

Web 技术开发

课程设计报告



计算机科学与技术学院

梁振铎

3013216096

2015 年 7 月 10 日

一、 课程设计的目的和意义

许多常用的 Web 上的服务都与 Web 数据库应用程序有很大关系。在线邮件收发系统、在线购物、论坛与 BBS、企业网站、门户网站等服务背后都有数据库。数据库应用程序是建立现代化网站的必需品。为了学习使用广泛、简单实用的 Web 数据库应用程序构建方法，MySQL 则是最合适的数据库管理系统，PHP 是最常被采用的 Apache Web 服务器相关模块。本课程设计的目的和意义在于学会创建应用所需要的技术，并学习如何创建一个安全、可靠、易维护、具扩充性的 Web 应用程序。

二、 课程设计的主要任务

- 1、在线商店是一个电子商务网站，其应用程序不管理统计、存货控制、薪资、订货与其他任务。
- 2、用户可以选择商品并放入购物车；从第一件物品放入购物车开始计算，购物车内只保留一天内放入的商品，超时即清空；用户若关闭浏览器或登出系统，购物车即清空。
- 3、每位用户有一辆购物车，可随时添加、清空车内商品，任意改变商品数目。
- 4、网站用户可以匿名，在决定购买之前可一直保持匿名状态；若要购买则必须登录；若要登录，必须已经注册。
- 5、注册时需要提供昵称、国家、省市、地址、邮编、电话等信息，并设置登录账号和密码；输入账号、密码即可登录。
- 6、用户购买商品时，其订单存放于数据库中；购物车是尚未与客户产生关联的订单，用户提交支付信息后该订单可转换为完整订单。
- 7、商品信息等存放于数据库中，可以通过信息检索对商品进行筛选。
- 8、用户只能购买有库存的商品；当用户将购物车的内容转换为订单时，系统会检查库存能否满足订单，若不足则通知用户修改购买数目；当用户确认订购后，库存随之减少。

三、 软件平台的选择和搭建

Apache 是 Apache Software Foundation 的开放源代码的服务器软件，由于其跨平台和安全性被广泛使用，是最流行的 Web 服务器之一；

PHP 是一种开源的通用计算机脚本语言，尤其适用于网络开发并可嵌入 HTML 中使用，也是最常用的 Apache Web 服务器相关模块。PHP 本身具有集成网站及数据库环境工具，尤其适合开发 Web 数据库应用程序；

PEAR 是 PHP 扩展应用程序库，它提供了标准化、结构化、经过整理的程序包，可应用于数据验证、访问数据库服务器、付款机制、web service 使用、图形处理、读写文件等众多常见任务；

MySQL 是最为普遍的数据库管理系统，它既快捷又容易使用，由其制造商 MySQL AB 以开放源代码的形式授权；

Zend Studio 是 Zend Technologies 开发的 PHP 语言集成开发环境，也是目前

市面上最优秀的 PHP 语言 IDE;

1、安装 Apache

下载解压 Apache 2.4, 将 httpd.conf 文件中的 SRVROOT 改为安装目录并安装;

2、安装 PHP

下载 PHP 5.6.7 并解压到指定路径下, 将 Apache 的路径信息等加入 php.ini 中;

3、安装 PEAR 软件包

下载最新的 go-pear.phar 文件到 PHP 目录下, 在该目录下执行 php go-pear.phar 命令安装, 并安装 DB、HTML_Template_IT、Mail、HTTP 等包;

4、安装 Zend Studio

下载并安装 Zend Studio 12.0, 配置 PHP 信息, 新建项目;

5、安装 MySQL

下载并安装 MySQL 5.6, 设置账户信息并建立新数据库, 项目中的 db.inc 文件中的 \$hostname、\$databasename、\$username、\$password 几项改为数据库信息以连接数据库;

四、数据库设计

1、表 users 包含用户的帐户信息, 有如下字段:

