Explain the difference between Java Persistence API, Hibernate and Spring Data JPA

* JPA (Java Persistence API), JPA is a specification (JSR 338), JPA does not have implementation, Hibernate is one of the implementation for JPA, Hibernate is a ORM tool, Spring Data JPA is an abstraction above Hibernate to remove boiler plate code when persisting data using Hibernate.
* Difference between Spring Data JPA and Hibernate - https://dzone.com/articles/what-is-the-difference-between-hibernate-and-sprin-1
* Intro to JPA - <https://www.javaworld.com/article/3379043/what-is-jpa-introduction-to-the-java-persistence-api.html>

**JPA (Java Persistence API)**

* Java EE (now Jakarta EE) established the Java Persistence API (JPA), a specification that standardizes the mapping of Java objects to relational database tables.
* It offers a collection of interfaces and annotations that specify the contract for ORM (Object Relational Mapping) in Java applications, including @Entity, EntityManager, and @Query.
* However, JPA does not offer a functional implementation; it is merely a specification. You need a JPA provider that implements these specifications in order to carry out persistence operations.

**Hibernate**

* Hibernate is a high-performance object/relational persistence and query service that is freely available for download under the terms of the GNU Lesser General Public License (LGPL). In addition to handling the mapping of Java classes to database tables and Java data types to SQL data types, Hibernate also offers tools for data retrieval and query.

**Spring Data JPA**

* Implementing JPA-based (Java Persistence API) repositories is made simple by Spring Data JPA, a member of the broader Spring Data family. It facilitates the development of data access technologies-based Spring-powered applications.
* It can be very difficult to implement a data access layer for an application. To run the most basic queries, too much boilerplate code must be written. You get lost when you include frequently required options like auditing and pagination.
* By limiting the effort to what is truly required, Spring Data JPA seeks to greatly enhance the implementation of data access layers. As a developer, you can use any number of techniques to write your repository interfaces, and Spring will automatically set them up for you.