American University of Ras Al Khaimah

Name and ID:

Marwan Sayed - 2021005030
Muhammed Irtiza - 2022005573
Rashid Ibrahim - 2022005506

Instructor: Dr. Imad Youssef Hoballah

AT & AI

Summer, 2025

Project: Career Path Analysis Tool for Market Trend and Certification Alignment .

1 Introduction & Information

The Career Path Analysis Web Application is a data-driven tool being developed by AURAK internship students to bridge the gap between academic learning outcomes and real-world job market demands. It is envisioned as a full-featured web application (beyond a simple dashboard) serving two primary user groups: students (undergraduate and graduate) and faculty. The system will ingest current job market data, analyze job postings for in-demand skills, and help identify skill gaps in students. Using this analysis, the application will recommend external certifications or university courses to address missing skills, thereby guiding students on their career pathway. For faculty and academic planners, the tool provides real-time insights into industry skill requirements and trends, informing curriculum design and updates to ensure programs remain aligned with labor market needs. Drivers for this project include the rapidly evolving job market and the need for academic institutions to keep pace with changing skill requirements. Research indicates that if educational. By leveraging real-time labor market information and analytics, universities can adapt programs to meet emerging skill demands and improve graduate employability. This Week 1 document outlines the requirements and design for the initial phase of development, focusing on clarifying project scope, goals, and system features for Phase 1, as well as the foundational architecture and team responsibilities.

2 Scope (In-Scope vs. Out-of-Scope)

2.1 In-Scope:

- **Job Data Ingestion:** Collect and import job posting data from external sources. This will include integration with at least one job market API (e.g. LinkedIn, Indeed) or use of provided datasets, and support for manual data upload (e.g. CSV files of job listings). The focus is on setting up a basic pipeline to gather current job descriptions into the system.

- **NLP-Driven Skill Extraction:** Apply Natural Language Processing (NLP) techniques to parse job descriptions and extract the key skills and competencies required. In this phase, a basic skill extraction engine will be implemented (for example, using keyword matching or an NLP library) to automatically identify technical skills, soft skills, and qualifications mentioned in job postings.

- **Skill Repository & Mapping**: Establish an initial repository of skills and map them to relevant learning resources. This involves curating a list of common skills from the job data and linking each to one or more external certifications (industry certificates, online courses) or university courses that teach or reinforce that skill. The mapping can be manually prepared or based on available data, and will serve as the knowledge base for recommendations.

- **Student Skill Gap Analysis:** Provide functionality for students to identify skill gaps. Each student user will be able to input or select their current skill set (or academic

profile/major) and a desired job role or career path. The system will compare the skills required by that target role (from the job data analysis) with the student's existing skills to highlight missing skills . The outcome is a personalized skills gap report for the student.

- **Recommendations for Skill Development:** Based on the identified gaps, the system will suggest specific courses or certifications to the student. These recommendations will draw from the skill-to-course mapping repository – for example, if "Data Analysis" is a missing skill, the system might recommend a relevant certification (like Google Data Analytics Professional Certificate) or a university course that covers that competency. The recommendations will be tailored to help students acquire in-demand skills.

- **Trend Analysis & Visualization:** Implement basic visualization of job market trends for faculty users. This includes generating charts or graphs that illustrate insights such as the most in-demand skills in a given field, trends in job postings over time, or skill demand by industry. The focus for Phase 1 is to create a simple but insightful dashboard for faculty, e.g. a trend line of how often a certain skill appears in postings, or a bar chart of top 10 skills required in a specific domain. These visualizations will help faculty spot industry skill shortages and emerging demands.

- **User Roles & Access Control:** Set up two primary user roles within the application – Student and Faculty. Each role will have an appropriate interface and permissions. Students can input their data and receive personal recommendations, whereas faculty can view aggregated data and trends (but not individual student profiles). Basic authentication (login system) will be implemented to distinguish between these roles.

2.2 Out-Scope:

- **Advanced Analytics & AI:** Any sophisticated machine learning or AI beyond basic NLP parsing is out-of-scope in the first phase. For example, predictive modeling of career trajectories or advanced AI-driven personalization will not be implemented in Week 1. The focus is on rule-based or library-supported NLP for skill extraction, not building custom ML models from scratch (those can be explored in later phases).

- **Complete Student Progress Module:** While progress tracking is conceptually considered, the full development of a progress tracking system (with dashboards of progress, analytics on improvement, etc.) is deferred to future phases. Phase 1 will not include features like badge award systems, detailed progress analytics, or deep integration with an official student record system.