#	名字	类型	排序规则	属性	空	默认
<input type="checkbox"/> 1	cust_id	int(5)			否	无
<input type="checkbox"/> 2	user_name	varchar(50)	utf8_general_ci		否	无
<input type="checkbox"/> 3	password	varchar(32)	utf8_general_ci		否	无

cust_id 表示客户号, user_name 表示用户名, password 表示密码;

2、表 customer 包含用户的各项信息, 有如下字段:

#	名字	类型	排序规则	属性	空	默认
<input type="checkbox"/> 1	cust_id	int(5)			否	无
<input type="checkbox"/> 2	nickname	varchar(50)	utf8_general_ci		是	NULL
<input type="checkbox"/> 3	address	varchar(50)	utf8_general_ci		是	NULL
<input type="checkbox"/> 4	city	varchar(50)	utf8_general_ci		是	NULL
<input type="checkbox"/> 5	state	varchar(20)	utf8_general_ci		是	NULL
<input type="checkbox"/> 6	zipcode	varchar(10)	utf8_general_ci		是	NULL
<input type="checkbox"/> 7	country_id	int(4)			是	NULL
<input type="checkbox"/> 8	phone	varchar(15)	utf8_general_ci		是	NULL

cust_id 表示客户号, nickname 表示昵称, address 表示地址, city 表示城市, state 表示省份, zipcode 表示邮编, country_id 表示国家代号, phone 表示电话;

3、表 wine 包含商品的各项信息, 有如下字段:

	#	名字	类型	排序规则	属性	空	默认
<input type="checkbox"/>	1	wine_id	int(5)			否	无
<input type="checkbox"/>	2	wine_name	varchar(50)	utf8_general_ci		否	无
<input type="checkbox"/>	3	wine_type	int(2)			否	无
<input type="checkbox"/>	4	year	int(4)			否	无
<input type="checkbox"/>	5	winery_id	int(4)			否	无
<input type="checkbox"/>	6	description	blob			是	NULL

wine_id 表示商品编号，wine_name 表示商品名称，wine_type 表示类型，year 表示年份，winery_id 表示厂商代号，description 表示描述；

4、表 orders 包含订单的各项用户信息，有如下字段：

	#	名字	类型	排序规则	属性	空	默认
<input type="checkbox"/>	1	cust_id	int(5)			否	无
<input type="checkbox"/>	2	order_id	int(5)			否	无
<input type="checkbox"/>	3	date	timestamp(6)			否	CURRENT_TIMESTAMP(6)
<input type="checkbox"/>	4	instructions	varchar(128)	utf8_general_ci		是	NULL
<input type="checkbox"/>	5	creditcard	char(16)	utf8_general_ci		是	NULL

cust_id 表示客户号，order_id 表示订单号，date 表示交易日期，instructions 表示备注，creditcard 表示支付账户；

5、表 items 包含订单的各项商品信息，有如下字段：

	#	名字	类型	排序规则	属性	空	默认
<input type="checkbox"/>	1	cust_id	int(5)			否	无
<input type="checkbox"/>	2	order_id	int(5)			否	无
<input type="checkbox"/>	3	item_id	int(3)			否	无
<input type="checkbox"/>	4	wine_id	int(4)			否	无
<input type="checkbox"/>	5	qty	int(3)			是	NULL
<input type="checkbox"/>	6	price	decimal(5,2)			是	NULL

