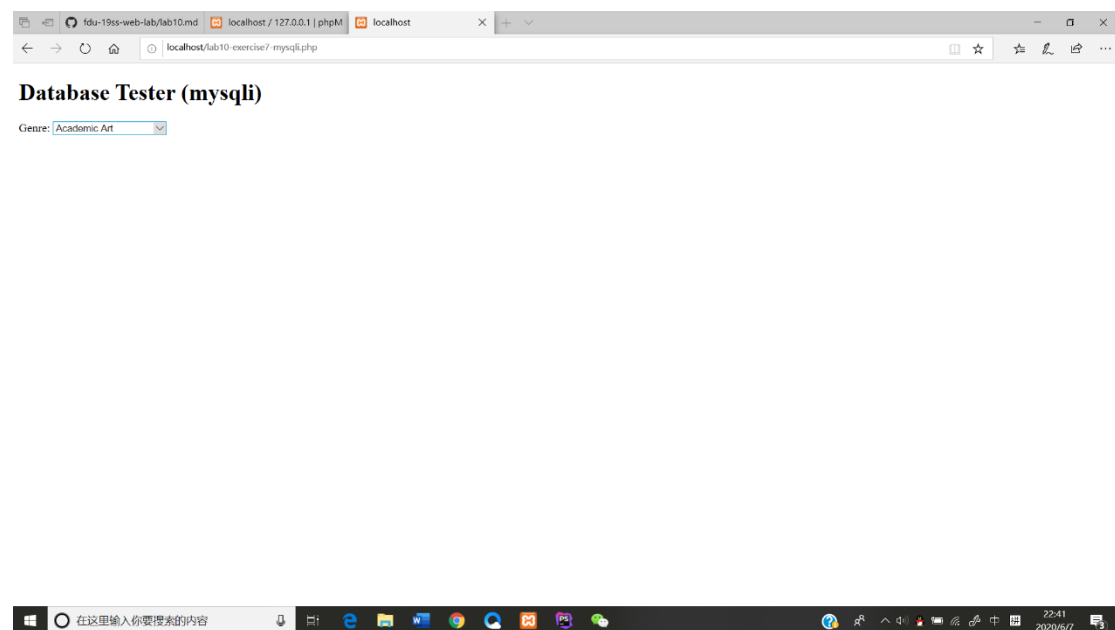
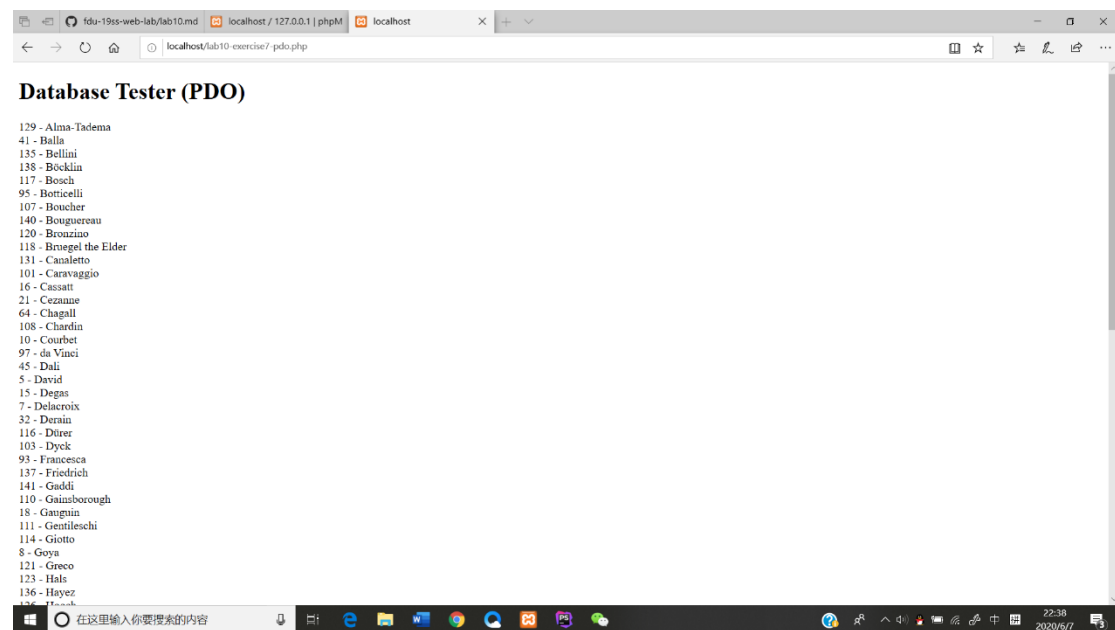


