Basics Of Spring Boot

1. What is Spring Boot?

  - Spring Boot is a framework for building Java applications.
  - It simplifies the process of setting up and configuring Spring applications.
  - With Spring Boot, you can quickly create standalone, production-ready applications.

2. Key Features of Spring Boot

  - **Auto-Configuration:** Spring Boot automatically configures your application based on its dependencies.
  - **Starter Dependencies:** Spring Boot provides starter dependencies that include commonly used libraries and frameworks, making it easy to get started with new projects.
  - **Embedded Servers:** Spring Boot includes embedded servers like Tomcat, Jetty, and Undertow, so you don't need to deploy your application to an external server.
  - **Actuator:** Spring Boot Actuator provides production-ready features like health checks, metrics, and monitoring.

3. Getting Started:
  - To start using Spring Boot, you need to have Java installed on your computer.
  - You can create a new Spring Boot project using Spring Initializr **(https://start.spring.io/)** or your preferred IDE.
  - Add dependencies like Web, JPA, Security, etc., based on your project requirements.

4. Project Structure:
  - The main Java source code goes into the `src/main/java` directory.
  - Configuration files, templates, and static resources are stored in the `src/main/resources` directory.
  - Dependencies and project settings are managed in the `pom.xml` (if using Maven) or `build.gradle` (if using Gradle) file.

5. Developing Your Application:
  - Use annotations like `@Controller`, `@RestController`, `@Service`, `@Repository` to define components.
  - Implement business logic, data access, and web endpoints in your application.

6. Testing:
  - Write unit tests and integration tests to verify the functionality of your application.
  - Spring Boot provides excellent support for testing with frameworks like JUnit and Mockito.

7. Running Your Application:
  - You can run your Spring Boot application from your IDE or using the `mvn spring-boot:run` command OR Press ctrl + F5;

 - Access your application through the browser or tools like Postman to test its functionality.

8. Deployment:
  - Once your application is ready, package it as a JAR or WAR file using Maven or Gradle.
  - Deploy the packaged artifact to your preferred environment, such as a cloud platform or an on-premises server.