考生姓名 准考证号

**2019年下半年 Web前端开发中级 实操考试**

（考试时间14:00-16:30 共150分钟）

1. 本试卷共4道题，满分100分。

2. 请在指定位置或开发环境下作答。

# 试题一（30分）

阅读下列说明、效果图和HTML代码，进行静态网页开发，填写（1）至（15）代码。

**【说明】**

这是“Web技术社区”网站，该网站致力于推广和分享各种前端开发技术，现在我们需要编写该网站首页。

项目名称为web\_skill，采用Bootstrap框架，包含首页index.html、css文件夹、js文件夹、img文件夹，其中，css文件夹包含index.css文件和bootstrap.min.css文件；js文件夹包含jquery-3.2.1.min.js文件和bootstrap.min.js文件；img文件夹包含bootstrap.png、php.png、mysql.png和laravel.png图片。

首页（index.html）使用Bootstrap响应式布局，PC端和移动端能够自适应显示，内容分为三部分：

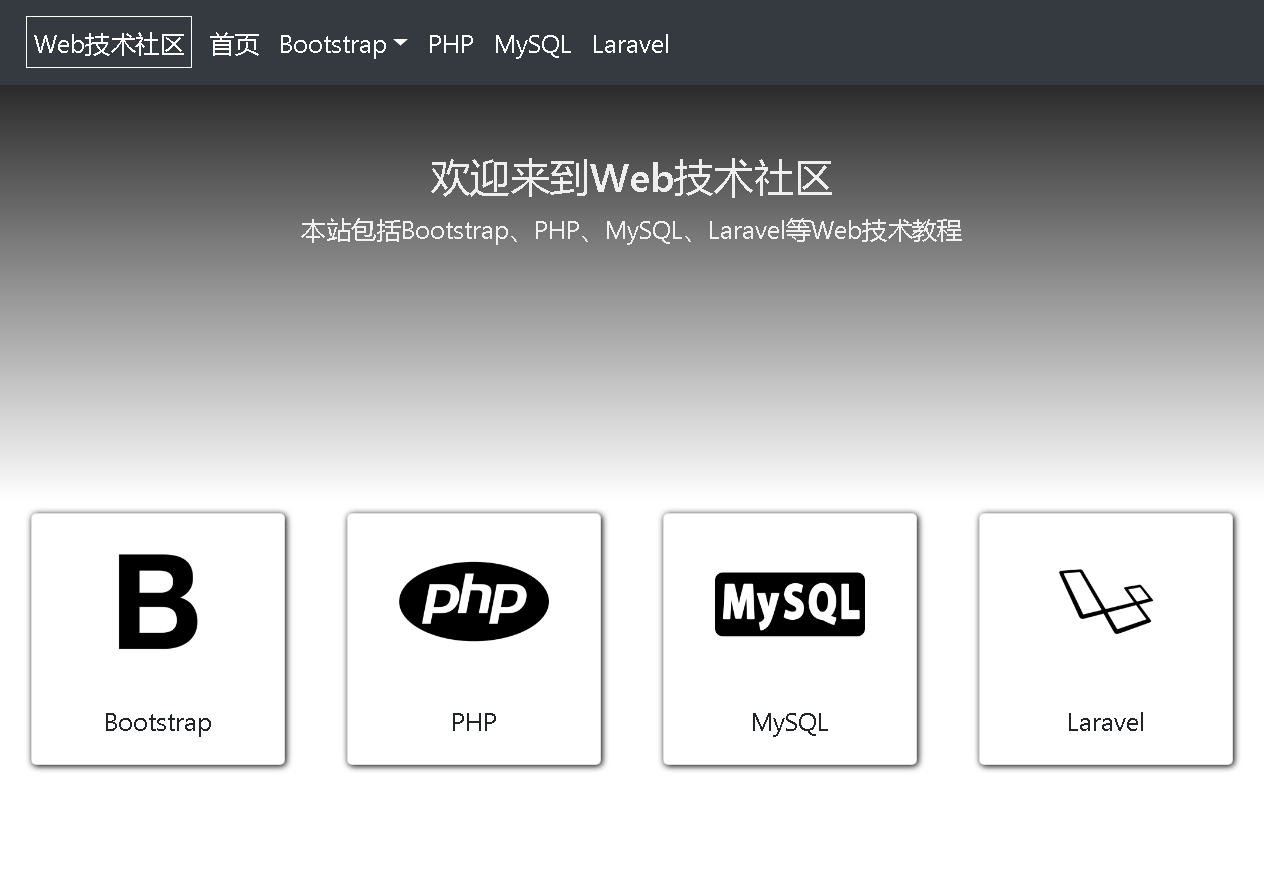
一是【页头】，包括网页标题和导航栏，网页标题“Web技术社区”使用盒模型，导航栏使用了下拉插件，显示时为菜单，在移动端显示为折叠导航栏；

