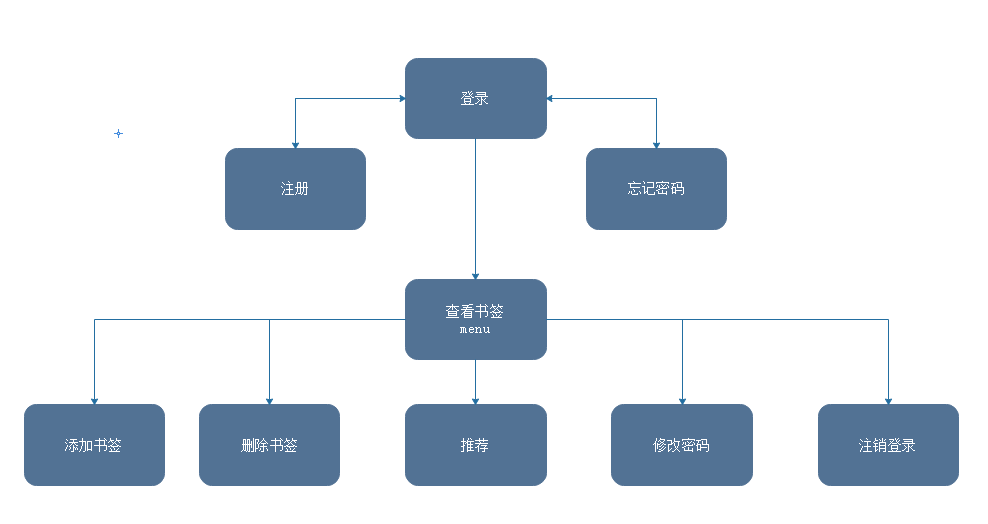
1. **项目描述**

**构建一个书签收藏夹：框架如下：（**<https://github.com/demonsheart/an-item-for-bookmark>**）**

****

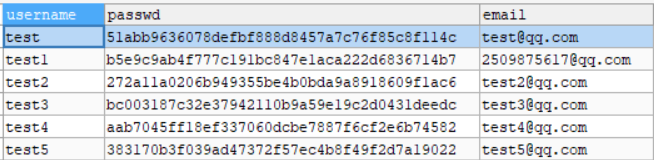
1. **开发环境**

**Laravel5.5框架**

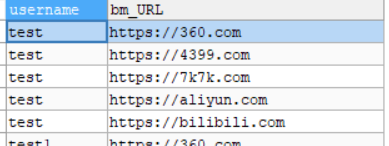
1. **详细设计**
2. **程序结构（对涉及的变量、类、程序结构进行说明）**

