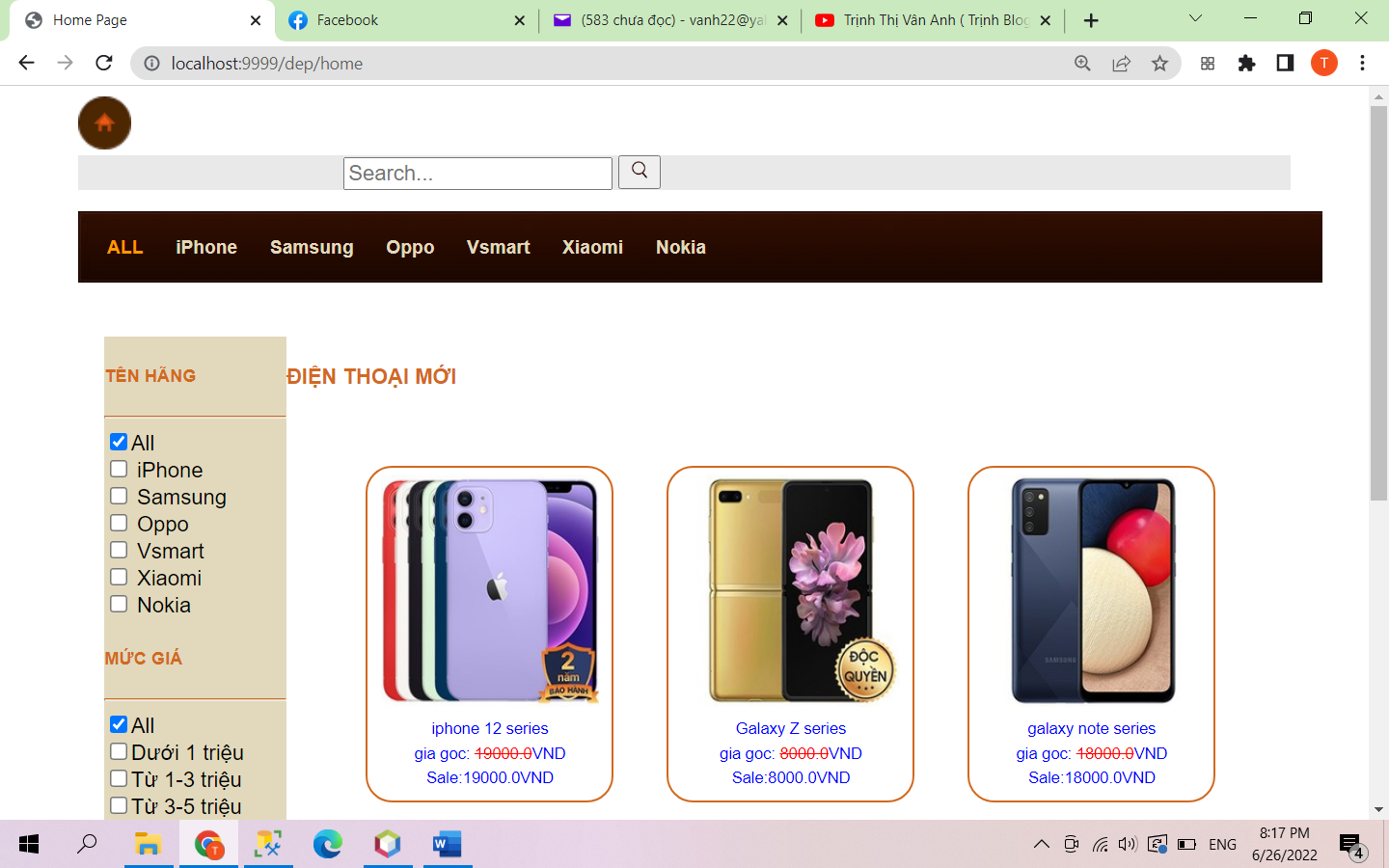
Hướng dẫn làm cái page Home đơn giản



Phần css

|  |  |
| --- | --- |
| Phần đầu | Phần search |
| #wrapper{  margin-left: 5%;  margin-right: 5%;  font-family: arial;  }  .clr {  clear:both;  } | .topnav {  background-color: #e9e9e9;  }  .topnav input[type=text] {  margin-left: 200px;  font-size: 16px;  } |
| Phần menu | |
| #menu\_tab {  width:100%;  }  ul.menu {  float:left;  background:url(../images/background\_menu.jpg) repeat-x;  display:block;  list-style-type:none;  font-weight:bold;  height: 54px;  padding:12px;  width: 100%;  }  ul.menu li {  display:inline;  font-size:14px;  padding-top: 15px;  text-align:center;  line-height:30px;  } | ul.menu li a{  margin: 10px;  padding-left:0px;  text-decoration:none;  text-align:center;  color:#E2D9BC;  position:relative;  }  ul.menu li a:hover{  color: #F90;  }  ul.menu li a.active{  color: #F90;  } |
| Tab trái | Contents |
