



计算机学院
互联网数据库开发

团队作业 2——设计文档

项目名称：抗战胜利 80 周年纪念网站

团队名称：抗战纪念开发组

组长：童汉鑫（学号：2311995）

组员：刘浩泽（学号：2212478）

组员：彭浩然（学号：2313314）

组员：董珺（学号：2212880）

2025 年 12 月 21 日

目录

| | |
|---|-----------|
| 1 系统架构设计 | 4 |
| 1.1 总体架构 | 4 |
| 1.1.1 架构层次 | 4 |
| 1.1.2 系统架构图 | 4 |
| 1.2 技术架构 | 5 |
| 1.2.1 开发框架 | 5 |
| 1.2.2 前端技术栈 | 5 |
| 1.2.3 后端技术栈 | 5 |
| 1.3 目录结构设计 | 5 |
| 1.4 模块划分 | 7 |
| 2 数据库设计 | 7 |
| 2.1 数据库概述 | 7 |
| 2.2 数据表清单 | 8 |
| 2.3 详细表结构设计 | 8 |
| 2.3.1 用户表 (pre_sys_user) | 8 |
| 2.3.2 文章分类表 (pre_news_category) | 9 |
| 2.3.3 文章主表 (pre_news_article) | 10 |
| 2.3.4 评论表 (pre_news_comment) | 10 |
| 2.3.5 留言板表 (pre_guestbook) | 11 |
| 2.3.6 统计表设计 | 11 |
| 2.4 数据库 E-R 图 | 12 |
| 2.5 数据完整性约束 | 12 |
| 2.5.1 实体完整性 | 12 |
| 2.5.2 参照完整性 | 12 |
| 2.5.3 域完整性 | 12 |
| 2.5.4 用户定义完整性 | 13 |
| 3 接口设计 | 13 |
| 3.1 前台接口设计 | 13 |
| 3.1.1 首页相关接口 | 13 |
| 3.1.2 文章相关接口 | 13 |
| 3.1.3 用户认证接口 | 14 |
| 3.1.4 评论接口 | 14 |
| 3.1.5 留言板接口 | 15 |
| 3.2 后台接口设计 | 15 |
| 3.2.1 文章管理接口 | 15 |
| 3.2.2 分类管理接口 | 15 |
| 3.2.3 用户管理接口 | 16 |
| 3.2.4 评论管理接口 | 16 |
| 3.2.5 留言管理接口 | 16 |

| | |
|---------------------------|-----------|
| 3.3 安全设计 | 16 |
| 3.3.1 CSRF 防护 | 16 |
| 3.3.2 XSS 防护 | 16 |
| 3.3.3 SQL 注入防护 | 17 |
| 3.3.4 权限控制 | 17 |
| 4 界面设计 | 17 |
| 4.1 前台界面设计 | 17 |
| 4.1.1 设计风格 | 17 |
| 4.1.2 导航栏设计 | 17 |
| 4.1.3 首页布局 | 18 |
| 4.1.4 文章列表页设计 | 19 |
| 4.1.5 文章详情页设计 | 19 |
| 4.1.6 团队展示页设计 | 19 |
| 4.1.7 留言板页设计 | 19 |
| 4.2 后台界面设计 | 20 |
| 4.2.1 设计风格 | 20 |
| 4.2.2 菜单结构 | 20 |
| 4.2.3 列表页设计 | 20 |
| 4.2.4 表单页设计 | 21 |
| 5 安全性设计 | 21 |
| 5.1 认证与授权 | 21 |
| 5.1.1 用户认证 | 21 |
| 5.1.2 权限控制 | 21 |
| 5.2 数据安全 | 21 |
| 5.2.1 输入验证 | 21 |
| 5.2.2 SQL 注入防护 | 22 |
| 5.2.3 XSS 防护 | 22 |
| 5.2.4 CSRF 防护 | 22 |
| 5.3 会话安全 | 22 |
| 6 性能优化设计 | 22 |
| 6.1 数据库优化 | 22 |
| 6.2 前端优化 | 23 |
| 6.3 代码优化 | 23 |
| 7 测试设计 | 23 |
| 7.1 测试策略 | 23 |
| 7.2 测试用例 | 24 |
| 7.2.1 用户认证测试 | 24 |
| 7.2.2 文章管理测试 | 24 |
| 7.2.3 安全测试 | 24 |

| | |
|---------------|-----------|
| 8 部署设计 | 24 |
| 8.1 部署环境 | 24 |
| 8.2 部署步骤 | 25 |
| 8.3 配置说明 | 25 |
| 8.3.1 数据库配置 | 25 |
| 8.3.2 URL 配置 | 26 |
| 9 维护设计 | 26 |
| 9.1 日志管理 | 26 |
| 9.2 数据备份 | 26 |
| 9.3 监控告警 | 27 |
| 10 总结 | 27 |