二是【网站介绍】，背景从上到下由黑到白渐变，鼠标悬停时“欢迎来到Web技术社区”标题字号变大为根元素大小的2.25倍；

三是【技术介绍】，采用栅格系统布局，以图片和标题的形式展示四项【技术】，屏幕宽度≥992px时显示四列，屏幕宽度<576px时显示两列。

**【效果图】**

（1）index.html在PC端效果如图1-1所示。



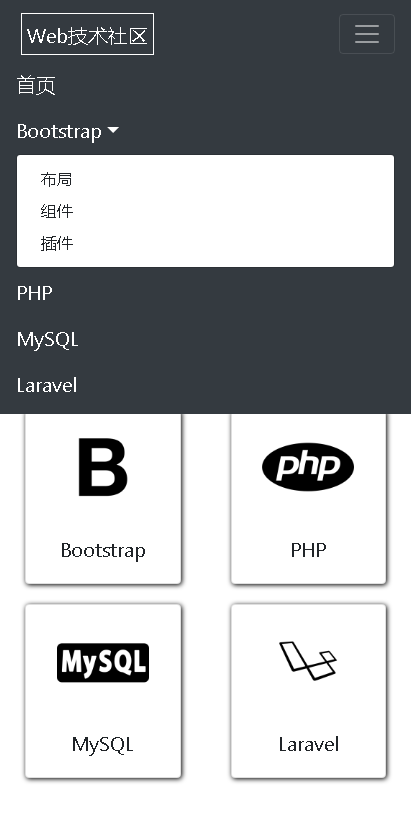
**图1-1**

（2）导航栏“Bootstrap”项使用下拉插件，点击显示下拉菜单，内容为“布局”、“组件”、“插件”，如图1-2所示。



**图1-2**

（3）index.html在移动端效果如图1-3所示，导航栏和“Bootstrap”下拉插件如图1-4所示。

**图1-3 图1-4**

**【代码：首页index.html】**

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<!-- 导入css文件夹中bootstrap.min.css基础样式文件 -->

<link rel="stylesheet" type="text/css" href="./css/bootstrap.min.css" />

<link rel="stylesheet" type="text/css" href="css/index.css" />

<title>Web技术社区</title>

</head>

<body>

<!-- 页头 -->

<!-- 响应式导航栏 -->

<nav class="navbar （2） navbar-dark bg-dark">

<a class="brand" href="#">Web技术社区</a>

<!-- 折叠导航栏 -->

<button class="（3）" type="button" data-toggle="collapse" data-target="（4）">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navToggler">

<ul class="navbar-nav mr-auto">

<li class="nav-item active">

<a class="nav-link" href="#">首页</a>

</li>

<!-- 下拉菜单 -->

<li class="nav-item （5）">

<a class="nav-link dropdown-toggle" href="#" data-toggle="dropdown">Bootstrap</a>