| Khung  #content{  margin-left: 20px;  width: 100%;  }  Tab trái  #tab1{  width: 15%;  float: left;  background-color: #E2D9BC;  }  ul.tab{  list-style: none;  }  ul.tab li{  display: block;  margin-top: 10px;  }  ul.tab li a{  text-decoration: none;  font-size: 16px;  }  ul.tab li a:hover{  background-color: #F90;  }  ul.tab li a.active{  background-color: #F90;  color: white;  } | #tab2{  width: 80%;  float: left;  }  /\*phần sản phẩm\*/  ul.item{  list-style: none;  float: left;  width: 100%;  }  ul.item li{  width: 25%;  height: 250px;  text-align: center;  float: left;  margin: 20px;  border: 2px solid chocolate;  border-radius: 20px;  }  ul.item li img{  margin-top: 5px;  width: 90%;  height: 70%;  }  ul.item li a{  text-decoration: none;  }  ul.item li:hover{  background: bisque;  }  ul.item p{  margin:5px;  color: blue;  font-size: 12px;  }  .old{  text-decoration: line-through;  color: red;  } |

Dựng Model:

|  |  |
| --- | --- |
| Category.java | Product.java |
| public class Category {  private int id;  private String name,describe; | public class Product {  private String id,name;  private int quantity;  private double price;  private String releaseDate,describe,image;  private Category category; |