cust_id 表示客户号，order_id 表示订单号，item_id 订单内商品号，wine_id 表示商品编号，qty 表示数量，price 表示价格；

6、表 inventory 包含库存信息，有如下字段：

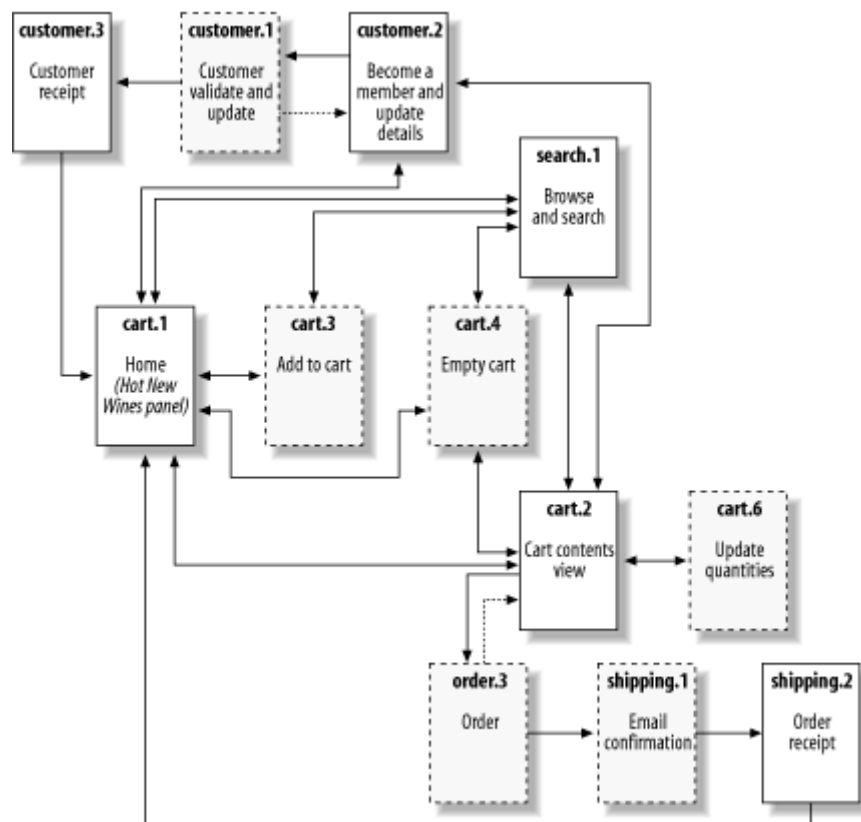
#	名字	类型	排序规则	属性	空	默认
<input type="checkbox"/> 1	wine_id	int(5)			否	无
<input type="checkbox"/> 2	inventory_id	int(3)			否	无
<input type="checkbox"/> 3	on_hand	int(5)			否	无
<input type="checkbox"/> 4	cost	decimal(5,2)			否	无

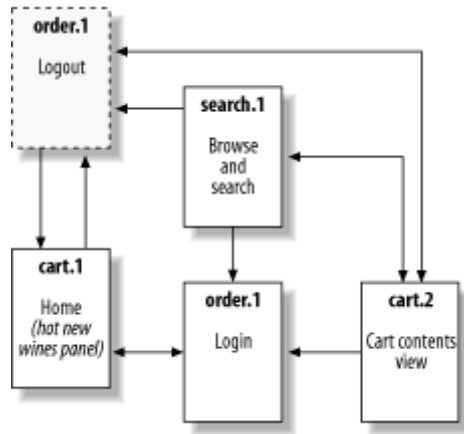
wine_id 表示商品编号，inventory_id 表示仓库编号，on_hand 表示库存量，cost 表示售价；

7、此外，表 countries 包含国家代号信息，表 region 包含商品产地信息，winrey 包含商品厂商信息，表 wine_type 包含商品种类信息。

五、软件设计

在线商店首页为 index.php，用户可在首页把推荐商品加入购物车或者使用其他功能。该应用程序由模板产生的可维护网页，并填入来自数据库的数据；用户驱动的查询及浏览，由用户提供限制搜索或浏览数据库的参数；数据输入与验证；以 session 管理技巧进行的用户追踪；用户验证与管理；错误处理的回执页面。





六、界面设计

整个站点的内容居中显示，左上角显示购物车按钮，右上角显示登录状态，底部显示功能按钮，各页面显示如下：

1、站点首页界面：

尚未登录

购物车总计: \$0.00 (0 件商品)

