



Hands UP

Coding Standards

Updated: 19/08/2025

Contents

1	Coding Standards	2
1.1	Formatting	2
1.2	Naming Conventions	2
1.3	Braces and Parentheses	2
1.4	Comments	2
1.5	Error Handling	2
1.6	Repository File Structure	3

1 Coding Standards

1.1 Formatting

- No specific formatting conventions are enforced.

1.2 Naming Conventions

- **File names:** `camelCase`
- **Directory names:** `snake_case`
- **Functions and variables:** `camelCase` and `snake_case`
- **Constant variables:** `ALL_CAPS`

1.3 Braces and Parentheses

- The first brace of a function is placed at the end of the function signature, and the closing brace is on a new, separate line.
- Example:

Listing 1: Example of Brace and Parentheses convention

```
1 signIn() {  
2     // logic  
3 }
```

1.4 Comments

- Comments are only added to code lines that may be ambiguous.

1.5 Error Handling

- Implement proper error handling with logs that detail what went wrong.
- All exceptions must be caught and handled appropriately.
- Server errors must be descriptive and use correct HTTP status codes.

1.6 Repository File Structure

The repository should adhere to the following file structure:

```
Hands-Up
|-- ai_model
|   |-- jz_model
|   |-- llm
|   |-- models
|   |-- pre_processing
|   |-- processed_data
|   |-- words
|
|-- backend
|   |-- api
|   |   |-- controllers
|   |   |-- models
|   |   |-- routes
|   |   |-- utils
|   |
|   |-- database
|       |-- sql
|
|-- cypress
|   |-- fixtures
|   |-- integration
|   |-- screenshots
|   |-- support
|
|-- diagrams
|
|-- frontend
|   |-- build
|   |-- public
|   |-- src
|       |-- components
|       |-- context
|       |-- hooks
|       |-- images
|       |-- media
|       |-- pages
|       |-- utils
|       |-- videos
```

```
__ tests
  __ coverage
  __ mocks
  __ unit
    __ api
    |   __ auth
    |   __ learn
    |   __ user
    |
    __ ui
```