****

Web应用与开发

实 验 报 告

|  |  |
| --- | --- |
| **学 院** |  |
| **专 业** |  |
| **班 级** |  |
| **学 号** |  |
| **学生姓名** |  |
| **教师姓名** |  |
| **完成日期** |  |
| **成 绩** |  |

**引用须知：**

**本文档已删除各类敏感信息，所有内容均可引用，但仅限学习交流。祝大家能顺利通过HWT的Web课程！**

**——F001**

**作业3 基于Laravel框架的用户注册和登录**

1. **实验目的和要求**
2. 实验目的：按照前后端混合的开发形式，基于Laravel框架，实现简单和安全的用户注册和登录。
3. **实验要求**：

（1）实现Email形式的注册功能和相应的登录功能，注册部分具备邮件激活功能（使用Laravel的邮件发送机制）；

（2）实现忘记密码时通过重置密码邮件设置新密码（使用Laravel的邮件发送机制）；

（3）存在数据库中的密码不能以明文形式存放，即建议在浏览器端发送请求前，调用js代码对用户的密码做md5加密（可以在网上找到这样的js代码文件），然后再传输加密后的密码，并分析这样做的好处（建议上网找资料进行总结，并在该实验报告里列出所参考的资料）；

（4）用Wireshark对注册信息和登录信息的传输过程做抓包，在“测试流程”中贴出带注册信息和登录信息的截图；

（5）寻找相关文献，分析仅用md5进行加密可能存在的问题（建议到中国知网寻找，并在实验报告里列出所参考的文献）；

（6）网页内容基于Laravel的Blade模板；

（7）在hosts文件里自定义一个域名，实现通过域名来访问；

（8）使用路由机制设计访问路径；

（9）提交作业时，把自己的邮箱隐私信息去除掉。

1. **实验内容**

根据**实验要求中的每条规定**，进行用户注册和登录实验。

提示：

1. 可以参考网络资料，但不能照搬照抄。

2. 作业报告的格式尽量整齐，字体要统一，内容要层次分明。

1. **程序设计思路**

按照（一）中的相关要求，进行相关的程序设计，采用PHP语言。应用Blade模板，设计相关的数据表。完成1、2要求后，不强制制作要求3，选择更加完备的加密方式进行。4中的抓包验证交给相关软件去做，6中的Blade做出相应的改动，7中的hosts已经在之前的课堂上完成。

1. **程序源代码**

**注意源代码要有详细的注释。同学们提交的每个程序都应该遵循Honor Code（诚实代码保证）的要求。**

**请大家特别注意一定要在每个程序首部的注释中加上以下保证：**

**# 我真诚地保证：**

**# 我自己独立地完成了整个程序从分析、设计到编码的所有工作。**

**# 如果在上述过程中，我遇到了什么困难而求教于人，那么，我将在程序实习报告中**

**# 详细地列举我所遇到的问题，以及别人给我的提示。**

**# 在此，我感谢 XXX, …, XXX对我的启发和帮助。下面的报告中，我还会具体地提到**

**# 他们在各个方法对我的帮助。**

**# 我的程序里中凡是引用到其他程序或文档之处，**

**# 例如教材、课堂笔记、网上的源代码以及其他参考书上的代码段,**

**# 我都已经在程序的注释里很清楚地注明了引用的出处。**

**# 我从未没抄袭过别人的程序，也没有盗用别人的程序，**

**# 不管是修改式的抄袭还是原封不动的抄袭。**

**# 我编写这个程序，从来没有想过要去破坏或妨碍其他计算机系统的正常运转。**

**# 你的名字**

1. **源代码（包含数据库表设计代码）**

注意：

(1) 可以在此处贴关键代码，也可以放在压缩包中跟本文档一起上传。

(2) 如果是压缩项目工程代码，建议把vendor目录排除掉，以减小压缩包大小。

此处不进行照搬照抄，用附件一起进行打包

1. **程序测试方法及测试结果记录（不能光截图，要有相应的文字说明）**
2. **测试方法**
3. 配置好相应的自定义域名；
4. 进入自定义域名的网站，进行注册操作；
5. 注册完成后查看邮箱是否被激活；
6. 注册完成后登出，重新登录，观察相应情况；
7. 使用命令行发送相应的邮件。
8. 查看邮箱观察发送情况
9. **测试流程**