今日推荐

1996 Lane Gully Premium Wines Patton Cabernet Sauvignon 简介: This is one shot out of the blue! With striking colour, this is a fruit-driven wine, with intense blueberry and blackberry fruit aromas, which carry on to the well crafted, full-flavoured palate, which has great persistence of flavour.	价格: \$24.93 / 瓶 加入购物车
1999 Greenfield Vineyard Mockridge Sparkling White 简介: A very good example of Margaret River chardonnay. Peach and melon scents are followed by a full palate of ripe fruit tastes and a long, dry finish. Chicken dishes, hot or cold, will match it well now or cellar it until 2003.	价格: \$27.17 / 瓶 加入购物车
1997 Anderson and Sons Wines Belcombe Grenache 简介: Rich ruby colour with purple tinges it has a suggestion of peppermint and ripe red berry aromas. It's a bigish wine with a well-structured, richly flavoured palate and firm tannins. Try it with rare beef dishes or cellar until 2005.	价格: \$17.29 / 瓶 加入购物车

[查找商品](#)
[新用户注册](#)
[用户登录](#)

2、注册界面：

尚未登录

用户信息

昵称:
 地址:
 市/县/州:
 省/区/市:
 邮编:
 国家:
 电话:
 邮箱:
 密码:
 确认密码:

提交

[查找商品](#)
[回到主页](#)

3、登陆界面：

尚未登录

登录

用户名:

密码:

提交

回到主页

4、购物车界面：

已登录 root@mysql.com

购物车总计: \$141.02 (6 件商品)

您的购物车

数量	品名	单价	总计
<input type="text" value="1"/>	1996 Lane Gully Premium Wines Patton Cabernet Sauvignon	\$24.93	\$24.93
<input type="text" value="3"/>	1999 Greenfield Vineyard Mockridge Sparkling White	\$27.17	\$81.51
<input type="text" value="2"/>	1997 Anderson and Sons Wines Belcombe Grenache	\$17.29	\$34.58

更新数量

6 件商品, 总计 \$141.02

清空购物车 立刻支付 查找商品 回到主页 退出登录

5、支付界面：

已登录 root@mysql.com

购物车总计: \$0.00 (0 件商品)

提交订单

支付账户:

备注:

提交

回到主页

6、交易成功界面：

已登录 root@mysql.com

交易成功

您的订单号为 1 - 7。请牢记您的订单号。

您购买的商品将配送至，
root
Afghanistan r r 123456

订单详情:

数量	品名	单价	总计
1	1996 Lane Gully Premium Wines Patton Cabernet Sauvignon	\$24.93	\$24.93
3	1999 Greenfield Vineyard Mockridge Sparkling White	\$27.17	\$81.51
2	1997 Anderson and Sons Wines Belcombe Grenache	\$17.29	\$34.58

订单总计: \$141.02

一封确认邮件已经发送到您的邮箱，感谢您在本店购物！

回到主页

七、算法设计（15 分）

主要算法包含输入检测算法（检测是否输入为空、输入是否符合要求）、数据库查询（查询商品信息）、数据库修改算法（购物车内商品与数据库的交互）等。具体代码详见第十一节。

八、编程实现

在线商店应用程序开发是为了满足前述需求，是完整的 PHP 与 MySQL Web 数据库应用程序，并使用 PEAR ITX 模板及 PEAR DB 以获得 HTML 显示及数据库层。本应用有五个独立模块：客户管理、购物车、下单与购物、浏览与搜索、验证，源文件分别放在/customer、/cart、/order、/search、/auth 下面。应用程序中有一组共享组件，包括扩展包 PEAR ITX、验证函数、自定义错误处理器、验证函数、通用函数与常量。模板文件在/templates 下，inc 文件在/include 下。具体代码详见第十一节。

九、所遇到的主要问题

模板中一些老版本 PHP 的函数已经弃用，ereg() 使用 preg_match() 替代，ereg_replace()和 eregi() 使用 preg_replace() 替代，mysql 替换为 mysqli 等。

