# List of basic conventions for writing code

## Naming Conventions:

* Use descriptive and meaningful names that accurately convey the purpose of the identifier.
* Use camelCase for variable names.
* Use camelCase for method parameters.
* Use PascalCase for class names, types and components.

## Declaration convention:

* All Imports are at the beginning of the component or file
* All Exports are in the bottom of the document or in the line where a function is declared.
* Typescript files export functions and Svelte components export arrow functions(lambda).

## Statement style:

* "1TBS" (One True Brace Style) or more generally, the "K&R" placing the opening brace at the end of the controlling statement.
* Indent code inside braces to improve readability.
* Use 1 tab for each indentation level.
* Tab size equal to 4 spaces.
* Curly braces should be at the same level.

## Comments:

* Comments should be in the same level as the code.
* Use comments to explain non-obvious code, especially for complex algorithms or business logic.

## Whitespace:

* Use whitespace to improve readability by adding spaces around operators and between keywords and identifiers.
* Do not use excessive whitespace.
* A line of empty space should be left between each method.

## Error Handling:

* Use try-catch blocks for handling exceptions, and handle exceptions at an appropriate level of abstraction.
* Use standard response object for retrieval massage for success or error.

## String Formatting:

* Use string interpolation {`${}`} or normal String “” for string formatting.

# Requirements Check List

## 1. Data- Declaration Errors

- Have all variables been explicitly declared? – Yes

- Where a variable is initialized in a declarative statement, is it properly initialized? – Yes

## 2. Computation Errors

- Are there any mixed-mode computations? – No

- Are there any computations using variables having different datatype? – No

- Is it possible for the divisor in a division operation to be zero? – No

## 3. Comparison Errors

- Are there any comparisons between variables having different

datatypes, such as comparing a character string to an address,

date, or number? – No

- Are the comparison operators correct? – Yes

## 4. Fetch requests

- Have all fetch request surrounded by trycath? – Yes

- Have all fetch request return status code with the generated response? – Yes

- Are there any mismatch between route endpoints? –No

- Are Fetch error conditions handled correctly? – Yes

- Are there spelling or grammatical errors in any text that is

fetched and displayed on the front-end? - No