/* /*@author P.A.S.S.
/* /:-
/* Q. What is hibernate?
ans:- Hibernate is a pure java object-relational mapping (ORM) and persistence framework that allows
you to map plan old java objects to relational database tables using (xml) configuration files.
Q. Advantages of hibernate?
ans:- 1) Hibernate is database independent.
2) It's provide automatic generate pk.
3) Automatic table creation.
4) less code to write.
5) No SQL to write. 6) Improved performance.
7) If there is change in database or in any table then the only need to change xml file properties.
Q.What is ORM?
ans:- ORM stands for object to Relational mapping.ORM is The automated persistence object in a java application
in a relational database.
Q. ORM Tools?-
ans:- 1)Hibernates 2) Ibatis 3) Toplink 4)JPA 5)JDU
Q. What is Bootstrap?
ans:- Bootstrap is the most popular HTML, CSS and JavaScript framework for developing a responsive and mobile
friendly website.
Q. What is JasperReport?
ans:- JasperReports is a Java class library, and it is meant for those Java developers who need to add reporting capabilities to their applications.
Se
Q.What is a responsive website
ans:- A website is called responsive website which can automatically adjust itself to look good on all devices, from
smart phones to desktops etc.
<meta content="width=device-width, initial-scale=1" name="viewport"/>
Q.What does ORM consists of ?
Ans. An ORM solution consists of the following four pieces->
o API for performing basic CRUD operations
o API to express queries referring to classes
o Facilities to specify metadata o Optimization facilities: dirty checking, lazy associations fetching
Optimization facilities : dirty checking, lazy associations fetching
Q. What are the ORM levels?
ans:- * Pure relational (stored procedure)

* Light object mapping (JDBC)

- * Medium object mapping
- * Full object mapping (composition, inheritance, polymorphism, persistence by reachability)

Q. Core interfaces of hibernate?

ans:- * session factory

- * session
- * Query
- * criteria
- * Transaction
- * Session factory sf = new configuration().configure().buildSessionFactory();
- * Session session = sf.openSession();
- * Query q = session.createQuery().
- * Criteria cr = session.createCriteria().

Q. Hibernate v/s JDBC.

ans:- Hibernate is database independent. No need to take care of mapping tables to database as hbm.xml file will do it.

less code to write, automatic table creation with the help of hbm.xml .

JDBC is database dependent, need to create tables in database, need to write SQL codes. No automatic table is created in JDBC it is done by SQL codes, more time consuming from hibernate.

Q. Hibernate.cfg.xml?

ans:- It is a configuration file that contain properties required to establish connection with database.

Q. How to use dialect in cfg file?

Q. How to establish connection with Hibernate.

ans:-add dependency in pom.xml file.

<dependency>

<groupId>org.hibernate

<artifactId>hibernate-core</artifactId>

<version>3.6.10.Final</version>

</dependency>

then inside the cfg.xml

<hibernate-configuration>

<session-factory>

```
<\!property name="hibernate.connection.driver\_class">\!com.mysql.jdbc.Driver<\!/property>
```

property name="hibernate.connection.password">pass1234/property>

 $<\!property name="hibernate.connection.url">\!jdbc:mysql://localhost:3306/project_3<\!/property>\!$

cproperty name="hibernate.connection.username">root/property>

cproperty name="hbm2ddl.auto">update/property>

cproperty name="show_sql">true

<mapping resource="in/co/rays/project_3/hbm/UserDTO.hbm.xml" />

</session-factory>

</hibernate-configuration>

ans:- 1. beginTransaction() 2.save() 3.update() 4.createQuery() 5.merge() 6.persist() 7.flush() 8.delete()	
Q. How to create object of Query? ans:- Query query = session.createQuery(hqlQuery);	
Q. Hbm.xml ? ans:- It is a mapping file which instructs hibernate to map defined class to database.	
Q. Dialect ? ans:- Dialect is an interpreter which convert HQL Query to the native Query(It depends which dialect you are using). It converts Objects to SQL and SQL to Object.	
Q. cascading? ans:- cascading removed child object when parent object is removed. In hibernate we define it is hbm.xml in <set> tag.</set>	
Q. What is show SQL Property doing in cfg.xml ans:- It logs all the generated SQL statement to the console.	
Q. What is <mapping resource=""> ? ans:- contains the path of hbm.xml mapping.</mapping>	
Q. What is session factory? ans:- It is a DCP (Database Connection Pool). It is created by using Config setting in hibernate.cfg.xml. This is created only one instance in their lifetime that is session. Create one Session factory for one database. It is thread safe (many thread can't access it) immutable (can't change memory area) and optionally hold second level cache It follows factory and Singleton Design pattern.	
session factory provides the instance of session. It is a factory of session. It holds the data of second level cache that is not enable by default.	
Session Factory sf = new configuration().configure().buildSessionFactory().	
Methods of session factory:- 1) openSession() 2) getCurrentSession() 3) getClassMetadata() 4) getCollectionMetadata() 5) getAllClassMetadata() 6) getAllCollectionMetadata() 7) getStatistics() 8) close() 9) isClosed() 10) evict()	
Q. Open session v/s current session.	

