MVC architecture

* The Model-View-Controller (MVC) architecture in Java is a design pattern that provides a structured approach for Web developing applications.
* It separates the application’s concerns into three main components: the model, the view, and the controller.
* Each component has a specific role and responsibility within the architecture.

***Model*** — Represents the business layer of the application

* ***View*** — Defines the presentation of the application
* ***Controller*** — Manages the flow of the application

A diagram of a browser

Description automatically generated

A diagram of a data flow

Description automatically generated

In the Java Programming context, Model-java classes,view -display the data (jsp,html) and controller – maintains the flow of the application (servlets).

In the context of the server-client architecture, the process of handling a page request can be described as follows:

* A client, typically a web browser, initiates a request and sends it to the server-side controller.
* The controller receives the request and interacts with the model component. It retrieves the necessary data from the model, which may involve processing and manipulating the data as required.
* Once the controller has gathered the requested data, it transfers this data to the view layer.
* The view layer, utilizing the provided data, generates the appropriate output or representation of the requested page. Finally, the generated result is sent back to the client’s browser, completing the request-response cycle.

MVC architecture offers a lot of advantages for a programmer when developing applications, which include:

* Multiple developers can work with the three layers (Model, View, and Controller) simultaneously
* Offers improved scalability, that supplements the ability of the application to grow
* As components have a low dependency on each other, they are easy to maintain
* A model can be reused by multiple views which provides reusability of code
* Adoption of MVC makes an application more expressive and easy to understand
* Extending and testing of the application becomes easy

Laravel,struts,spring mvc,ruby on rails examples of frameworks build on mvc architecture.