STARK – Student Advising & Resource Knowledge Platform

STARK is an AI-powered academic and career assistant designed to help students with advising, resume analysis, skill gap insights, FAQs, event discovery, and job matching. It integrates multiple datasets, LLM-powered responses, and a unified dashboard experience.

Features

1. AI Chatbot

- Answers student FAQs using a fine-tuned LLM.
- Provides academic guidance, course clarification, and general queries.
- Integrates context from university datasets.

2. Resume Analyzer

- Extracts key skills, experience, and ATS-relevant metrics from resumes.
- Generates a match score for job descriptions.
- Provides suggestions to improve ATS compatibility.

3. Skill Gap Assessment

- Compares student profiles with job descriptions.
- Identifies missing skills and recommends relevant courses or clubs.

4. Career & Academic Data Integration

- Uses CSV datasets including:
    - advising notes
    - course catalog
    - jobs and events
    - clubs and announcements
    - student profile datasets
- Supports SQLite database (`stark.db`) for structured storage.

5. Student Dashboard (Web Interface)

- Built using HTML/CSS.
- Provides:

    - role-based login preview
    - live chatbot previews
    - an integrated student portal UI

6. Utility Modules

- Automated database seeding
- Context injection for the LLM
- Data preprocessing and cleaning scripts

Files & Structure

- `STARK/`
- `│`
- `├── app.py                        # Main backend orchestrator`
- `├── main_chatbot.py               # Chatbot logic + LLM integration`
- `├── resume_analyzer_integrated.py # Resume/ATS module`
- `├── summary_module.py             # Summaries and NLU helpers`
- `├── stark_context.py              # Knowledge base + system prompts`
- `├── ats_checker_updated.py`
- `├── student_faq_chatbot_llm.py`
- `├── init_db.py                    # Database initialization`
- `├── seed_more_data.py             # Populate DB with datasets`
- `├── stark.db                      # SQLite DB (if uploaded)`
- `│`
- `├── data/                         # Datasets (CSV + metadata)`
- `│   ├── advising_notes.csv`
- `│   ├── ats_jobs.csv`
- `│   ├── career_data.csv`
- `│   ├── chat_logs.csv`
- `│   ├── resumes_index.csv`
- `│   └── … (more datasets)`
- `│`
- `├── index.html`
- `├── stark_full_spa.html           # Full portal mockup`
- `├── stark_agent_live_preview.html`

- `├── stark_ai_portal_role_based_login_dashboards.html`
- `│`
- `├── notes/                        # Development notes`
- `└── Rishabh_Resume_Final.pdf`

How to Run

1. Install dependencies

- `pip install -r requirements.txt`

(You can generate this file later using `pip freeze > requirements.txt`.)

2. Initialize the database

- `python init_db.py`

(Optional if `stark.db` already exists.)

3. Start the application

- `python app.py`

Future Enhancements

- Convert static HTML pages into a full React/Flask frontend.
- Add OAuth or university SSO login support.
- Deploy chatbot as a standalone API endpoint.
- Add analytics dashboard for student progress.
- Improve job matching with embeddings + vector search.

Author

Rishabh Srinivasan
MIS | AI | Data Science
University of Maryland

-