1.注册与登录

首先进入[www.webdev.com](http://www.webdev.com)的网页，发现右上角存在登录和注册两个按钮。

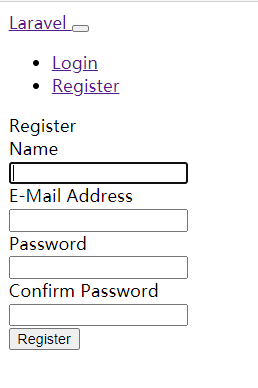
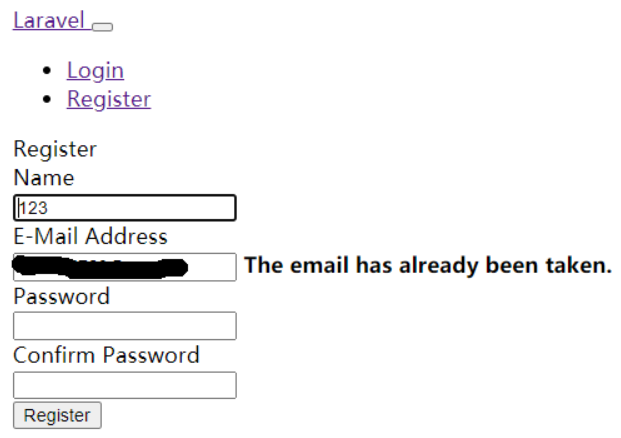
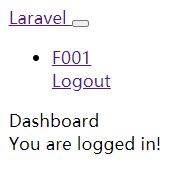


图1 图2 图3

注册部分图像如上（图一），输入相关信息后，在检测邮箱是否被重复使用，检测密码长度是否大于等于8位（图二），检测通过后自动激活（图三）。

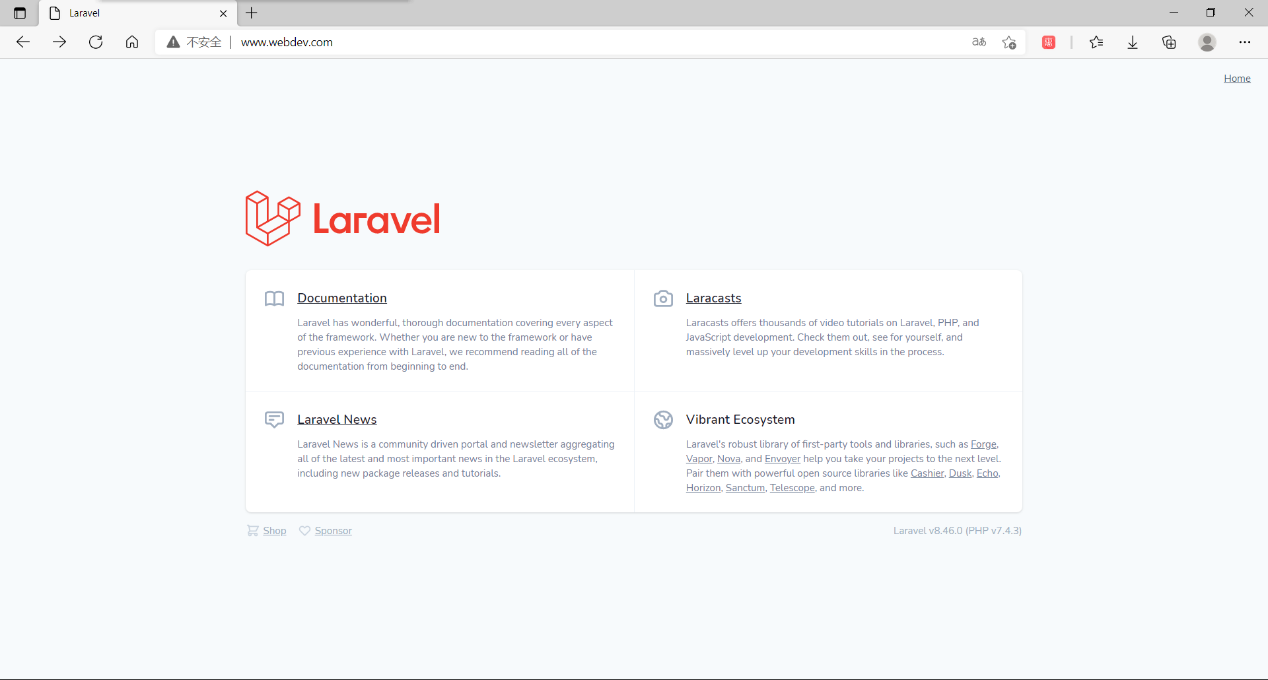


图4

此时进入到[www.webdev.com](http://www.webdev.com)中时右上角提示已经登陆成功，并且点击Home按钮后进入上述图三界面。

此时我们点击log out退出按钮，重新登陆，如下图所示

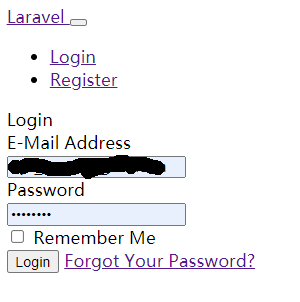
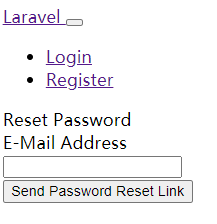
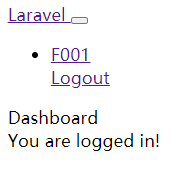


图5 图6 图7

通过图5图6可见输入相应的信息之后登录成功。

2.忘记密码

通过图5中的登录信息我们可以看到有忘记密码的选项，点击忘记密码的选项后可以看到图7所示的界面.

