5.00 %**
The app **does not run** via Node.js in the terminal, requiring **significant** modifications to get it running.	**0.00 %**	The app **does not run** via Node.js in the terminal, but only requires **minor** modifications to get it running.	**2.50 %**	The app **runs** via Node.js in the terminal, without any issues.	**5.00 %**
The user-facing portion of the app demonstrates **little to none** of the expected functionality.	**0.00 %**	The user-facing portion of the app demonstrates **most** of the expected functionality.	**2.50 %**	The user-facing portion of the app demonstrates **all** of the expected functionality.	**5.00 %**

Rubric

Final JS App: Functional Programming **(30%)**

No Marks		Half Marks		Full Marks	
Does not demonstrate an understanding of functional programming techniques, relying **too heavily** on imperative logic over declarative logic.	**0.00 %**	Demonstrates **some** understanding of functional programming techniques, with **occasional** reliance on imperative over declarative logic.	**3.75 %**	Demonstrates a **strong** understanding of functional programming techniques, **balancing** the use of declarative & imperative logic where appropriate.	**7.50 %**
Does not demonstrate an understanding of declaring, defining & transforming immutable data, relying **too heavily** on redefining values.	**0.00 %**	Demonstrates **some** understanding of declaring, defining & transforming immutable data, with **occasional** reliance on redefining values.	**3.75 %**	Demonstrates a **strong** understanding of declaring, defining & transforming immutable data.	**7.50 %**
Does not demonstrate an understanding of recursive techniques, relying **too heavily** on iterative logic over recursive logic.	**0.00 %**	Demonstrates **some** understanding of recursive techniques, with **occasional** reliance on iterative over recursive logic.	**3.75 %**	Demonstrates a **strong** understanding of recursive techniques, **balancing** the use of recursive & iterative logic where appropriate.	**7.50 %**
Does not demonstrate an understanding of modularity through nested functions, where variable scoping is **too broad**.	**0.00 %**	Demonstrates **some** understanding of modularity through nested functions, but variable scoping could be **narrowed**.	**3.75 %**	Demonstrates a **strong** understanding of modularity through nested functions, including **appropriate** variable scoping.	**7.50 %**

Rubric

Final JS App: Asynchronous Programming & Integrations **(25%)**

No Marks		Half Marks		Full Marks	
Does not use Node.js modules to support prompting, Fetch API, CSV parsing, and file system access.	**0.00 %**	**Uses** Node.js modules to support prompting, Fetch API, CSV parsing, and file system access, but **could be better** integrated into the code.	**4.16̄ %**	**Integrates** Node.js modules, or demonstrates suitable **bespoke** alternatives, to support prompting, Fetch API, CSV parsing, and file system access.	**8.33̄ %**
Demonstrates **little to no** use of asynchronous programming techniques, or **does not** use Promises and appropriate syntax (e.g. async/await) to action them.	**0.00 %**	Demonstrates an **occasional** use of asynchronous programming techniques, **using** Promises, and appropriate syntax (e.g. async/await) to action them.	**4.16̄ %**	Demonstrates a **holistic** approach to asynchronous programming, **using** Promises, and appropriate syntax (e.g. async/await) to action them.	**8.33̄ %**
Does not integrate live data derived from the web API proxy (using the Fetch API) and local data via CSV files.	**0.00 %**	**Integrates** live data derived from the web API proxy (using the Fetch API) and local data via CSV files, but it could be **more selectively** filtered.	**4.16̄ %**	**Integrates** live data derived from the web API proxy (using the Fetch API) and local data via CSV files, with **selective** filtering applied.	**8.33̄ %**

Rubric

Final JS App: Code Style **(25%)**

No Marks		Half Marks		Full Marks	
Demonstrates **little to no** consideration of an easy to understand program structure, with variables & functions **not following** a consistent naming scheme.	**0.00 %**	Demonstrates **some** consideration of an easy to understand program structure, with variables & functions **following** a consistent naming scheme.	**3.12̄ %**	Demonstrates **consideration** of an easy to understand program structure, with variables & functions **following** a consistent naming scheme.	**6.25 %**
Demonstrates **little to no** consideration of code neatness (e.g. tabbing, whitespace) or consistency (similar to common style guides).	**0.00 %**	Demonstrates **some** consideration of code neatness (e.g. tabbing, whitespace) and consistency (similar to common style guides).	**3.12̄ %**	Demonstrates **consideration** of code neatness (e.g. tabbing, whitespace) and consistency (similar to common style guides).	**6.25 %**
Demonstrates **little to no** understanding of the code through comments, with comments above function declarations **excluding** @param and @returns.	**0.00 %**	Demonstrates **some** understanding of the code through comments, with comments above function declarations **including** @param and @returns.	**3.12̄ %**	Demonstrates **understanding** of the code through comments, with comments above function declarations **including** @param and @returns.	**6.25 %**
Demonstrates **little to no** use of shorthand syntax (e.g. arrow function expressions) where appropriate.	**0.00 %**	Demonstrates **some** use of shorthand syntax (e.g. arrow function expressions) where appropriate.	**3.12̄ %**	Demonstrates **regular** use of shorthand syntax (e.g. arrow function expressions) where appropriate.	**6.25 %**