<div class="dropdown-menu">

<a class="dropdown-item" href="#">布局</a>

<a class="dropdown-item" href="#">组件</a>

<a class="dropdown-item" href="#">插件</a>

</div>

</li>

<li class="nav-item">

<a class="nav-link" href="#">PHP</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#">MySQL</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#">Laravel</a>

</li>

</ul>

</div>

</nav>

<div id="introduction">

<div id="content">

<h2>欢迎来到Web技术社区</h2>

<p>本站包括Bootstrap、PHP、MySQL、Laravel等Web技术教程</p>

</div>

</div>

<!-- 栅格系统 -->

<!-- 最外层容器100%宽度-->

<div class="（6）">

<!-- 行 -->

<div class="row">

<!--第一张图片和标题-->

<!-- 屏幕宽度≧992px时显示四列，屏幕宽度<576px时显示两列 -->

<div class="（7）">

<div class="card">

<img class="card-img-top" src="img/bootstrap.png">

<div class="card-body">

<p class="card-text">Bootstrap</p>

</div>

</div>

</div>

<!--第二张图片和标题-->

<!-- 屏幕宽度≧992px时显示四列，屏幕宽度<576px时显示两列 -->

<div class="（7）">

<div class="card">

<img class="card-img-top" src="img/php.png">

<div class="card-body">

<p class="card-text">PHP</p>

</div>

</div>

</div>

<!--第三张图片和标题-->

<!-- 屏幕宽度≧992px时显示四列，屏幕宽度<576px时显示两列 -->

<div class="（7）">

<div class="card">

<img class="card-img-top" src="img/mysql.png">

<div class="card-body">

<p class="card-text">MySQL</p>

</div>

</div>

</div>

<!--第四张图片和标题-->

<!-- 屏幕宽度≧992px时显示四列，屏幕宽度<576px时显示两列 -->

<div class="（7）">

<div class="card">

<img class="card-img-top" src="img/laravel.png">

<div class="card-body">

<p class="card-text">Laravel</p>

</div>

</div>

</div>

</div>

</div>

<!-- 导入js文件夹里的jquery-3.2.1.min.js文件和bootstrap.min.js文件 -->

<script src="js/jquery-3.2.1.min.js"></script>

<script src="（8）"></script>

</body>

</html>

**【代码：CSS文件index.css】**

body { font-size: 20px;}

.navbar-dark .navbar-nav .nav-link{

color: white;

}

.navbar {

position: absolute;

top:0px;

width: 100%;

z-index: 99;

}

.brand {

color: white;

/\* 盒模型 \*/

（9）;/\* 边框1px，白色，实线 \*/

padding: 5px;/\* 内边距5px \*/

（10）;/\* 外边距5px \*/

}

.brand:hover {

color: white;

text-decoration: none;

}

#introduction {

background: （11）(to （12）,black,white);/\* 背景从上到下颜色渐变 \*/

width:100%;

height:400px;

color: rgba(255,255,255,0.9);/\* 字体颜色 \*/

}

#content {

position: absolute;

top: 16%;

left: 16%;

width: 68%;

text-align: center;

}

#content h2:hover {

font-size: 2.25（13）;/\* 设置字体大小为根元素大小的2.25倍 \*/

（14）: font-size 1s;/\* 使用过渡，1s内标题字号变大 \*/

}

.card{

margin: 10px;

（15）: 1px 1px 5px black;/\* 盒阴影 \*/

text-align: center;

align-items: center;

}

.card-img-top{

margin-top: 10px;

width: 60%;

}

.card-text { text-align: center;}

**【问题】**（30分，每空2分）

进行静态网页开发，补全代码，在（1）至（15）处填入正确的内容。

# 试题二（30分）

阅读下列说明、效果图、MySQL数据库操作和代码，进行动态网页开发，填写（1）至（20）代码。

**【说明】**

该程序为一个用户管理模块，使用PHP编程，采用MySQL数据库和mysqli编程。项目名称为project，包括登录login.php、处理登录请求check.php、用户管理主页index.php、用户管理数据库操作user.php，以及创建数据库脚本db.sql和初始化数据脚本init.sql。

