

Website Development ---XXBBS

- 项目动机

受到MCBBS论坛的启发，想要了解BBS论坛运行的机制，并且目前同济大学好像缺少相关的校园论坛，因此想做一个基于Flask的BBS论坛

- 简介

此网站（暂名为XXBBS）为2019级计算机科学导论小组项目作业，网站定位为小型校园论坛BBS

XXBBS 使用 Flask 和 Bootstrap 前后端开发，使用 MySQL 作为数据库

由部分小组成员@Yaozhtj @oscarab @sSaltgrey @crazy-toy @Kururinnpa 和其他小组成员共同完成了此网站

体验地址：<http://wenweb.ahy1.top/>

Github：<https://github.com/oscarab/BBS>

感谢@oscarab的部署

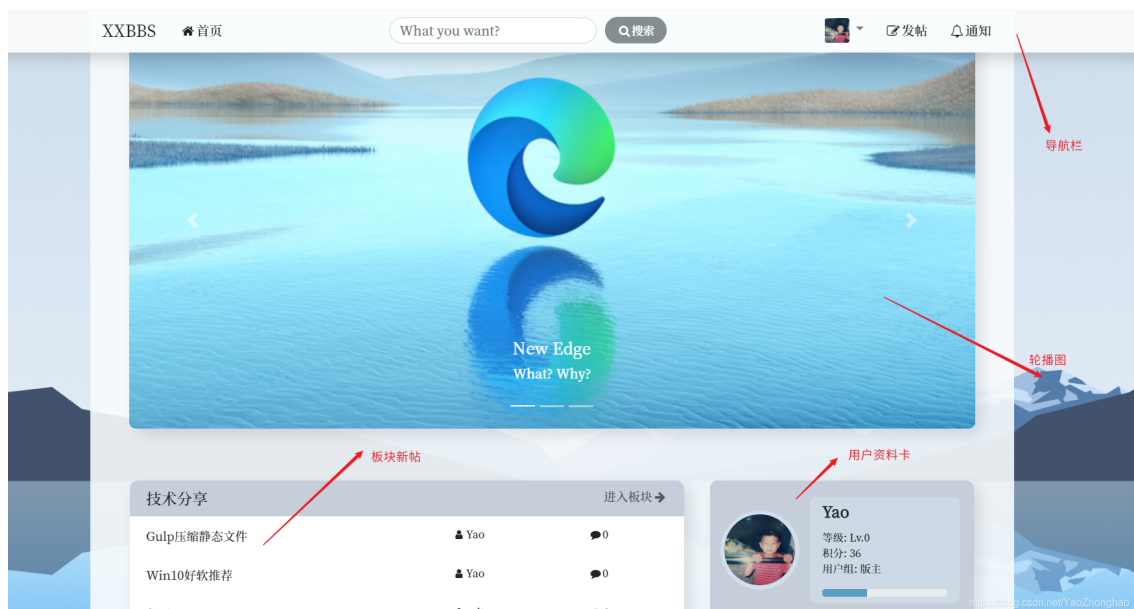
- 分工

- 前端：姚忠豪（组长）袁梓涵 杨凡 尹浩天
- 后端：文昕颢（组长）严嘉辉 赵炳哲

- 功能

- 首页

界面如下：



- 注册登录功能

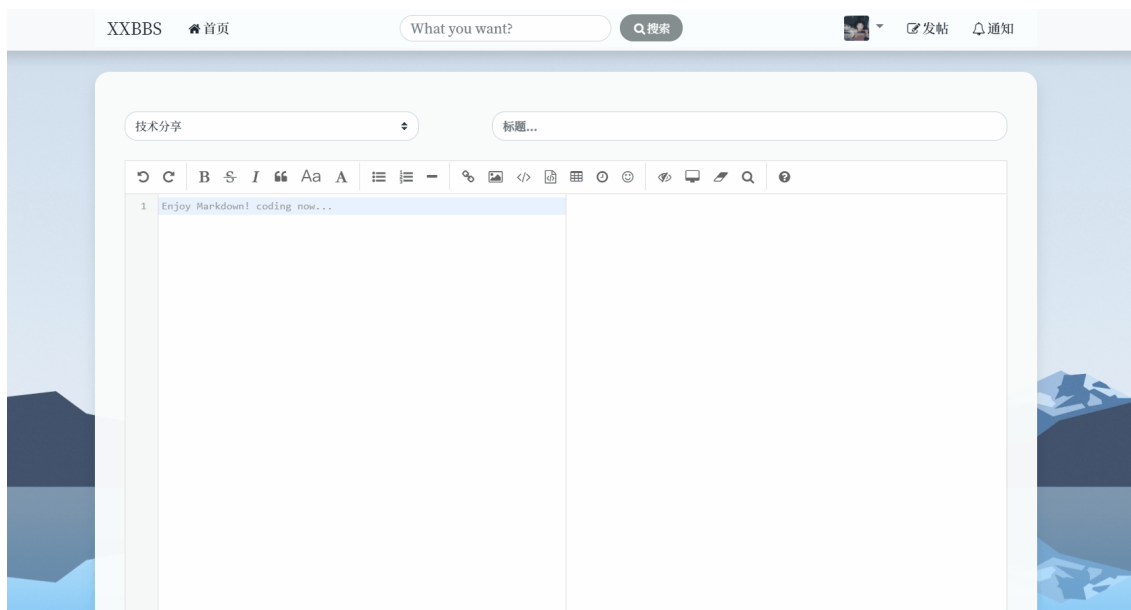
登录支持错误提示，注册界面如下(支持邮箱验证码发送和错误提示):