Xử lý DAO

|  |  |
| --- | --- |
| Lấy các dòng điện thoại (bảng Categories) | Lấy Category khi biết id |
| public List<Category> getAll() {  List<Category> list = new ArrayList<>();  String sql = "select \* from Categories";  try {  PreparedStatement st = connection.prepareStatement(sql);  ResultSet rs = st.executeQuery();  while (rs.next()) {  Category c = new Category(rs.getInt("id"),  rs.getString("name"),  rs.getString("describe"));  list.add(c);  }  } catch (SQLException e) {  System.out.println(e);  }  return list;  } | public Category getCategoryById(int id) {  String sql = "select \* from Categories where id=?";  try {  PreparedStatement st = connection.prepareStatement(sql);  st.setInt(1, id);  ResultSet rs = st.executeQuery();  if (rs.next()) {  Category c = new Category(rs.getInt("id"),  rs.getString("name"),  rs.getString("describe"));  return c;  }  } catch (SQLException e) {  System.out.println(e);  }  return null;  } |
| Lấy ds sản phẩm theo category id | Lấy ds sản phẩm theo danh sách category id |
| public List<Product> getProductsByCid(int cid) {  List<Product> list = new ArrayList<>();  String sql = "SELECT [id]\n"  + " ,[name]\n"  + " ,[quantity]\n"  + " ,[price]\n"  + " ,[releaseDate]\n"  + " ,[describe]\n"  + " ,[image]\n"  + " ,[cid]\n"  + " FROM [dbo].[Products]\n"  + " where 1=1 ";  if (cid != 0) {  sql += " and cid=" + cid;  }  try {  PreparedStatement st = connection.prepareStatement(sql);  ResultSet rs = st.executeQuery();  while (rs.next()) {  Product p = new Product();  p.setId(rs.getString("id"));  p.setName(rs.getString("name"));  p.setQuantity(rs.getInt("quantity"));  p.setPrice(rs.getDouble("price")); p.setReleaseDate(rs.getString("releaseDate"));  p.setDescribe(rs.getString("describe"));  p.setImage(rs.getString("image"));  Category c = getCategoryById(rs.getInt("cid"));  p.setCategory(c);  list.add(p);  }  } catch (SQLException e) {  }  return list;  } | public List<Product> searchByCheck(int[] cid) {  List<Product> list = new ArrayList<>();  String sql = "SELECT [id]\n"  + " ,[name]\n"  + " ,[quantity]\n"  + " ,[price]\n"  + " ,[releaseDate]\n"  + " ,[describe]\n"  + " ,[image]\n"  + " ,[cid]\n"  + " FROM [dbo].[Products]\n"  + " where 1=1 ";  if (cid!=null && cid[0]!=0) {  sql += " and cid in(";  for (int i = 0; i < cid.length; i++) {  sql += cid[i] + ",";  }  if (sql.endsWith(",")) {  sql = sql.substring(0, sql.length() - 1);  }  sql += ")";  }  try {  PreparedStatement st = connection.prepareStatement(sql);  ResultSet rs = st.executeQuery();  while (rs.next()) {  Product p = new Product();  p.setId(rs.getString("id"));  p.setName(rs.getString("name"));  p.setImage(rs.getString("image"));  p.setPrice(rs.getDouble("price"));  p.setQuantity(rs.getInt("quantity"));  p.setDescribe(rs.getString("describe")); p.setReleaseDate(rs.getString("releaseDate"));  Category c = getCategoryById(rs.getInt("cid"));  p.setCategory(c);  list.add(p);  }  } catch (SQLException e) {  System.out.println(e);  }  return list;  } |
| Đưa ds sản phẩm theo tên sp và miêu tả | Đưa ra ds sản phẩm có giá từ… đến… |