**【效果图】**

**（1）登录login.php**

在登录页面进行登录，并进行登录请求处理，用户登录页面如下2-1所示。

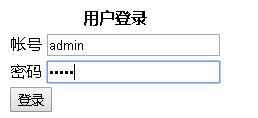


图2-1

**（2）用户管理主页index.php**

登录成功时，用Session保存用户账号，并跳转到“用户管理主页”，主页布局如下：

在头部显示“欢迎+用户账号”信息，在中间区域用表格显示用户列表信息，每一个用户信息显示为一行，内容包含“序号、帐号、密码、手机号”。用户管理主页如2-2所示。



图2-2

**【MySQL数据库操作：创建数据库脚本db.sql】**

系统使用MySQL数据库，数据库名为user\_center，表名为用户表users。user表包含序号、用户帐号、密码、手机号字段，其中，序号为自增字段和主键。补充下列（1）和（2）。

DROP DATABASE IF EXISTS user\_center;

CREATE DATABASE user\_center;

USE user\_center;

CREATE TABLE `users`

(

`id` int (11) NOT NULL （1） COMMENT '序号',

`account` varchar (16) NOT NULL COMMENT '用户帐号,16位字符',

`password` char (64) NOT NULL COMMENT '密码',

`telphone` varchar (32) DEFAULT '' COMMENT '手机号',

（2） (`id`)

)ENGINE= InnoDB DEFAULT CHARSET= utf8;

**【MySQL数据库操作：初始化数据脚本init.sql】**

在MySQL数据库中，向user\_center数据库中users表插入用户信息数据，以便在用户管理主页index.php上显示用户列表信息，插入数据脚本如下。

USE user\_center;

insert into users(account, password, telphone) value('admin', 'admin', "16871054787");

insert into users(account, password, telphone) value('admin2', 'admin2', "13871456841");

insert into users(account, password, telphone) value('admin3', 'admin3', "80041224771");

insert into users(account, password, telphone) value('admin4', 'admin4', "15874581652");

insert into users(account, password, telphone) value('admin5', 'admin5', "25417522145");

insert into users(account, password, telphone) value('admin6', 'admin6', "40014125155");

insert into users(account, password, telphone) value('admin7', 'admin7', "15847455514");

insert into users(account, password, telphone) value('admin8', 'admin8', "16854711223");

**【代码：用户管理数据库操作user.php】**

在user.php文件中，定义User类，进行数据库操作。在进行数据库编程中，应用mysqli。

（1）在类中定义静态属性，设置数据库连接基本参数。

（2）在类中定义静态的方法conn()，连接mysql数据库。

（3）定义checkLogin()方法，验证用户登录。

（4）定义queryAll()方法，查询全部用户信息，并返回$users二维数组。

根据要求填写（3）至（10）。

<?php

（3）User {

static $host = "127.0.0.1";

static $dbname = "user\_center";

static $user = "root";

static $pass = "123456";

static function conn() {

$db = （4）(self::$host, self::$user, self::$pass, self::$dbname);

if(mysqli\_connect\_errno()) {

return null;

}

return $db;

}

public （5）checkLogin($account, $password) {

$db = null;

$stmt = null;

try {

$db = self::conn();

if($db) {

$sql = "（6）";

$stmt = $db->prepare($sql);

$stmt->bind\_param("（7）", $account, $password);

$stmt->bind\_result($id);

$stmt->（8）();

if($stmt->fetch()) {

return true;

}

}

} catch(Exception $e) {

echo "Message:" . $e->getMessage();

} finally {

if($stmt != null) {

$stmt->free\_result();

}

if($db != null) {

$db->close();

}

}

return false;