输入之前注册的邮箱后点击发送按钮，过一段时间会发现输入的信息被清空了，此时即可认为已经向邮箱中发送了重置密码的邮件。打开邮箱如图8所示，可见邮件发送成功。

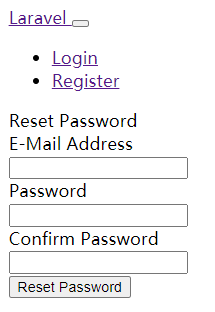
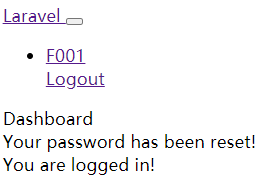


图8 图9 图10

点击进入后，可以看到如图9所示的界面，这里的Email地址应该是自动填写的，但是因为隐藏信息删去了。我们输入新的密码之后可以得到如图10所示的界面。

3.加密处理

题目的要求告诉我们不能在数据库中以明文的形式存储密码，但是在完成上述操作后我们查看PHP admin的数据库，发现如下图11所示，password密码一栏中所有的数据存储均不是明文，也就是说Laravel自身已经完成了相应的加密。

查阅相关文献后，了解到Laravel的加密其是使用AES-256和AES-128进行的相关加密，并且在底层通过MAC签名，使得一经加密就无法修改，因此其安全性得到了相应的保障。AES的相关加密标准相较于MD5而言更加安全，因为MD5已经由王小云院士所破解。

那为何老师并不推荐进行这样的操作呢？我们知道，Laravel是一种后端上使用的框架，因此尽管成功完成了加密，但是数据需要经历从前端（收集）->后端（加密）-前端（显示）的过程。因此从前端到后端的这一数据传输的过程中一旦被监听，相关的安全性难以保证。因此老师给出的建议在前端调用js代码进行MD5加密，再传输加密完成后的代码，可以减少服务器在MD5加密负担的同时，保障了密码的安全性。因为密码是直接在客户端就进行加密，不像php要传到服务器端才加密，这样传输过程的安全也得到进一步的加强。显然的，它的缺点是相关加密安全性并不是特别高。

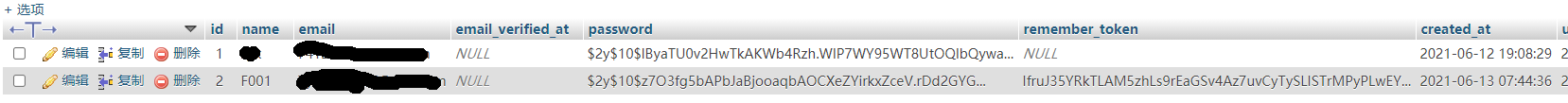


图11

4.抓包分析

在实验的过程中，尝试通过wireshark对登录信息和注册信息进行相应的抓包操作，但是由于相关网络协议调用飞速，导致在几秒的时间内出现了大量的报文，通过过滤器的筛选依次通过DNS和HTTP协议进行相应的分析。但是也仍然因为相关数据存在大量的重复操作难以实现详细的分析。