| public List<Product> searchByKey(String key) {  List<Product> list = new ArrayList<>();  String sql = "SELECT [id]\n"  + " ,[name]\n"  + " ,[quantity]\n"  + " ,[price]\n"  + " ,[releaseDate]\n"  + " ,[describe]\n"  + " ,[image]\n"  + " ,[cid]\n"  + " FROM [dbo].[Products]\n"  + " where name like ? or describe like ?";  try {  PreparedStatement st = connection.prepareStatement(sql);  st.setString(1, "%" + key + "%");  st.setString(2, "%" + key + "%");  ResultSet rs = st.executeQuery();  while (rs.next()) {  Product p = new Product();  p.setId(rs.getString("id"));  p.setName(rs.getString("name"));  p.setImage(rs.getString("image"));  p.setPrice(rs.getDouble("price"));  p.setQuantity(rs.getInt("quantity"));  p.setDescribe(rs.getString("describe")); p.setReleaseDate(rs.getString("releaseDate"));  Category c = getCategoryById(rs.getInt("cid"));  p.setCategory(c);  list.add(p);  }  } catch (SQLException e) {  System.out.println(e);  }  return list;  } | public List<Product> getProductsByPrice(double from,double to) {  List<Product> list = new ArrayList<>();  String sql = "SELECT [id]\n"  + " ,[name]\n"  + " ,[quantity]\n"  + " ,[price]\n"  + " ,[releaseDate]\n"  + " ,[describe]\n"  + " ,[image]\n"  + " ,[cid]\n"  + " FROM [dbo].[Products]\n"  + " where price between ? and ? ";  try {  PreparedStatement st = connection.prepareStatement(sql);  st.setDouble(1, from);  st.setDouble(2, to);  ResultSet rs = st.executeQuery();  while (rs.next()) {  Product p = new Product();  p.setId(rs.getString("id"));  p.setName(rs.getString("name"));  p.setQuantity(rs.getInt("quantity"));  p.setPrice(rs.getDouble("price")); p.setReleaseDate(rs.getString("releaseDate"));  p.setDescribe(rs.getString("describe"));  p.setImage(rs.getString("image"));  Category c = getCategoryById(rs.getInt("cid"));  p.setCategory(c);  list.add(p);  }  } catch (SQLException e) {  }  return list;  } |
| Đưa ra 3 sản phẩm mới nhất | Đưa ra 3 sản phẩm cũ nhất |
| public List<Product> getNewProducts() {  List<Product> list = new ArrayList<>();  String sql = "SELECT TOP 3 [id]\n"  + " ,[name]\n"  + " ,[quantity]\n"  + " ,[price]\n"  + " ,[releaseDate]\n"  + " ,[describe]\n"  + " ,[image]\n"  + " ,[cid]\n"  + " FROM [dbo].[Products]\n"  + " order by releaseDate desc";  try {  PreparedStatement st = connection.prepareStatement(sql);  ResultSet rs = st.executeQuery();  while (rs.next()) {  Product p = new Product();  p.setId(rs.getString("id"));  p.setName(rs.getString("name"));  p.setQuantity(rs.getInt("quantity"));  p.setPrice(rs.getDouble("price"));  p.setReleaseDate(rs.getString("releaseDate"));  p.setDescribe(rs.getString("describe"));  p.setImage(rs.getString("image"));  Category c = getCategoryById(rs.getInt("cid"));  p.setCategory(c);  list.add(p);  }  } catch (SQLException e) {  }  return list;  } | public List<Product> getOldProducts() {  List<Product> list = new ArrayList<>();  String sql = "SELECT TOP 3 [id]\n"  + " ,[name]\n"  + " ,[quantity]\n"  + " ,[price]\n"  + " ,[releaseDate]\n"  + " ,[describe]\n"  + " ,[image]\n"  + " ,[cid]\n"  + " FROM [dbo].[Products]\n"  + " order by releaseDate";  try {  PreparedStatement st = connection.prepareStatement(sql);  ResultSet rs = st.executeQuery();  while (rs.next()) {  Product p = new Product();  p.setId(rs.getString("id"));  p.setName(rs.getString("name"));  p.setQuantity(rs.getInt("quantity"));  p.setPrice(rs.getDouble("price"));  p.setReleaseDate(rs.getString("releaseDate"));  p.setDescribe(rs.getString("describe"));  p.setImage(rs.getString("image"));  Category c = getCategoryById(rs.getInt("cid"));  p.setCategory(c);  list.add(p);  }  } catch (SQLException e) {  }  return list;  } |

