Core Java Task

1. Complete customer dashboard
2. Create customer table in database with following columns and all varchar  
   email, custname, city, phone, password;
3. Add some dummy customers in customer table
4. Add DbConnection class in CMS java project and make sure it is connected to Db

Hibernate

<https://docs.jboss.org/hibernate/orm/6.1/userguide/html_single/Hibernate_User_Guide.html#architecture>

1. ORM framework => object relational mapping
   1. Mapping of java objects to database table
   2. Mapping of data types -> automatic type conversion [ XXXDialect ]
   3. Mapping of different types of relationships
      1. Java -> isa hasa, equals
      2. DB -> PK, FK, CK, UK, Joins
   4. Creation of sql queries is taken care
   5. DB connection, loading driver and executing queries
   6. Handing exceptions
   7. By default commit false
2. Creating a simple hibernate maven project
   1. Create a quickstart Maven project
   2. Add dependencies in pom.xml file
      1. Hibernate-core
      2. MYSQL [ DB driver ]
   3. Hibernate configuration file => hibernate.cfg.xml [src/main/resources]
      1. Connection parameters
      2. Dialect
      3. Other DB related configurations
   4. Create entity classes
   5. To map java class with database tables
      1. XML based configuration
      2. Annotation based configurations  
         <mapping class=”” or resources=”.xml”>
   6. Load the configuration file
   7. Tables were created automatically by hibernate => No  
      DB first approach
   8. CRUD operations
      1. Save or persist [IMP] => insert
      2. Merge[IMP] or update => update
      3. Delete or remove [IMP] =>delete
      4. Get(<name of the class>, id) or load=> Select by id
      5. createQuery(<string>, return type) => Select all
   9. HQl queries => where, order by, group by, having or aggregate functions

Or palceholders :id

1. OneToOne => 1 User has 1 Vehicle
   1. Unidirectional
   2. Bidirectional => mappedBy
   3. Canchange the column name : @JoinColumn
2. Embeddable (on the class) and Embedded (on the property)
3. ElementCollection => can be used on collections of type primitives
   1. @JoinTable => to specify the name of the joined table
4. OneToMany => can be used on collections of type user defined data types [ class ]
   1. By default it created a 3rd table
   2. @JoinColumn -> will not create the 3rd table