



Collegra 开发文档

李巴蒂

23336104

libd5@mail2.sysu.edu.cn

卢嘉聪

23336167

2716559553@qq.com

张珈瑞

23336303

2990383675@qq.com

雷颜玮

23336103

leiyanwei2005@163.com

刘靖海

23336154

3093952785@qq.com

1 引言

随着信息技术的飞速发展，软件在大学生日常生活中的作用愈发重要。无论是在课程管理、任务规划，还是实习求职、生活记录等方面，学生们都越来越倾向于借助各种软件工具来提高效率。然而，当前大多数相关应用往往仅聚焦于某一功能点，例如课程表、待办事项或简历制作等。功能分散、界面不统一、数据难以联通的问题，成为用户体验的瓶颈，也提高了操作成本。

为此，我们设计并实现了 Collegra ——一款面向大学生的一体化轻量级辅助软件。Collegra 旨在整合学习、生活与职业发展的多个常用功能，通过模块化设计和简洁界面，为用户提供统一、便捷的操作体验。与传统工具相比，Collegra 不仅具备更高的任务关联性和内容联动性，也便于用户根据自身需求进行个性化扩展。

本项目采用敏捷开发方法进行需求分析与系统迭代，文档将详细介绍 Collegra 的开发过程，包括软件需求分析（第2节）、核心功能设计（第3节）、测试流程（第4节）以及成员分工（第5节）等内容。我们希望该软件能为大学生用户带来实质性便利，并为未来的校园工具集成提供一个可行的探索路径。

Collegra 开源于 <https://github.com/Hearth-Li/Collegra>

2 需求说明

由于 Collegra 集成了多个不同功能的子模块，我们将首先描述软件整体层面的共性需求（2.1节），然后再分别说明各子模块的具体功能要求（2.2至2.8节）。

2.1 整体需求

Collegra 的核心目标是支持大学生在学习、生活和职业发展等方面的多样化需求。因此，各个子模块的功能设计应始终围绕该主题展开。在整体设计层面，系统需满足以下共性要求：

- **功能一致性**：各子模块的功能应聚焦于促进大学生的学习效率、生活规划与职业能力发展，避免功能冗余或偏离主题。

- **语言适配**：考虑到英语在国际学术交流和职业发展中的重要地位，以及大学生在英语学习、出国交流中的实际需求，系统应支持中英文双语界面。用户可根据偏好自由切换语言，所有页面内容均需具备完整的中英版本。
- **界面统一性**：系统应提供统一、协调的图形用户界面（GUI），保证各子模块之间在风格、布局、交互逻辑上的一致性，以提升整体用户体验。

Collegra 当前共实现了七个功能模块，分别为：学习路径推荐器、简历生成器、待办清单、课程表管理、AI 助手、资源归纳库和知识卡片。以下将分别介绍每个子模块的具体需求。

2.2 学习路径推荐

在信息爆炸的互联网时代，大学生虽然可以便捷地获取各类学习资源，但同时也面临信息冗杂、质量参差不齐的挑战。本模块旨在为学生提供围绕特定方向（如机器学习、前端开发、科研写作等）的高质量学习路径，帮助用户高效构建知识体系并有序推进学习过程。模块的功能需求如下：

- **学习路径展示**：模块通过一组静态页面展示精选的学习路径，每条路径包含多个按顺序组织的学习节点。每个节点配有推荐资源链接（如课程、文章、视频等），并附带简要说明，便于用户理解内容层次与进阶关系。
- **交互功能支持**：为提升用户体验，系统提供节点的“完成”状态标记功能，并可自动记录与更新学习路径的整体完成进度。用户可以随时查看当前学习进度，便于持续管理学习节奏。

2.3 简历生成器

在升学、求职等重要阶段，简历是展示个人背景和能力的核心材料。然而，许多大学生虽然具备丰富的经历，却缺乏将其组织成规范、美观简历的能力。本模块旨在提供一个结构化、用户友好的信息输入界面，并自动生成格式统一、内容完整的个人简历。具体功能需求如下：

- **用户信息录入**：系统应支持用户输入并管理个人基本信息（如姓名、联系方式）、教育经历、科研或项目经历、论文发表、技能特长等多个简历常见模块。各类信息应支持重复添加、编辑与删除，保证灵活性。
- **编译方式选择**：简历输出基于 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 模板生成。考虑到用户环境的差异，系统应允许用户选择两种导出方式：一是生成 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 源码，由用户复制至 Overleaf 等在线平台自行编译；二是提供本地 PDF 编译选项，并配套用户友好的环境配置引导教程，降低使用门槛。