1 系统架构设计

1.1 总体架构

本系统采用经典的三层架构设计，基于 Yii2 框架的 MVC 模式开发，分为前台展示系统和后台管理系统两个子系统。

1.1.1 架构层次

1. 表现层 (View Layer)

- 前台：响应式页面，适配 PC/平板/手机
- 后台：基于 Bootstrap 的管理界面
- 使用 HTML5 + CSS3 + JavaScript 实现
- 采用 ECharts 进行数据可视化

2. 业务逻辑层 (Controller Layer)

- 处理用户请求和业务逻辑
- 前台控制器：SiteController、GuestbookController 等
- 后台控制器：ArticleController、UserController、CategoryController 等
- 实现权限控制和访问验证

3. 数据访问层 (Model Layer)

- 封装数据库操作
- ActiveRecord 模式进行 ORM 映射
- 数据验证和业务规则
- 模型关联关系定义

4. 数据层 (Database Layer)

- MySQL 数据库
- 19 张数据表
- 存储过程和触发器（可选）

1.1.2 系统架构图

系统采用前后端分离的架构设计：

- **前台系统**：面向普通用户，提供内容浏览、互动功能
- **后台系统**：面向管理员，提供内容管理、数据统计功能
- **共享层**：通用模型、配置、组件

1.2 技术架构

1.2.1 开发框架

- **Yii2 Advanced Template**

- 前后台分离的应用模板
- 内置用户认证、RBAC 权限管理
- 强大的数据库查询构建器
- 完善的表单验证机制

1.2.2 前端技术栈

- **HTML5**: 语义化标签，提升 SEO 和可访问性
- **CSS3**: Flexbox 布局、Grid 布局、动画效果
- **JavaScript (ES6+)**: AJAX 异步请求、DOM 操作
- **Bootstrap 5**: 响应式栅格系统、UI 组件
- **ECharts 5**: 数据可视化图表库
- **Font Awesome**: 图标库

1.2.3 后端技术栈

- **PHP 8.2**: 面向对象编程、命名空间
- **Yii2 Framework**: MVC 框架、ActiveRecord ORM
- **Composer**: 依赖管理工具
- **MySQL 5.7+**: 关系型数据库

1.3 目录结构设计

```
advanced/
    backend/          # 后台应用
        assets/        # 资源包
        config/        # 配置文件
        controllers/   # 控制器
            ArticleController.php
            CategoryController.php
            UserController.php
            CommentController.php
            GuestBookController.php
        models/         # 模型
        views/          # 视图
            article/
            category/
```

```
user/
layouts/
web/          # Web 根目录
index.php

frontend/      # 前台应用
assets/        # 资源包
config/        # 配置文件
controllers/   # 控制器
    SiteController.php
    GuestbookController.php
models/         # 模型
views/         # 视图
site/
    index.php    # 首页
    news.php     # 文章列表
    view.php     # 文章详情
    about.php    # 团队展示
    login.php    # 登录
    signup.php   # 注册
guestbook/
layouts/
web/          # Web 根目录
index.php

common/        # 共享代码
config/        # 共享配置
models/        # 共享模型
    PreSysUser.php
    PreNewsArticle.php
    PreNewsCategory.php
    PreNewsComment.php
    GuestBook.php
    TeamMember.php
console/       # 控制台应用
migrations/   # 数据库迁移
data/
    install.sql # 数据库安装脚本
    crawler/    # 爬虫脚本
    team/        # 团队作业文档
    personal/   # 个人作业文档
vendor/        # 第三方依赖
```

1.4 模块划分

表 1: 系统模块划分表

| 模块名称 | 功能描述 | 负责人 |
|--------|--------------------|-----|
| 用户管理模块 | 用户注册、登录、注销、权限控制 | 彭浩然 |
| 文章管理模块 | 文章 CRUD、筛选、排序、批量操作 | 刘浩泽 |
| 分类管理模块 | 分类 CRUD、分类关联文章管理 | 刘浩泽 |
| 评论管理模块 | 评论发表、审核、删除 | 童汉鑫 |
| 留言管理模块 | 留言发表、查看、已读标记 | 彭浩然 |
| 团队展示模块 | 团队成员信息展示和管理 | 童汉鑫 |
| 数据统计模块 | 访问统计、内容分布图表 | 刘浩泽 |
| 首页展示模块 | 热门文章、最新文章、时间线 | 刘浩泽 |