}

public function queryAll() {

$db = null;

$stmt = null;

$users = array();

try {

$db = self::conn();

$sql = "select id, account, password, telphone from users";

$stmt = $db->prepare($sql);

（9）

$stmt->execute();

while($stmt->fetch()) {

（10）($users, array(

"id" => $id,

"account" => $account,

"password" => $password,

"telphone" => $telphone

));

}

} catch(Exception $e) {

echo "Message:" . $e->getMessage();

} finally {

if($stmt != null) {

$stmt->free\_result();

}

if($db != null) {

$db->close();

}

}

return $users;

}

}

**【代码：登录login.php】**

点击“登录”按钮，将表单以POST方式提交，处理该请求的文件为check.php，填写（11）。

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>登录</title>

</head>

<body>

<form action="（11）" method="post">

<table>

<tr><th>用户登录</th></tr>

<tr>

<td>

<label>帐号</label>

<input name="account" type="text" value="" />

</td>

</tr>

<tr>

<td>

<label>密码</label>

<input name="password" type="password" value="" />

</td>

</tr>

<tr>

<td><input type="submit" value="登录" /></td>

</tr>

</table>

</form>

</body>

</html>

**【代码：处理登录请求check.php】**

在check.php文件中，导入User类文件，并创建该类的对象$user，调用user.php中checkLogin()方法，对用户账号和密码进行验证。当登录成功时，使用Session保存用户账号，并跳转到用户管理主页index.php，当登录失败时跳转到登录页面login.php。根据要求填写（12）至（14）。

<?php

include\_once "user.php";

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$account = $\_POST["account"];

$password = $\_POST["password"];

$user= （12）;

if ($user->（13）($account,$password)) {

session\_start();

$\_SESSION["user"] = $account;

（14）("location:index.php");

} else {

（14）("location:login.php");

}

}

**【代码：用户管理主页index.php】**

在index.php文件中，将Session中保存用户账号显示到页面头部。导入User类文件，并创建该类的对象$user，调用user.php中queryAll()方法，查询全部用户信息。将查询返回用户信息，通过for循环，以表格形式进行显示，根据要求填写（15）至（20）。

<?php include\_once "user.php"; ?>

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

</head>

<body>

<header>

<div>

<?php

session\_start();

if (isset(（15）)) {

echo "欢迎 " . （15）;

} else {

echo "<a href='login.php'>未登录</a>";

}

?>

</div>

</header>

<article>

<?php

$user= new User();

$users = （16）;

?>

<table>

<tr><th>帐号</th> <th>密码</th><th>手机号</th></tr>

<?php （17） ?>

<tr>

<td>（18）</td>

<td><?php echo $u["password"];?></td>

<td><?php echo （19）;?></td>

</tr>

（20）

</table>

</article>

</body>

</html>

【**问题**】（30分，每空1.5分）

进行动态网页开发，补全代码，在（1）至（20）处填入正确的内容。

# 试题三（20分）

阅读下列说明、效果图和代码，进行动态网页开发，回答问题1至问题4。

**【说明】**

随着互联网发展，越来越多的人开始阅读电子书，本项目实现在网页上阅读JSON格式书籍，应用HTML、CSS、AJAX、JSON、PHP等技术。项目名称为book，包含主页index.html、index.css和loadJSON.php文件，其中，主页面index.html显示书籍内容；index.css为主页样式文件；loadJSON.php为主页提供数据，返回JSON格式书籍。

主页分为两个部分：左侧为按钮和“目录”，中间为章对应“内容”。点击“开始阅读”按钮，通过ajax发送请求到loadJSON.php，loadJSON.php返回JSON格式的书籍。在主页上通过ajax拿到数据后，使用JavaScript进行DOM操作，实现动态构建“目录”和“内容”。同时，为每一级“目录”绑定点击方法，当点击对应的“目录”中章时，对应“内容”部分会随之更新，如图3-1所示。