The image shows a registration form on a forum website. The form is titled "注册" (Register) and is set against a background of a blue mountain range. The form fields include: "电子邮箱" (Email) with the value "Yaozhtj@yeah.net", "用户名" (Username) with the value "yao", "密码" (Password) with six dots, "确认密码" (Confirm Password) with six dots, and "验证码" (Verification Code). There is a "发送验证码" (Send Verification Code) button next to the verification code field. At the bottom of the form is a large "注册" (Register) button. The website header includes "XXBBS", a home icon, a search bar with "What you want?", a search button, and links for "登录" (Login) and "注册" (Register).

- 发帖功能

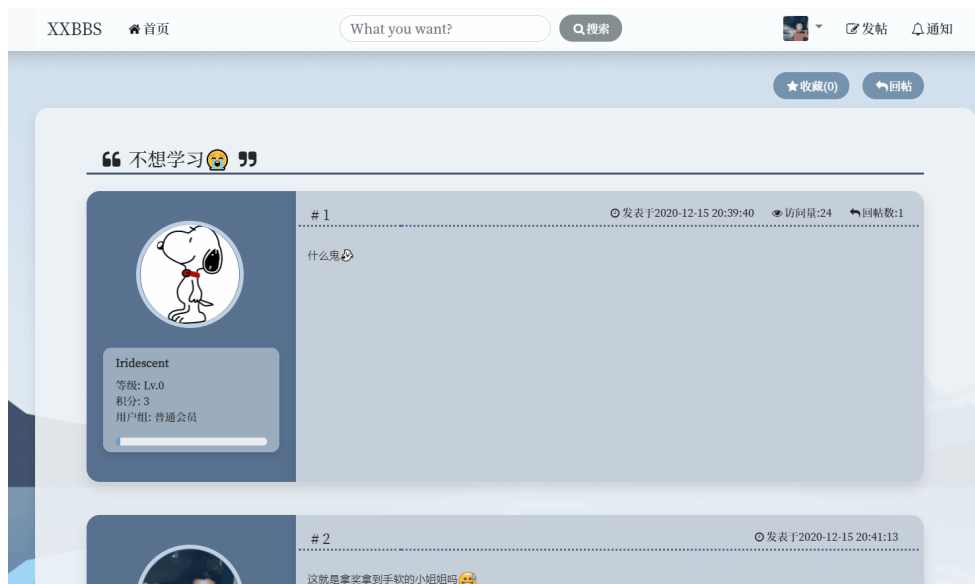
发布帖子使用Markdown编辑器，使用开源的 [Editormd](#)，虽然此工具已经没人维护，还有些 Bug(可参见其 [issue](#))，但功能算是比较好的了



The image shows the post creation interface on the forum website. The interface includes a header with "XXBBS", a home icon, a search bar with "What you want?", a search button, and links for "发帖" (Post) and "通知" (Notification). The main content area is a form for creating a post. It has a "技术分享" (Technical Sharing) category selector and a "标题..." (Title...) field. Below these is a rich text editor with a toolbar containing icons for bold, italic, underline, link, unlink, list, quote, code, image, video, table, and other formatting options. The editor area shows a single line of text: "1 Enjoy Markdown! coding now...".

