

西电新生学习导航智能体（XDU-FreshmenAssistant）

项目方案书

1. 项目概述

1.1 项目名称

西电新生学习导航智能体（XDU-FreshmenAssistant）

1.2 项目背景

进入大学后，西安电子科技大学大一新生在学习安排、竞赛选择、资源获取、政策理解以及阶段性发展规划等方面，普遍面临信息来源分散、规则理解成本高、经验传递不稳定等问题。相较于高中阶段较为统一的学习节奏，大学新生需要在短时间内适应学院培养方案、课程体系、校园平台、竞赛体系与综合发展路径，若缺乏及时、准确、可交互的引导，往往容易出现“知道要做什么，但不知道从哪里开始”的现实困境。

基于上述问题，本项目提出“西电新生学习导航智能体”方案。项目面向西安电子科技大学大一新生，目标是构建一个可本地运行、可本地演示、具备基础校园知识增强能力的智能问答原型系统，为新生提供学习建议、竞赛导航、校园资源说明、政策事项参考和阶段性学习规划支持。

本项目并非面向大规模生产部署的成熟校园平台，而是一个**可运行的本地智能体原型 + 初步知识资源底座 + 后续可扩展的知识增强架构**。项目来源于 Coze 零代码智能体的二次开发，现已逐步演进为基于 Python 服务的本地校园智能助手原型，具备进一步向知识库化、工具化、结构化演进的技术基础。

1.3 项目定位

本项目当前定位如下：

- 原型验证型项目：**验证“校园知识整理 + 智能问答交互 + 本地部署演示”这一方案的可行性。
- 可持续迭代型项目：**为后续引入更完善的知识库检索、工具链调用和多模态能力提供代码基础。
- 服务新生导向型项目：**围绕大一新生的真实高频问题，提供聚焦学习与竞赛导航的辅助支持。

因此，项目方案书将重点说明其应用场景、技术路线、原型能力、现阶段成果、创新价值与后续建设方向，而不将尚未完成的能力表述为已实现内容。

2. 应用场景与需求分析

2.1 目标用户画像

项目主要面向西安电子科技大学大一新生。该群体具有以下特点：

- 刚进入大学，对学院培养体系、课程安排和资源平台认知不足；
- 对竞赛、科研、奖学金、升学路径等内容关注度高，但缺乏系统理解；
- 常通过学长学姐经验、群聊消息、零散文档获取信息，辨别成本较高；
- 在入学初期和学期转换节点，问题呈现集中、高频、重复的特点。

2.2 典型需求场景

结合新生实际情况，本项目主要服务以下场景：

（1）学习路径咨询场景

新生常会提出如下问题：

- “本专业大一应该重点学什么？”
- “培养方案里哪些课程最关键？”
- “如何在不脱离课程要求的前提下做阶段规划？”

这类问题并不只是简单的信息查询，更需要系统根据已有资料进行整理、解释与建议输出，帮助新生形成可执行的学习认识。

（2）竞赛导航场景

新生对竞赛的认知通常片面，常见困惑包括：

- “有哪些适合大一参加的竞赛？”
- “电子类、算法类、创新创业类竞赛分别适合什么基础的人？”
- “竞赛与课程学习、保研、综合测评之间有什么关系？”

这类问题需要对竞赛清单、报名路径、准备建议进行统一梳理，并以适合新生理解的方式给出分层说明。

（3）校园资源查询场景

校园中的图书馆、教务平台、学工系统、学院通知渠道、奖助政策、请销假流程等内容往往分散于不同入口。新生在实际使用中容易出现以下问题：

- 知道有该项资源，但不知道入口和用途；
- 能找到文件，但不理解规则适用范围；
- 获取的是零散经验，难以确认可靠性。

因此，需要一个能够整合校园资源说明并进行问答式解释的入口。

（4）政策事项参考场景

对于请销假、奖学金、培养方案、保研政策等事项，新生通常并不需要抽象说明，而是需要“基于资料的可理解解释”。当前校园信息虽然可能已经存在于通知、手册或规章文件中，但缺乏统一面向新生的交互式组织方式。