十、参考文献

《PHP & MySQL Web 数据库应用开发指南》，
Hugh E. Williams, David Lane 著，东南大学出版社

十一、源代码

主页源码：

```
1 <?php
2 require_once "DB.php";
3 require_once "includes/winestore.inc";
4 require_once "includes/template.inc";
5
6 set_error_handler("customHandler");
7
8 function showPanel($connection, &$amp;template)
9 {
10     $query = "SELECT wi.winery_name, w.year,
11               w.wine_name, w.wine_id, w.description
12               FROM wine w, winery wi, inventory i
13               WHERE w.winery_id = wi.winery_id
14               AND w.wine_id = i.wine_id
15               AND w.description IS NOT NULL
16               GROUP BY w.wine_id
17               ORDER BY i.date_added DESC LIMIT 3";
18 }
```



```

19  $result = $connection->query($query);
20  if (DB::isError($result))
21      trigger_error($result->getMessage(), E_USER_ERROR);
22
23  while ($row = $result->fetchRow(DB_FETCHMODE_ASSOC))
24  {
25      $template->setCurrentBlock("row");
26      $template->setVariable("YEAR", $row["year"]);
27      $template->setVariable("WINERY", $row["winery_name"]);
28      $template->setVariable("WINE", $row["wine_name"]);
29      $template->setVariable("DESCRIPTION",
30          $row["description"]);
31      $template->setVariable("VARIETIES", showVarieties
32          ($connection, $row["wine_id"]));
33      $price = showPricing($connection, $row["wine_id"]);
34      $template->setVariable("BOTTLE_PRICE",
35          sprintf("%.2f", $price));
36      $template->setCurrentBlock("link");
37      $template->setVariable("SCRIPT", S_ADDTOCART);
38      $template->setVariable("QTY", "1");
39      $template->setVariable("WINE_ID", $row["wine_id"]);
40      $template->setVariable("STRING", "加入购物车");
41      $template->parseCurrentBlock("link");
42      $template->setCurrentBlock("row");
43      $template->parseCurrentBlock("row");
44  }
45 }
46
47 session_start();
48
49 $template = new winestoreTemplate(T_HOME);
50 $connection = DB::connect($dsn, true);
51 if (DB::isError($connection))
52     trigger_error($connection->getMessage(), E_USER_ERROR);
53
54 showPanel($connection, $template);
55 $template->showWinestore(SHOW_ALL, B_ALL & ~B_UPDATE_CART &
56     ~B_HOME & ~B_PASSWORD &
57     ~B_PURCHASE & ~B_EMPTY_CART);
58 ?>

```

组件定义:

```

1 <?php
2 require_once 'db.inc';