练习 7



Config.php 定义了它的用户名、密码。

lab10-exercise7-pdo.php 是将艺术家按 last name 的字母表顺序排列

```
(echo $row['ArtistID'] . " - " . $row['LastName'])
```

用 config 中定义的用户名、密码等创建了一个 pdo 类的对象。

\$sql 是筛选数据库，筛选 artist 中的所有人并按 last name 排序。

\$result 是 pdo 根据 \$sql 筛选后返回的结果数组，用循环一行一行打印出。

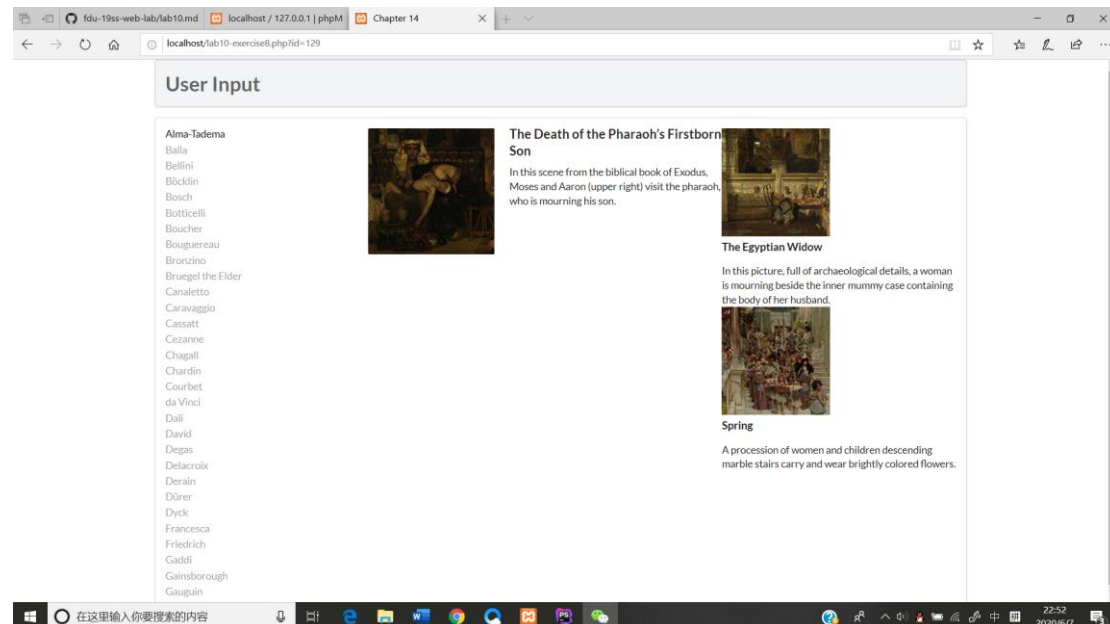
lab10-exercise7-mysqli.php 通过 mysqli 与数据库连接，通过 mysqli_connect 函数创建了连接 \$connection。

\$sql 筛选 Genres 中的全部按 GenreName 排序。

\$result 是通过上述的连接和筛选方式，利用 mysqli_query 函数得到的结果们，通过循环将他们依次加入 select 中的 option。做完后将 result 的内存清空，并断开连接。

