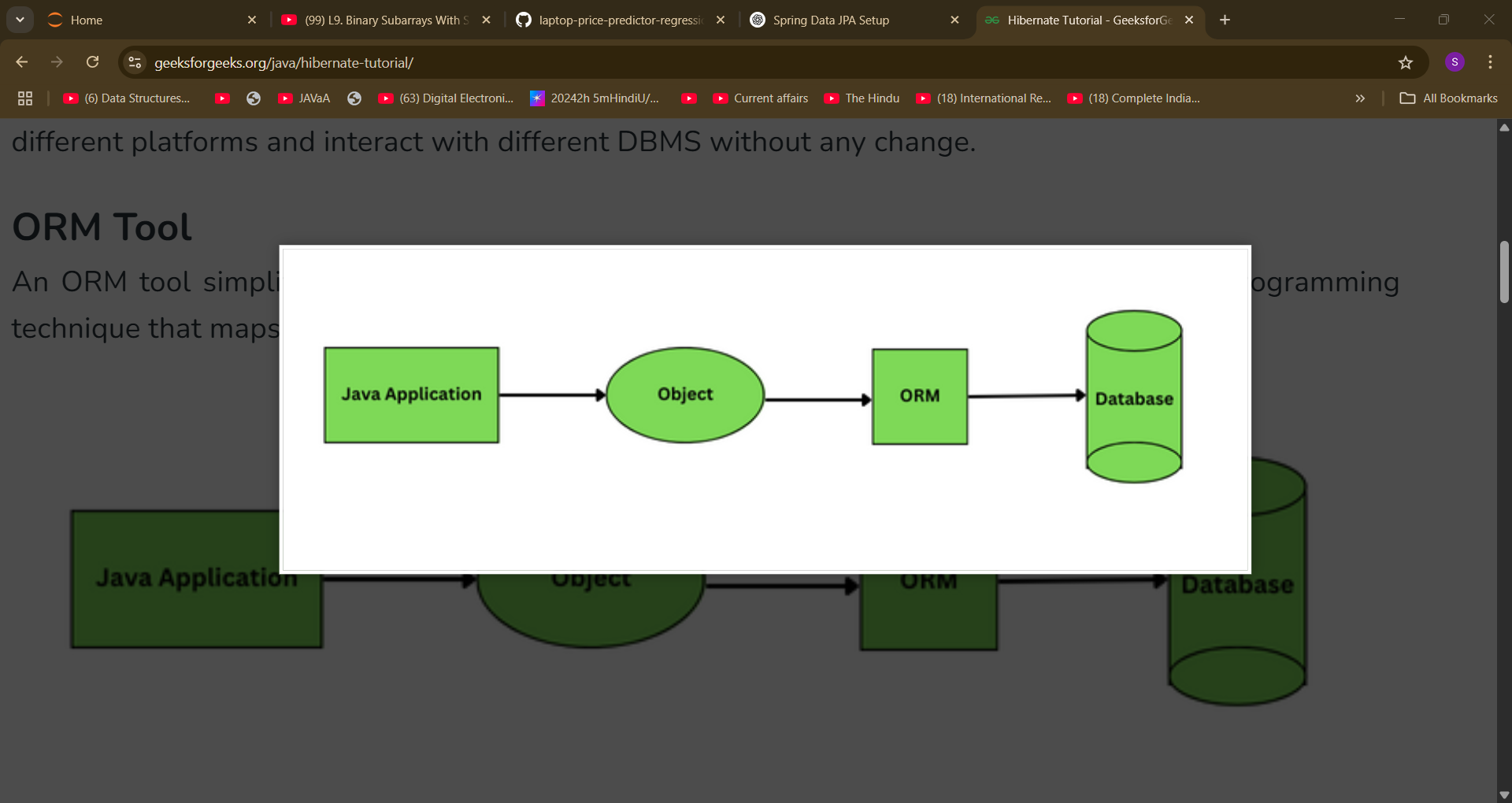
**Hands On 4**

Java Persistence API (JPA)

* The Java Persistence API (JPA) is the Java API specification that bridges the gap between relational databases and object-oriented programming by handling the mapping of the Java objects to the database tables and vice-versa.
* Does not contain concrete implementation of the specification.
* Provides a specification for persisting, reading, and managing data from your Java object to relational tables in the database.

Hibernate

* a Java ORM (Object-Relational Framework) , maps Java objects to database tables, streamlines data persistence and retrieval without the need for complex SQL queries.
* Hibernate is one of the implementations of JPA.
* Persistence of data is handled by Session.
* It uses HQL (Hibernate Query Language).
* In Hibernate, implementation classes are provided.



Spring Data JPA

* Spring Data JPA is a JPA Data Access Abstraction.
* Spring Data JPA is not an implementation or JPA provider, it's just an abstraction used to significantly reduce the amount of boilerplate code required to implement data access layers for various persistence stores.
* Persistence of data is handled by EntityManager.
* It uses JPQL (Java Persistence Query Language).
* In JPA, no implementation classes are provided.

Example – Saving a Product Object

**Hibernate Code: Manual boilerplate: session handling, transaction, try-catch, rollback**

public void saveProduct(Product product) {

Session session = sessionFactory.openSession();

Transaction tx = null;

try {

tx = session.beginTransaction();

session.save(product);

tx.commit();

} catch (Exception e) {

if (tx != null) tx.rollback();

e.printStackTrace();

} finally {

session.close();

}

}

**Spring Data JPA Code: Much simpler: No transaction code, session, or factory required. Handled by Spring.**

* ProductRepository.java

public interface ProductRepository extends JpaRepository<Product, Integer> {

}

* ProductService.java

@Service

public class ProductService {

@Autowired

private ProductRepository productRepository;

public void saveProduct(Product product) {

productRepository.save(product);

}

}