```

```

3 require_once 'customHandler.inc';
4 define("D_INSTALL_PATH", "D:/Program Files/Apache24/htdocs");
5 define("D_WEB_PATH", "/store/");
6 define("D_CART", D_WEB_PATH . "cart/");
7 define("D_CARTIMAGES", D_CART . "images/");
8 define("D_CUSTOMER", D_WEB_PATH . "customer/");
9 define("D_AUTH", D_WEB_PATH . "auth/");
10 define("D_ORDER", D_WEB_PATH . "order/");
11 define("D_SEARCH", D_WEB_PATH . "search/");
12 define("D_TEMPLATES", D_INSTALL_PATH . D_WEB_PATH .
    "templates/");
13 define("S_MAIN", D_WEB_PATH . "index.php");
14 define("S_ADDTOCART", D_CART . "addtocart.php");
15 define("S_EMPTYCART", D_CART . "emptycart.php");
16 define("S_SHOWCART", D_CART . "showcart.php");
17 define("S_UPDATECART", D_CART . "updatecart.php");
18 define("S_ORDER_1", D_ORDER . "order-step1.php");
19 define("S_ORDER_2", D_ORDER . "order-step2.php");
20 define("S_ORDER_3", D_ORDER . "order-step3.php");
21 define("S_ORDER_4", D_ORDER . "order-step4.php");
22 define("S_ORDERRECEIPT", D_ORDER . "receipt.php");
23 define("S_SEARCH", D_SEARCH . "search.php");
24 define("S_SEARCHFORM", D_SEARCH . "searchform.php");
25 define("S_DETAILS", D_CUSTOMER . "details.php");
26 define("S_VALIDATE", D_CUSTOMER . "validate.php");
27 define("S_CUSTRECEIPT", D_CUSTOMER . "receipt.php");
28 define("S_LOGOUT", D_AUTH . "logout.php");
29 define("S_LOGIN", D_AUTH . "login.php");
30 define("S_LOGINCHECK", D_AUTH . "logincheck.php");
31 define("S_PASSWORD", D_AUTH . "password.php");
32 define("S_CHANGEPASSWORD", D_AUTH . "changepassword.php");
33 define("S_PASSWORDRECEIPT", D_AUTH . "receipt.php");
34 define("T_SKELETON", "winestore.tpl");
35 define("T_HOME", "index.tpl");
36 define("T_SHOWCART", "showcart.tpl");
37 define("T_DETAILS", "details.tpl");
38 define("T_CUSTRECEIPT", "custreceipt.tpl");
39 define("T_LOGIN", "login.tpl");
40 define("T_PASSWORD", "password.tpl");
41 define("T_PASSWORDRECEIPT", "passwordreceipt.tpl");
42 define("T_EMAIL", "email.tpl");
43 define("T_ORDERRECEIPT", "orderreceipt.tpl");
44 define("T_SEARCH", "search.tpl");
45 define("T_SOURCE", "source.tpl");

```

```

46 define("I_CART_OFF", D_CARTIMAGES . "cart_off.jpg");
47 define("I_CART_ON", D_CARTIMAGES . "cart_on.jpg");
48 define("B_EMPTY_CART", 1);
49 define("B_SHOW_CART", 2);
50 define("B_UPDATE_CART", 4);
51 define("B_PURCHASE", 8);
52 define("B_SEARCH", 16);
53 define("B_HOME", 32);
54 define("B_DETAILS", 64);
55 define("B_LOGINLOGOUT", 128);
56 define("B_PASSWORD", 256);
57 define("B_ALL", 511);
58 define("NO_CART", 1);
59 define("SHOW_ALL", 2);
60 define("SEARCH_ROWS", 12);
61 define("ERROR_FILE", "/tmp/php_error_log");
62 define("FILE_ERRORS", false);
63 define("SCREEN_ERRORS", true);
64 $dsn = "mysqli://{ $username }:";
65     { $password } @ { $hostname } / { $databasename }";
66
67 function pearclean($array, $index, $maxlength, $connection)
68 {
69     if (isset($array["{ $index }"]))
70     {
71         $input = trim(substr($array["{ $index }"], 0, $maxlength));
72         $link = mysqli_connect("localhost", "root",
73                               "19950608", "winestore");
74         $input = mysqli_real_escape_string($link, $input);
75         return ($input);
76     }
77     return NULL;
78 }
79
80 function getCust_id($user_name, $connection = null)
81 {
82     global $dsn;
83     if (!isset($connection))
84     {
85         $connection = DB::connect($dsn, false);
86         if (DB::isError($connection))
87             trigger_error($connection->getMessage(), E_USER_ERROR);
88     }
89

