Color.java

```
package com.ecommerce;


public class Color {


 private long COLORID;
    private String name;


    public Color() {


    }
    public Color(String name) {
        this.COLORID = 0;
        this.name = name;
    }


    public long getCOLORID() {return this.COLORID; }
    public String getName() { return this.name;}
    public void setCOLORID(long id) { this.COLORID = id;}
    public void setName(String name) { this.name = name;}



}
```

EProduct.java

```
package com.ecommerce;


import java.math.BigDecimal;
import java.util.*;
```

```
public class EProduct {


private long ID;

private String name;

private BigDecimal price;

private Date dateAdded;

 //Component object

private ProductParts parts;



public ProductParts getParts() {

return parts;

}


public void setParts(ProductParts parts) {

this.parts = parts;

}



private List<Color> colors;

    private Collection<ScreenSizes> screenSizes;

    private Set<OS> os;

    private Map finance ;


public EProduct() {


}
```

```
public EProduct(long id, String name, BigDecimal price, Date dateAdded) {
this.ID = id;
this.name = name;
this.price = price;
this.dateAdded = dateAdded;
}
public long getID() {
return ID;
}

public void setID(long iD) {
ID = iD;
}

public String getName() {
return name;
}

public void setName(String name) {
this.name = name;
}

public BigDecimal getPrice() {
return price;
}

public void setPrice(BigDecimal price) {
this.price = price;
}
```

```
public Date getDateAdded() {

return dateAdded;

}


public void setDateAdded(Date dateAdded) {

this.dateAdded = dateAdded;

}


public List<Color> getColors() {

return colors;

}

public void setColors(List<Color> colors) {

this.colors = colors;

}


public Collection<ScreenSizes> getScreenSizes() {

return screenSizes;

}


public void setScreenSizes(Collection<ScreenSizes> screenSizes) {

this.screenSizes = screenSizes;

}


public Set<OS> getOs() {

return os;

}


public void setOs(Set<OS> os) {
```

```
this.os = os;
}

public Map getFinance() {
return finance;
}

public void setFinance(Map finance) {
this.finance = finance;
}




}
```

Finance.java

```
package com.ecommerce;

public class Finance {

 private long FINANCEID;
    private String name;
    private String ftype;

    public Finance() {

    }
    public Finance(String name, String ftype) {
        this.FINANCEID = 0;
        this.name = name;
        this.ftype = ftype;
    }

    public long getFINANCEID() {return this.FINANCEID; }
    public String getName() { return this.name;}
    public String getFtype() { return this.ftype;}
    public void setFINANCEID(long id) { this.FINANCEID = id;}
    public void setName(String name) { this.name = name;}
    public void setFtype(String ftype) { this.ftype= ftype;}

}
```

OS.java

```
package com.ecommerce;

public class OS {

 private long OSID;
    private String name;

    public OS() {

    }
    public OS(String name) {
        this.OSID = 0;
        this.name = name;
    }
    public long getOSID() {return this.OSID; }
    public String getName() { return this.name;}

    public void setOSID(long oSID) {
OSID = oSID;
}
public void setName(String name) {
this.name = name;
}

}
```

ProductParts.java

```
package com.ecommerce;

public class ProductParts {

 private String hdd;
    private String cpu;
    private String ram;

    public String getHdd() { return this.hdd;}
    public String getCpu() { return this.cpu;}
    public String getRam() { return this.ram;}

    public void setHdd(String value) { this.hdd= value;}
    public void setCpu(String value) { this.cpu= value;}
    public void setRam(String value) { this.ram= value;}
}



package com.ecommerce;

public class ScreenSizes {

 private long SCREENID;
    private String size;

    public ScreenSizes() {

    }
    public ScreenSizes(String size) {
        this.SCREENID = 0;
        this.size = size;
    }

    public long getSCREENID() {return this.SCREENID; }
    public String getSize() { return this.size;}
    public void setSCREENID(long id) { this.SCREENID = id;}
    public void setSize(String size) { this.size = size;}

}
```

ScreenSizes.java

```
package com.ecommerce;
```

```
public class ScreenSizes {

 private long SCREENID;
    private String size;

    public ScreenSizes() {

    }
    public ScreenSizes(String size) {
            this.SCREENID = 0;
            this.size = size;
    }

    public long getSCREENID() {return this.SCREENID; }
    public String getSize() { return this.size;}
    public void setSCREENID(long id) { this.SCREENID = id;}
    public void setSize(String size) { this.size = size;}

}
```

Color.hbm.xml

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC
"-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">
<hibernate-mapping package="com.ecommerce">
   <class name="Color" table="colors">
```

