Major Information Consulting System (MICS)

Track: Development Track

Team Member: Qinghong Cai (qc32@illinois.edu)

1. Functions and Users

I am developing a web-based application that helps students explore and compare academic majors using AI-powered insights. The system scrapes professional advice from Quora and similar forums, processes it with OpenAI, and provides structured answers through an interactive interface.

Key Features:

- AI-Powered Major Consultation – Answers questions about majors, career paths, and academic transitions.
- Comparative Analysis – Highlights differences and overlaps between majors.
- Personalized Suggestions – Recommends suitable majors based on user interests.
- Knowledge Base Search – Retrieves curated answers from Quora and other forums.

Target Users:

- High school students deciding on a college major.
- University students considering a major change.
- Academic advisors seeking data-driven insights.

2. Significance

Pain Points Addressed:

- Students struggle to find reliable, structured information about majors.
- Manual research on forums (Quora, Reddit) is time-consuming and inconsistent.
- Many lack access to personalized guidance when choosing a major.

Impact:

- Saves time by automating data collection from trusted sources.
- Reduces uncertainty with AI-curated, evidence-based advice.
- Democratizes access to professional insights, reducing reliance on advisors.

3. Approach

Technologies Used:

- **Frontend:** React (interactive UI, search, and recommendations).
- **Backend:** Express + Node.js (API handling, data processing).
- **Data Pipeline:**

  Puppeteer – Scrapes Quora and other forums for Q&A.

  OpenAI Embeddings – Converts text into searchable vectors.

  FastGPT Knowledge Base – Stores and retrieves structured answers.

Workflow:

1. **Data Collection:** Puppeteer scrapes Quora discussions on majors.
2. **Embedding & Storage:** OpenAI processes text into vectors; stored in FastGPT.

3. **User Query:** A student asks, e.g., "Which major is better for AI careers: CS or Data Science?"

4. **Retrieval & Response:** FastGPT searches the knowledge base and returns a structured answer.

Risks & Mitigation:

Risk	Mitigation
Quora blocking scrapers	Use rate-limiting, proxies, and ethical scraping practices.
OpenAI API costs	Cache responses, optimize token usage, and explore cheaper alternatives.
Slow search performance	Pre-compute embeddings and use vector indexing (e.g., FAISS).

4. Evaluation

Usefulness Testing:

- ➢ **User Surveys:** Collect feedback from 50+ students on answer quality and usability.
- ➢ **A/B Testing:** Compare manual vs. AI-assisted major research efficiency.
- ➢ **Accuracy Check:** Validate answers against expert-curated data.

Technical Validation:

- ➢ **Speed:** Ensure responses load in <2s (benchmark API latency).
- ➢ **Scalability:** Test with 100+ concurrent users (Locust/Artillery).
- ➢ **Robustness:** Handle malformed queries gracefully (e.g., "Should I switch majors?").

5. Timeline

Week 1: Setup & Scraping

Set up React + Express, design UI, and build Quora scraper with Puppeteer.

Week 2: Data & Backend

Store scraped data, integrate OpenAI embeddings, and connect FastGPT for search.

Week 3: Search & UI

Implement search functionality, refine UI, and deploy a test version.

Week 4: Testing & Demo

Gather user feedback, fix bugs, and prepare final demo/presentation.