编辑框部分功能(支持跨域上传图片，表情，代码，链接等等):

- 回帖功能(回帖部分也使用了相对简洁的markdown编辑):



- 用户主页

- 用户可以查看通知(自己的帖子有无新回复), 自己发的帖子, 收藏的帖子



- 支持用户的积分制与升级, 积分规则: 参与发帖或者回复均可增加指定积分, 积分累加可升级
- 支持上传头像
- 支持积分排行榜

帖子详情页面

支持 markdown 转 html 渲染

XXBBS 首页 What you want? 搜索

Yao
等级: Lv.0
积分: 36
用户组: 版主

安装依赖

首先安装gulp,在git bash中执行以下命令:

```
1. $npm install gulp -g
```

要实现gulp压缩需要安装以下几个模块:

```
gulp-htmlclean // 清理html  
gulp-htmlmin // 压缩html  
gulp-minify-css // 压缩css  
gulp-uglify // 混淆js
```

通过以下命令安装模块:

```
1. npm install gulp-htmlclean gulp-htmlmin gulp-minify-css gulp-uglify gulp-imagemin --save
```

安装完成之后可以在博客根目录的package.json文件中可以看到:

```
1. "gulp": "^4.0.2",  
2. "gulp-htmlclean": "^2.7.22",  
3. "gulp-htmlmin": "^5.0.1",  
4. "gulp-imagemin": "^7.1.0",  
5. "gulp-minify-css": "^1.2.4",  
6. "gulp-uglify": "^3.0.2",
```

添加gulpfile.js

在博客根目录下,新建gulpfile.js文件,内容如下:

<https://blog.csdn.net/YaoZhonghao>

- 权限功能

用户组分为 管理员 , 版主 , 普通会员

管理员权限最高,可增删用户帖子,移动帖子所属的板块,可禁言版主和普通会员,删除回复

版主可增删本板块的帖子,移动本板块的帖子,可以禁言普通用户

“ Gulp压缩静态文件 ”

1 发表于2020-12-15 22:49:00 访问量:39 回帖数:1

删除 编辑

引言

hexo博客的访问速度有时很感人。这里介绍一种gulp压缩的方法,将自己的博客静态文件压缩,以此来提高网站的性能。

安装依赖

首先安装gulp,在git bash中执行以下命令:

```
1. $npm install gulp -g
```

要实现gulp压缩需要安装以下几个模块:

```
gulp-htmlclean // 清理html  
gulp-htmlmin // 压缩html
```

<https://blog.csdn.net/YaoZhonghao>

- 支持移动端



- 开发过程

- 前端

实现： `html` + `css` + `js`

使用前端框架: `Bootstrap`

```
<link href="https://cdn.bootcdn.net/ajax/libs/twitter-bootstrap/4.5.3/css/bootstrap.min.css" rel="stylesheet">
<script src="{{ url_for('static', filename='js/jquery.min.js') }}"></script>
<script src="{{ url_for('static', filename='js/popper.min.js') }}"></script>
<script src="https://cdn.bootcdn.net/ajax/libs/twitter-bootstrap/4.5.0/js/bootstrap.min.js"></script>
```

模板引擎: `jinja2`

编辑器插件: `Editormd`

```
<link rel="stylesheet" href="{{ url_for('static', filename='editormd/css/editormd.min.css') }}">
```

```

<script src="https://cdn.bootcdn.net/ajax/libs/editor-
md/1.5.0/editormd.min.js"></script>

<script type="text/javascript">
    $(function () {
        var editor = editormd("test-editor", {
            emoji: true,
            height: 600,
            imageUpload: true,
            imageFormats: ["jpg", "jpeg", "gif", "png", "bmp", "webp"],
            tex: false,
            flowChart: false,
            sequenceDiagram: false,
            imageUploadURL: "{{ url_for('edit.upload') }}",
            path: "{{ url_for('static',filename='editormd/lib/') }}",
            toolbarIcons: function () {
                return ["undo", "redo", "|", "bold", "del", "italic",
"quote", "ucwords", "uppercase", "|", "list-ul", "list-ol", "hr", "|",
"link", "image", "code", "code-block", "table", "datetime", "emoji", "|",
"watch", "preview", "clear", "search", "|", "help"]
            }
        });
    });
</script>

```

- 后端

框架: Flask

```

def create_app():
    app = Flask(__name__, instance_relative_config=True)

    app.config.from_mapping( #默认配置
        SECRET_KEY="woaixuexiaaajikedao666",
        MAX_CONTENT_LENGTH=1 * 1024 * 1024, #限制上传文件大小1M
        PERMANENT_SESSION_LIFETIME = timedelta(hours=24),

        HOST="127.0.0.1", #Mysql数据库设置
        PORT=3306,
        USER="root",
        PASSWORD="root",
        DATABASE="bbs",

        MAIL_SERVER='smtp.example.com', #smtp服务器设置
        MAIL_USERNAME="root",
        MAIL_PASSWORD="root",
        MAIL_DEFAULT_SENDER="root",
        MAIL_USE_TLS=True,
        MAIL_PORT=587
    )