**【效果图】**

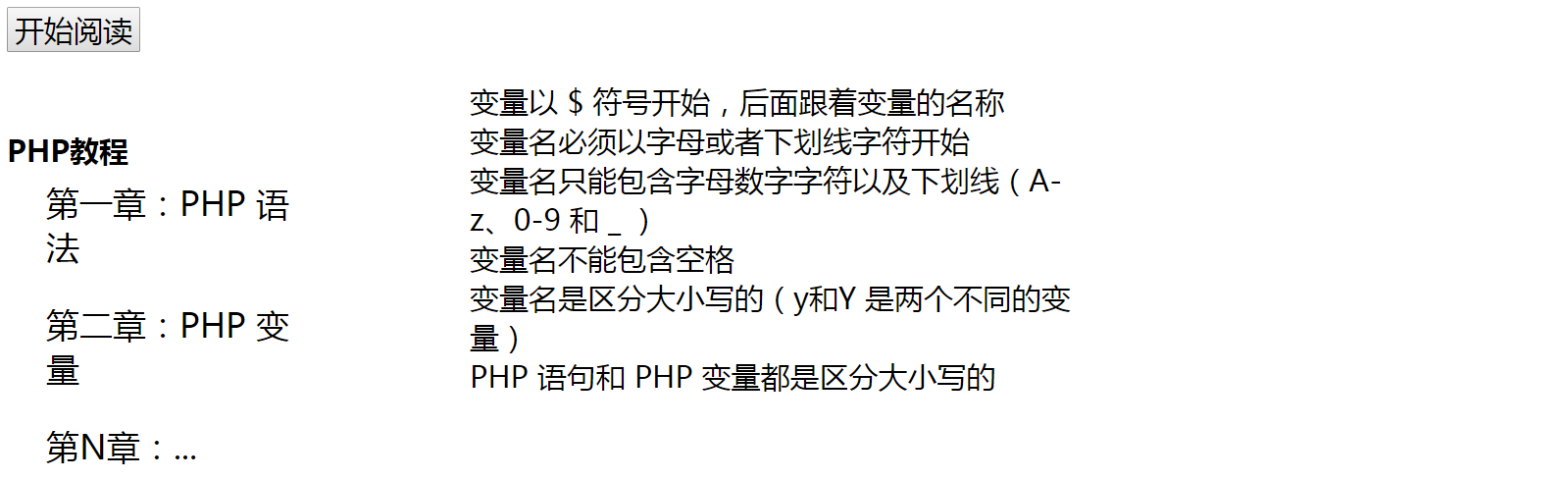


图3-1

**【代码：主页index.html】**

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>Web阅读器</title>

<link rel="stylesheet" type="text/css" href="index.css"/>

</head>

<body>

<div>

<!-- 相对路径 -->

<button onclick=" （1） ">开始阅读</button>

</div>

<header></header>

<aside class="list">

</aside>

<article class="content">

<p></p>

</article>

</body>

<script type="text/javascript">

var json = {};

/\* 获取json格式文章 \*/

function loadJSON(url){

var xmlhttp;

if (window.XMLHttpRequest) { // for IE7+, Firefox, Chrome, Opera, Safari

xmlhttp = new （2） ;

} else { // for IE6, IE5

xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");

}

xmlhttp. （3） ("GET", url, true);

xmlhttp. （4） ();

xmlhttp. （5） =function(){

if (xmlhttp. （6） == 4 && xmlhttp. （7） == 200) {

var result = （8） (xmlhttp.responseText);

json = result;

/\* 构建标题 \*/

var dom = document.getElementsByTagName("header")[0];

var h1 = document.createElement("h1");

h1.innerHTML = （9） ;

dom.appendChild(h1); /\* 添加节点 \*/

/\* 构建目录 \*/

var dom = document.getElementsByTagName("aside")[0];

for(var data in result){

if(data.search("subject") != -1){

var ul = document.createElement("ul");

ul.innerHTML = result[data];

ul.value = data;

