

课程评分系统微型网站项目报告

1 项目简介

1.1 项目技术栈

1. 前端:

- Vue: 由于本项目为小型网站,而Vue.js作为一个轻量级,响应式并且支持单文件组件的前端框架,是最合适的选择。
 - Vite: Vite是Vue官方推出的新一代前端构建工具
 - Vue Router: Vue Router是Vue官方的路由管理插件
 - Pinia: Pinia是Vue官方的状态管理插件
 - vitest: vitest是Vue官方推出的单元测试框架
 - Cypress: Cypress是一款开源的端到端测试框架
- Sass: Sass是一种CSS预处理语言,它扩展了CSS语言,增加了变量、嵌套、混合、导入等功能,使CSS更加强大和方便维护。

2. 后端:

- Python: 由于Python是一门高级语言,并且有丰富的第三方库,可以快速开发后端服务。
- Flask: Flask是Python的一个轻量级Web框架,它可以快速开发Web应用。
- PyMySQL: PyMySQL是Python的一个MySQL数据库驱动。

3. 数据库:

- MySQL: MySQL是一款开源的关系型数据库管理系统,它是最流行的数据库管理系统之一。

2 系统架构

1. 前端:

- 页面:
 - 主页: 显示课程列表
 - 登录/注册: 允许用户登录/注册
 - 个人中心: 显示用户信息, 允许用户修改个人信息

3 项目结构

```
{
  "ccs": {
    "server": {
      // 后端文件(backend)
      "api": {
        "_types.py": "类型声明文件",
        "sqlTools.bk": "备用源文件",
        "sqlTools.pyd": "py动态链接库",
        "sqlTools.pyi": "类型提示文件",
        "utils.py": "工具",
        "views.py": "flask蓝图文件",
        "README.md": "库文档"
      },
      "static": {
        "css": "...",
        "img": "...",
        "js": "...",
        "font": "..."
      },
      "template": {
        "index.html": "主页模板"
      },
      "app.py": "后端入口文件",
      "sqlTools.py": "数据库连接工具"
    },
    "client": {
      // 前端文件(frontend)
      // vue标准目录结构,省略处不言自明
      "cypress": "e2e测试用例",
      "public": "...",
      "src": {
        "assets": "...",
        "components": {
          "__tests__": "vitest单元测试",

```

```
    "genHeader.vue": "头部组件",
    "home.vue": "主页组件",
    "login.vue": "登录组件",
    "profile.vue": "个人中心组件"
  },
  "router": "...",
  "store": "...",
  "mock": {
    // mock模拟数据
    "index.ts": "...",
    "api.ts": "API接口模拟"
  },
  "App.vue": "...",
  "main.ts": "..."
},
".gitignore": "...",
".prettierrc.json": "prettier代码格式化配置",
"cypress.config.js": "...",
"env.d.ts": "...",
"eslint.config.js": "eslint代码检查配置",
"index.html": "...",
"openapi.json": "API接口文档",
"package.json": "...",
"pnpm-lock.yaml": "pnpm依赖管理文件",
"tsconfig.app.json": "...",
"tsconfig.json": "...",
"tsconfig.node.json": "...",
"tsconfig.vitest.json": "vitest单元测试配置",
"typedoc.json": "typescript文档生成配置",
"vite.config.ts": "...",
"vitest.config.ts": "vitest单元测试配置"
},
"utils": "TS工具包(confunc)"
"README.md": "项目说明文档",
"LICENSE": "项目许可证",
"CCS.spec": "pyinstaller构建文件",
```

