=> Spring MVC :

Spring => ✓
SpringBoot => ✓ } Autoconfigroation

=> Spring MVC => Web Application can be Developed
(servlets | spring web Mvc)

=> Spring 1.x => spring MVC & spring web MVC
spring 2. => Spring WebMVC =>

=> Distributed Application

C → B → B

User → GPay → Banking App

=> SOAP (Webservices) }
=> RESTfull services }

| Web Application

C <=> B

=> iMoon => UI

Spring web MVC :

Spring Web MVC module is developed based on 2 design patterns
a) MVC Design Pattern
b) Front Controller Design Pattern ✓

Spring MVC Components :

1) Front Controller (Dispatcher Servlet)
2) Handler Mapper
3)Controller ✓
4) ModelAndView
5) ViewResolver
6)View

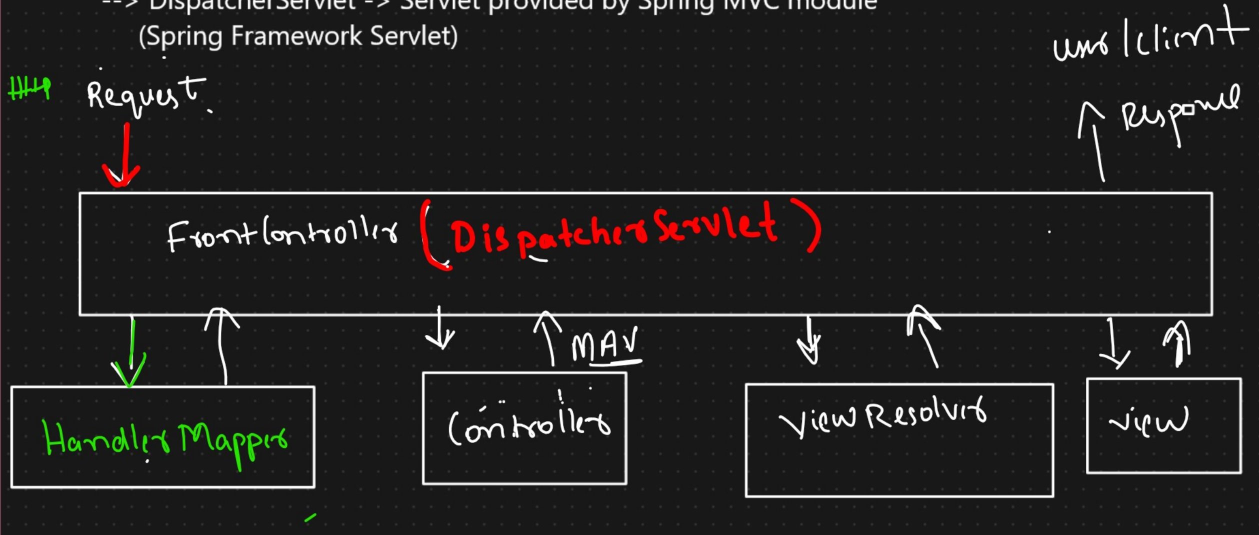
1) Front Controller is responsible to perform - pre processing and post processing of incoming request of an app

(pre processing)
ex: capturing form data }
taking request ✓

(Post processing)
ex: sending response back to client --> client understandable format

--> Spring Web MVC based application -> DispatcherServlet as a front Controller

--> DispatcherServlet -> Servlet provided by Spring MVC module
(Spring Framework Servlet)



Handles Mapper --> pre defined class -- Spring Web MVC

--HM-> to identify Request Handler(Controller)
--Request Handler --> Program/app which handles incoming request
--RequestHandlers==Controllers
--Handler Mapper -> identify request handler(Controller) and return details to DispatcherServlet

--Controller --> responsible to handle request
-- Spring WebMVC -- We can create our own Controllers

-- There some predefined controller classes --> MultiActionController , AbstracttCommandController etc

-- We can create Controllers using @Controller annotation

ModelAndView --->

-->Once Request Processing is done--> Controller will return ModelAndView object to DispatcherServlet

--> ModelAndView --> predefined class avalaible in --> Spring WebMVC module

--> ModelAndView --> Model represents Data and view represents logical view name

View Resolver --

--> Responsible to identify view files (extension and location)

(Many internal predefined view resolvler classes -- XmlViewResolver, URLBasedViewResolver etc)

DispatcherServlet will give model date and vew file to view component then view component is responsible to rendering the data in vew files (responding back to client in client understandable format)

Spring Web MVC -- Summary

↓

DispatcherServlet acts as Front Controller

↓

HandlerMapper would identify Request Handler

↓

Conroller will process the request and return the model data and logical view name
Controller also refered as Request handler

↓

View Resolver identifies view file (location and Extention) Prefix and Suffix

↓

View Component will render model data on the vew file

↓

DispatcherServlet will send response to client in client understandable format

Developing Spring Web MVC application:

springboot-starter-web dependency is used to build web-app

@Controller annotation is sued to represent our class as Spring Controller or Request Handler

@RequestMapping annotations are used to bind Controller method to URL pattern

Controller method returning ModelAndView Object

for ModelAndView object --a)data in form of key and value b)logical view name (no extension)

Model data can be accessed in view file using key

@GetMapping is used to bind our data to request (HTTP GET Request)

Model is an interface -- to store data in form of key and value pair
Model is used to send data from Controller to View(UI)