ans:- If you use session factory.getCurrentSession(), you'll obtain a "currrent session" which is bound to the lifecycle of the transaction and will be automatically flushed and closed when the transaction

end (commit or rollback).

In case of sessionfactory.opensession(), you will have to manage the session yourself and to flush and close it "manually".

Q. What is session?

ans:- session is an interface. It's the main run-time interface between a Java application and Hibernate.

A session is used to get physically connection with database.

Session session = sf.opensession();

Method of session:-

- * begin Transaction()
- * save()
- * update()
- * createQuery()
- * merge()
- * persist()
- * flush()
- * delete()

Q. What is Transaction?

ans:- Transaction is an interface. Allows the application to define unit of work. It is a set of database calls.

Transaction trans = session.beginTransaction

begin a unit of work and return associated transaction object.

Q. session.save v/s session.persist.

ans:- Both make the transient instance persist but "save" return serializable object where as "persist" return void. [return type int]

Q. session.update v/s session.merge.

ans:- At first look both methods seems similar because both of then are used to convert the object which is in detached state into

persistence state. but (The major difference between update and merge is that update method cannot be used when same object

exists in the session. update method "No unique record exception" thrown an exception).

use "update" method If you are sure that session does not contain persistent object with the same identifier. In case of merge we don't need to worry about state of session.

Q. Session.get() v/s Session.load().

ans:- Both are used for retrieving object from database. But in case of get() we got original record and if no record will found then

it will return Null.

In case of Load() we will get proxy object and in case if no record is found then it will get "object not found" exception.

Q. evict(); clear(); and close(); ?

ans:- evict(dto):- An object is renamed or removed from first level cache, It used to remove particular object from session.

Clear():- It is called when we can removed all the objects from first level cache, session is open but data is clear from cache.

Close():- It is used for destroy memory area and session.

Q. hbm2ddl.auto? ans:- hbm2ddl.auto contain database schema and perform automatic table create and drop operation (1) validate (2) update (3) create (4) create.drop (5) None.
Q. What is DOCTYPE ? ans:- A DOCTYPE is a document type declaration is an instruction that associates a particular xml file with a document type definition.
Q. hibernate.cfg.xml ? ans:- It is configuration file that contain properties required to establish connection with database.
Q. What is criteria? ans:- It is an interface for retrieving the entities by composing criteria object. It is used to perform search operation. Criteria c = session.createCriteria(abc.class)
Q. criteria v/s detached criteria? ans:- Criteria is attached with session at compile time where as detached criteria is attach (can be) with session at run time. Detached criteria allows to create Queries without session. Detached criteria DC = Detachedcriteria.forclass()
Q. What is Model Factory? ans:- Model Factory is a factory of model classes also called DAO, It follows factory design pattern, if there are
implementation then it provides access to the user that which implementation class to instantiate. Q.which design patterns are followed by Model Factory? ans:- 1) Factory design:- the class which have ability to create instance of another class that is called factory class and follow factory design pattern. 2) Bridge design:- Bridge Design pattern decouples an Interface from its implementation class .Interface acts as a bridge which has multiple implementation classes.
Q. How to use bootstrap? ans:- using library:-
Q. How to apply header? ans:- <div class="header" style="background-color: #43302E; width: 100%;"></div>
Q. How to show Hi,Guest in right side? ans:- <div class="collapse navbar-collapse" id="navbarNav"></div>

```
Q. How to create Login Box?
ans:-<div="row log1">
     <div class="col-md-4"></div>
         <div class="col-md-4">
           <div class="card">
                <div class="card-body"></div>
               </div>
            </div>
         </div>
       <div class="col-md-4"></div>
  </div>
Q. How to display success box?
ans:- <div class="alert alert-success alert-dismissible">
        <button type="button" class="close" data-dismiss="alert">&times;</button>
                <%=ServletUtility.getSuccessMessage(request)%></div>
Q. How to display error box?
  <div class="alert alert-danger alert-dismissible">
      <button type="button" class="close" data-dismiss="alert">&times;</button>
        <%=ServletUtility.getErrorMessage(request)%>
   </div>
O. How to Concat icon and text field?
ans:- <div class="col-sm-12">
          <div class="input-group">
              <div class="input-group-prepend">
                 <div class="input-group-text">
                    <i class="fa fa-envelope grey-text" style="font-size: 1rem;"></i>
                </div>
            </div>
                     type="text"
           <input
                                     class="form-control"
                                                             name="login"
                                                                              placeholder="Enter
                                                                                                     email"
value="<%=DataUtility.getStringData(dto.getLogin())%>">
        </div>
     </div>
Q. How to show fields in header with icon?
ans:- 
             <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-</p>
toggle="dropdown" aria-haspopup="true"
                                         aria-expanded="false"> <font style="color: white;">User</font></a>
          <div class="dropdown-menu" aria-labelledby="navbarDropdown">
               <a class="dropdown-item" href="<%=ORSView.USER_CTL%>">
                    <i class="fas fa-user-circle"></i>Add User</a>
                <a class="dropdown-item" href="<%=ORSView.USER LIST CTL%>">
                     <i class="fas fa-user-friends"></i>User List</a>
           </div>
```