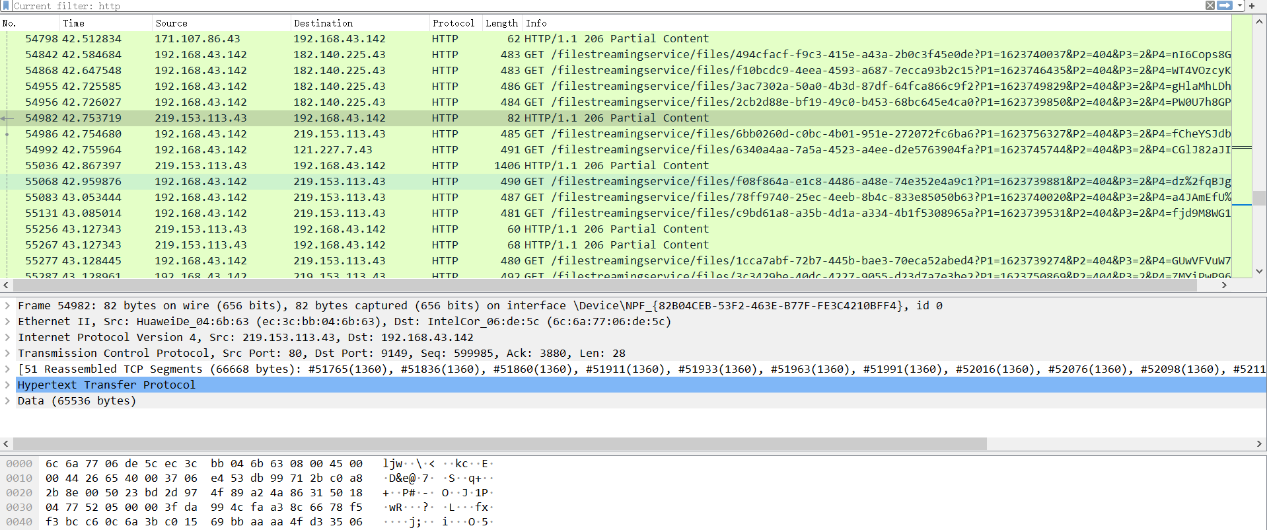


图12

可以看到如图12所示，在进行登录操作的过程中，192.168.43.142发送了大量的请求报文。可见正在进行相关登录的验证等。

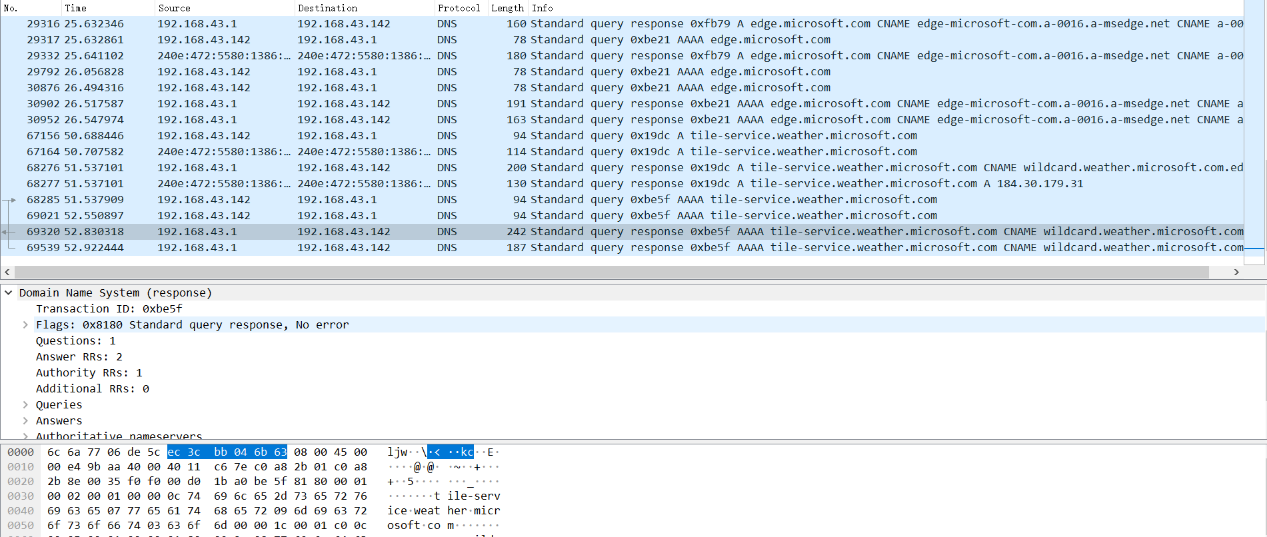


图13

此处为DNS报文的相关请求。可以看到在对应的事件节点存在相应的DNS请求，只不过请求的网址并不是我们的[www.webdev.com](http://www.webdev.com)，这里我也猜测可能是因为我使用的微软的edge浏览器，因此只显示请求到edge上，而之前访问过webdev，相应的网址存在了浏览器的缓存中。此外我也猜测，由于该部分是在课堂上完成，连接的是手机的热点，因此可能产生相关的影响。在报文中出现了我的华为手机的相关信息，因此猜想正确。