```

```

90     $query = "SELECT cust_id FROM users WHERE
91               user_name = '{$user_name}'";
92
93     $result = $connection->query($query);
94     if (DB::isError($result))
95         trigger_error($result->getMessage(), E_USER_ERROR);
96
97     $row = $result->fetchRow(DB_FETCHMODE_ASSOC);
98
99     return($row["cust_id"]);
100 }
101
102 function showWine($wineId, $connection = null)
103 {
104     global $dsn;
105
106     $wineQuery = "SELECT year, winery_name, wine_name
107                  FROM winery, wine
108                  WHERE wine.winery_id = winery.winery_id
109                  AND wine.wine_id = {$wineId}";
110
111     if (!isset($connection))
112     {
113         $connection = DB::connect($dsn, false);
114         if (DB::isError($connection))
115             trigger_error($connection->getMessage(), E_USER_ERROR);
116     }
117
118     $result = $connection->query($wineQuery);
119
120     if (DB::isError($result))
121         trigger_error($result->getMessage(), E_USER_ERROR);
122
123     $row = $result->fetchRow(DB_FETCHMODE_ASSOC);
124     $output = "{$row["year"]}
125               {$row["winery_name"]} {$row["wine_name"]}";
126     $output .= showVarieties($connection, $wineId);
127     return $output;
128 }
129
130 function showVarieties($connection, $wineID)
131 {
132     $query = "SELECT gv.variety
133              FROM grape_variety gv, wine_variety wv, wine w

```

```

134         WHERE w.wine_id = wv.wine_id
135         AND wv.variety_id = gv.variety_id
136         AND w.wine_id = {$wineID}
137         ORDER BY wv.id";
138
139     $result = $connection->query($query);
140
141     if (DB::isError($result))
142         trigger_error($result->getMessage(), E_USER_ERROR);
143
144     $varieties = "";
145     while ($row = $result->fetchRow(DB_FETCHMODE_ASSOC))
146         $varieties .= " {$row["variety"]}";
147
148     return $varieties;
149 }
150
151 function showPricing($connection, $wineID)
152 {
153     $query = "SELECT min(cost) FROM inventory
154             WHERE wine_id = {$wineID}";
155
156     $result = $connection->query($query);
157
158     if (DB::isError($result))
159         trigger_error($result->getMessage(), E_USER_ERROR);
160
161     $row = $result->fetchRow(DB_FETCHMODE_ASSOC);
162
163     $price = $row["min(cost)"];
164
165     return $price;
166 }
167
168 function showCountry($country_id, $connection)
169 {
170     $query = "SELECT country FROM countries WHERE
171             country_id = {$country_id}";
172
173     $result = $connection->query($query);
174
175     if (DB::isError($result))
176         trigger_error($result->getMessage(), E_USER_ERROR);
177

```

```

178 $countryRow = $result->fetchRow(DB_FETCHMODE_ASSOC);
179
180 return($countryRow["country"]);
181 }
182
183 function showTitle($title_id, $connection)
184 {
185     $query = "SELECT title FROM titles WHERE
186             title_id = {$title_id}";
187
188     $result = $connection->query($query);
189
190     if (DB::isError($result))
191         trigger_error($result->getMessage(), E_USER_ERROR);
192
193     $titleRow = $result->fetchRow(DB_FETCHMODE_ASSOC);
194
195     return($titleRow["title"]);
196 }
197
198 ?>

```

输入验证:

```

1 <?php
2 function checkMandatory
3 ($field, $errorString, $errors, $formVars)
4 {
5     if (!isset($_SESSION["{$formVars}"][${"$field"}]) ||
6         empty($_SESSION["{$formVars}"][${"$field"}]))
7     {
8         $_SESSION["{$errors}"][${"$field"}] =
9             "The {$errorString} field cannot be blank.";
10         return false;
11     }
12     return true;
13 }
14
15 function checkMinAndMaxLength
16 ($field, $minlength, $maxlength,
17 $errorString, $errors, $formVars)
18 {
19     if (isset($_SESSION["{$formVars}"][${"$field"}]) &&
20 (strlen($_SESSION["{$formVars}"][${"$field"}])<$minlength||
21 strlen($_SESSION["{$formVars}"][${"$field"}])>$maxlength))