Q. How to show toggler on header?

```
ans:- <nav class="navbar navbar-expand-lg"> <a class="navbar-brand"
                          href="<%=ORSView.WELCOME CTL%>"><img
                          src="<%=ORSView.APP CONTEXT%>/img/custom.png" width="150px"
                          height="40px"></a>
                 <button class="navbar-toggler" type="button" data-toggle="collapse"
                          data-target="#navbarNav" aria-controls="navbarNav"
                          aria-expanded="false" aria-label="Toggle navigation">
                          <span class="navbar-toggler-icon"></span><i class="fas fa-bars"</pre>
                                  style="color: #fff; font-size: 28px;"></i>
                 </button>
                 <div class="collapse navbar-collapse" id="navbarNav">
 Q. What is the difference between load() and get()?
ans:-
get():-
 1) If you are not sure that the object exists, then use one of the get() methods.
 2) get() method will return null if the unique id is not found in the database.
 3) get() will hit the database immediately.
load():-
 1) Only use the load() method if you are sure that the object exists.
 2) load() method will throw an exception if the unique id is not found in the database.
 3) load() just returns a proxy by default and database won't be hit until the proxy is first invoked.
Q.What is the general flow of Hibernate communication with RDBMS?
ans:- The general flow of Hibernate communication with RDBMS is:
        Load the Hibernate configuration file and create configuration object. It will automatically load
        all hbm mapping files
        Create session factory from configuration object
        Get one session from this session factory
        Create HQL Query
        Execute query to get list containing Java objects
Q.What is HQL?
Ans. It is Hibernate Query Language. It is database independent means queries written in HQL can be used for any
database.
Q.What is difference between attached and detached criteria in hibernate?
Ans. The detached criteria allows you to create the query without Session.
     Using a DetachedCriteria is exactly the same as a Criteria except you can do the initial creation and setup of
your query without having access to the session.
     When it comes time to run your query, you must convert it to an executable query with
getExecutableCriteria(session).
     What is Session?
Q.
Ans. Session is an interface. The main runtime interface between a Java application and Hibernate.
     A Session is used to get a physical connection with a database.
```

Q.

Criteria vs HQL?