2.3 现有方式的主要痛点

综合上述场景，新生当前主要痛点可概括为以下五个方面：

痛点	具体表现
信息分散	通知、文档、群聊、网站、经验贴分布在不同位置，查找成本高
表述复杂	官方文本偏制度性、说明性，新生理解门槛较高
获取路径不统一	同类问题可能需要在不同平台查询，用户操作链条长
经验依赖明显	许多问题依赖口口相传，信息一致性和时效性难保证

痛点	具体表现
缺少阶段性引导	新生往往缺乏针对入学初期的系统性导航工具

2.4 为什么需要“学习导航智能体”

本项目不是为了替代学校官方渠道，而是针对新生早期高频、重复、解释性强的问题，提供一个本地可运行、可演示、可扩展的智能问答入口。其价值主要体现在：

- 降低信息理解门槛：**将分散资料整理为更适合问答交互的表达形式。
 - 提升问题响应效率：**围绕新生常见问题形成集中支持。
 - 构建统一入口原型：**为后续校园知识服务系统建设提供实验基础。
 - 促进知识结构化沉淀：**将竞赛、资源、培养方案、政策类资料从零散文本逐步整理为可复用知识底座。
-

3. 项目目标与建设意义

3.1 项目目标

本项目的总体目标是构建一个面向西电大一新生的本地智能问答原型系统，使其能够围绕学习建议、竞赛导航、校园资源说明、政策事项参考和阶段性学习规划等任务，提供初步的交互式支持。

在当前阶段，项目目标具体包括：

- 完成从 Coze 零代码智能体到 Python 本地服务原型的迁移；
- 建立可本地运行的前后端基础架构；
- 形成初步校园知识资料目录与分层管理机制；
- 通过轻量知识增强方式提升问答与校园资料的关联度；
- 支持本地展示、课程设计汇报、项目答辩等场景下的稳定演示。

3.2 建设意义

（1）面向新生服务的实际意义

项目聚焦新生高频需求，具有明确的服务导向。一方面，可以帮助新生更快理解大学学习环境；另一方面，也为校园信息的组织方式提供新的探索路径。

（2）面向项目孵化的工程意义

本项目并非停留在概念验证层面，而是已经形成了可运行的本地原型、接口体系和知识目录结构。这种从零代码原型向代码化项目演进的过程，具有较强的工程训练价值。

（3）面向后续扩展的研究意义

本项目已经具备向知识库问答、工具调用、多源数据接入、结构化规则校验等方向继续扩展的可能性。对于校园智能助手方向而言，该项目可作为“小场景、可验证、易迭代”的建设起点。

3.3 为什么当前阶段强调本地部署

在项目现阶段，本地部署具有较强的现实合理性，主要体现在以下几点：

1. **降低搭建门槛：** 无需额外建设云端网站与复杂运维环境，更适合课程设计与竞赛原型阶段。
2. **提升展示稳定性：** 现场答辩、实验室演示时，本地运行可减少对外部网络和远程服务环境的依赖。
3. **便于快速迭代：** 开发者可以在本机环境中直接修改配置、替换知识资料、调试接口。
4. **突出工程可控性：** 对项目结构、接口调用、日志与配置的掌控度更高。
5. **适合原型验证阶段：** 在功能尚未完全完善前，本地部署能以较低成本完成验证与展示。

因此，本项目将“本地可运行、可演示、可扩展”作为当前阶段的主要建设策略，而非盲目追求上线规模化系统。

4. 总体方案设计

4.1 方案思路

项目总体设计遵循“**可运行原型优先、知识底座先行、能力渐进增强**”的思路展开。具体而言：

- 在系统层面，先完成前后端可运行闭环；
- 在智能体层面，以提示词驱动问答为基础；
- 在知识层面，优先整理本地校园资料，并通过轻量方式实现问答增强；
- 在架构层面，为后续工具调用、向量检索和规则校验预留扩展空间。

4.2 建设原则

1. **真实性原则：** 以当前实际能力为边界，不夸大原型成熟度。
2. **可演示原则：** 保证项目在普通 Windows 环境下能够启动并展示。
3. **可维护原则：** 从零代码平台导出后，逐步转化为结构清晰、可配置、可迭代的代码项目。
4. **渐进式增强原则：** 优先建设知识资源底座，再逐步引入更复杂的检索与工具能力。
5. **容错与兼容原则：** 在本地环境条件不稳定的情况下，保证系统具备基本运行能力。