**数据库设计：**

**表一：user表，储存用户名、密码散列值、邮箱。**

****

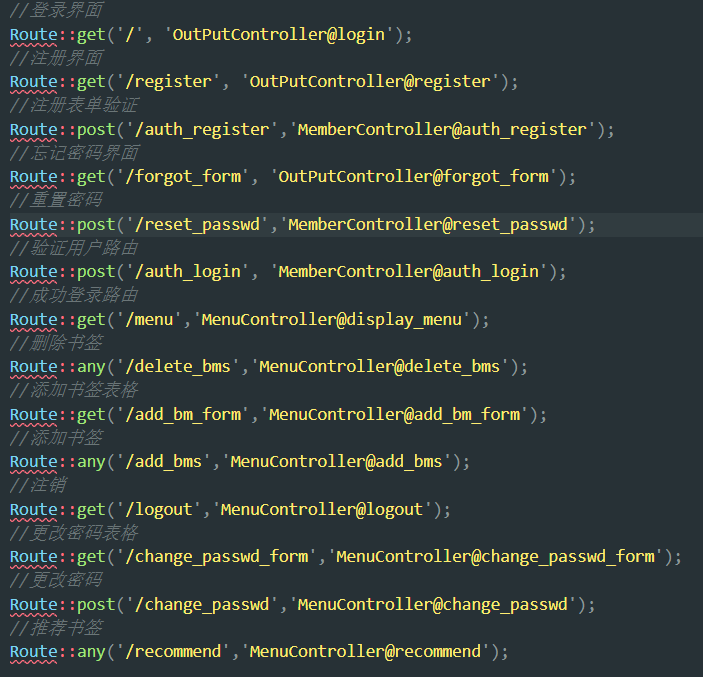
**表二：bookmark表，储存用户名、收藏网址**

****

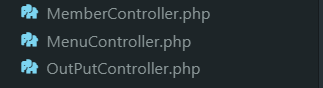
1. **主要功能（对实现的系统功能进行阐述，哪个函数完成什么功能）**

**概述：**

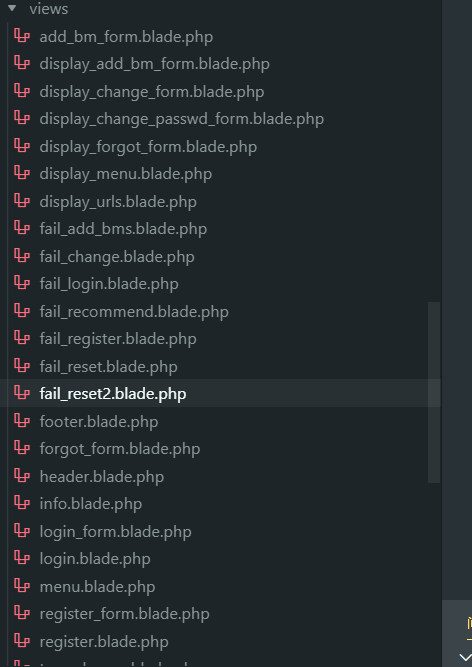
**路由器：负责各页面的跳转**

****

**控制器：唤起视图，并处理表格数据，与数据库交互**

****

**视图：显示页面**

****

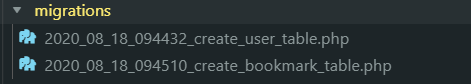
**模型：为连接数据库做准备**

****

**验证规则：自定义URL字段的验证规则**

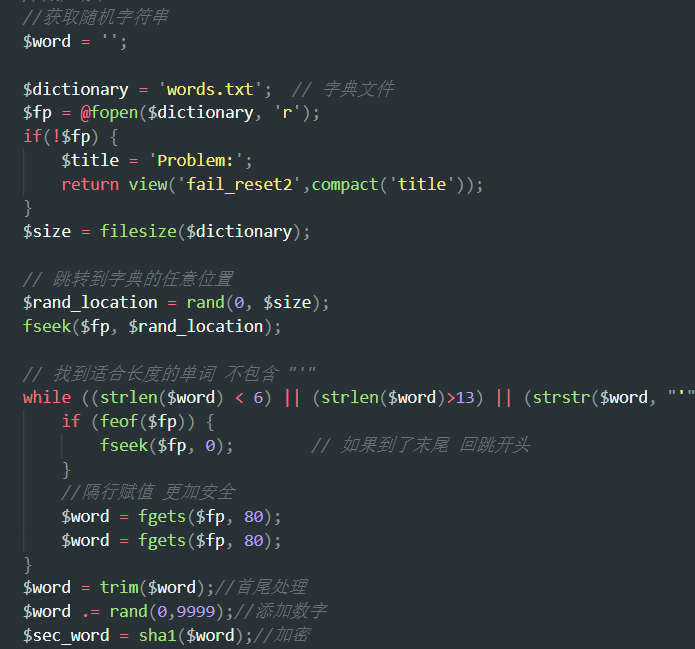