欽� data,	HQL is to perform both select and non-select operations on the data, but Criteria is only for selecting the
uutu,	we cannot perform non-select operations using criteria.
欽◆	HQL is suitable for executing Static Queries, where as Criteria is suitable for executing Dynamic Queries
欽◆	HQL doesn't support pagination concept, but we can achieve pagination with Criteria.
欽 ◆ 	Criteria used to take more time to execute than HQL.
	ference between Hibernate VS JDBC?
	Hibernate:- Hibernate is database independent, no need to take care of mapping tables to database as (ML file will do it.
	JDBC:- JDBC is database dependent, need to create tables in DB.
Q. Wh	at is DOCTYPE?
	OOCTYPE :- a document type declaration is an instruction that associates a particular XML file with a nent type definition.
	nat is Dialect?
using)	Dialect is an interpreter which converts HQL Query to the native query. (It depends which dialect you are
	It converts Object to SQL and SQL to Object.
Q. Wh	at is Cascading?
Ans. C <set></set>	Cascading removed Child object when Parent Object is removed. In Hibernate we define it in hbm.xml in tag.
	nat is show_sql property doing in cfg.xml?
	: logs all the generated SQL statements to the console. hibernate.c3p0.min_size 欽�Minimum number of JDBC connections in the pool. Hibernate default: 1
	hibernate.c3p0.max_size ************************************
	hibernate.c3p0.timeout 欽� When an idle connection is removed from the pool (in second). Hibernate
defaul	t: 0, never expire.
	at is the use of Hibernate.cfg.xml?
Ans. It	: Is a configuration file that contain properties required to establish connection with database.
Q. Wh	nat is <mapping resource=""> ?</mapping>
Ans. C	Contains the path of hbm.xml mapping
Q. wh	at is hbm.xml?
Ans. It	is mapping file which instructs hibernate to map defined class to database.
Q. Ses	sion.save vs Session.persist?
ans:-B void.	oth make the transient instance persist but save 欽�return serializable object where as persistent returns
	cian undata ve Socian marga?

Q. Session.update vs Session.merge?

ans:-At first look both update() and merge() methods seems similar because both of them are used to convert the object which is in detached state into persistence state,

object exists in the session. Use Update ≰ of you are sure that session does not contain persistent object with the same identifier. In case of merge we don't need worry about state of session. Q. In which case Rollback occurs? 飲**�**Loss of communication with participant in case of distributed transactions 欽**�**An invalid or out of range parameter. Q. What is Criteria? Ans. It is an interface for retrieving entities by composing criteria objects. It is used for performing Search 欽� operations. Q. Difference between Session.get() vs Session.load()? Ans. Both are used for retrieving object from database. But in case of get() we get original record and if no record is found than it will return NULL. In case of Load() we will get proxy object and in case if no record is found then we will get ObjectNotFound 欽� exception. Q. What is state of object? ans:- 1.Transient 2.Persistence 3.Detached. Q. Why Caching? Ans. Application Performance Optimization. A cache is designed to reduce the traffic between our application and database. Q. What is query cache? ans:- We use query cache in case if a query is executed again and again because it holds the quary along with its result set. cache contain original result rather than persistent object. second level cache should be enable. need to set permission query.setCacheable(true); ______ Q. How to enable second level cache? Ans:- # Add 2 configuration setting in hibernate.cfg.xml file <property name="cache.provider_class">org.hibernate.cache.EhCacheProvider/property>