4.3 总体建设内容

本项目当前总体建设内容包括：

- 本地前端交互页面；
 - 基于 FastAPI 的后端服务；
 - 基于 LangChain / LangGraph 的 Agent 执行框架；
 - OpenAI 兼容方式的大模型调用接口；
 - 本地知识目录及轻量知识增强模块；
 - 本地 Coze 转接后端原型；
 - 会话记忆持久化与降级容错机制；
 - 文件文本提取能力，为后续知识库扩展做准备。
-

5. 技术路线与系统架构

5.1 技术演进路线

本项目的技术路线并非从零开始独立构建，而是经历了从“零代码智能体原型”到“本地代码化可运行系统”的演进。其过程可概括为：

第一阶段：Coze 零代码智能体原型阶段

项目最初基于 Coze 进行校园问答类智能体尝试，优点是搭建速度快、适合快速验证场景需求，但在本地控制、接口扩展、知识管理和工程可维护性方面存在局限。

第二阶段：导出骨架并转向 Python 服务化阶段

在保留原有问答思路的基础上，项目逐步迁移至 Python 后端，采用 FastAPI 承担接口服务，形成更加可维护、可本地运行的工程骨架。

第三阶段：本地演示与知识增强阶段

为适应课程设计、项目展示和答辩需求，项目重点强化本地一键启动、本地静态前端、OpenAI 兼容调用接口以及本地知识目录组织，并在 Coze 转接原型中加入了轻量级知识增强能力。

第四阶段：后续可扩展阶段

在当前原型基础上，后续可进一步引入向量检索、结构化知识库、规则核验、工具调用链和多模态输入能力，逐步向更完整的校园智能助手系统演进。

5.2 系统架构关系说明

本项目当前系统关系可从“前端 - 本地后端 - 大模型/Coze 接口 - 本地知识资料”四部分理解。

（1）前端层

前端为独立静态页面，基于 HTML、CSS、JavaScript 实现，主要承担用户输入、消息展示和接口调用功能。用户通过浏览器输入问题后，前端调用本地部署的 OpenAI 兼容接口，实现文本对话交互。

（2）本地后端层

本地后端基于 Python + FastAPI 构建，是整个系统的核心中枢。当前已提供

`/run`、`/stream_run`、`/cancel`、`/v1/chat/completions`、`/health` 等接口，用于支持普通问答、流式输出、任务取消、兼容聊天接口以及健康检查等能力。

后端负责以下工作：

- 接收前端请求；
- 调用主 Agent 或本地 Coze 转接逻辑；
- 管理 system prompt 与模型参数配置；
- 维护会话状态与记忆；
- 在需要时读取本地知识资料并拼接上下文；
- 将结果返回前端显示。

(3) 大模型 / Coze 接口层

当前系统存在两类问答实现路径：

1. **本地主 Agent 路径：**基于 LangChain / LangGraph 调度，通过 OpenAI 兼容方式调用大模型，当前以提示词驱动问答为主。
2. **本地 Coze 转接路径：**前端调用本地 OpenAI 兼容接口，本地服务再向 Coze 的流式接口发起请求，并在请求前引入轻量知识增强。

这表明项目处于过渡演进阶段：既保留了原有 Coze 原型经验，又逐步向本地自主可控的智能体架构迁移。

(4) 本地知识资料层

本地知识资料存储于 `assets/knowledge` 目录下，并按照 `raw / curated / processed / indexes / templates` 分层管理。当前知识内容包括竞赛清单、校园资源清单，以及关于学院介绍、培养方案、图书馆、奖学金、请销假、保研、竞赛介绍等多份 Markdown 文档。

在现有轻量知识增强方案中，系统会从本地知识目录中读取 JSON/Markdown 资料，进行简单关键词匹配，将匹配结果拼接为上下文后再发送至模型或 Coze 接口，以增强回答与校园资料的相关性。