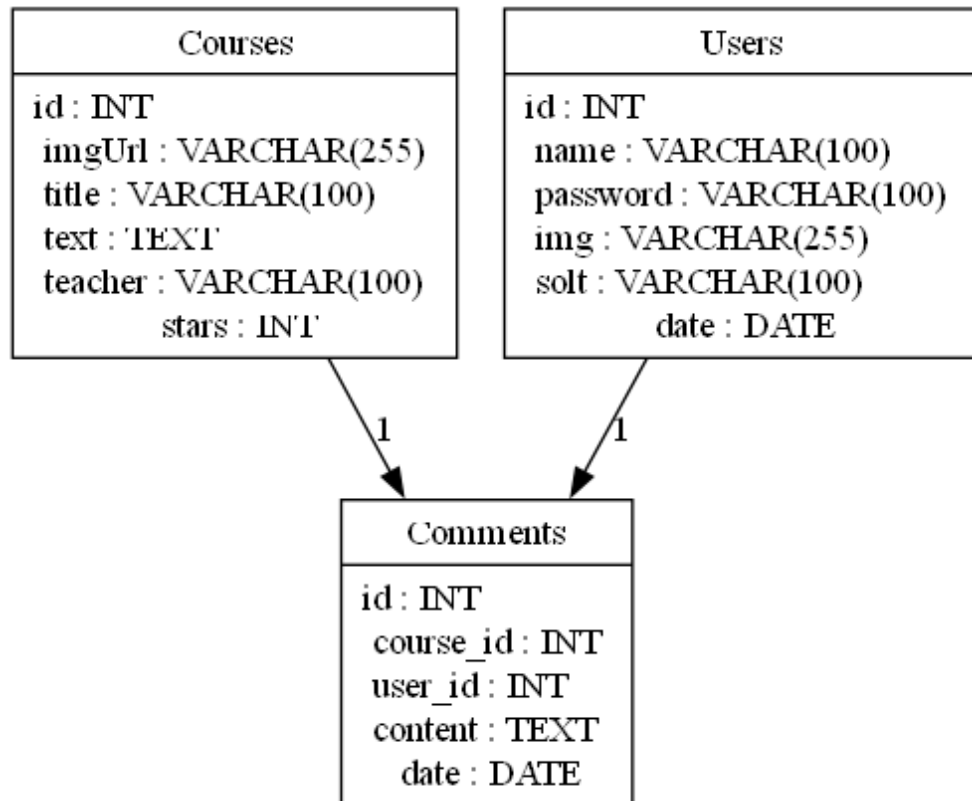
```

        "ccs.sql": "mysql数据库文件",
        "ccs.dot": "数据库ER图源码"
    }
}

```

4 数据库配置

4.1 基本数据库结构



4.2 MySQL表示

```

CREATE TABLE Courses (
    id INT PRIMARY KEY AUTO_INCREMENT,
    imgUrl VARCHAR(255),
    title VARCHAR(100) NOT NULL,
    text TEXT,
    teacher VARCHAR(100) NOT NULL,
    stars INT DEFAULT 0
);

```

```

CREATE TABLE Users (

```

```

    id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100) NOT NULL,
    password VARCHAR(100) NOT NULL,
    img VARCHAR(255),
    salt VARCHAR(100) NOT NULL,
    date DATE NOT NULL
);

CREATE TABLE Comments (
    id INT PRIMARY KEY AUTO_INCREMENT,
    course_id INT NOT NULL,
    user_id INT NOT NULL,
    content TEXT,
    date DATE NOT NULL,
    FOREIGN KEY (course_id) REFERENCES Courses(id),
    FOREIGN KEY (user_id) REFERENCES Users(id)
);

```

5 项目部署

5.1 前端开发

① Note

1. 确保您的计算机上具有Nodejs且版本不低于20.12,并且配有对应包管理器
npm或pnpm

1. 进入工作路径

```
cd /path/to/your/CCS/client
```

2. 自动安装依赖

- npm

```
npm install
```

- pnpm

```
pnpm install
```

3. 运行项目

- npm

```
npm run dev
```

- pnpm

```
pnpm run dev
```

4. 构建项目

- npm

```
npm run build
```

- pnpm

```
pnpm run build
```

5.2 后端部署

5.2.1 导入MySQL数据

```
mysql -u 用户名 -p < ccs.sql
```

5.2.2 启动服务(开发环境)

📘 Note

1. 确保您的计算机上具有**Python**解释器且版本不低于3.11,并且配有对应包管理器
2. 确保您安装有Flask和PyMySQL包

```
pip install flask pymysql tabulate pandas
```

1. 进入项目目录

```
cd path/to/project/CCS/server
```

🚨 Important

在文件 `/CCS/server/api/views.py` 中修改字段 `PASSWORD` 为您的MySQL用户密码

2. 启动服务

```
python app.py
```

3. 打开浏览器访问 <http://localhost:5000>