2 数据库设计

2.1 数据库概述

- **数据库名称:** news_system
- **字符集:** utf8mb4
- **排序规则:** utf8mb4_unicode_ci
- **存储引擎:** InnoDB
- **数据表数量:** 19 张 (>10 张, 满足课程要求)

2.2 数据表清单

表 2: 数据库表清单

| 序号 | 表名 | 说明 | 设计者 |
|----|--------------------------|---------|-----|
| 1 | pre_sys_user | 系统用户表 | 彭浩然 |
| 2 | pre_news_category | 文章分类表 | 刘浩泽 |
| 3 | pre_news_article | 文章主表 | 刘浩泽 |
| 4 | pre_news_comment | 文章评论表 | 童汉鑫 |
| 5 | pre_guestbook | 留言板表 | 彭浩然 |
| 6 | pre_article_tag | 文章标签表 | 刘浩泽 |
| 7 | pre_article_tag_relation | 文章标签关联表 | 刘浩泽 |
| 8 | pre_article_view_log | 文章浏览日志表 | 刘浩泽 |
| 9 | pre_article_like | 文章点赞记录表 | 刘浩泽 |
| 10 | pre_article_favorite | 文章收藏表 | 刘浩泽 |
| 11 | pre_article_attachment | 文章附件表 | 刘浩泽 |
| 12 | pre_team_department | 团队部门表 | 童汉鑫 |
| 13 | pre_team_member | 团队成员表 | 童汉鑫 |
| 14 | pre_daily_stats | 每日统计表 | 刘浩泽 |
| 15 | pre_category_stats | 分类统计表 | 刘浩泽 |
| 16 | pre_link | 友情链接表 | 彭浩然 |
| 17 | pre_tag | 标签表 | 彭浩然 |
| 18 | pre_visit_log | 访问日志表 | 彭浩然 |
| 19 | pre_config | 系统配置表 | 彭浩然 |

2.3 详细表结构设计

2.3.1 用户表 (pre_sys_user)

表 3: 用户表结构

| 字段名 | 类型 | 空 | 键 | 说明 |
|---------------|--------------|-----|-----|------------------|
| uid | INT(11) | NO | PRI | 用户 ID (主键, 自增) |
| username | VARCHAR(50) | NO | UNI | 用户名 (唯一) |
| password_hash | VARCHAR(255) | NO | | 密码哈希值 |
| email | VARCHAR(100) | YES | | 邮箱地址 |
| phone | VARCHAR(20) | YES | | 手机号码 |
| avatar | VARCHAR(255) | YES | | 头像 URL |
| role | TINYINT(1) | YES | | 角色: 0 普通用户 1 管理员 |
| status | TINYINT(1) | YES | | 状态: 0 禁用 1 正常 |
| created_at | DATETIME | YES | | 创建时间 |
| updated_at | DATETIME | YES | | 更新时间 |

索引设计：

- PRIMARY KEY: uid
- UNIQUE KEY: username
- INDEX: email

2.3.2 文章分类表 (pre_news_category)

表 4: 文章分类表结构

| 字段名 | 类型 | 空 | 键 | 说明 |
|-------------|-------------|-----|-----|----------------|
| cid | INT(11) | NO | PRI | 分类 ID (主键, 自增) |
| name | VARCHAR(50) | NO | UNI | 分类名称 (唯一) |
| description | TEXT | YES | | 分类描述 |
| sort | INT(11) | YES | | 排序值 (越小越靠前) |
| status | TINYINT(1) | YES | | 状态: 0 禁用 1 启用 |
| created_at | DATETIME | YES | | 创建时间 |
| updated_at | DATETIME | YES | | 更新时间 |

预设分类数据：

1. 重大战役
2. 英雄人物
3. 历史遗址
4. 抗战文化
5. 纪念活动
6. 历史档案
7. 老兵故事
8. 国际视角

2.3.3 文章主表 (pre_news_article)

表 5: 文章主表结构

| 字段名 | 类型 | 空 | 键 | 说明 |
|-------------|--------------|-----|-----|----------------------|
| aid | INT(11) | NO | PRI | 文章 ID (主键, 自增) |
| cid | INT(11) | NO | FOR | 分类 ID (外键) |
| title | VARCHAR(200) | NO | | 文章标题 |
| summary | TEXT | YES | | 文章摘要 |
| content | LONGTEXT | NO | | 文章内容 (富文本) |
| author | VARCHAR(50) | YES | | 作者 |
| source | VARCHAR(100) | YES | | 来源 |
| cover_image | VARCHAR(255) | YES | | 封面图片 URL |
| views | INT(11) | YES | | 浏览量 (默认 0) |
| likes | INT(11) | YES | | 点赞数 (默认 0) |
| is_top | TINYINT(1) | YES | | 是否置顶: 0 否 1 是 |
| is_hot | TINYINT(1) | YES | | 是否热门: 0 否 1 是 |
| status | TINYINT(1) | YES | | 状态: 0 草稿 1 已发布 2 已下架 |
| created_at | DATETIME | YES | | 创建时间 |
| updated_at | DATETIME | YES | | 更新时间 |

