The MVC-Web development pattern is a model for how the MVC pattern is being interpreted in web application frameworks. MVC-Web reflects the evolutionary changes that have occurred in the MVC development pattern as it has been implemented in web frameworks.

The MVC-Web model component is generally responsible for maintaining the application state. Its responsibilities include:

* Data persistence: maintain a database or an abstract database interface
* Transaction processing: execute application logic that operates on the application state
* External interface: manage interactions with external agents, such as web services or legacy systems
* Query handling: provide information to view and controller elements in response to queries.

The MVC-Web view component presents a user interface, including data presentation and input devices. Its responsibilities include:

* Information retrieval and display: present information to the user; query the model as necessary to obtain information to be displayed
* User input: present input forms and controls that allow the user to interact with the application
* Client-side dynamic behavior: provide an interactive client-side experience for the user (using JavaScript, Ajax, or other means); This may include input completion, input validation or other implementations of application-specific rules or functions that are otherwise a responsibility of the model component.

The MVC-Web controller component has three primary responsibilities.

* Front controller: receive incoming requests and route them to the appropriate handler
* Action handlers: receive request parameters; validate request parameter syntax (this may repeat validation done by view elements); orchestrate request handling by invoking appropriate model elements
* Control flow: invoke the appropriate view element as a response to the request being processed, depending upon the outcome of the action invoked.

Components of the MVC-Web pattern interact in the following ways.

* Model-view: View elements may query the model to obtain information to be displayed to the user
* Model-controller: Controller action elements call on model elements to carry out requested transactions. Model functions can include executing application logic, updating the database, or invoking services of external agents. Controller elements may request data from the model to be passed to view elements.
* Controller-view: Controller elements respond to requests originating with view elements. The controller also determines which view element will be presented to the user in response to a request. Controller elements may prepare information for use by view elements.