Short answer questions:

1. Describe the Spring MVC workflow.

* Spring MVC utilizes the Front-End-Controller pattern. All incoming requests are handled by the Dispatcher Servlet. The steps are as follows:
* 1.) The client sends an HTTP Request to the Dispatcher Servlet.
* 2.) The Dispatcher Servlet receives the request an communicates with the Handler Mapping to map the request to a controller according to its action verb.
* 3.) The Dispatcher Servlet sends the request to the controller.
* 4.) The controller processes the request and sends a view name back to the Dispatcher Servlet. Note: can jump to step 7 if using only static resources (client side rendering).
* 5.) The Dispatcher Servlet communicates with the View Resolver to map the view name with an existing implementation of the view.
* 6.) The View Resolver renders the view.
* 7.) The response is sent back to the client.
* Configuration: deployment descriptor, root context, dispatcher servlet

1. Describe how to integrate Spring and Hibernate using Spring ORM.

* In your beans.xml file, create the following two beans to integrate Spring with Hibernate: HibernateJpaVendorAdapter, EntityManagerFactory bean

1. What is a RESTful web service?

* A web service is the interpolation of data for software applications that have many platforms and frameworks.
* REST – Representational State Transfer
* Useful for working with API’s
* URL end points for data transfer (of many different types). These endpoints are exposed so they should be properly secured.

1. What is SOA and why is it useful?

* S O API

1. Describe step-by-step how to expose a REST API connected to persistent data source using Spring Boot, Spring Data, and Spring REST.

* Use the @RestController which combines @Controller and @ResponseBody for your Controllers (beans with service methods)
* Define your endpoint urls

Give specific technical details. You may refer to source code but not your notes.

Submit your answers in a text document via Slack.