索引设计:

- PRIMARY KEY: aid
- FOREIGN KEY: cid REFERENCES pre_news_category(cid)
- INDEX: cid, status, is_top, is_hot
- INDEX: created_at, views, likes

2.3.4 评论表 (pre_news_comment)

表 6: 评论表结构

| 字段名 | 类型 | 空 | 键 | 说明 |
|------------|-------------|-----|-----|-----------------------|
| comment_id | INT(11) | NO | PRI | 评论 ID (主键, 自增) |
| aid | INT(11) | NO | FOR | 文章 ID (外键) |
| uid | INT(11) | NO | FOR | 用户 ID (外键) |
| content | TEXT | NO | | 评论内容 |
| parent_id | INT(11) | YES | | 父评论 ID (用于回复) |
| status | TINYINT(1) | YES | | 状态: 0 待审核 1 已通过 2 已拒绝 |
| ip_address | VARCHAR(50) | YES | | IP 地址 |
| created_at | DATETIME | YES | | 创建时间 |
| updated_at | DATETIME | YES | | 更新时间 |

索引设计：

- PRIMARY KEY: comment_id
- FOREIGN KEY: aid REFERENCES pre_news_article(aid)
- FOREIGN KEY: uid REFERENCES pre_sys_user(uid)
- INDEX: aid, status

2.3.5 留言板表 (pre_guestbook)

表 7: 留言板表结构

| 字段名 | 类型 | 空 | 键 | 说明 |
|------------|--------------|-----|-----|-----------------|
| id | INT(11) | NO | PRI | 留言 ID (主键, 自增) |
| nickname | VARCHAR(50) | NO | | 留言者昵称 |
| email | VARCHAR(100) | YES | | 邮箱地址 |
| content | TEXT | NO | | 留言内容 |
| ip_address | VARCHAR(50) | YES | | IP 地址 |
| is_read | TINYINT(1) | YES | | 是否已读: 0 未读 1 已读 |
| created_at | DATETIME | YES | | 创建时间 |

2.3.6 统计表设计

每日统计表 (pre_daily_stats)

表 8: 每日统计表结构

| 字段名 | 类型 | 空 | 键 | 说明 |
|----------------|----------|-----|-----|----------------|
| id | INT(11) | NO | PRI | 统计 ID (主键, 自增) |
| stat_date | DATE | NO | UNI | 统计日期 (唯一) |
| total_views | INT(11) | YES | | 总浏览量 |
| total_articles | INT(11) | YES | | 总文章数 |
| total_comments | INT(11) | YES | | 总评论数 |
| created_at | DATETIME | YES | | 创建时间 |
| updated_at | DATETIME | YES | | 更新时间 |

分类统计表 (pre_category_stats)

表 9: 分类统计表结构

| 字段名 | 类型 | 空 | 键 | 说明 |
|---------------|----------|-----|-----|----------------|
| id | INT(11) | NO | PRI | 统计 ID (主键, 自增) |
| cid | INT(11) | NO | UNI | 分类 ID (唯一, 外键) |
| article_count | INT(11) | YES | | 文章数量 |
| created_at | DATETIME | YES | | 创建时间 |
| updated_at | DATETIME | YES | | 更新时间 |

2.4 数据库 E-R 图

主要实体关系:

1. **用户-文章**: 一对多关系 (一个用户可以发布多篇文章)
2. **分类-文章**: 一对多关系 (一个分类包含多篇文章)
3. **文章-评论**: 一对多关系 (一篇文章可以有多条评论)
4. **用户-评论**: 一对多关系 (一个用户可以发表多条评论)
5. **文章-标签**: 多对多关系 (通过 pre_article_tag_relation 关联)
6. **分类-统计**: 一对一关系 (每个分类对应一条统计数据)

2.5 数据完整性约束

2.5.1 实体完整性

- 所有表都有主键约束
- 主键自增, 确保唯一性

2.5.2 参照完整性

- pre_news_article.cid 外键引用 pre_news_category.cid
- pre_news_comment.aid 外键引用 pre_news_article.aid
- pre_news_comment.uid 外键引用 pre_sys_user.uid
- 设置级联操作: ON DELETE CASCADE, ON UPDATE CASCADE

