Views

Views are part of the MVC architecture. They are code responsible for presenting data to end users. In a Web application, views are usually created in terms of view templates which are PHP script files containing mainly HTML code and presentational PHP code. They are managed by the view application component which provides commonly used methods to facilitate view composition and rendering. For simplicity, we often call view templates or view template files as views.

Best Practices

Views are responsible for presenting models in the format that end users desire. In general, views

- should mainly contain presentational code, such as HTML, and simple PHP code to traverse, format and render data.
- should not contain code that performs DB queries. Such code should be done in models.
- should avoid direct access to request data, such as `$_GET`, `$_POST`. This belongs to controllers. If request data is needed, they should be pushed into views by controllers.
- may read model properties, but should not modify them.

To make views more manageable, avoid creating views that are too complex or contain too much redundant code. You may use the following techniques to achieve this goal:

- use layouts to represent common presentational sections (e.g. page header, footer).
- divide a complicated view into several smaller ones. The smaller views can be rendered and assembled into a bigger one using the rendering methods that we have described.
- create and use widgets as building blocks of views.
- create and use helper classes to transform and format data in views.