****

**数据迁移：建表**

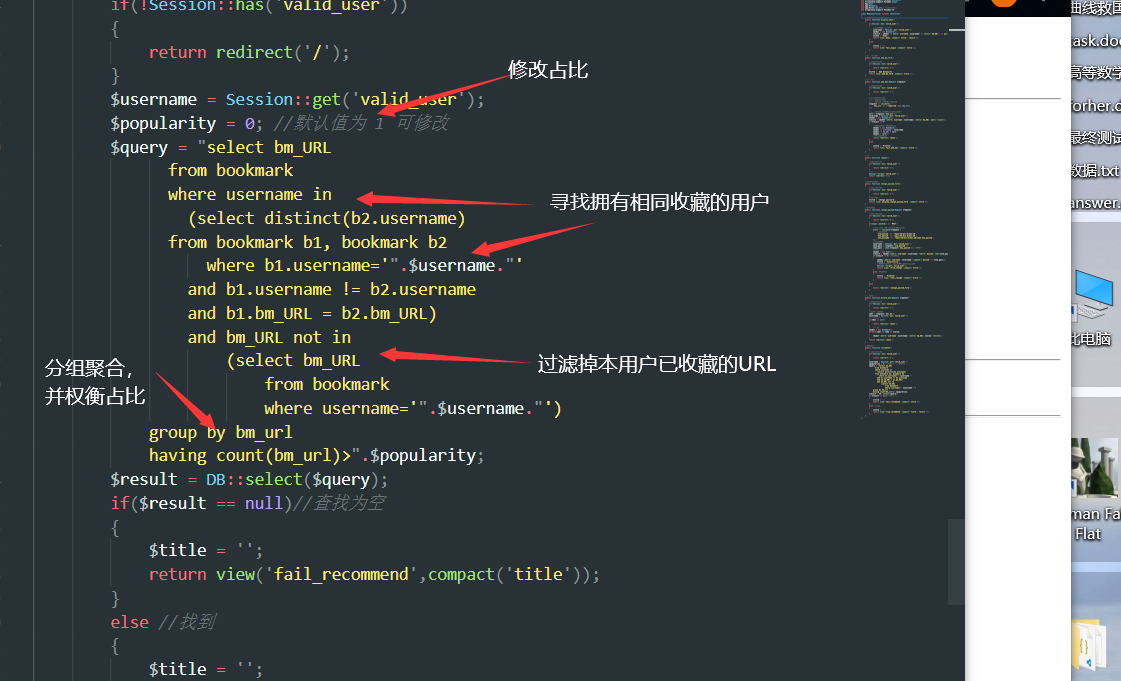
****

**※整体细节考虑：**

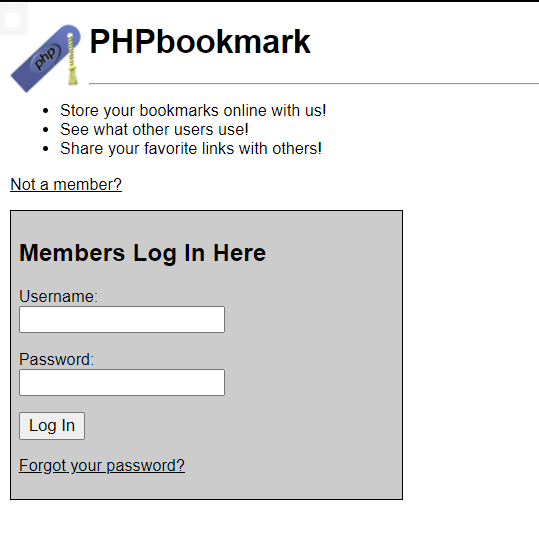
1. **所有出现表单的地方都调用了laravel自带的表单验证**
2. **所有密码都使用sha1()函数加密以及验证**
3. **所有表单都调用了csrf验证机制**
4. **视图中将header以及footer模板单独编写文件，实现复用代码**
5. **成功登录将username保存进全局变量Session(‘valid\_user’)中，注销或者更改密码则注销全局变量Session(‘valid\_user’)，验证机制也是基于此。**
6. **生成随机密码函数：从一个英文字典文件中读取单词，并进行处理：**