- **Large-Scale Data or Real-Time Continuous Ingestion:** Initial development will use a limited dataset of job postings (for example, focusing on a few industries or a sample of recent postings). Real-time continuous scraping or ingestion of massive job data is out-of-scope. The system will not yet handle high-frequency updates or very large data volumes until performance is tuned in later phases.

- **Mobile Application:** The development of a dedicated mobile app is not included in Phase 1. The focus is on a web-based interface accessible via desktop (and possibly mobile browsers if the web UI is responsive). Native Android/iOS apps or other platforms are beyond the first phase scope.

- **Integration with University Systems:** Direct integration with existing university IT systems (e.g. student information systems, learning management systems, or official course catalogs through APIs) is not part of Phase 1. Any course or student data used will be entered or uploaded manually. Future phases might integrate with these systems for automated data retrieval, but initially the approach is standalone.

- **Extensive User Profile Management:** Apart from basic login and skill/career inputs, comprehensive user profile features (such as resume uploads, detailed resume parsing, social logins, etc.) are not included in this phase. Students will not have a full profile management system in Week 1 – they will interact with the tool mainly to get their skill gap analysis and recommendations.

- **Administrative Tools:** Management interfaces for administrators (to curate data, manage users, etc.) are minimal or nonexistent in Phase 1. Any needed data uploads or adjustments can be handled directly via the development team or backend database. Building a friendly admin panel is out-of-scope initially.

3 Project Goals

The primary goal of the Career Path Analysis Web Application is to empower students and faculty at AURAK with actionable insights drawn from real-time job market data, thereby enhancing student career preparedness and aligning academic offerings with industry needs. The high-level objectives of the project are:

- **Empower Students in Career Planning:** Provide a platform where students can explore current popular job roles and understand the skills required for those careers. By seeing the real-world demands, students can gauge where they stand. The tool will highlight the gap between the student's current skill set and the target job requirements, which guides the student's personal development plan. Ultimately, this will help students focus their efforts on gaining high-value skills, making them more competitive in the job market upon graduation.

- **Personalized Skill-Building Recommendations:** Offer tailored recommendations for each student to bridge their skill gaps. Instead of generic career advice, the system will use data to suggest specific courses or certifications that match the student's needs. For example, if a student in computer science aspires to be a data scientist but lacks knowledge in cloud computing, the system might recommend an AWS Cloud certification. These personalized pathways ensure that students have a clear roadmap to acquire in-demand skills. An important goal is to integrate these recommendations seamlessly into the student's academic or extracurricular plan (e.g., suggesting an elective course for next semester, or an online certification during summer).

- **Inform and Enhance Curriculum Design:** Enable faculty to ground their curriculum development decisions in evidence from the job market. By analyzing thousands of job postings, the tool will reveal trends such as emerging technologies or methodologies gaining traction in industry. Faculty can use this information to update course content, introduce new courses, or emphasize certain skills within existing courses. The goal is to ensure the university's programs remain relevant and produce graduates with skills that match employer expectations. For instance, if the data shows a surge in demand for data privacy skills in IT jobs, the Computer Science department might incorporate data privacy modules in appropriate courses.

- **Real-Time Labor Market Analytics:** Provide up-to-date labor market intelligence in an accessible format. Rather than relying on outdated surveys or anecdotal feedback, the system continuously (or periodically) ingests fresh job data so that analyses reflect the current state of the market. One objective is to visualize these analytics clearly – e.g., a faculty member could see at a glance what skills are most requested in the finance sector this quarter. By having real-time data, the university can be proactive rather than reactive in both advising students and adjusting curricula.

- **Support Data-Driven Career Services:** Augment the efforts of career counselors and academic advisors with data. The tool's outputs (skill gap reports, trend charts, etc.) can support one-on-one advising sessions. For example, an advisor can generate a report for a student's desired career path and together discuss which skills to work on. This makes career guidance more concrete and personalized. It also supports the Career Services department's mission by highlighting skill development opportunities (workshops, internships, etc.) that align with identified gaps.

- **Modular and Scalable Design:** From a project development perspective, an objective is to build the system in a modular way so it can be extended in future weeks. Each key feature (ingestion, NLP analysis, recommendation engine, visualization) should be relatively independent. This modular design not only allows different team members to work in parallel during development, but also makes it easier to upgrade components (for example, swapping in a more advanced NLP model later) without overhauling the entire system. Scalability is also an objective – while Phase 1 might handle a modest amount of data and users, the architecture should allow scaling up (to more data sources, more users, more features) in subsequent phases or deployments.