需要明确的是：**该过程属于轻量知识增强，而非完整的向量检索式 RAG 系统。**

5.3 系统架构组成

层次	主要技术	当前作用
前端展示层	HTML / CSS / JavaScript	文本对话、输入输出交互、接口调用
接口服务层	FastAPI	提供聊天、流式返回、取消、健康检查等接口
Agent 编排层	LangChain / LangGraph	管理问答流程与会话逻辑
模型调用层	OpenAI 兼容接口 / Coze 流式接口	承担文本生成与对话响应
知识增强层	本地知识目录 + 关键词匹配	提供轻量上下文补充
记忆管理层	Postgres / MemorySaver	实现持久化优先、失败回退的记忆策略
数据准备层	文件文本提取能力	支持 PDF、DOCX、XLSX、PPTX、TXT、MD 等格式解析

5.4 关键技术说明

(1) 配置化模型管理

模型参数和 system prompt 通过配置文件统一管理，便于在不改动核心代码逻辑的情况下调整模型行为、提示词策略和接口参数，提高系统可维护性。

(2) 会话记忆降级容错

系统会优先使用 Postgres 持久化保存会话记忆；当本地数据库环境不可用时，自动回退到内存 MemorySaver，以保证基本对话功能可运行。该设计对本地演示场景具有较强实用性。

(3) 知识目录分层管理

通过 `raw / curated / processed / indexes / templates` 的分层组织方式，项目为后续知识清洗、加工、索引和模板生成预留了结构基础，这有利于从“资料堆积”逐步走向“知识工程化”。

6. 核心功能模块说明

6.1 文本对话模块（已实现）

项目已具备基础文本对话能力。用户可通过前端输入问题，系统通过本地后端调用模型接口并返回问答结果。前端支持 Enter 发送，并可通过 OpenAI 兼容接口完成交互。

6.2 接口服务模块（已实现）

项目后端已提供以下主要接口：

接口	作用
<code>/run</code>	普通运行接口
<code>/stream_run</code>	流式输出接口
<code>/cancel</code>	取消当前任务
<code>/v1/chat/completions</code>	OpenAI 兼容聊天接口
<code>/health</code>	系统健康检查

这些接口使得项目具备较好的调试性、兼容性和展示性。

6.3 主 Agent 问答模块（当前原型能力）

当前主 Agent 基于 LangChain / LangGraph 搭建，但整体仍以**提示词驱动问答**为主，`tools=[]`，说明完整工具调用链尚未真正建成。也就是说，系统目前主要依赖模型与提示词完成对话，而不是基于外部工具进行检索、执行和校验。

因此，该模块当前应被视为“**可运行原型能力**”，而非成熟的智能决策代理系统。

6.4 本地知识增强模块（当前原型能力）

在本地 Coze 转接后端中，系统已经实现基于 `assets/knowledge` 目录的轻量级知识增强：读取本地 JSON/Markdown 资料，进行简单关键词匹配，并将匹配到的知识片段拼接到模型请求上下文中。

这一能力能够在一定程度上提升回答对校园资料的依赖性和贴近性，但由于当前尚未引入向量索引、召回排序、片段切分和来源校验等完整机制，因此仍属于过渡性方案。

6.5 本地一键启动模块（已实现）

项目已提供 Windows 环境下的一键启动脚本，可同时拉起前端与后端服务并自动打开浏览器。这一功能对于课程答辩、项目展示、实验室演示等场景具有较高实用价值。

6.6 文件文本提取模块（已实现基础能力）

项目已具备通用文件文本提取能力，可处理 PDF、DOCX、XLSX、PPTX、TXT、MD 等常见格式。这为后续知识库导入、资料整理和内容加工提供了必要前提。

6.7 知识库扩展与工具升级（后续规划）

后续可在现有基础上继续扩展：

- 向量检索与完整 RAG 流程；
- 结构化政策规则库；
- 网站信息同步与来源追踪；
- 工具调用链（如网页检索、知识库查询、规则判定）；
- 多模态输入支持。

上述能力均属于**后续可扩展方向**，当前尚未完整实现。

7. 创新点说明

本项目的创新点强调务实性和工程可行性，主要体现在以下几个方面。

7.1 从零代码智能体向可维护代码项目迁移

项目并非简单停留在零代码平台使用层面，而是将原有 Coze 智能体思路进一步迁移到 Python + FastAPI 的代码工程中。该过程实现了从“快速搭建原型”向“可维护、可扩展、可本地控制”的技术演进，具有较强的工程实践价值。