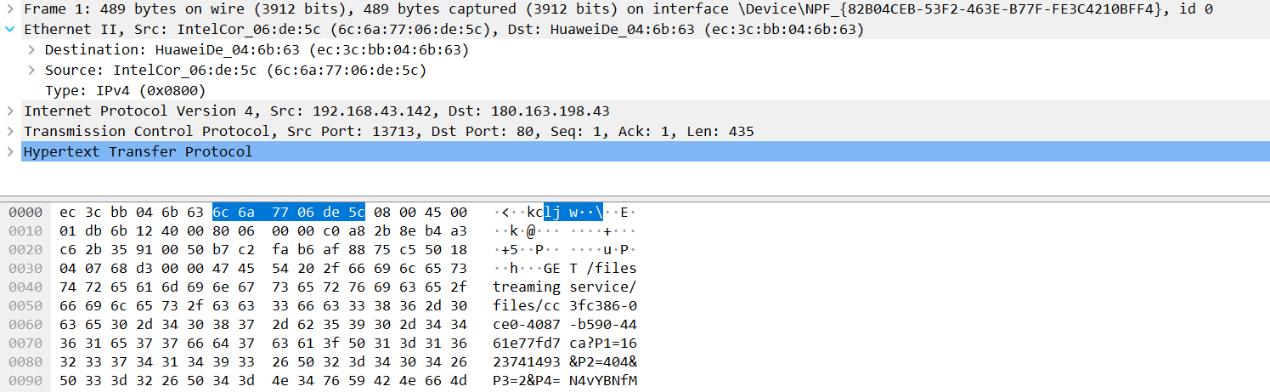


图14

在进行注册的过程中DNS报文分析就更为清晰。通过图 可以通过分析source与destination的地址即为请求段和服务器的通讯。相关请求调用了smtp协议发送邮件至注册的邮箱中。调用了相关的设置、相关的网站，都可以在报文的详情中看到。

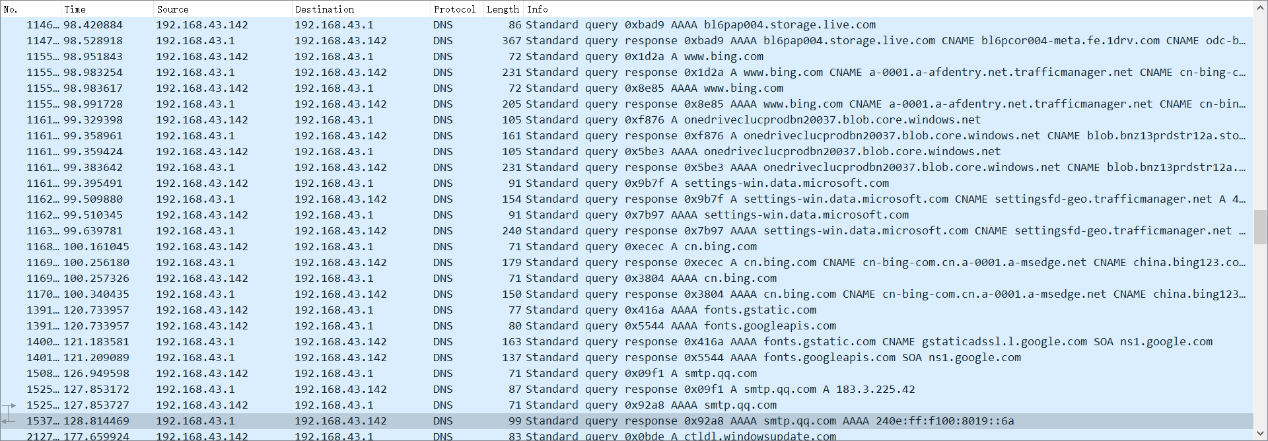
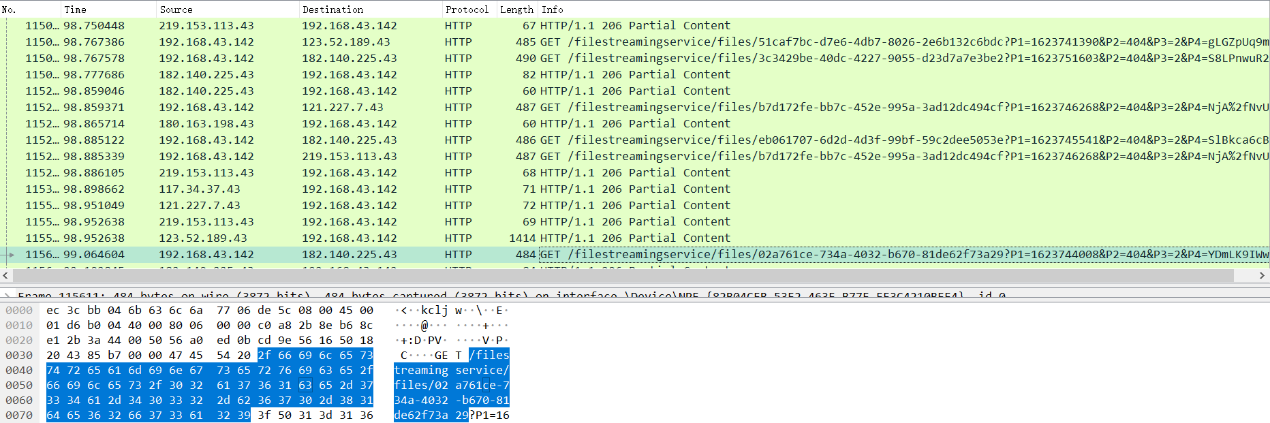


