**SAP UI5 best Practices**

* Follow MVC(Model-View-Controller) architecture.
* Use responsive design techniques to ensure your application works well on different devices and screen sizes.
* Never Develop Application by hard coding the description texts. Apply i18n wherever applicable.
* Create keys for crud operations.
* Minimize the use of global variables and functions to reduce the risk of naming conflicts.
* Use camel case for functions and variables.
* Use base controller for bigger applications.
* Never make OData calls in loops.
* Minimize the number of individual HTTP requests by using batch requests when fetching data from the server.
* Provide meaningful error messages for users and log detailed error information for debugging.
* Implement lazy loading and virtualization for large lists or tables to reduce the number of rendered elements in the DOM.
* Use browser developer tools to monitor network requests, JavaScript execution, and memory usage while testing your application.
* Use Promises and async/await for asynchronous operations to prevent blocking the UI thread.
* Ensure that you are using the latest version of the SAPUI5 library to take advantage of performance improvements and bug fixes.
* Create reusable UI components (e.g., custom controls, fragments) to maintain consistency and reduce redundancy.
* Implement server-side pagination and filtering to request only the data needed for the currently visible items. This reduces the amount of data transferred between the server and the client.
* Use OData v2 for it improves features and better querying capabilities compared to v1.
* Remove all the commented codes – Otherwise, all the time it needs to load them from server to client and it is a waste of bandwidth usage.
* Enable Asynchronous loading in the bootstrap.
* Destroy complex controls after the usage.
* Add a semicolon (;) after each statement, even if it’s optional.
* No spaces before and after round parentheses (function call, function parameters).
* Ensure that opening curly brace (function, for, if-else, switch) appear on the same line as the conditional statement they belong to. For example, if (a === true) {.
* Use “===” and “!==” instead of “==” and “!=”;
* Don’t define properties that are set to their default values.
* Do not use console.log ().
* Use XML view for the view code.
* Defines if the rule check function will contain asynchronous operations.
* Give Title as short as possible, as descriptive as possible.
* Avoiding hard coded Id’s.
* Use the function for reuse code and call the function wherever it requires.