2.5.3 域完整性

- 必填字段设置 NOT NULL 约束
- 枚举类型字段使用 TINYINT 存储
- 时间字段默认值为 CURRENT_TIMESTAMP
- 计数字段 (views, likes) 默认值为 0

2.5.4 用户定义完整性

- username 唯一约束
- email 格式验证（应用层）
- password 强度验证（应用层）

3 接口设计

3.1 前台接口设计

3.1.1 首页相关接口

表 10: 首页接口

| URL | 方法 | 功能说明 |
|-------------|-----|------------------------|
| /site/index | GET | 获取首页数据（热门文章、最新文章、统计数据） |

响应数据示例：

```
[
  'hotArticles' => [
    ['aid' => 1, 'title' => '...', 'views' => 5000, ...],
    ...
  ],
  'latestArticles' => [...],
  'categories' => [...],
  'dailyStats' => [...],
  'categoryStats' => [...]
]
```

3.1.2 文章相关接口

表 11: 文章接口

| URL | 方法 | 功能说明 |
|---------------------|------|----------------|
| /site/news | GET | 获取文章列表，支持筛选和排序 |
| /site/view?id={aid} | GET | 获取文章详情 |
| /site/like?id={aid} | POST | 文章点赞（AJAX） |

/site/news 请求参数：

| 参数名 | 类型 | 说明 |
|---------|--------|--|
| cid | int | 分类 ID (可选) |
| keyword | string | 搜索关键词 (可选) |
| sort | string | 排序方式: created_at/views/likes (默认 created_at) |
| page | int | 页码 (默认 1) |

/site/like 响应数据:

```
{
  "success": true,
  "message": "点赞成功",
  "likes": 123
}
```

3.1.3 用户认证接口

表 12: 用户认证接口

| URL | 方法 | 功能说明 |
|--------------|------|--------|
| /site/signup | GET | 显示注册页面 |
| /site/signup | POST | 提交注册表单 |
| /site/login | GET | 显示登录页面 |
| /site/login | POST | 提交登录表单 |
| /site/logout | POST | 用户登出 |

注册表单数据:

```
[
  'username' => 'string',
  'email' => 'string',
  'password' => 'string',
  'password_repeat' => 'string'
]
```

3.1.4 评论接口

表 13: 评论接口

| URL | 方法 | 功能说明 |
|---------------|------|------------|
| /site/comment | POST | 提交评论 (需登录) |

评论表单数据:

```
[
  'aid' => 'int',
  'content' => 'string',
```

```

'parent_id' => 'int' // 可选，用于回复
]

```

3.1.5 留言板接口

表 14: 留言板接口

| URL | 方法 | 功能说明 |
|-------------------|------|--------------|
| /guestbook/index | GET | 显示留言板页面和留言列表 |
| /guestbook/create | POST | 提交留言 |

3.2 后台接口设计

3.2.1 文章管理接口

表 15: 后台文章管理接口

| URL | 方法 | 功能说明 |
|------------------------------|----------|------------------|
| /article/index | GET | 文章列表（支持搜索、筛选、排序） |
| /article/create | GET/POST | 创建文章 |
| /article/update?id={aid} | GET/POST | 编辑文章 |
| /article/delete?id={aid} | POST | 删除文章 |
| /article/view?id={aid} | GET | 查看文章详情 |
| /article/batch-update | POST | 批量操作（置顶/热门/删除） |
| /article/toggle-top?id={aid} | POST | 切换置顶状态 |
| /article/toggle-hot?id={aid} | POST | 切换热门状态 |

批量操作参数：

```

[
  'ids' => [1, 2, 3],
  'action' => 'top' // 或 'untop', 'hot', 'unhot', 'delete'
]

```

3.2.2 分类管理接口

表 16: 后台分类管理接口

| URL | 方法 | 功能说明 |
|---------------------------|----------|--------|
| /category/index | GET | 分类列表 |
| /category/create | GET/POST | 创建分类 |
| /category/update?id={cid} | GET/POST | 编辑分类 |
| /category/delete?id={cid} | POST | 删除分类 |
| /category/view?id={cid} | GET | 查看分类详情 |

3.2.3 用户管理接口

表 17: 后台用户管理接口

| URL | 方法 | 功能说明 |
|------------------------|----------|--------|
| /user/index | GET | 用户列表 |
| /user/create | GET/POST | 创建用户 |
| /user/update?uid={uid} | GET/POST | 编辑用户 |
| /user/delete?uid={uid} | POST | 删除用户 |
| /user/view?uid={uid} | GET | 查看用户详情 |