图15

同样的通过时间信息找到对应的HTTP协议。可见也同样的结论。发送了大量的请求报文，最终成功注册的一个过程。



5.分析仅用MD5加密可能存在的问题

此处查阅了大量论文（9-16），得出相关的结论。其实简单来说，仅使用MD5进行加密不可取，最主要的原因是不安全。在2004年我国的王小云院士率领的团队就宣布破解了MD5加密。MD5是一种摘要算法，在MD4的基础指上加入了“安全带”的概念用于提高安全性，但是由于不能够预防碰撞的攻击，因此其安全性是不够的。

由于MD5是一种密码散列函数，因此其不可逆。但是正如上文所说，它无法防止碰撞。假设现在有明文A，经过MD5加密后成为了密文B，尽管找不到一个逆函数使得在拿到密文B的情况下得出明文A，但是可以通过伪造一个新的明文C，使得在经过MD5加密后同样成为密文B。在这种情况下，系统会认为操作者拥有的就是明文A，从而得到相应的管理员权限。这种情况在加密大文件时尤其容易发生。也就是说文件相对容易发生碰撞。

那么为什么老师仍然推荐我们使用MD5进行加密呢？原因是MD5尽管存在碰撞的现象，但是在长度相对较短，或者定长的数据存贮中，碰撞的现象并不容易发生。这就意味着相对而言提高了安全性。但是MD5也容易被进行字典撞库，这在查阅的几篇论文中均有提到。

正是因为MD5加密有着大文件易发生撞库，不可逆，短数据、定长数据加密性相对较好的特点，MD5现在不用于加密、校验等，而更多用于刑侦案件的解密中。例如对操作系统中的文件进行哈希碰撞可以获得犯罪嫌疑人使用电脑的相关信息。这在相关文献中均有阐述。

此外，由于MD5早在2004年就被宣布破解，近几年来，研究其使用范围也从加密到了具体案件使用中。在查阅相关文献的过程中，我也发现，php使用的人数相对较少，在web开发中使用MD5的人数也并不多，主要体现在论文数量少，论文时间陈旧等。仍然在进行MD5加密的原因更多是因为大多数开发软件均有相应的函数，调用起来相对更加方便。

6.Blade模板分析

网页内容基于Blade模板，省去了较多的花里胡哨的背景等。单纯保留了登陆界面的导航词等。如图12所示，可见有在路由收到login或者register的请求时，进行相应的相应。Blade模板主要用于进行相应的继承。这里仅展示了部分Blade，具体代码详见附录文件。