```

```

app.config.from_pyfile('config.py', silent=True) #载入配置文件
try:
    os.makedirs(app.instance_path)
except OSError:
    pass

global mail
mail = Mail(app)

from . import database
database.init_app(app) #注册数据库初始化命令

from . import auth
app.register_blueprint(auth.authbp) #登陆注册
from . import userpage
app.register_blueprint(userpage.userpagebp) #用户主页
from . import index
app.register_blueprint(index.indexbp) #主页
from . import edit
app.register_blueprint(edit.editbp) #帖子编辑
from . import posts
app.register_blueprint(posts.postsbp) #主题页面
from . import parts
app.register_blueprint(parts.partsbp) #主题页面

@app.route('/')
def hello():
    return redirect(url_for('index.index'))

return app

```

数据库: MySQL

```

def initDatabase():
    db = getDatabase()
    cursor = db.cursor()

    cursor.execute('CREATE TABLE IF NOT EXISTS post('
        'id INT UNSIGNED NOT NULL PRIMARY KEY AUTO_INCREMENT,'
        'title TEXT NOT NULL,'
        'type TINYINT NOT NULL,'
        'content LONGTEXT NOT NULL,'
        'userid INT UNSIGNED NOT NULL,'
        'posttime DATETIME NOT NULL,'
        'updatetime DATETIME NOT NULL,'
        'reply MEDIUMTEXT NOT NULL,'
        'visit BIGINT UNSIGNED NOT NULL,'
        'collect INT UNSIGNED NOT NULL'
        ');')

#主题表单
#主题id
#标题
#主题分类 1 2 3
#内容
#发主题的用户id
#发表时间
#更新时间
#主题回复
#阅读量
#收藏量

```

```

cursor.execute('CREATE TABLE IF NOT EXISTS reply('
    'id INT UNSIGNED NOT NULL PRIMARY KEY AUTO_INCREMENT,' #主题回复id
    'userid INT UNSIGNED NOT NULL,' #用户id
    'content LONGTEXT NOT NULL,' #内容
    'posttime DATETIME NOT NULL' #发表时间
    ');')

cursor.execute('CREATE TABLE IF NOT EXISTS user ('
    'uuid INT UNSIGNED NOT NULL PRIMARY KEY AUTO_INCREMENT,' #用户id
    'name varchar(128) BINARY NOT NULL,' #昵称
    'ip varchar(32) NOT NULL,' #ip
    'password varchar(256) NOT NULL,' #密码
    'email varchar(32) NOT NULL,' #邮箱
    'registertime DATETIME NOT NULL,' #注册时间
    'lastlogin DATETIME NOT NULL' #最后登录时间
    ');')

cursor.execute('CREATE TABLE IF NOT EXISTS userinfo ('
    'uuid INT UNSIGNED NOT NULL PRIMARY KEY AUTO_INCREMENT,' #用户id
    'warn TEXT NOT NULL,' #提醒
    'permission varchar(32) NOT NULL,' #权限列表 管理员
    'collect TEXT NOT NULL,' #收藏的主题
    'point MEDIUMINT UNSIGNED NOT NULL)' #积分
)

cursor.execute('CREATE TABLE IF NOT EXISTS mail ('
    'destination varchar(32) NOT NULL,' #发送邮箱
    'posttime INT UNSIGNED NOT NULL);' #发送时间
)

```

版主1/2/3/4 普通会员 禁言

第三方库：

Pymysql：操作数据库

Flask_mail：发送邮件验证

Pillow：生成验证码图片

关键技术实现：

- **Pymysql**：使用该库进行Mysql数据库的连接与操作。
- **Flask_mail**：为了在注册时实现邮箱验证码的发放，使用该库进行发送邮件，发送邮件的邮箱采用阿里云的邮箱，验证码储存于 **sessions** 中，用户进行输入时即可进行比对。
- **Pillow**：为了在登陆时实现随机验证码图片的生成，使用该库生成图片，首先随机生成包含字母或数字的验证码，然后绘至图片上，再传送至前端。
- 图片上传：头像上传处采用表单的方式传至后端，后端进行图片大小判断，避免传入过大图片，再裁剪至正方形图片，最后储存到以用户 **uuid** 命名的文件夹下。发表帖子时上传图片的前端功能由 **editor.md** 提供，后端接收图片后，用上传时间对其重命名，再储存在相应位置。

- 分页：在 `url` 出增加一个整型参数表示页数，后端按发表顺序获取所有帖子后均分为若干组，每组即为一页的所有帖子，再根据需要的页数回传相应的帖子。