Professor and Class,

**Architecture model 1:** The first architecture model makes hefty use of JSP pages as the focal points of the application. All client requests are sent directly to the JSP. There’s a clear tie here between the visual display of the page and the business logic that precedes it. Yet, they do not lend well to large-scale applications. These applications use JSP pages to handle HTTP parameters, parse business logic, and manage HTTP sessions. This model is efficient and quick to build, but it lacks scalability, a necessary feature for many enterprise applications.

**Architecture model 2:**

This second architecture model for Java EE applications implements the use of Servlet classes, which sit between requests for pages and the application logic that processes requests. This servlet performs front-end processing of requests and authorization parameters, before determining which JSP page will be sent back to the client. This architecture model is often referred to as the “MVC” (model-view-controller) architecture, and it’s become highly popular with enterprise team since its release. Applications that use this model tend to be easier to maintain over time.

I'm not entirely sure if the discussion question is referring to the second or first model here. I believe that the question is an attempt to get us to consider the second model as better overall compared to the first architecture setup. Those kinds of servlets, which I've also seen called "scriptlets", tend to spawn exponentially: all these files that are a jumble of HTML and Java JSP code mixed together, hard-coded into a mess of links and business logic which remains inseparable from the presentation layer. The newer MVC is certainly better on the mere merits of its separation of concerns. I couldn't imagine the hassle of managing an application built with the first model on a team of dozens of developers.

Refs:

<https://www.javatpoint.com/model-1-and-model-2-mvc-architecture>

<https://download.oracle.com/otn_hosted_doc/jdeveloper/1012/developing_mvc_applications/adf_aboutmvc2.html>

Regards,

Daniel