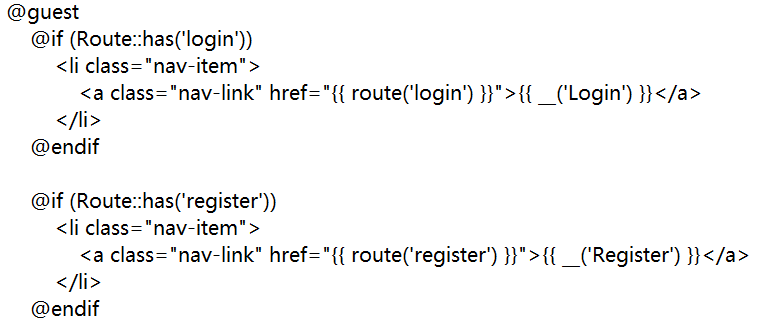


图12

7.自定义域名

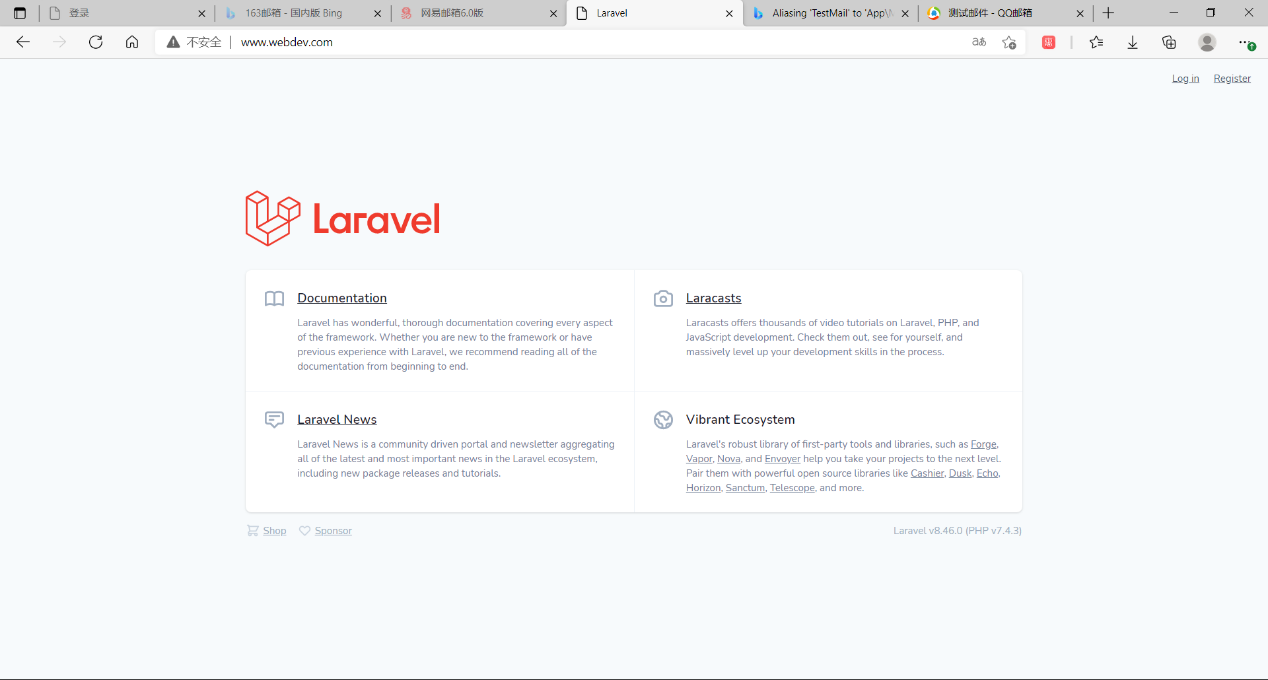
这里通过对Laravel网站的重定向进行自定义域名的设置。

第一步我们在对应的路径名称为C:\Windows\System32\drivers\etc文件目录中，找到文件hosts，在最后添加。相关的格式为127.0.0.1+自定义域名，在本例中，即为127.0.0.1 [www.webdev.com](http://www.webdev.com)。

第二步通过对httpd-vhosts.conf进行相应的修改。在末尾添加语句DocumentRoot和Directory之后指向域名对应的初始页面，ServerName后面接自定义的域名，要和host文件中相同。

第三步重新启动我们的服务器。

第四步输入自定义网页的名字，如图13所示，即可进入我们所定义的网站。



8.路由机制访问

路由机制的访问原理如图13所示。通过首页的.php文件，调用路由信息，对相关控制器进行访问。可以把 index.php 类比成网线；route/web.php 类比成路由器设备；controller 控制器类比成多部手机设备。Route作为一个类，通过::这一静态方法对类进行访问。在Route::后跟着一个请求方法，包括了get、post、delete等多种请求。两个参数一个表示路径，另一个表示闭包函数。在项目中，主要使用了路由中间件使得未验证邮箱的用户不能访问路由。在路由发送的过程中，调用相关token进行。

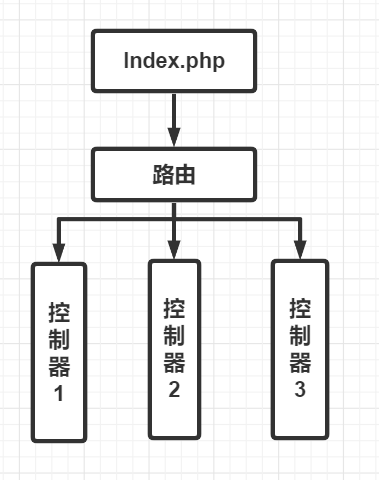


图13

9.邮件发送

通过cmd命令行的相关调用，成功发送了相应的测试文件。分别向两个邮箱发送测试文件，经测试均成功收到。图14和图15的邮件发送分别对应图15和图16的文件接收，这里使用了两个账号进行测试，遇到了相关问题。可以看到图16由于发送和接受的是同一个邮箱，因此相关的ID显示并不正常，而图17就没有这样的问题。