```
        <id name="COLORID" type="long" column="ID">
           <generator class="identity"/>
        </id>
        <property name="name" type="string" column="COLOR_NAME"/>
```

EProduct.hbm.xml

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC
"-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">
<hibernate-mapping package="com.ecommerce">
<class name="EProduct" table="eproduct">
<id name="ID" column="ID">
<generator class="increment" />
</id>
<property name="name" type="string" column="NAME" />
<property name="price" type="big_decimal" column="PRICE" />
<property name="dateAdded" type="timestamp"
column="DATE_ADDED" />

<component name="parts" class="com.ecommerce.ProductParts">
<property name="hdd" column="parts_hdd" type="string" />
<property name="cpu" column="parts_cpu" type="string" />
<property name="ram" column="parts_ram" type="string" />
</component>
```

```
<list name="colors" cascade="all">

<key column="product_id" />

<list-index column="idx" />

<one-to-many class="com.ecommerce.Color" />

</list>


<set name="os" cascade="all">

<key column="product_id" />

<one-to-many class="com.ecommerce.OS" />

</set>


<bag name="screenSizes" cascade="all">

<key column="product_id"></key>

<one-to-many class="com.ecommerce.ScreenSizes" />

</bag>

<map name = "finance"  cascade="all">

        <key column = "product_id"/>

        <index column = "ftype" type = "string"/>

        <one-to-many class="com.ecommerce.Finance"/>

     </map>



</class>

</hibernate-mapping>
```

Finance.hbm.xml

```
<?xml version="1.0"?>

<!DOCTYPE hibernate-mapping PUBLIC
```

```
"-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">
<hibernate-mapping package="com.ecommerce">
  <class name="Finance" table="finance" >
    <id name="FINANCEID" type="long" column="ID">
      <generator class="identity"/>
    </id>
    <property name="name" type="string" column="NAME"/>
    <property name="ftype" type="string" column="FTYPE"/>
  </class>
</hibernate-mapping>
```

OS.hbm.xml

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC
"-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">
<hibernate-mapping package="com.ecommerce">
  <class name="Finance" table="finance" >
    <id name="FINANCEID" type="long" column="ID">
      <generator class="identity"/>
    </id>
    <property name="name" type="string" column="NAME"/>
    <property name="ftype" type="string" column="FTYPE"/>
  </class>
</hibernate-mapping>
```

ScreenSizes.hbm.xml

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC
```

```
"-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">
<hibernate-mapping package="com.ecommerce">
   <class name="OS" table="os">
      <id name="OSID" type="long" column="ID">
         <generator class="identity"/>
      </id>
      <property name="name" type="string" column="NAME"/>
   </class>
</hibernate-mapping>
```

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC
"-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">
<hibernate-mapping package="com.ecommerce">
   <class name="ScreenSizes" table="screensizes">
      <id name="SCREENID" type="long" column="ID">
         <generator class="identity"/>
      </id>
      <property name="size" type="string" column="SIZE"/>
   </class>
</hibernate-mapping>
```

ComponenetMappingDemo.java

```
package com.simpli;
```

```
import java.io.*;
import java.io.PrintWriter;
import java.util.List;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.*;

import org.hibernate.*;

import com.ecommerce.EProduct;
import com.ecommerce.ProductParts;

@WebServlet("/component_mapping-demo")
public class ComponentMappingDemo extends HttpServlet {

protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
PrintWriter out = response.getWriter();
out.println("<html><body>");

SessionFactory factory = HibernateUtil.getSessionFactory();
Session session = factory.openSession();


// STEP 3 Query the DB and get the data
List<EProduct> eproducts = session.createQuery("from EProduct").list();
out.println("<table border=1>");
for (EProduct prod : eproducts) {
out.println("<tr>" + "<td>" + prod.getID() + "<td>" + prod.getName() + "<td>" + prod.getPrice() + "<td>"
+ prod.getDateAdded());

ProductParts parts = prod.getParts();
out.println("<td> Parts =" + parts.getCpu() + ", " + parts.getHdd() + ", " + parts.getRam());

}

out.println("</table>");

session.close();

out.println("</body></html>");
}

}
```

HibernateInit.java

```
package com.simpli;

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.*;
import org.hibernate.*;


@WebServlet("/init")
public class HibernateInit extends HttpServlet {


protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
PrintWriter out = response.getWriter();
out.println("<html><body>");


// STEP 1: Get a Session (connection) from the Session Factory class
SessionFactory factory = HibernateUtil.getSessionFactory();


