1. **Naming Conventions:**

**Page Names:**

1. Use descriptive **names** for pages, reflecting their purpose or functionality. Use camelCase or underscores for multi-word page names (e.g., myPage or my\_page).
2. **Page Title** Use descriptive names for pages, reflecting their purpose or functionality. Remember your page Title show in Browser Tab.
3. You also put Title as your page Type as Like: Form for Customer Entry.
4. You can use **Page Group** for Identifying Easily you pages. It’s better for Authorization.
5. You can reserved Page ID as like 1-10 for Transaction and 10-20 setup etc.

**Item Names:**

1. Use meaningful **names** for items, if your Item name is related to DB object then describing is useful for assign.
2. Make sure which type field allowed for this field (text, date, radio, check, number etc.)
3. Use Text Case for input value (Lower or Upper)
4. Make best use of Item properties.
5. Prefix items with a three-letter abbreviation indicating the item type (e.g., txtName for a Text Field).
6. If you’re develop this for various kind of screen as pc, mobile, tab then need items choose template to be responsive.

**Region Names:**

1. Use clear and descriptive names for regions.
2. Avoid using default names like "Region 1"; instead, use names like "Employee Information."
3. Every properties has a comment section you can use this for identifying use of this objects.

**Dynamic Actions:**

1. Use meaningful **names** for dynamic Actions, it’s organized as sequentially. Like: DA for show hide mobile Number.
2. **Action Name also** Try to be specific.

**Process Name:**

Use server process name with valid and unique name as like: “Process for Customer Request Approval”

**Validation Names:**

1. When Use validation in page level with a descriptive name as Like P1\_CUSTOMER\_ID is not null.
2. Don’t repeat any validation which are similar to others.

**Buttons Name:**

Use descriptive and valid name for every button that’s represent to which are related to your process action. As Like: BTN\_CUSTOMER\_SUBMIT.

**2. Formatting and Indentation:**

**Consistent Indentation:**

1. Any PL-SQL or SQL Block follow a consistent and readable indentation style for every use. (e.g., 2 or 4 spaces).
2. Indent nested components for better readability.
3. In queue, list, report etc use proper indention for readability.
4. Use Responsive report for small screen.

**Whitespace:**

Use whitespace effectively to enhance code readability. Separate logical sections of code with empty lines for better visual separation.

**3. CSS Styles:**

**Consistent Naming:**

1. Common CSS can be store as a Shared component static filed or in server.
2. Use CSS variables for color and Class which are Oracle apex default.
3. Avoid inline styles; instead, use external or page-level CSS.
4. Avoid using CSS in Sql Query.

**Responsive Design:**

Ensure that your UI is responsive and works well on different devices and screen sizes.

Leverage APEX's responsive features and grid layouts.

**4. JavaScript and Dynamic Actions:**

i ) **Modular JavaScript:**

Organize JavaScript code into functions and modules for reusability.

Use proper commenting to explain complex logic.

**ii.) Use Dynamic Actions Wisely:**

Avoid cluttering your pages with too many dynamic actions.

Consolidate actions where possible for better maintainability.

**5. Page Design:**

**Logical Grouping:**

1. Group related items, regions, and buttons logically on your pages.
2. Use tabs and navigation menus to organize content effectively.
3. Add Responsive page items template if this Application use web and Mobile also.

**Consistent Themes:**

1. Stick to a consistent theme for your application for a professional look.
2. Customize the theme if necessary to match your organization's branding.

**6. Comments and Documentation:**

**Inline Comments:**

1. Add inline comments to explain complex logic or provide context.
2. Document the purpose of dynamic actions and processes.

**Page-level Documentation:**

1. Provide high-level documentation for each page, explaining its purpose and functionality.

**7. Security:**

**Input Validation:**

1. Implement proper input validation to prevent security vulnerabilities.
2. Use bind variables in SQL queries to prevent SQL injection.

**8. Version Control:**

**Use Version Control:**

1. Use version control systems like Git to track changes to your APEX application.