****

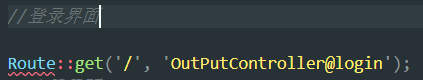
1. **推荐算法**

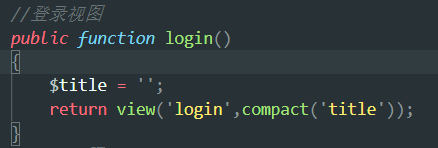


**（1）、登录界面**

****

**路由唤起图表：**

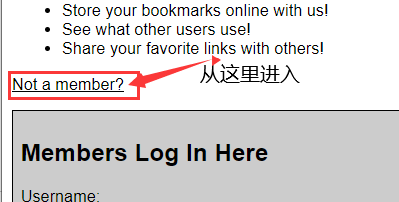
****

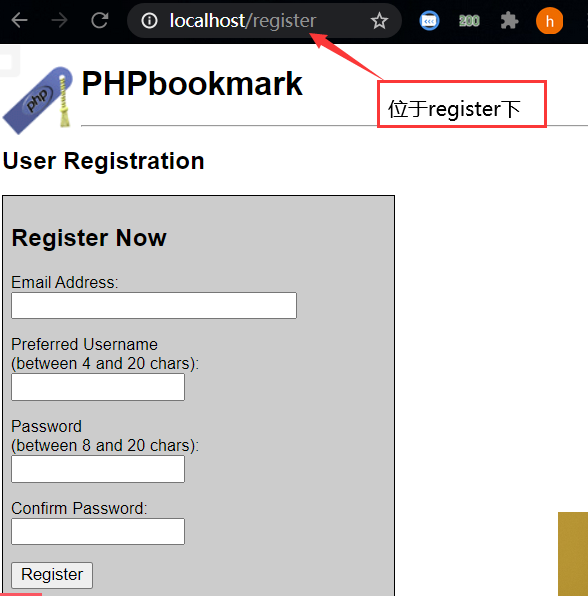
****

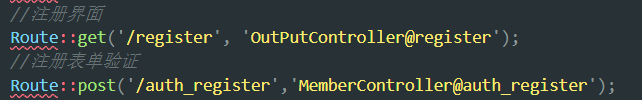
****

**auth\_login()将验证输入表单数据、用户存在与否、用户与密码是否匹配、成功通过验证则设置开始session会话并跳转menu界面**

**（2）、注册界面**

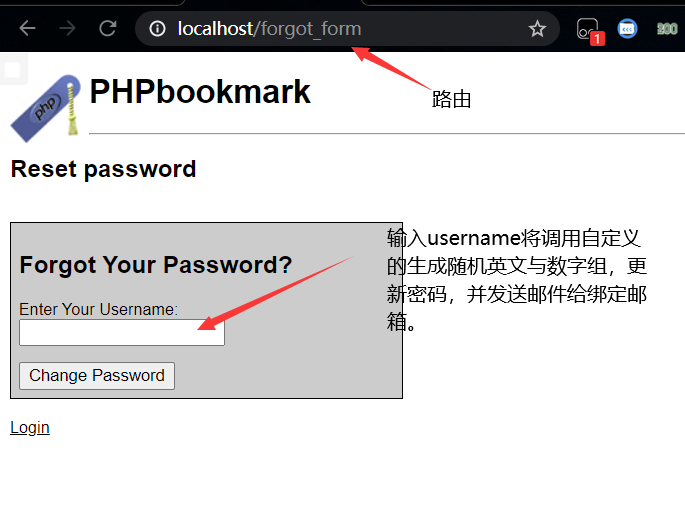


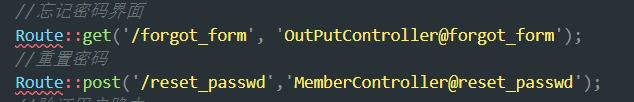




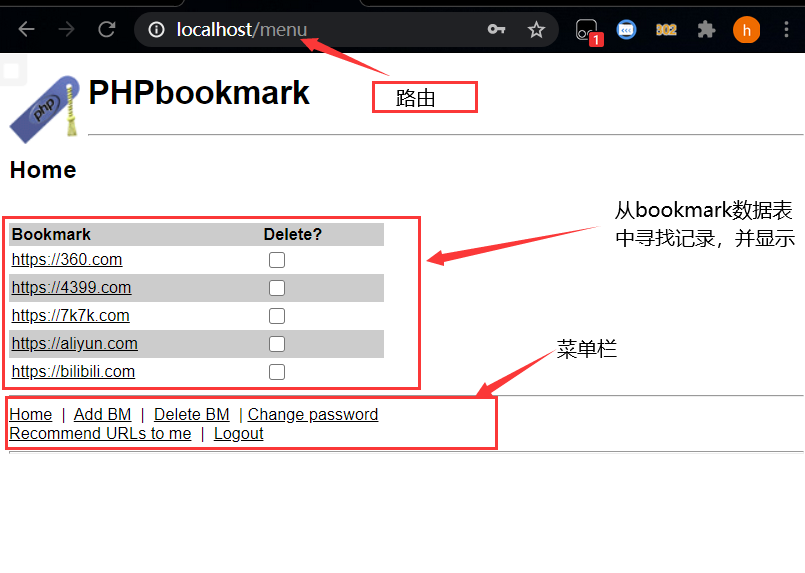
**register将唤起视图呈现表单，auth\_register将检验输入数据的有效性并将帐户信息存进数据库**