- **Deliver a Functional Prototype by Phase 1 End:** A practical goal for the internship's first phase is to have a working prototype that demonstrates end-to-end functionality on a basic level. This means by the end of Week 1, we aim to have at least one example scenario working: e.g., a test student can log in, select a target job role (with data ingested from somewhere), see required skills vs. their input skills, and receive one or two course/cert recommendations; a test faculty user can log in and see a simple chart of skill demand trends in that field. This prototype will validate

the concept and provide a foundation for user feedback and iterative improvement in future weeks.

Overall, the project is driven by the vision of data-informed career path planning – helping students chart their path to employment and helping faculty chart the path of curriculum evolution, both anchored in real-world evidence. If successful, this tool will enhance student success (by guiding them to become well-skilled graduates) and bolster the university's academic quality (by ensuring programs meet contemporary professional standards).

4 Initial System Architecture

The initial system architecture is designed to support a skill-gap analysis web application that helps students identify the skills they lack for their target careers, and enables faculty to align curriculum content with real-world market demands. The architecture follows a modular design, with clear separation between data ingestion, processing, storage, and user interaction.

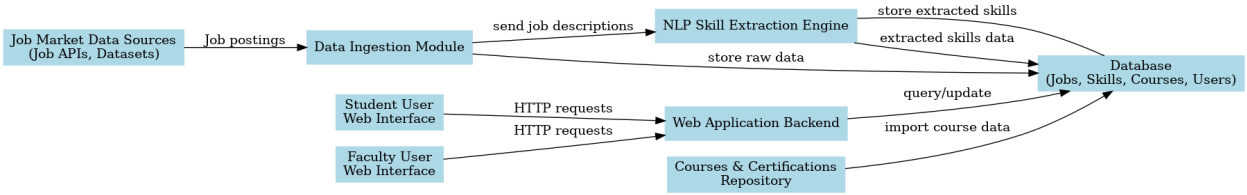

figure : Initial System Architecture

1. Job Market Data Sources

- External platforms such as LinkedIn, Bayt, and Indeed provide job postings via APIs or datasets (CSV/JSON).
- These sources are the foundation of the system's input data.

2. Data Ingestion Module

- Responsible for fetching job descriptions and preparing them for analysis.
- Performs data cleaning, formatting, and storage into the database.
- Sends descriptions to the NLP engine for skill extraction.

3. NLP Skill Extraction Engine

- Applies NLP techniques to identify relevant hard and soft skills from job descriptions.

- Uses tools like spaCy or keyword dictionaries.
- Outputs structured skill data to the central database.

4. Central Database

- Stores all core data: job postings, extracted skills, user profiles, courses, and certifications.
- Ensures consistency across all modules through structured queries and updates.

5. Courses & Certifications Repository

- Contains curated academic courses and industry certifications.
- Each course or certification is mapped to one or more skills.
- Powers recommendation features based on identified skill gaps.

6. Web Application Backend

- Handles all logic, business rules, and server-side functionality.
- Manages user authentication, data processing, API interactions, and visualization support.
- Interfaces between the database and the front-end.

7. Student Web Interface

- Allows students to select a major, input skills, and choose a target job.
- Displays skill-gap results and personalized learning recommendations.
- Provides a user-friendly platform for career planning.

8. Faculty Web Interface

- Offers faculty access to labor market analytics and trend visualizations.
- Supports curriculum enhancement and academic decision-making.
- Focuses on aggregated data rather than individual student profiles.

Key Architecture Highlights

- Modular design enables flexibility and future scalability.
- NLP integration automates real-time skill identification.
- Distinct user roles (student, faculty) provide tailored experiences.
- Centralized database ensures reliable and consistent data access.