2.4 待办清单

本模块集任务管理与智能决策辅助于一体，旨在帮助大学生更高效地规划、管理和完成个人事务。应实现以下功能：

- **任务管理 (CRUD)**：用户可以创建新的待办任务，并对现有任务执行编辑、删除和搜索等操作。每个任务应至少包含标题、截止时间、优先级、状态（待完成/已完成）等基本字段。
- **优先级分类管理**：系统应支持为任务设置优先级（高、中、低三档），并在界面中进行视觉标识，便于用户识别和排序。

- **智能选择辅助**：当存在多个优先级相同的任务时，用户可能难以确定处理顺序。系统应提供基于“轮盘赌”算法的推荐机制，为用户智能推荐一个任务，辅助决策的同时也增强交互的趣味性。
- **数据持久化存储**：任务数据应保存在本地文件或数据库中，确保用户在关闭和重新打开软件后仍可访问原有任务列表，保障使用的连续性和稳定性。

2.5 课程表管理

2.6 AI 助手

2.7 资源归纳库

2.8 知识卡片

3 详细设计与实现

类似于软件需求的分析（第2节），本节对系统实现进行说明，亦采用“总体基本实现”与“各子模块实现”相结合的方式描述。

3.1 基础实现

为实现统一风格的图形用户界面（GUI），我们基于 HTML5、Tailwind CSS 和 JavaScript 设计了基础网页模板 `base.html`。该模板通过 Jinja2 模板继承支持子模块页面扩展，提供语言切换、统一视觉风格、导航系统和交互式聊天侧边栏，满足功能需求（见第 2.1 节）。

核心功能包括：

- **语言切换**：支持中英文切换，状态持久化。
- **统一视觉风格**：动态背景和一致排版。
- **导航系统**：下拉菜单支持模块跳转。
- **聊天侧边栏**：提供 AI 交互功能。

3.1.1 实现概述

语言切换 通过顶部导航栏的下拉菜单，用户可切换中英文。JavaScript 函数 `toggleLanguage()` 将语言状态存入 `localStorage`，通过 CSS 类动态显示对应语言内容。核心代码如下：

```
<select id="languageSelect" onchange="toggleLanguage()">
  <option value="zh">中文</option>
  <option value="en">English</option>
</select>
<script>
function toggleLanguage() {
  const newLang = document.getElementById('languageSelect').value;
  localStorage.setItem('language', newLang);
  document.querySelectorAll('.lang-en').forEach(el =>
    el.classList.toggle('hidden', newLang !== 'en'));
  document.querySelectorAll('.lang-zh').forEach(el =>
```

```
        el.classList.toggle('hidden', newLang !== 'zh'));
        location.reload();
    }
</script>
```

统一视觉风格 页面采用 Tailwind CSS 和自定义 CSS，背景使用动态渐变动画，英文和中文分别使用 Playfair Display 和 Noto Serif SC 字体。顶部导航栏为半透明白色，适配移动端。样式示例：

```
body {
    font-family: 'Inter', sans-serif;
    background: linear-gradient(135deg, #d6e4ff, #c3d4fe, #e0d2ff, #f4d2e0, #e0f0ff);
    animation: gradientShift 15s ease-in-out infinite;
}
```

导航系统 顶部下拉菜单支持子模块跳转（如首页、简历生成器），使用 `url_for` 生成动态 URL，语言通过 CSS 类切换。结构如下：

```
<div class="nav-dropdown">
    <button class="dropdown-btn">菜单</button>
    <div class="dropdown-content">
        <a href="{{url_for('index')}}">首页</a>
        <!-- 其他链接 -->
    </div>
</div>
```

聊天侧边栏 右侧可折叠侧边栏提供 AI 聊天功能，包含输入框和发送按钮，通过 `chat.js` 处理交互逻辑。结构如下：

```
<div id="chat-sidebar" class="chat-sidebar">
    <div class="sidebar-header">
        <h2>快速聊天</h2>
        <button id="close-sidebar"></button>
    </div>
    <div id="sidebar-chat-window"></div>
    <div class="sidebar-input-section">
        <input type="text" id="sidebar-chat-input">
        <button id="sidebar-send-button">发送</button>
    </div>
</div>
```

总结 `base.html` 提供统一的 GUI 框架，支持语言切换、视觉一致性、导航和 AI 交互。子模块通过 `% block content %` 扩展功能，完整代码位于 `templates/base.html`。

3.2 学习路径推荐

3.3 简历生成器

3.4 待办清单

3.5 课程表管理

3.6 AI 助手

3.7 资源归纳库

3.8 知识卡片

4 软件测试

提供测试案例展现软件功能

5 成员分工说明