HomeServlet.java

|  |
| --- |
| DAO d = new DAO();  List<Category> list = d.getAll();  String[] pp={"Dưới 1 triệu",  "Từ 1-3 triệu",  "Từ 3-5 triệu",  "Từ 5-10 triệu",  "Trên 10 triệu"};  boolean[] pb=new boolean[pp.length+1];  pb[0]=true;  List<Product> news=d.getNewProducts();  List<Product> olds=d.getOldProducts();  boolean[] chid = new boolean[list.size()+1];  chid[0]=true;  request.setAttribute("data", list);  request.setAttribute("news", news);  request.setAttribute("olds", olds);  request.setAttribute("pp", pp);  request.setAttribute("pb", pb);  request.setAttribute("cid", 0);  request.setAttribute("chid", chid);  request.getRequestDispatcher("list.jsp").forward(request, response); |

Giao diện: list.jsp

|  |
| --- |
|  |
| <a href="home"> <img src="images/home\_icon.png"  width="40px" height="40px" alt="home" /> </a>  <div class="topnav">  <form action="home1" method="get">  <input type="text" placeholder="Search..." name="key"/>  <button onclick="this.form.submit()">  <img src="images/search\_icon.png"  width="16px" height="16px"/>  </button>  </form>  </div>  <div class="clr"></div> |
|  |
| <div id="menu\_tab">  <c:set var="cat" value="${requestScope.data}"/>  <c:set var="cid" value="${requestScope.cid}"/>  <ul class="menu">  <li><a class="${cid==0?"active":""}" href="home1?cid=${0}">ALL</a></li>  <c:forEach items="${cat}" var="c">  <li><a class="${c.id==cid?"active":""}" href="home1?cid=${c.id}">${c.name}</a></li>  </c:forEach>  </ul>  </div>  <div class="clr"></div> |
|  |
| <div id="content">  <div id="tab1">  <c:set var="chid" value="${requestScope.chid}"/>  <h5 style="color: chocolate">TÊN HÃNG</h5>  <hr style="border-top: 1px solid chocolate "/>  <form id="f1" action="home1">  <input type="checkbox" id="c0" name="cidd"  ${chid[0]?"checked":""}  value="${0}" onclick="setCheck(this)"/>All<br/>  <c:forEach begin="0" end="${cat.size()-1}" var="i">  <input type="checkbox" id="cm" name="cidd"  ${cat.get(i).getId()==cid?"checked":""}  value="${cat.get(i).getId()}"  ${chid[i+1]?"checked":""} onclick="setCheck(this)" />  ${cat.get(i).getName()}  <br/>  </c:forEach>  </form> |
| Javascript |
| function setCheck(obj){  var fries = document.getElementsByName('cidd');  if ((obj.id=='c0') && (fries[0].checked==true))  {  for (var i = 1; i < fries.length; i++)  fries[i].checked = false;  } else{  for (var i = 1; i < fries.length; i++) {  if (fries[i].checked==true) {  fries[0].checked=false;  break;  }  }  }  document.getElementById('f1').submit();  } |
|  |
| <h5 style="color: chocolate">MỨC GIÁ</h5>  <hr style="border-top: 1px solid chocolate "/>  <c:set var="pp" value="${requestScope.pp}"/>  <c:set var="pb" value="${requestScope.pb}"/>  <form id="f2" action="home1">  <input type="checkbox" id="g0" name="price"  ${pb[0]?"checked":""}  value="0" onclick="setCheck1(this)"/>All<br/>  <c:forEach begin="0" end="${4}" var="i">  <input type="checkbox" id="g1" name="price"  ${pb[i+1]?"checked":""}  value="${(i+1)}" onclick="setCheck1(this)"/>${pp[i]}<br/>  </c:forEach>  </form>  </div> |
| function setCheck1(obj){  var fries = document.getElementsByName('price');  if ((obj.id=='g0') && (fries[0].checked==true))  {  for (var i = 1; i < fries.length; i++)  fries[i].checked = false;  } else{  for (var i = 1; i < fries.length; i++) {  if (fries[i].checked==true) {  fries[0].checked=false;  break;  }  }  }  document.getElementById('f2').submit();  } |
|  |
| <div id="tab2">  <c:set var="news" value="${requestScope.news}"/>  <c:if test="${news!=null}">  <h4 style="color: chocolate">ĐIỆN THOẠI MỚI </h4>  <ul class="item">  <c:forEach items="${news}" var="p">  <li>  <a href="#">  <img src="${p.image}" width="80px" height="80px"/>  <p>${p.name}</p>  <p>gia goc: <span class="old">${(p.price)}</span>VND</p>  <p>Sale:${(p.price)}VND</p>  </a>  </li>  </c:forEach>  </ul>  <hr/>  </c:if> |
|  |
| <c:set var="olds" value="${requestScope.olds}"/>  <c:if test="${olds!=null}">  <h4 style="color: chocolate">ĐIỆN THOẠI KHUYẾN MẠI </h4>  <ul class="item">  <c:forEach items="${olds}" var="p">  <li>  <a href="#">  <img src="${p.image}" width="80px" height="80px"/>  <p>${p.name}</p>  <p>gia goc: <span class="old">${(p.price)}</span>VND</p>  <p>Sale:${(p.price)}VND</p>  </a>  </li>  </c:forEach>  </ul>  <hr/>  </c:if> |