cproperty name="hibernate.cache.use_second_level_cache">true/property>

but the major difference between update and merge is that update method cannot be used when the same

```
(means the current session has all the permissions on object but other session can only read it.
# There are four ways to use second level cache. <cache usage="read-only" />
   read-only: caching will work for read only operation.
   Non strict-read-write: caching will work for read and write but one at a time.
   read-write: caching will work for read and write, can be used simultaneously.
Q. which cache you can use instead of Ehcache?
ans:- 1)Ehcache, jboss, swarm, Oscache
public CollegeDTO findByPK(long pk) throws ApplicationException {
                 Session session = null;
                 CollegeDTO dto = null;
                 try {
                          session = HibDataSource.getSession();
                          dto = (CollegeDTO) session.get(CollegeDTO.class, pk);
                          System.out.println(dto);
                 } catch (HibernateException e) {
                          throw new ApplicationException("Exception: Exception in getting course by pk");
                 } finally {
                          session.close();
                 }
                 return dto;
        }
public CollegeDTO fingByName(String name) throws ApplicationException {
                 Session session=null;
                 CollegeDTO dto=null;
                 try {
                          session=HibDataSource.getSession();
                          Criteria criteria=session.createCriteria(CollegeDTO.class);
                          criteria.add(Restrictions.eq("name", name));
                          List list=criteria.list();
                          if(list.size()==1){
                                   dto=(CollegeDTO) list.get(0);
                 } catch (HibernateException e) {
              throw new ApplicationException(
                        "Exception in getting User by Login " + e.getMessage());
         } finally {
              session.close();
                 return dto;
```

```
}
public UserDTO findByLogin(String login) throws ApplicationException {
                  Session session = null;
                  UserDTO dto = null;
                  try {
                           session = HibDataSource.getSession();
                           Criteria criteria = session.createCriteria(UserDTO.class);
                           criteria.add(Restrictions.eq("login", login));
                           List list = criteria.list();
                           if (list.size() == 1) {
                                    dto = (UserDTO) list.get(0);
                           }
                  } catch (HibernateException e) {
                           e.printStackTrace();
                           throw new ApplicationException("Exception in getting User by Login " + e.getMessage());
                  } finally {
                           session.close();
                  return dto;
         }
public UserDTO authenticate(String login, String password) throws ApplicationException {
                  Session session = null;
                  UserDTO dto = null;
                  session = HibDataSource.getSession();
                  Query q = session.createQuery("from UserDTO where login=? and password=?");
                  q.setString(0, login);
                  q.setString(1, password);
                  List list = q.list();
                  if (list.size() > 0) {
                           dto = (UserDTO) list.get(0);
                  } else {
                           dto = null;
                  return dto;
         }
public List getMeritList(int pageNo, int pageSize) throws ApplicationException {
                  Session session = null;
                  List list = null;
                  try {
                           session = HibDataSource.getSession();
```

```
StringBuffer hql = new StringBuffer("from MarksheetDTO where (physics>33 and
chemistry>33 and maths>33) order by (physics+chemistry+maths) desc");
                         if (pageSize > 0) {
                                 pageNo = ((pageNo - 1) * pageSize) + 1;
                                 hql.append(" limit" + pageNo + "," + pageSize);
                         }
                         Query query = session.createQuery(hql.toString());
                         list = query.list();
                } catch (Exception e) {
                         throw new ApplicationException("Exception in marksheet list" + e.getMessage());
                } finally {
                         session.close();
                return list;
           }
ModelFactory():-
     public class ModelFactory{
     private static ResourceBundle rb = ResourceBundle.getBundle("in.co.rays.project 3.bundle.system");
     private static final string DATABASE =rb.getString("DATABASE");
     private static ModelFactory mFactory=null;
     private static HashMap modelCache = new HashMap();
    private ModelFactory(){}
   public static ModelFactory getInstance(){
        if(mFactory == null){
                mFactory = new ModelFactory();
        }
       return mFactory;
   }
 public MarksheetModelInt getMarksheetModel(){
       MarksheetModelInt marksheetModel=(MarksheetModelInt)modelCache.get("marksheetModel");
    if(marksheetModel==null){
        if("Hibernate".equals(DATABASE)){
               marksheetModel=new MarksheetModelHibImp();
       if("JDBC".equals(DATABASE)){
            marksheetModel = new MarksheetModelJDBCImpl();
        }
           modelCache.put("marksheetModel",marksheetModel);
```

```
}
     return marksheetModel;
public CollegeModelInt getCollegeModel(){
       CollegeModelInt collegeModel = (ColegeModelInt)modelCache.get("collegeModel");
       if(collegeModel==null){
            if("Hibernate".equals(DATABASE){
                collegeModel = new CollegeModelHibImp();
            if("JDBC".equals(DATABASE){
               collegeModel = new CollegeModelJDBCImpl();
        modelCache.put("collegeModel",collegeModel);
        return collegeModel;
public RoleModel getRoleModel(){
        RoleModelInt roleModel = (RoleModelInt)modelCache.get("roleModel");