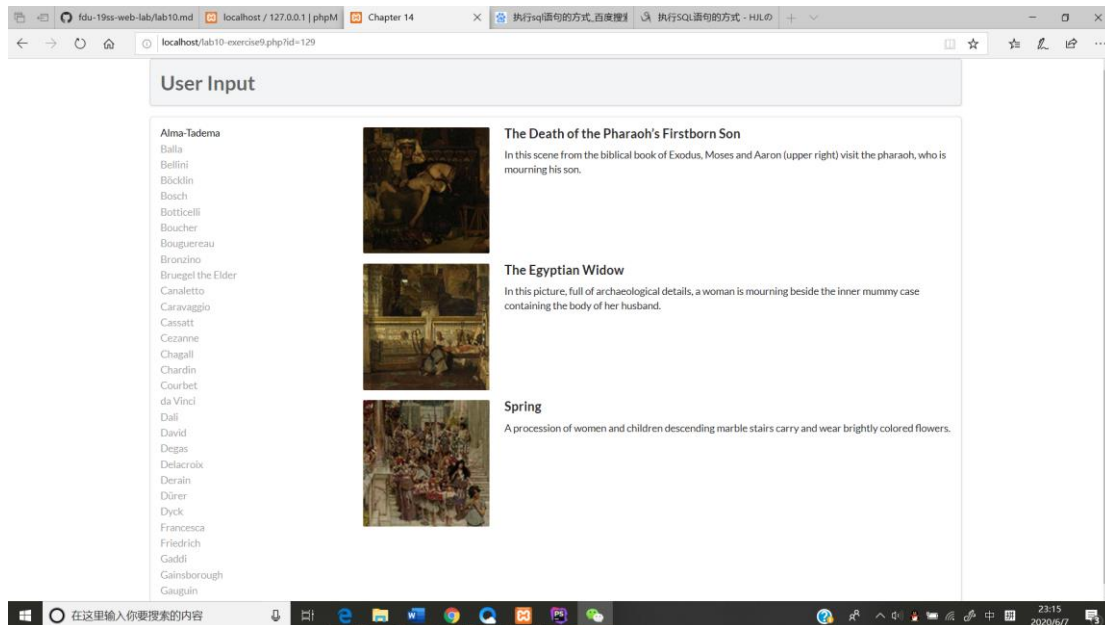
两个方法的区别就是一个是创建 pdo 类的实例，通过对它操作来操作数据库，另一个是创建一个与数据库的连接，操作数据库需要用到这个连接。

练习 8



练习 8 中用了练习 7 中第一个 pdo 类的方式操作数据库，outputArtists()函数通过\$筛选到 Artist 中、按 lastName 进行排序结果的前 30 个，通过循环将每个艺术家的名字设置为一个 <a>，放在下面的 html 部分中的<div class="ui link list"></div>中。而 outputPaintings()函数是用来在点击左边艺术家名字后显示对应的图片的。它通过\$sql 筛选所点击的艺术家 id 为条件索引其名下的所有图片，得到的结果，并按行循环依次输出图片。输出图片的方法为调用 outputSinglePainting(\$row)函数，通过 echo 函数在页面上输出一块内容，具体结构是 class 为 item 的 div 中包含了图片 div（通过\$row 的属性得到位置和名称），包含名字和描述的内容 div。这一整块在点击左边链接后显示的图片位于<div class="ui items"></div>中，动态生成。

练习 9



执行 sql 语句的方法有 4 种，E9 中用的是 prepare()

Exec()函数: `$res=$pdo->exec($query)`, 返回值是受影响的行数; 通常进行的是 insert、delete 等操作。

Query()函数: `$res=$pdo->query($sql);` \$sql 是筛选条件 (如 `select * from Artists order by LastName`), 返回值是搜索的结果。Query()可以实现所有 exec()的功能。

预处理语句: 包括 prepare()和 execute()两步。prepare()方法负责查询准备, execute()方法负责执行查询, 此外 bindParam()方法可以绑定参数给 execute()方法

```
$res=$pdo->prepare($query)
```

```
$res->execute();
```

而使用预处理语句的好处在于:

- 1、只解析查询一次, 然后可以使用相同或不同的参数执行多次。Prepare()函数执行之后, 数据库就会分析、编译并优化它要执行查询的计划。如果是结构相同而参数不同的复杂查询, 那多次解析就会很占时间、执行速度较慢, 而通过使用一个预处理语句就可以避免重复分析、编译、优化的环节。
- 2、传给预处理语句的参数不需要使用引号。
- 3、减少 sql 注入的机会。