```

```

22     {
23         $_SESSION["{$errors}"][${"$field"}] =
24 "The {$errorString} field must be greater than or equal to".
25 "{$minlength} and less than or equal to {$maxlength} ".
26 "characters in length.";
27         return false;
28     }
29     return true;
30 }
31
32 function checkZipcode
33 ($field, $errorString, $errors, $formVars)
34 {
35     if (isset($_SESSION["{$formVars}"][${"$field"}]) &&
36         !preg_match("/^([0-9]{5,6})$/",
37             $_SESSION["{$formVars}"][${"$field"}]))
38     {
39         $_SESSION["{$errors}"][${"$field"}] =
40             "The zipcode must be 5 or 6 digits in length";
41         return false;
42     }
43     return true;
44 }
45
46 function checkPhone
47 ($field, $errorString, $errors, $formVars)
48 {
49     $validPhoneExpr =
50         "/^([0-9]{2,3}[ ])?[0-9]{4}[ ]?[0-9]{4}$/";
51
52     if (isset($_SESSION["{$formVars}"][${"$field"}]) &&
53         !preg_match($validPhoneExpr,
54             $_SESSION["{$formVars}"][${"$field"}]))
55     {
56         $_SESSION["{$errors}"][${"$field"}] =
57             "The {$field} field must be 8 digits in length, " .
58             "with an optional 2 or 3 digit area code";
59         return false;
60     }
61     return true;
62 }
63
64 function emailCheck
65 ($field, $errorString, $errors, $formVars)

```

```

66 {
67     $validEmailExpr =
68         "^[0-9a-z~!#$%&_-](.[0-9a-z~!#$%&_-])*".
69         "@[0-9a-z~!#$%&_-](.[0-9a-z~!#$%&_-])*$";
70
71     if (!eregi($validEmailExpr,
72         $_SESSION["{$formVars}"][${"$field"}]))
73     {
74         $_SESSION["{$errors}"][${"$field"}] =
75             "The email must be in the name@domain format.";
76         return false;
77     }
78
79     if (function_exists("getmxrr") &&
80         function_exists("gethostbyname"))
81     {
82         $maildomain =
83             substr(strstr($_SESSION["{$formVars}"]
84                 [{"$field"}], '@'), 1);
85
86         if (!(getmxrr($maildomain, $temp) ||
87             gethostbyname($maildomain) != $maildomain))
88         {
89             $_SESSION["{$errors}"][${"$field"}] =
90                 "The email domain does not exist.";
91             return false;
92         }
93     }
94
95     return true;
96 }
97
98 function checkCard($field, $errors, $formVars)
99 {
100     if (!preg_match("/^[0-9 ]*$/",
101         $_SESSION["{$formVars}"][${"$field"}]))
102     {
103         $_SESSION["{$errors}"][${"$field"}] =
104             "Card number must contain only digits and spaces.";
105         return false;
106     }
107
108     $_SESSION["{$formVars}"][${"$field"}] = preg_replace
109         ('[ ]', '', $_SESSION["{$formVars}"][${"$field"}]);

```



```

110
111 $firstFour = intval(substr(
112     $_SESSION["{$formVars}"][${"$field"}], 0, 4));
113 $type = "";
114 $length = 0;
115
116 if ($firstFour >= 8000 && $firstFour <= 8999)
117 {
118     $type = "SurchargeCard";
119     $length = 16;
120 }
121
122 if (empty($type))
123 {
124     $_SESSION["{$errors}"][${"$field"}] =
125         "Please check your card details.";
126     return false;
127 }
128
129 if (strlen($_SESSION["{$formVars}"]
130     [{"$field"}]) != $length)
131 {
132     $_SESSION["{$errors}"][${"$field"}] =
133         "Card number must contain {$length} digits.";
134     return false;
135 }
136
137 $check = 0;
138
139 for($x=$length-1;$x>=0;$x-=2)
140     $check += intval(substr(
141         $_SESSION["{$formVars}"][${"$field"}], $x, 1));
142
143 for($x=$length-2;$x>=0;$x-=2)
144 {
145     $double = intval(substr($_SESSION["{$formVars}"]
146         [{"$field"}], $x, 1)) * 2;
147     if ($double >= 10)
148         $check += $double - 9;
149     else
150         $check += $double;
151 }
152 ?>

```