/\* 构建内容 \*/

ul. （10） = function(){

document.getElementsByTagName("p")[0].innerHTML = json["content" + this.value.split("subject")[1]];

};

dom.appendChild(ul);

}

}

}

}

}

</script>

</html>

**【代码：loadJSON.php】**

<?php

$arr = array(

"title" => "PHP教程",

"subject1" => "第一章：PHP 语法",

"content1" => "PHP 脚本可以放在...",

"subject2" => "第二章：PHP 变量",

"content2" => "变量以 $ 符号开始，后面...",

"subject3" => "第N章：...",

"content3" => "未完待续",

);

echo json\_encode($arr);

?>

**【代码：index.css】**

\*{

font-size: 30px;

}

header{

position: absolute;

top:15%;

width:20%;

}

.list{

position: absolute;

top:20%;

width:20%;

}

.list ul{

font-size: 34px;

}

.list li{

font-size: 32px;

}

.content{

position: absolute;

left:30%;

width:40%;

}

**【问题】（20分，每空2分）**

进行动态网页开发，补全代码，在（1）至（10）处填入正确的内容。

# 试题四（20分）

阅读下列说明、效果图和代码，进行动态网页开发，回答问题1至问题4。

**【说明】**

该程序为一个问卷调查系统，使用PHP的Laravel框架编程，项目名称为survey，核心文件包括路由文件web.php、模板文件（问卷调查模板paper.blade.php和调查结果模板result.blade.php）、控制器文件SurveyController.php。页面需要的数据采用硬编码方式，存放在问卷调查控制类的静态属性中。

**【效果图】**

**（1）问卷调查页面：对应模板文件paper.blade.php**

用户访问问卷调查页面，勾选问题答案，点击“提交”按钮，如图4-1所示。

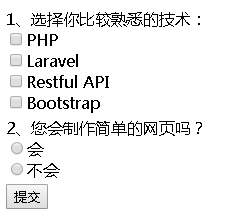


图4-1

**（2）调查结果页面：对应模板文件result.blade.php**

接收问卷调查页面提交数据，并进行显示，结果页面效果如图4-2所示。



图4-2

**【问题1】基于Laravel框架，回答下列问题，填写（1）至（3）。**

**1、包管理工具**

安装Laravel框架，需要使用Composer 工具：通过命令“（1） create-project laravel/laravel survey --prefer-dist”创建survey工程，创建完毕后，程序目录如图4-3所示。

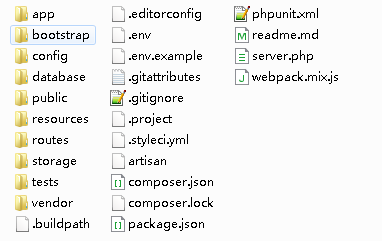


图4-3

**2、Laravel框架MVC结构**

Laravel遵循model-view-controller(MVC)模式，MVC模式的3个组件分别为模型（Model）、视图（View）和控制器（Controller）。项目核心文件如下表所示，请分析项目相关核心MVC文件，使用MVC关键词“Model、View或Controller”，填写（2）至（3）。

|  |  |  |
| --- | --- | --- |
| **文件** | **说明** | **MVC关键词** |
| routes/web.php | 应用路由文件 |  |
| app/Http/Controllers/SurveyController.php | 问卷调查控制类 | （2） |
| resources/views/paper.blade.php | 问卷调查模板 | （3） |
| resources/views/result.blade.php | 调查结果模板 |  |

**【问题2】分析项目路由，回答下列问题，填写（4）至（5）。**

在浏览器中输入网址，访问问卷调查页面，用户填写问卷后，点击“提交”按钮，通过POST方式提交问卷表单，在路由文件web.php中，路由配置如下：

<?php

// 访问问卷调查页面

Route::get("/survey", "SurveyController@paper");