3.2.4 评论管理接口

表 18: 后台评论管理接口

| URL | 方法 | 功能说明 |
|----------------------------------|------|------|
| /comment/index | GET | 评论列表 |
| /comment/approve?id={comment_id} | POST | 通过评论 |
| /comment/reject?id={comment_id} | POST | 拒绝评论 |
| /comment/delete?id={comment_id} | POST | 删除评论 |
| /comment/batch-approve | POST | 批量通过 |

3.2.5 留言管理接口

表 19: 后台留言管理接口

| URL | 方法 | 功能说明 |
|-------------------------------|------|--------|
| /guest-book/index | GET | 留言列表 |
| /guest-book/view?id={id} | GET | 查看留言详情 |
| /guest-book/delete?id={id} | POST | 删除留言 |
| /guest-book/mark-read?id={id} | POST | 标记已读 |

3.3 安全设计

3.3.1 CSRF 防护

- 所有 POST 请求需要包含 CSRF Token
- Yii2 自动生成和验证 CSRF Token
- Token 存储在 Meta 标签和表单隐藏字段中

3.3.2 XSS 防护

- 输出时使用 Html::encode() 转义

- 富文本编辑器使用 HtmlPurifier 过滤
- 严格验证用户输入

3.3.3 SQL 注入防护

- 使用参数化查询 (Prepared Statements)
- 通过 ActiveRecord 进行数据库操作
- 避免直接拼接 SQL 语句

3.3.4 权限控制

- 后台操作需要管理员权限
- 使用 Yii2 的 access control filter
- Session 管理，防止会话劫持

4 界面设计

4.1 前台界面设计

4.1.1 设计风格

- 主题色：深红色 (#8B0000)、金色 (#FFD700)
- 辅助色：深褐色 (#4a0000)、米色 (#F5F5DC)
- 字体：微软雅黑、思源黑体
- 布局：响应式栅格布局，适配 PC/平板/手机
- 风格：庄重、简洁、现代化

4.1.2 导航栏设计

- Logo：网站名称 + 图标
- 导航菜单：首页、新闻资讯、团队介绍、留言板
- 用户菜单：未登录显示“登录/注册”，已登录显示用户名和“退出”
- 固定在页面顶部，滚动时保持可见

4.1.3 首页布局

1. Hero Banner

- 全屏宽度，深红渐变背景
- 主标题：“铭记历史珍爱和平”
- 副标题：“纪念中国人民抗日战争暨世界反法西斯战争胜利 80 周年”
- 年份标识：1945-2025

2. 纪念标语

- 居中显示：“勿忘国耻 · 振兴中华 · 缅怀先烈 · 开创未来”
- 金色五角星装饰

3. 分类导航

- 横向滚动的标签栏
- 显示所有文章分类
- 点击跳转到对应分类的文章列表

4. 热门专题

- 卡片式布局，3 列网格（响应式）
- 每个卡片显示封面图、标题、摘要、分类、统计数据
- Hover 效果：阴影加深、轻微放大

5. 最新发布

- 布局同热门专题

6. 抗战历史时间线

- 垂直时间轴
- 每个节点显示日期、标题、描述
- 使用深红色圆点标记
- 最后节点（抗战胜利）金色高亮

7. 数据统计图表

- 两列布局
- 左侧：近 30 天访问趋势折线图
- 右侧：分类内容分布饼图
- 使用 ECharts 渲染

4.1.4 文章列表页设计

- 顶部横幅：背景图 + 标题
- 筛选栏：分类下拉、排序选项
- 文章卡片：3列网格（响应式）
- 底部分页导航

4.1.5 文章详情页设计

- **顶部：**阅读进度条（固定在页面最顶部）
- **文章头部：**标题、发布时间、作者、分类、统计数据
- **文章正文：**富文本内容，段落间距适中
- **浮动工具栏（右侧）：**
 - 点赞按钮（心形图标）
 - 分享按钮（分享图标）
 - 返回顶部按钮（箭头图标）
- **相关文章推荐：**底部横向卡片列表
- **评论区：**
 - 评论输入框（需登录）
 - 评论列表（时间倒序）
 - 每条评论显示：用户头像、昵称、时间、内容

4.1.6 团队展示页设计

- **团队简介：**项目介绍、开发目标
- **成员展示：**卡片式布局，4列网格
- 每个卡片显示：照片、姓名、角色、负责模块
- **分工表格：**详细的分工和工作量说明