7.2 本地一键启动，降低部署与展示门槛

在项目尚处于原型验证阶段时，采用本地部署而非复杂云端环境，可以显著降低部署成本与答辩展示门槛。Windows 一键启动脚本的加入，使系统更适合教学展示、竞赛路演和本地调试，提升了原型的可用性。

7.3 采用“本地知识目录 + 轻量检索”的渐进式实现路径

项目没有直接跳到复杂的完整 RAG 系统，而是先构建本地知识资料目录，并通过简单关键词匹配实现基础知识增强。这种路径符合原型开发的阶段规律：先形成可用闭环，再逐步升级检索能力，避免一次性堆叠过多复杂模块而导致系统难以维护。

7.4 具备知识库扩展与工具化升级空间

当前项目已经形成较清晰的知识目录结构、文件提取能力和 Agent 基础框架，为后续接入更完善的知识检索、工具调用链和结构化规则判断打下基础。这种“原型先行、架构预留”的设计，体现了较好的可扩展性。

7.5 会话记忆支持降级容错

系统在会话记忆层面采用“Postgres 持久化优先，失败时回退到内存 MemorySaver”的策略。这种设计虽然不复杂，但对本地环境适配与原型稳定运行具有直接意义，体现了面向真实演示场景的工程考虑。

8. 使用示例

以下示例基于项目当前定位进行说明，展示的是系统可支持的典型提问形式及输出摘要。示例中的回答应理解为“结合已有知识资料与提示词能力生成的说明型输出”，而非对学校官方政策的最终认定。

8.1 示例一：新生咨询竞赛方向

用户提问：

大一新生适合参加哪些竞赛？我现在基础一般，想尽快找到一个可入门的方向。

系统输出摘要：

系统可结合已有竞赛清单与导航资料，先按竞赛类型进行分类说明，如程序设计类、电子设计类、创新创业类、数学建模类等；随后给出适合新生入门的建议，例如先从基础门槛相对明确、训练路径较清晰的竞赛入手，同时提醒用户结合自身专业基础、时间投入和团队情况做选择。系统还可补充“先学课程基础、再做竞赛积累”的阶段性建议。

8.2 示例二：新生咨询学院培养方案

用户提问：

我想了解自己学院的大一培养方案，哪些课程最重要？应该怎么安排学习重点？

系统输出摘要：

系统可基于学院介绍和培养方案相关资料，说明大一阶段通常以数学基础、程序设计基础、专业导论类课程为核心；同时对“培养方案中的课程结构”“基础课与后续专业课之间的关系”进行解释。若知识资料中已有相关 Markdown 文档，系统可进一步归纳出建议性的学习重点，例如重视数学与编程基础、建立课程之间的知识连接，而不是仅关注单门课程成绩。

8.3 示例三：新生咨询请销假与校园事项

用户提问：

如果我要请假，应该怎么理解请销假相关流程？需要注意什么？

系统输出摘要：

系统可根据已有请销假说明类资料，对请销假的一般流程、注意事项、信息来源渠道进行解释，并提醒用户以学院或学校最新正式通知为准。若当前知识库中对应条目较为基础，系统可给出“建议进一步核验对应学院规定、时间要求和审批路径”的提示，避免将未核验内容表述为最终规则。

8.4 示例四：新生咨询校园资源使用

用户提问：

学校图书馆、校园资源平台和竞赛信息一般去哪里看？我怕漏掉重要信息。

系统输出摘要：

系统可基于校园资源清单说明图书馆、教务平台、学院通知渠道、竞赛信息来源等常见入口，并建议用户建立固定的信息获取路径，如定期查看学院通知、关注教学平台和竞赛公告。该类回答的重点在于帮助新生建立“资源地图”，而不是仅给出零散链接。