     if(roleModel==null){
          if("Hibernate".equals(DATABASE)){
               roleModel = new RoleModelHibImp();
          if("JDBC".equals(DATABASE)){
              roleModel = new RoleModelJDBCImpl();
          modelCache.put("roleModel",roleModel);
   } return roleModel;
}
public CourseModel getCourseModel(){
    CourseModelInt courseModel = (CourseModelInt)modelCache.get("courseModel");
    if(courseModel == null){
        if("Hibernate".equals(DATABASE){
             courseModel = new CourseModelHibImp();
        if("JDBC".equals(DATABASE){
               courseModel = new CourseModelHibImp();
       modelCache.put("courseModel",courseModel);
     return courseModel;
}
public SubjectModel getSubjectModel(){
   SubjectModelInt subjectModel = (SubjectModelInt)modelCache.get("subjectModel");
    if(subjectModel == null){
         if("Hibernate".equals(DATABASE){
            subjectModel = new SubjectModeHibImp();
           }
```

```
if("JDBC".equals(DATABASE){
                subjectModel = new SubjectModelHibImp();
           modelCache.put("subjectMode",subjectMode);
       } return subjectModel;
}
   #pagination:-
 -----
                                 if(pageSize>0){
                                   criteria.setFirstResult((pageNo-1)*pageSize);
                                   criteria.setMaxResults(pageSize);
#List:-
-----
              public List list(int pageNo, int pageSize) throws ApplicationException {
                 // TODO Auto-generated method stub
                 Session session = null;
                 List list = null;
                 try {
                          session = HibDataSource.getSession();
                          Criteria criteria = session.createCriteria(CollegeDTO.class);
                          if (pageSize > 0) {
                                   pageNo = ((pageNo - 1) * pageSize) + 1;
                                   criteria.setFirstResult(pageNo);
                                   criteria.setMaxResults(pageSize);
                          list = criteria.list();
                 } catch (HibernateException e) {
                          throw new ApplicationException("Exception: Exception in College list");
                 } finally {
                          session.close();
                 }
                 return list;
        }
public List search(UserDTO dto, int pageNo, int pageSize) throws ApplicationException {
                 // TODO Auto-generated method stub
                 System.out.println(
                                   "hellllo" + pageNo + "....." + pageSize + "......" + dto.getId() + "....." +
dto.getRoleId());
                 Session session = null;
                 ArrayList<UserDTO> list = null;
                 try {
                          session = HibDataSource.getSession();
                          Criteria criteria = session.createCriteria(UserDTO.class);
```

```
if (dto != null) {
                                     if (dto.getId() != null) {
                                              criteria.add(Restrictions.like("id", dto.getId()));
                                     if (dto.getFirstName() != null && dto.getFirstName().length() > 0) {
                                              criteria.add(Restrictions.like("firstName", dto.getFirstName() + "%"));
                                    }
                                     if (dto.getLastName() != null && dto.getLastName().length() > 0) {
                                              criteria.add(Restrictions.like("lastName", dto.getLastName() + "%"));
                                     if (dto.getLogin() != null && dto.getLogin().length() > 0) {
                                              criteria.add(Restrictions.like("login", dto.getLogin() + "%"));
                                     if (dto.getPassword() != null && dto.getPassword().length() > 0) {
                                              criteria.add(Restrictions.like("password", dto.getPassword() + "%"));
                                     if (dto.getGender() != null && dto.getGender().length() > 0) {
                                              criteria.add(Restrictions.like("gender", dto.getGender() + "%"));
                                     if (dto.getDob() != null && dto.getDob().getDate() > 0) {
                                              criteria.add(Restrictions.eq("dob", dto.getDob()));
                                     if (dto.getLastLogin() != null && dto.getLastLogin().getTime() > 0) {
                                              criteria.add(Restrictions.eq("lastLogin", dto.getLastLogin()));
                                     if (dto.getRoleId() > 0) {
                                              criteria.add(Restrictions.eq("roleId", dto.getRoleId()));
                                     if (dto.getUnSuccessfullLogin() > 0) {
                                              criteria.add(Restrictions.eq("unSuccessfulLogin",
dto.getUnSuccessfullLogin()));
                           // if pageSize is greater than 0
                           if (pageSize > 0) {
                                     pageNo = (pageNo - 1) * pageSize;
                                     criteria.setFirstResult(pageNo);
                                     criteria.setMaxResults(pageSize);
                           list = (ArrayList<UserDTO>) criteria.list();
                  } catch (HibernateException e) {
                           throw new ApplicationException("Exception in user search");
                  } finally {
                           session.close();
                  return list;
         }
Q. How to create object of model?
```

CollegeModelInt model = ModelFactory.getInstance().getCollegeModel();

Q. What is role of session?
ans:- 1) wrap JDBC connection.
2) factory of transaction.
3) holds first level cache.
dependencies:- java mail API 1.4.7 , log4j 1.2.17 , mysql 5.1.6 , c3p0 0.9.5.2 , servlet 4.0 , hibernate 3.6 , jasper 6.11
technologies:- jdk 1.8, jsp 2.2 , servlet 4.0 , java mail API 1.4.7 , jdbc 3.0 , jquery 1.11 , mysql 5.1.6 , bootstrap4,hibernate 3.6

tools:- ERWin, Eclipse, ArgoUMI, MySqlYog, tibcotool,