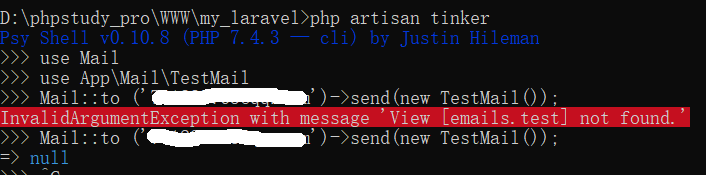


图14

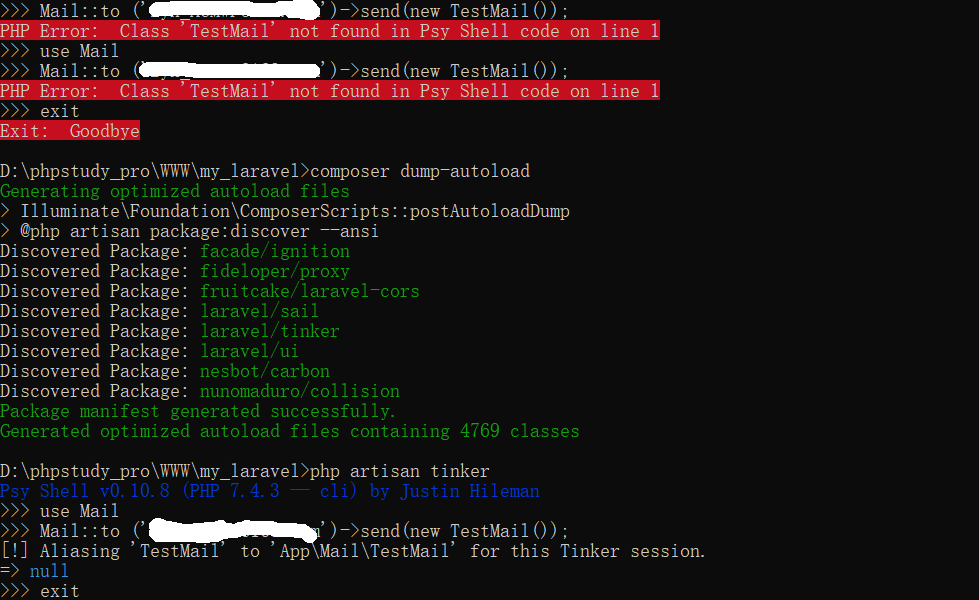


图15

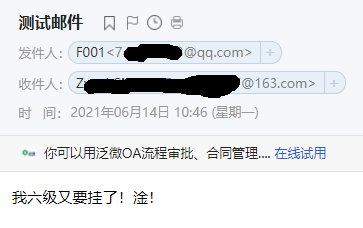
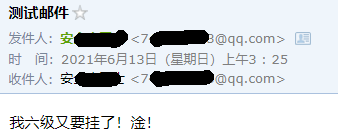


图16 图17

1. **实验分析总结及心得（该部分也是评分的一个重点）**

（结合所学知识对实验过程中观察到的实验结果进行分析总结，以便加深对知识的理解，并总结通过实验学到的知识或技术）

特别推荐学生写出做实验遇到的问题，以及从原理分析得到解决方案的过程。

本次实验遇到的问题着实不小，前前后后的实验一共做了3次。本次实验很难说有什么特别明显的亮眼之处，更多是一个处在别人背后重复造轮子的过程。但是轮子成功造出来了而且跑的很好，这就是成功之处。本次实验我参考了较多的网上资料和书籍，我们的教材再相应的章节存在较多经典的描述，起到了很好的帮助作用。

本次实验的过程中，在发送邮件的部分，遇到了邮件不能正常发送的问题。经过检查后我们发现是由于在发送邮件部分的代码传入的参数与另一个参数输入混淆了，说明在项目编写时各个变量的独立性和识别性还有待提高。此外遇到的问题是一开始甚至都不能进行php artisan 遇到问题后我查阅了相关资料，但是难以解决问题，因此我向群内同学提问，发现是composer的版本和laravel/ui的版本不匹配，通过对composer的一系列操作之后解决了相关的问题。分别是更新-清理缓存-自我更新-自动跳转载入这四个步骤（composer update\composer clearcache\composer selfupdate\composer dumpautoload）。因此解决了相关的问题。另外由于个人的一些不良习惯所致，未能及时记录做实验过程中遇到的相关错误，在实验四中将会进行相应改正，希望在下一个实验中可以更快更好地完成相关任务。

1. **参考文献**
2. <https://blog.csdn.net/low5252/article/details/97007172>
3. <https://blog.csdn.net/tingliting/article/details/53538602>
4. <https://laravelacademy.org/post/8920>
5. <https://blog.csdn.net/tingliting/article/details/53538602>
6. <https://github.com/appstract/laravel-blade-directives>
7. <https://blog.csdn.net/xiao_xiao_lan/article/details/106215001>
8. <https://blog.csdn.net/minose/article/details/80567202?utm_medium=distribute.pc_relevant.none-task-blog-2%7Edefault%7EBlogCommendFromMachineLearnPai2%7Edefault-4.control&depth_1-utm_source=distribute.pc_relevant.none-task-blog-2%7Edefault%7EBlogCommendFromMachineLearnPai2%7Edefault-4.control>
9. <https://learnku.com/docs/laravel/5.6>
10. 基于PHP的web应用中的SQL注入及防御措施\_王娅
11. 基于PHP数据加密安全性探讨\_刘凯凌
12. WEB平台设计缺陷分析\_汪永生
13. MD5值的电子取证应用研究\_王聪
14. 哈希算法在电子数据取证中的应用研究\_王冠
15. Web数据库中信息的安全策略\_张翠玲
16. <https://cloud.tencent.com/developer/article/1805749>
17. <https://blog.csdn.net/christ2/article/details/51661011>
18. <https://www.jb51.net/article/184669.htm>
19. <https://blog.csdn.net/xiao_xiao_lan/article/details/106174841>
20. PHP和MySQL Web开发