9. 当前成果、存在问题与改进方向

9.1 当前成果

根据现有项目状态，当前已形成的主要成果包括：

1. 已完成从 Coze 零代码智能体到 Python 服务项目的初步迁移；
2. 已搭建基于 FastAPI 的本地后端服务，具备标准接口；
3. 已实现独立静态前端页面，支持文本对话交互；
4. 已具备 Windows 环境本地一键启动能力；
5. 已搭建主 Agent 基础框架，并支持 OpenAI 兼容调用方式；
6. 已建立 `assets/knowledge` 分层知识目录；
7. 已收集并整理竞赛清单、校园资源清单及多类校园 Markdown 资料；
8. 已在本地 Coze 转接原型中实现轻量知识增强；
9. 已支持 PDF、DOCX、XLSX、PPTX、TXT、MD 等格式的通用文件文本提取；
10. 已实现会话记忆的持久化优先与降级容错机制。

这些成果表明，项目已经具备“本地运行 + 基础知识支持 + 结构可扩展”的原型系统雏形。

9.2 当前存在的问题

需要明确指出的是，项目目前仍存在以下不足：

（1）主 Agent 工具链尚未完成

当前主 Agent 仍以提示词驱动问答为主，`tools=[]`，说明完整工具调用链尚未真正建立。因此，系统在复杂任务分解、外部知识调用、规则核验等方面能力仍有限。

（2）尚不是完整 RAG 系统

当前知识增强采用的是本地知识目录读取与简单关键词匹配方案，并未实现向量化索引、召回排序、分块检索、来源追踪等完整 RAG 机制，因此知识增强能力仍处于初级阶段。

（3）知识资料仍需持续人工核验

虽然项目已初步整理学院介绍、培养方案、图书馆、奖学金、请销假、保研、竞赛介绍等资料，但部分条目的时间信息、官网链接、规则解释仍需进一步人工核验，不能直接将其视为最终权威依据。

（4）当前主要支持文本对话

项目当前交互形式主要为文本问答，尚未实现语音输入、图像解析、表单引导等更丰富交互方式。

（5）数据清洗与结构化程度仍不足

现有知识资料虽然已形成目录管理，但在规范命名、来源标注、字段抽取、结构化整理、模板化生成方面还有较大提升空间。

9.3 改进方向

后续项目可从以下方向继续推进：

1. **完善 Agent 工具链：**引入检索工具、网页工具、规则校验工具等，增强系统可执行性。
2. **升级知识增强方案：**由简单关键词匹配逐步过渡到完整 RAG 流程。
3. **推进知识结构化：**为竞赛、培养方案、政策事项建立标准化字段。
4. **加强来源与时效管理：**补充文档来源标记、更新时间和核验机制。
5. **增强交互形式：**探索多轮引导式问答、表单化咨询、多模态输入等能力。
6. **优化展示与日志能力：**提升演示可视化程度与调试记录能力。

上述方向均属于**后续改进规划**，并不代表当前已全部实现。

10. 结论

“西电新生学习与导航智能体（XDU-FreshmenAssistant）”项目立足于西安电子科技大学大一新生在学习规划、竞赛选择、校园资源理解和政策事项认知方面的真实需求，提出了一个以本地可运行、知识可积累、架构可扩展为核心特征的校园智能问答原型方案。

从项目现状来看，其已经完成了从 Coze 零代码智能体向 Python 本地服务项目的初步迁移，形成了前后端可运行闭环、初步知识目录体系、轻量知识增强能力和本地一键启动机制，具备较好的原型展示价值和后续扩展基础。尤其是在当前阶段强调本地部署，既符合课程设计与创新项目的实际条件，也体现了原型验证阶段“低门槛、强可控、重演示”的工程取向。

同时，项目也清晰地暴露出原型阶段的限制：主 Agent 工具链尚未建成，知识增强还不是完整 RAG 系统，知识资料仍需持续核验与结构化整理，当前交互方式也主要限于文本对话。正因如此，本项目更适合被定义为一个**面向校园智能服务方向的可运行原型与基础平台**，而不是已经完成全面落地的成熟产品。

总体而言，本项目方案具有明确的问题导向、清晰的技术演进逻辑和务实的实现边界。其价值不仅在于完成一个可展示的智能体原型，更在于为后续建设更完善的校园知识服务系统提供了可持续迭代的工程基础。对于“星火杯”类创新实践项目而言，该项目具有较强的现实针对性、可实施性与继续深化空间。