**（3）、忘记密码**





**（4）、主菜单**

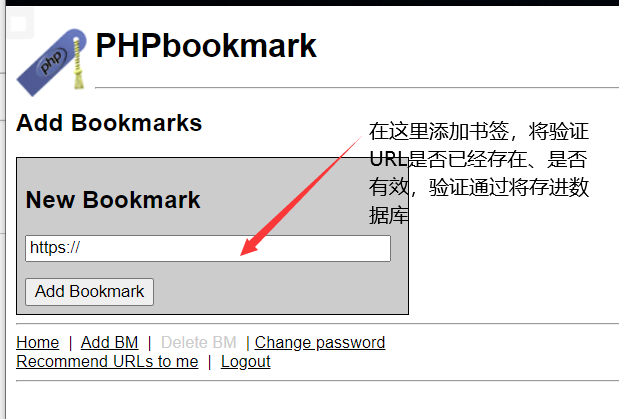


**（6）、菜单功能：**

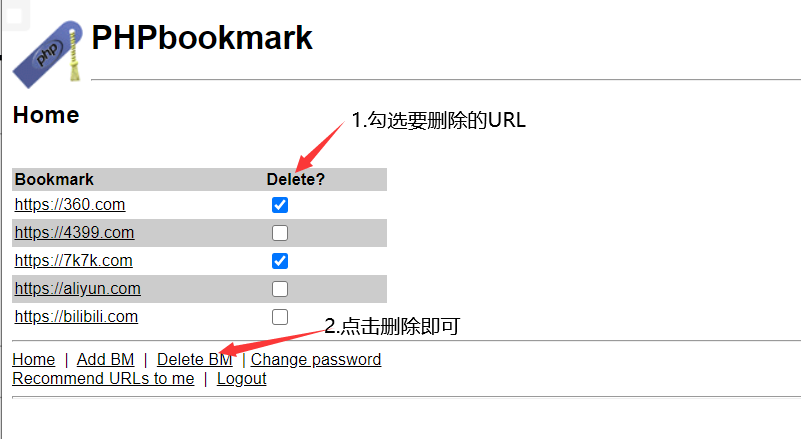
**1.Home**

**将回到Home页面**

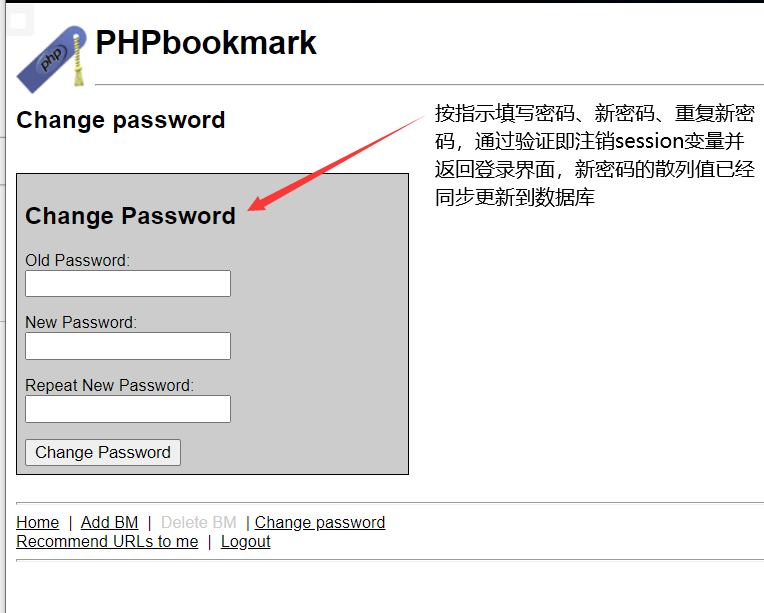
**2.Add BM**



1. **Delete BM**



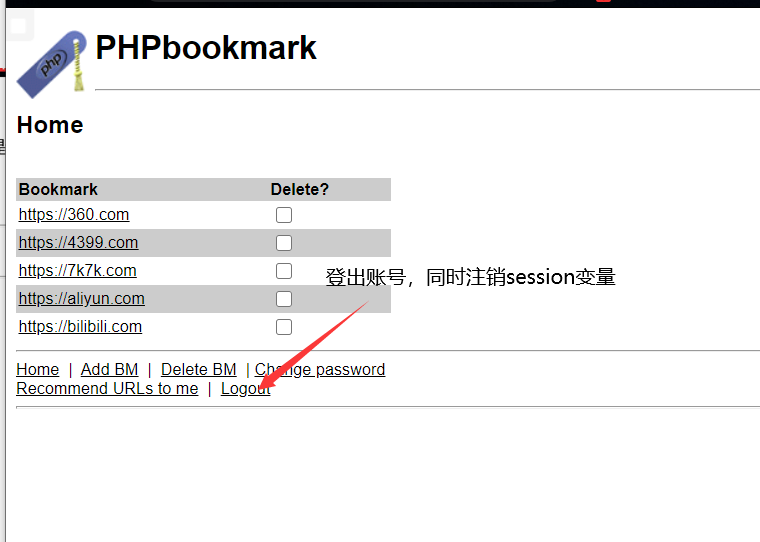
1. **Change password**



1. **Recommend**



1. **Logout**



1. **程序使用说明**
2. **项目打包**
3. **安装composer之后在项目根目录下使用composer install命令安装vendor目录**
4. **Database目录下必须有factories、migrations、seeds目录**
5. **必须先配置.env文件，链接数据库、填写key（php artisan key:generate）**
6. **必须先建表（建议使用php artisan migrate）**
7. **如果第一次建表，则需要先执行php artisan migrate:install**
8. **本项目的缺陷在于mail函数不能真正的发送邮件（可能与防火墙、配置、协议问题暂时无法解决），可在如下目录中查看邮件**

****

1. **测试及结果截图**

**测试不再一一罗列**

1. **总结**

**开发过程中，真正体会laravel框架的实用，将路由、控制、视图分开，使得开发逻辑清晰明了，laravel其封装的方法也极大地降低了开发难度。**