

# JavaScript 在线编程平台 - 功能说明文档

组员：李志帆、姚嘉峻、甄泽榕

## 1. 项目简介

这是一个基于 Web 的在线编程教学平台，专为 JavaScript、HTML 和 CSS 教学设计。系统采用\*\*双角色（教师/学生）\*\*设计，提供沉浸式的代码编辑体验、完善的作业提交与批改流程，以及本地化的数据存储方案。界面采用现代化的 Cyberpunk 风格，兼具科技感与实用性。

## 2. 技术栈

- 前端核心:** 原生 HTML5, CSS3, JavaScript (ES6+)
- 代码编辑器:** Monaco Editor (VS Code 的核心编辑器组件)
- 数据存储:** IndexedDB (浏览器本地数据库，无需后端服务器)
- 运行环境:**
  - JavaScript: 浏览器沙箱 ( new Function )
  - Web 项目: iframe 隔离环境 (支持 HTML/CSS/JS 混合预览)
- 样式风格:** 自定义 CSS (Cyberpunk 主题) + VS Code Dark Theme

## 3. 系统架构

系统完全运行在客户端浏览器中，通过 IndexedDB 实现数据的持久化存储。主要模块包括：

- File Manager:** 虚拟文件系统，支持文件/文件夹的增删改查及上传。
- Compiler:** 负责代码的实时编译、运行及控制台日志 (Log/Error/Warn) 捕获。
- Database:** 封装 IndexedDB 操作，管理用户、作业、提交记录和代码库。
- Editor:** 集成 Monaco Editor，提供多标签页、代码高亮、智能提示等功能。

# 4. 核心功能详解

## 4.1 公共功能

### • 用户认证:

- 支持教师和学生两种角色的注册与登录。
- 登录/注册界面采用动态模态框设计。

### • 代码编辑器:

- **多标签页管理:** 支持同时打开编辑多个文件。
- **文件树操作:** 支持新建文件/文件夹、重命名、删除。
- **本地上传:** 支持拖拽或点击上传本地代码文件/文件夹。
- **实时预览:** 提供独立的 Preview 面板查看 HTML/CSS 效果。
- **控制台:** 模拟浏览器 Console，实时显示 `console.log` 输出及报错信息。

## 4.2 学生端功能 ( StudentApp )

学生端专注于代码练习与作业提交。

### 1. 编辑器 (Editor):

- 提供完整的代码编写环境。
- 支持创建个人项目并保存。

### 2. 我的作业 (Homework):

- 查看教师发布的作业列表（包含标题、描述、截止时间、附件）。
- **开始作业:** 自动创建作业相关的文件模板。
- **提交作业:** 将当前编辑器中的**整个项目结构**（所有文件和文件夹）打包提交。
- 支持在截止日期前重新提交。

### 3. 成绩查询 (Grades):

- 查看已批改作业的得分、评级（S/A/B/C）及教师评语。
- 提供仪表盘统计（平均分、最高分、及格率）。

### 4. 代码库 (Code Library):

- 管理个人代码片段或项目。
- 支持从本地导入项目到代码库。
- 支持将代码库中的项目加载回编辑器进行修改。

## 4.3 教师端功能 ( TeacherApp )

教师端专注于教学管理与资源建设。

### 1. 仪表盘 (Dashboard):

- 实时概览：学生总数、作业数、待批改作业数。
- 待办事项：列出待批改的作业快捷入口。
- 最新动态：显示学生最新的提交记录。

## 2. 学生管理 (Students):

- **列表管理**: 查看、搜索、删除学生。
- **手动添加**: 单个添加学生账号。
- **批量导入**: 支持 CSV 文件批量导入学生名单 (格式: `username,password,name,studentId`)。
- **导出名单**: 将学生数据导出为 CSV。

## 3. 作业管理 (Homework):

- **发布作业**: 设置标题、描述、截止时间，支持上传多个附件文件。
- **查看提交**: 监控全班提交进度，查看提交列表。

## 4. 批改作业 (Grading):

- **在线批改**: 直接在浏览器中预览学生提交的代码项目 (自动加载学生的文件结构)。
- **评分反馈**: 录入分数 (0-100) 和详细评语。
- 支持重新批改。

## 5. 代码库与模板 (Code Library & Templates):

- **公共代码库**: 创建可供演示的示例项目。
- **自定义模板**: 在编辑器中将当前代码保存为模板 (如 HTML5 骨架、D3 图表模板等)，供后续快速复用。

## 6. 数据备份 (Data Management):

- **导出数据**: 将整个数据库 (用户、作业、提交、代码库) 导出为 JSON 备份文件。
- **恢复数据**: 从 JSON 文件恢复系统数据 (用于跨设备迁移或数据回滚)。

# 5. 数据结构说明

主要数据存储在 IndexedDB 的以下 Store 中：

- `teachers / students` : 用户信息。
- `homework` : 作业详情及附件数据 (Base64编码)。
- `submissions` : 学生提交记录，包含代码内容、项目结构、分数和评语。
- `codeLibrary` : 保存的代码片段或项目。
- `projects` : 用户在编辑器中当前保存的工作区状态 (即“自动保存”功能)。
- `templates` : 自定义代码模板。

# 6. 使用流程示例

1. **初始化**: 教师登录，导入学生名单，发布“第一次作业”。

2. **学生操作**: 学生登录，在“我的作业”中点击“开始作业”，在编辑器中完成代码编写，点击“提交作业”。
3. **批改**: 教师在仪表盘看到待批改提醒，点击进入，预览学生代码，打分并写评语。
4. **反馈**: 学生在“成绩查询”中看到分数和评语。