// STE2 Create the session object
Session session = factory.openSession();


out.println("Hibernate Session opened.<br>");


session.close();


out.println("Hibernate Session closed.<br>");


out.println("</body></html>");
}}
```

HibernateUtil.java

```
package com.simpli;
```

```java
import org.hibernate.SessionFactory;
import org.hibernate.boot.*;
import org.hibernate.boot.registry.*;

public class HibernateUtil {

private static final SessionFactory sessionFactory;

static {
try {
StandardServiceRegistry standardRegistry = new StandardServiceRegistryBuilder()
.configure("hibernate.cfg.xml").build();

Metadata metaData = new MetadataSources(standardRegistry).getMetadataBuilder().build();

sessionFactory = metaData.getSessionFactoryBuilder().build();

} catch (Throwable th) {
throw new ExceptionInInitializerError(th);
}
}

public static SessionFactory getSessionFactory() {
return sessionFactory;
}

}
```

ProductDetailsServlet.java

```
package com.simpli;

import java.io.*;
import java.util.*;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.*;
import org.hibernate.*;
import com.ecommerce.*;

@WebServlet("/product-details")
public class ProductDetailsServlet extends HttpServlet {

protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
PrintWriter out = response.getWriter();
out.println("<html><body>");

SessionFactory factory = HibernateUtil.getSessionFactory();
Session session = factory.openSession();

// STEP 3 Query the DB and get the data
List<EProduct> eproducts = session.createQuery("from EProduct").list();
```

```
out.println(
"<table border=1><th>ID<th>NAME<th>PRICE<th>DATE_ADDED<th>COLORS<th>OS<th>SCREENSIZES<th>FINANCE</th>");

for (EProduct prod : eproducts) {
out.println("<tr>" + "<td>" + prod.getID() + "<td>" + prod.getName() + "<td>" + prod.getPrice() + "<td>"+
prod.getDateAdded());
// Display the colors the product is available in
List<Color> colors = prod.getColors();
out.println("<td>");
for (Color color : colors)
out.println(color.getName());

// Display the OS the product is available in
Set<OS> Oses = prod.getOs();
out.println("<td>");
for (OS os : Oses)
out.println(os.getName());

// Display the Screen Sizes the product is available in
Collection<ScreenSizes> screenSizes = prod.getScreenSizes();
out.println("<td>");
for (ScreenSizes ss : screenSizes)
out.println(ss.getSize());

//                              Map finances = prod.getFinance();
//                              out.println("<td>");
//          if (finances.get("CREDITCARD") != null) {
//                Finance f = (Finance) finances .get("CREDITCARD");
//                out.println(f.getName() + " &nbsp;");
//          }
//          if (finances .get("BANK") != null) {
//                Finance f = (Finance) finances .get("BANK");
//                out.println(f.getName() + " &nbsp;");
//          }

}
out.println("</table>");

session.close();
out.println("</body></html>");
}
}
```

Hibernate.cfg.xml

```
<?xml version='1.0' encoding='utf-8'?>
<!DOCTYPE hibernate-configuration PUBLIC
"-//Hibernate/Hibernate Configuration DTD 3.0//EN"
"http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
  <session-factory>
```

```
    <!-- Database connection settings -->
    <property name="connection.driver_class">com.mysql.cj.jdbc.Driver</property>
    <property name="connection.url">jdbc:mysql://localhost:3306/ecommerce</property>
    <property name="connection.username">root</property>
    <property name="connection.password">Simplilearn</property>
    <property name="dialect">org.hibernate.dialect.MySQL8Dialect</property>
   <mapping  resource="com/ecommerce/Color.hbm.xml"/>
   <mapping  resource="com/ecommerce/OS.hbm.xml"/>
   <mapping  resource="com/ecommerce/ScreenSizes.hbm.xml"/>
   <mapping  resource="com/ecommerce/Finance.hbm.xml"/>
   <mapping  resource="com/ecommerce/EProduct.hbm.xml"/>
     </session-factory>
</hibernate-configuration>
```

Log4j.properties

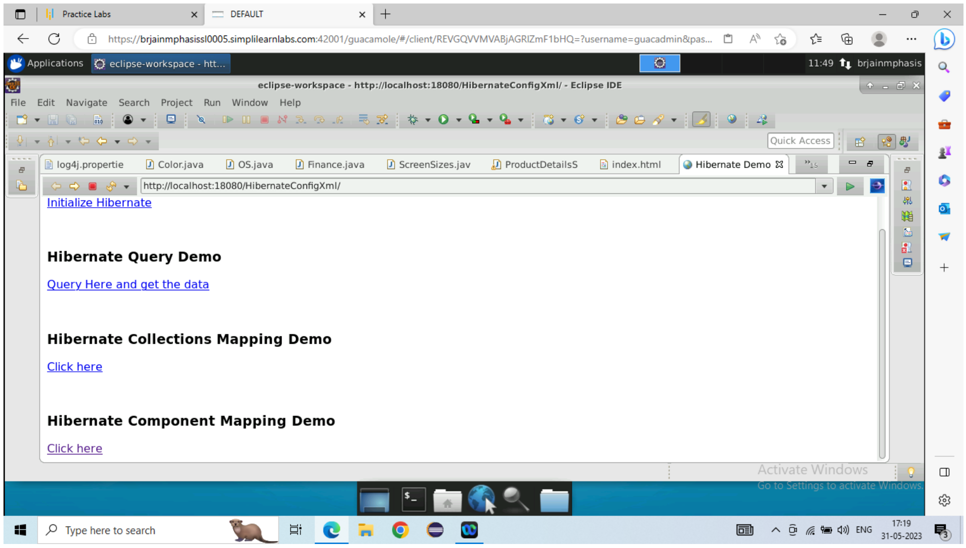
```
# LOG4J configuration
log4j.rootLogger=DEBUG, Appender1,Appender2
log4j.appender.Appender1=org.apache.log4j.ConsoleAppender
log4j.appender.Appender1.layout=org.apache.log4j.PatternLayout
log4j.appender.Appender1.layout.ConversionPattern=%-7p %d [%t] %c %x - %m%n