4.1.7 留言板页设计

- **留言表单：**昵称、邮箱（可选）、留言内容
- **留言列表：**卡片式布局
- 每条留言显示：昵称、时间、内容
- 支持分页显示

4.2 后台界面设计

4.2.1 设计风格

- **主题色**: 蓝色 (Bootstrap 默认主题)
- **布局**: 左侧菜单 + 右侧内容区
- **风格**: 简洁、高效、功能优先

4.2.2 菜单结构

1. 首页 (仪表盘)
2. 内容管理
 - 文章管理
 - 分类管理
3. 互动管理
 - 评论管理
 - 留言管理
4. 系统管理
 - 用户管理
 - 团队管理

4.2.3 列表页设计

- **顶部**: 页面标题 + 操作按钮 (如“添加”)
- **筛选栏**: 搜索框、下拉筛选、排序选项
- **数据表格**: 使用 Yii2 GridView 组件
 - 序号列
 - 数据列 (可排序)
 - 操作列 (查看、编辑、删除)
- **批量操作**: 复选框 + 批量操作按钮
- **分页**: 底部分页导航

4.2.4 表单页设计

- **顶部:** 页面标题 + 面包屑导航
- **表单区域:** 使用 Yii2 ActiveForm 组件
 - 标签在输入框上方
 - 必填字段标记红色星号
 - 实时验证提示
- **底部按钮:** 提交、取消

5 安全性设计

5.1 认证与授权

5.1.1 用户认证

- 使用 Yii2 的 User 组件管理用户认证
- 密码使用 bcrypt 算法加密 (Yii2 Security 组件)
- Session 超时时间: 2 小时
- 支持“记住我”功能 (30 天)

5.1.2 权限控制

- 前台:
 - 游客: 可浏览所有公开内容
 - 注册用户: 可点赞、评论
- 后台:
 - 仅管理员可访问
 - 使用 access control filter 拦截未授权访问

5.2 数据安全

5.2.1 输入验证

- 前端: HTML5 表单验证、JavaScript 验证
- 后端: Yii2 Model rules 验证 (必填、格式、长度等)
- 富文本内容: HtmlPurifier 过滤

5.2.2 SQL 注入防护

- 使用 ActiveRecord 进行数据库操作
- 动态条件使用参数绑定
- 禁止拼接原始 SQL

5.2.3 XSS 防护

- 输出时使用 Html::encode()
- 富文本内容使用 HtmlPurifier 过滤
- CSP (Content Security Policy) 头部设置

5.2.4 CSRF 防护

- Yii2 自动生成 CSRF Token
- 所有 POST 请求验证 Token
- AJAX 请求在 Header 中携带 Token

5.3 会话安全

- Session ID 使用 httponly 和 secure 标志
- Session 超时自动注销
- 登录后重新生成 Session ID
- 防止 Session 固定攻击

6 性能优化设计

6.1 数据库优化

- **索引优化:** 为常用查询字段建立索引
- **查询优化:**
 - 使用预加载 (eager loading) 减少 N+1 查询
 - 分页查询减少数据量
 - 避免 SELECT *，只查询需要的字段
- **缓存优化:**
 - 使用 Yii2 的数据缓存 (文件缓存)
 - 分类列表缓存 (1 小时)
 - 统计数据缓存 (30 分钟)

6.2 前端优化

- **资源压缩:** CSS/JS 文件压缩合并
- **图片优化:**
 - 使用 WebP 格式
 - 图片懒加载
 - 响应式图片
- **CDN 加速:**
 - ECharts、Font Awesome 使用 CDN
 - 静态资源 CDN 分发
- **浏览器缓存:** 设置合理的 Cache-Control 头

6.3 代码优化

- 避免重复代码，提取公共方法
- 使用异步加载（AJAX）提升交互体验
- 合理使用 Yii2 的延迟加载

7 测试设计

7.1 测试策略

1. **单元测试:** 测试 Model 层的业务逻辑
2. **功能测试:** 测试 Controller 层的功能实现
3. **集成测试:** 测试前后端集成、数据库交互
4. **安全测试:** SQL 注入、XSS、CSRF 测试
5. **性能测试:** 并发访问、响应时间测试
6. **兼容性测试:** 多浏览器、多设备测试

7.2 测试用例

7.2.1 用户认证测试

表 20: 用户认证测试用例

| 测试项 | 测试步骤 | 预期结果 |
|-------|-------------|-------------|
| 用户注册 | 填写有效信息并提交 | 注册成功, 自动登录 |
| 重复用户名 | 使用已存在的用户名注册 | 提示用户名已存在 |
| 密码强度 | 输入弱密码 | 提示密码强度不足 |
| 用户登录 | 输入正确的用户名和密码 | 登录成功, 跳转到首页 |
| 错误密码 | 输入错误的密码 | 提示用户名或密码错误 |

