Coding Conventions and Guidelines

Naming Convention

1. General:

- Avoid using names that are too general or too wordy. For example:
  □ Bad Practice: *data_structure*, *my_list*, *info_map*
  Good Practice: user_profile, event_list

2. Packages & Modules:

- Use lowercase with underscores. For example- user_routes, admin_views, event_controller.

3. Classes:

- Use PascalCase. For example- UserModel, EventForm, DashboardView
- Class Attributes:
  Models – m_prefix. For example- m_user_id, m_email
  Forms- f_prefix. Form example- f_password, f_name

4. Variables:

- Use Snake_case for all variables. For example- event_name, user_email
- Use is_prefix for Boolean. For example- is_attending, is_admin
- Use plural for collections. For example- users, teams etc.
- Use n_prefix for Count. For example- n_events, n_teams

5. Functions & Methods:

- Use lowercase with snake_case. For example- create_event(), get_user()
- Use prefix with _ for non-public. For example- _validate_input()
- First argument in class methods should be self.

6. Constants

- Use UPPERCASE with underscores for constants. For example- MAX-LOGIN-ATTEMPTS, DEFAULT_ROLE.
-

Code Layout

1. Indentation:

- Use 4 spaces per indentation level.
- No tabs, only spaces.

2. Line Length:

- Max: 79 characters per line.
- Break long lines after operators when needed

3. Blank Lines:

- Two blank lines between class/functions.
- One blank line inside class methods.
- Use blank line inside class methods.

4. Whitespace Rules:

- Spaces inside () or []
- Spaces before commas, semicolons, colons
- Uneven spaces around =, +, etc