log4j.appender.Appender2=org.apache.log4j.FileAppender
log4j.appender.Appender2.File/home/brjainmphasis/Downloads/mylog.txt
log4j.appender.Appender2.layout=org.apache.log4j.PatternLayout
log4j.appender.Appender2.layout.ConversionPattern=%-7p %d [%t] %c %x - %m%n
```

```
# Log everything. Good for troubleshooting
log4j.logger.org.hibernate=INFO

# Log all JDBC parameters
log4j.logger.org.hibernate.type=ALL
```

Index.html

```
<title>Hibernate Demo  </title>
<h3>Hibernate Configuration Example </h3>

<a href="init">Initialize Hibernate</a><br>

<br><br>
<h3>Hibernate Query Demo </h3>

<a href="query-demo">Query  Here and get the data</a><br>

<br><br>
<h3>Hibernate Collections Mapping  Demo </h3>
<a href="product-details">Click here</a><br>

<br><br>
<h3>Hibernate Component Mapping  Demo </h3>
<a href="component_mapping-demo">Click here</a><br>
```

Output:



eclipse-workspace - http://localhost:18080/HibernateConfigXml/ - Eclipse IDE

http://localhost:18080/HibernateConfigXml/

Initialize Hibernate

Hibernate Query Demo

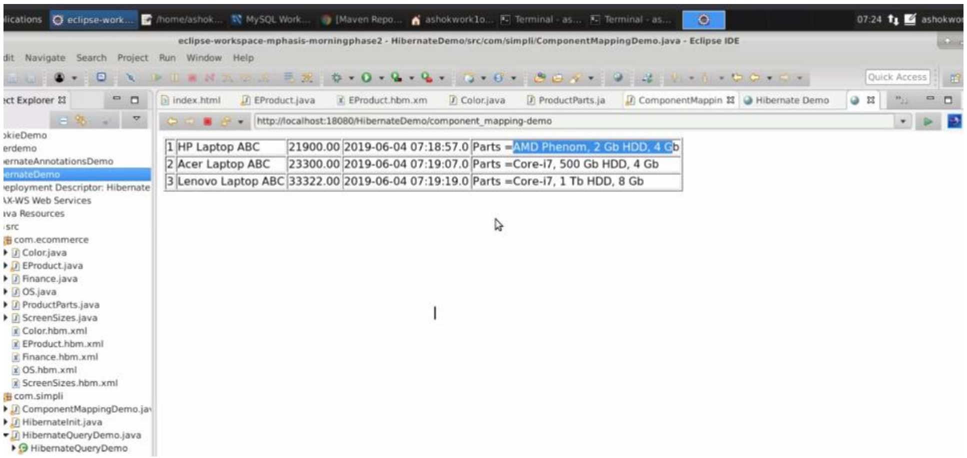
Query Here and get the data

Hibernate Collections Mapping Demo

Click here

Hibernate Component Mapping Demo

Click here



eclipse-workspace-mphasis-morningphase2 - HibernateDemo/src/com/simpli/ComponentMappingDemo.java - Eclipse IDE

http://localhost:18080/HibernateDemo/component_mapping-demo

1	HP Laptop ABC	21900.00	2019-06-04 07:18:57.0	Parts =AMD Phenom, 2 Gb HDD, 4 Gb
2	Acer Laptop ABC	23300.00	2019-06-04 07:19:07.0	Parts =Core-i7, 500 Gb HDD, 4 Gb
3	Lenovo Laptop ABC	33322.00	2019-06-04 07:19:19.0	Parts =Core-i7, 1 Tb HDD, 8 Gb