ProductsServlet.java (url /home1)

|  |
| --- |
| Có chọn checkbox không? |
| private boolean ischeck(int d, int[] id) {  if (id == null) {  return false;  } else {  for (int i = 0; i < id.length; i++) {  if (id[i] == d) {  return true;  }  }  return false;  }  } |
| String[] pp = {"Dưới 1 triệu",  "Từ 1-3 triệu",  "Từ 3-5 triệu",  "Từ 5-10 triệu",  "Trên 10 triệu"};  boolean[] pb = new boolean[pp.length + 1];  DAO d = new DAO();  List<Category> categories = d.getAll();  boolean[] chid = new boolean[categories.size() + 1];  List<Product> products = new ArrayList<>();  String key = request.getParameter("key");  if (key != null) {  products=d.searchByKey(key);  }  String cid\_raw = request.getParameter("cid");  String[] cidd\_raw = request.getParameterValues("cidd");  String[] price = request.getParameterValues("price");  int cid = 0;  int[] cidd = null;  if (cid\_raw != null) {  cid = Integer.parseInt(cid\_raw);  products = d.getProductsByCid(cid);  if (cid == 0) {  chid[0] = true;  }  }  if (cidd\_raw != null) {  cidd = new int[cidd\_raw.length];  for (int i = 0; i < cidd.length; i++) {  cidd[i] = Integer.parseInt(cidd\_raw[i]);  }  products = d.searchByCheck(cidd);  }  if (price != null) {  double from = 0, to = 0;  for (int i = 0; i < price.length; i++) {  List<Product> temp = new ArrayList<>();  if (price[i].equals("0")) {  from = 0;  to = 20000;  products = d.getProductsByPrice(from, to);  pb[0] = true;  break;  } else {  if (price[i].equals("1")) {  from = 0;  to = 1000;  temp = d.getProductsByPrice(from, to);  products.addAll(temp);  pb[1] = true;  }  if (price[i].equals("2")) {  from = 1000;  to = 3000;  temp = d.getProductsByPrice(from, to);  products.addAll(temp);  pb[2] = true;  }  if (price[i].equals("3")) {  from = 3000;  to = 5000;  temp = d.getProductsByPrice(from, to);  products.addAll(temp);  pb[3] = true;  }  if (price[i].equals("4")) {  from = 5000;  to = 10000;  temp = d.getProductsByPrice(from, to);  products.addAll(temp);  pb[4] = true;  }  if (price[i].equals("5")) {  from = 10000;  to = 20000;  temp = d.getProductsByPrice(from, to);  products.addAll(temp);  pb[5] = true;  }  }  }  }  if (price == null) {  pb[0] = true;  }  if (cid\_raw == null) {  chid[0] = true;  }  if ((cidd\_raw != null) && (cidd[0] != 0)) {  chid[0] = false;  for (int i = 1; i < chid.length; i++) {  if (ischeck(categories.get(i - 1).getId(), cidd)) {  chid[i] = true;  } else {  chid[i] = false;  }  }  }  request.setAttribute("data", categories);  request.setAttribute("products", products);  request.setAttribute("pp", pp);  request.setAttribute("key", key);  request.setAttribute("pb", pb);  request.setAttribute("cid", cid);  request.setAttribute("chid", chid);  request.getRequestDispatcher("list.jsp").forward(request, response); |
| Viết tiếp vào list.jsp |
| <c:set var="list" value="${requestScope.products}"/>  <c:if test="${list!=null}">  <h4 style="color: chocolate">ĐIỆN THOẠI (${list.size()} sản phẩm)</h4>  <ul class="item">  <c:forEach items="${list}" var="p">  <li>  <a href="#">  <img src="${p.image}" width="80px" height="80px"/>  <p>${p.name}</p>  <p>gia goc: <span class="old">${(p.price)}</span>VND</p>  <p>Sale:${(p.price)}VND</p>  </a>  </li>  </c:forEach>  </ul>  </c:if> |

Chỉ cho truy nhập trang từ servlet (không từ file jsp)

|  |
| --- |
| HomeFilter.java url /\* |
| HttpServletRequest req=(HttpServletRequest)request;  HttpServletResponse res=(HttpServletResponse)response;  String uri=req.getServletPath();  if(uri.endsWith(".jsp")){  res.sendRedirect("home");  } |