7.2.2 文章管理测试

表 21: 文章管理测试用例

| 测试项 | 测试步骤 | 预期结果 |
|------|------------|----------|
| 创建文章 | 填写完整信息并保存 | 文章创建成功 |
| 必填验证 | 不填写标题直接提交 | 提示标题不能为空 |
| 编辑文章 | 修改文章内容并保存 | 更新成功 |
| 删除文章 | 点击删除并确认 | 文章删除成功 |
| 批量置顶 | 选择多篇文章批量置顶 | 选中文章置顶成功 |

7.2.3 安全测试

表 22: 安全测试用例

| 测试项 | 测试方法 | 预期结果 |
|---------|----------------------|-------------|
| SQL 注入 | 在输入框中输入 SQL 语句 | 被转义或过滤, 不执行 |
| XSS 攻击 | 在评论中输入脚本代码 | 被转义, 不执行 |
| CSRF 攻击 | 伪造 POST 请求 (无 Token) | 请求被拒绝 |
| 未授权访问 | 游客访问后台管理页面 | 跳转到登录页 |

8 部署设计

8.1 部署环境

- **开发环境:** Windows + XAMPP
- **测试环境:** Linux + Apache + MySQL
- **生产环境:** Linux + Nginx + MySQL (或 Apache)

8.2 部署步骤

1. 环境准备

- 安装 PHP 8.0+
- 安装 MySQL 5.7+
- 安装 Composer
- 配置 Web 服务器 (Apache/Nginx)

2. 代码部署

- 从 Git 仓库拉取代码
- 运行 composer install 安装依赖
- 配置数据库连接 (common/config/main-local.php)

3. 数据库初始化

- 创建数据库: CREATE DATABASE news_system
- 导入表结构: mysql < data/install.sql
- 导入测试数据 (可选)

4. 权限设置

- 设置 runtime 目录可写权限
- 设置 assets 目录可写权限
- 设置 uploads 目录可写权限 (如有)

5. Web 服务器配置

- 配置虚拟主机
- 设置 Document Root 为 frontend/web 和 backend/web
- 配置 URL 重写规则

6. 验证部署

- 访问前台首页
- 访问后台登录页
- 测试主要功能

8.3 配置说明

8.3.1 数据库配置

```
// common/config/main-local.php
return [
    'components' => [
        'db' => [
            'class' => 'yii\db\Connection',
            'dsn' => 'mysql:host=localhost;dbname=news_system',
            'username' => 'root',
            'password' => '',
            'charset' => 'utf8mb4',
        ],
    ],
];
```

8.3.2 URL 配置

```
// frontend/config/main.php 或 backend/config/main.php
'components' => [
    'urlManager' => [
        'enablePrettyUrl' => true,
        'showScriptName' => false,
        'rules' => [],
    ],
],
```

9 维护设计

9.1 日志管理

- **应用日志:** runtime/logs/app.log
- **错误日志:** runtime/logs/error.log
- **访问日志:** Web 服务器日志
- 日志级别: error, warning, info, trace
- 日志轮转: 按日期或文件大小

9.2 数据备份

- **数据库备份:**
 - 每日自动备份
 - 备份文件命名: news_system_YYYYMMDD.sql
 - 保留最近 30 天的备份
- **代码备份:** Git 版本控制
- **文件备份:** 定期备份 uploads 目录

9.3 监控告警

- 服务器性能监控 (CPU、内存、磁盘)
- 应用错误监控 (500 错误、异常日志)
- 数据库性能监控 (慢查询、连接数)

10 总结

本设计文档详细描述了“抗战胜利 80 周年纪念网站”的系统设计方案，包括：

1. **系统架构设计**: 三层架构 + MVC 模式，前后台分离
2. **数据库设计**: 19 张表，满足课程要求，ER 图清晰
3. **接口设计**: RESTful 风格，前后台接口完整
4. **界面设计**: 抗战主题风格，响应式布局
5. **安全性设计**: 认证授权、CSRF/XSS/SQL 注入防护
6. **性能优化设计**: 数据库、前端、代码三层优化
7. **测试设计**: 单元、功能、集成、安全、性能测试
8. **部署设计**: 详细部署步骤和配置说明
9. **维护设计**: 日志、备份、监控方案

本设计方案技术先进、架构合理、安全可靠，为项目实施提供了完整的技术指导。