// 提交问卷

Route::post("/finish", "SurveyController@finish");

项目发布在本机上，本机地址为localhost，根据web.php文件中定义的路由信息，当输入下列访问地址时，分析路由情况，选择相应A至F内容，填写（4）至（5）。

在浏览器中输入“http://localhost/survey”时，路由会（4）。

在浏览器中输入“http://localhost/finish”时，路由会（5）。

A. 显示paper.blade.php模板。

B. 抛出异常，未定义该方法。

C. 将字符串“SurveyController@finish”返回并显示到页面。

D. 将字符串“SurveyController@paper”返回并显示到页面。

E. 调用SurveryController类的paper()方法。

F. 调用SurveryController类的finish()方法。

**【问题3】分析项目控制器，回答下列问题，填写（6）至（7）。**

在SurveyController.php文件中，定义SurveyController类，该类继承基类控制器。在SurveyController类中定义静态属性$questions，以硬编码形式保存问卷调查数据，并定义paper()方法，处理访问问卷调查页面请求，使用paper.blade.php模板文件返回问卷调查页面，将$questions传给模板文件。分析控制类SurveyController，填写（6）至（7）。

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class SurveyController extends （6） {

// 问卷内容，静态

private static $questions = array(

"Q1" => array(

"issue" => "选择你比较熟悉的技术：",

"type" => "checkbox",

"options" => array(

"PHP",

"Laravel",

"Restful API",

"Bootstrap"

)

),

"Q2" => array(

"issue" => "您会制作简单的网页吗？",

"type" => "radio",

"options" => array(

"会",

"不会"

)

)

);

// 初始问卷调查数据，并跳转到paper页面

public function paper() {

return view("（7）", ["questions" => self::$questions]);

}

// 处理用户提交的调查问卷

public function finish(Request $request) {

$input = $request->all();

$result = array();

foreach(self::$questions as $key => $question) {

if(isset($input[$key])) {

array\_push($result, array(

"issue" => $question["issue"],

"answer" => join(",", $input[$key])

));

}

}

return view("result", ["result" => $result]);

}

}

**【问题4】分析模板文件，回答下列问题，填写（8）至（9）。**

**1、问卷调查模板paper.blade.php**

（1）在问卷调查模板文件paper.blade.php中，使用for循环显示问题，显示需要数据由SurveyController类中paper()返回时传递，用户填写完问卷数据后，使用POST方式提交。分析问卷调查模板和web.php中的路由信息，填写（8）空。

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

</head>

<body>

<table>

<form action="（8）" method="post">

（9）

@foreach($questions as $key=>$question)

<tr>

<td>

<label>{{ ($loop->index+1). "、".$question["issue"] }}</label><br />

@foreach($question["options"] as $option)

<input type="{{$question['type']}}" name="{{ $key.'[]'}}"

value="{{ $option }}">{{ $option }}</input>

<br />

@endforeach

</td>

</tr>

@endforeach

<tr>

<td><input type="submit" value="提交" /></td>

</tr>

</form>

</table>

</body>

</html>

（2）在上述代码中，当使用POST方式提交表单时，Laravel框架需要添加CSRF Token字段。对于下列选项描述，不正确的是（9）

A. <input type="hidden" name="\_token" value="{{csrf\_token()}}">

B. {{ csrf\_field() }}

C. {{ csrf@ }}

D. {!! csrf\_field() !!}

**2、调查结果模板result.blade.php**

在调查结果模板文件result.blade.php中，使用for循环显示用户填写的问题和答案，显示需要数据由SurveyController类中finish()返回时传递，分析finish()方法和调查结果模板文件，填写（10）。

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

</head>

<body>

<section>

@foreach($result as $q)

<label>{{ ($loop->index+1). "、".$q["issue"] }}</label><br />

<label>（10）</label><br />

@endforeach

</section>

</body>

</html>