5 Functional & Non-Functional

ID	Functional Requirement Description
FR-1	Job Data Ingestion: Import job postings from at least one external API (LinkedIn, Indeed, Bayt) and via manual CSV/JSON uploads. Store raw postings in the database.
FR-2	NLP-Driven Skill Extraction: Automatically parse each job description with NLP (e.g., spaCy NER) to extract a structured list of required skills.
FR-3	Course & Certification Mapping: Maintain a repository of university courses and external certifications, each linked to one or more skills.
FR-4	Student Profile Input: Allow students to declare their existing skills or load their transcript/major.
FR-5	Skill-Gap Analysis: Compare a student's profile against the aggregated skills required by a chosen job role to identify missing skills.
FR-6	Personalized Recommendations: For each missing skill, suggest relevant university courses or external certifications drawn from the mapping repository.
FR-7	Faculty Dashboard & Visualizations: Provide faculty with interactive charts (e.g., top N skills, skill-demand trends) based on aggregated job data.
FR-8	Generate Reports: Allow students and faculty to download CSV or PDF reports: skill-gap analysis reports (students) and market-trend summaries (faculty).
FR-9	User Authentication & Roles: Support secure login; enforce role-based access (Student vs. Faculty vs. Administrator).
FR-10	Student Progress Tracking (Conceptual): Enable students to mark completed courses/certifications and update their skill profile over time.
FR-11	Administrator Functions: Allow an admin user to import or update curriculum skill lists and maintain the course/certification repository.

ID	Non-Functional Requirement Description
NFR-1	Performance: API responses (e.g., gap analysis, visualizations) must complete within 500 ms for typical queries (1 000 skills).
NFR-2	Scalability: Architecture must support horizontal scaling of ingestion workers and database sharding or vector-store integration.
NFR-3	Reliability & Availability: System uptime 99.5 %; handle transient failures with automatic retries.
NFR-4	Security: All data in transit encrypted (HTTPS); input validation and sanitization to prevent injection and XSS attacks.
NFR-5	Usability: Clean, intuitive web UI; clear navigation; responsive layout for desktop and tablet browsers.
NFR-6	Maintainability: Modular codebase with clear separation (ingestion, NLP, backend, frontend); consistent coding standards and documentation.
NFR-7	Compatibility: Support latest versions of major browsers (Chrome, Firefox, Safari, Edge); no client-side plugins required.
NFR-8	Extensibility: Easily add new features (e.g., resume parsing, mobile app) by following the existing modular architecture.
NFR-9	Testability: Critical logic (e.g., skill extraction, gap analysis) must have unit tests or manual test cases defined.
NFR-10	Documentation: Provide API docs (Swagger/OpenAPI), user guides for students/faculty, and developer setup instructions.

6 Technology Stack & Justification

To successfully implement the Career Path Analysis Web Application, we selected a technology stack that balances development efficiency, future scalability, and ease of use especially considering the experience level of the development team and the short timeline of Phase 1. The chosen tools and platforms support rapid prototyping, robust data processing, and clean front-end interactivity, while remaining modular enough to evolve over the course of the internship project.

6.1 Backend

- **Programming Language: Python 3.11**
  Python was chosen due to its extensive ecosystem for data analysis and NLP, ease of learning, and strong community support. It is well-suited for building APIs, performing natural language processing, and handling backend logic with minimal boilerplate.

- **Web Framework: FastAPI (or Flask as an alternative)**
  FastAPI offers asynchronous request handling, built-in data validation with Pydantic, and automatic generation of API documentation (Swagger/OpenAPI). Its performance and developer-friendly syntax make it ideal for a modern backend. Flask remains an alternative if the team is more familiar with it.

- **NLP Library: NLP Library: spaCy (or/ NLTK, Sentence Transformers)**
  SpaCy provides efficient, production-grade natural language processing, which is essential for skill extraction from job descriptions. For semantic similarity or advanced matching, we may incorporate Sentence Transformers or pre-trained BERT models in later phases.

- **Data Ingestion: Custom Python scripts + API integration**
  We plan to write lightweight ingestion scripts using Python's requests and Pandas libraries, with flexibility to integrate job posting APIs or load static datasets (e.g., CSV/JSON).

6.2 Frontend

- **Framework: React.js**
  React is a popular, component-based JavaScript library ideal for building dynamic, responsive interfaces. Its modular architecture supports maintainability and reusability, which is beneficial as the dashboard evolves.

- **Charting Library: Chart.js (or Recharts)**
  For visualizing trends and data analytics, we chose Chart.js due to its simplicity and solid integration with React. It allows us to render line charts, bar graphs, and pie charts with minimal overhead.

- **State Management: React Context API**
  For managing user roles and session state, React Context API suffices in early phases.

6.3 Database

Primary Storage: PostgreSQL (or MongoDB as a flexible alternative)
PostgreSQL is a robust relational database with strong support for structured queries, indexing, and schema design. It fits well with our need to store job data, skill mappings, and user profiles. MongoDB remains an option for unstructured or flexible document storage, especially if job data formats vary significantly.

6.4 Authentication

Authentication Method: Email + Password login
For simplicity and control, we will implement a custom login system using hashed passwords. Role-based access control will restrict views (student vs. faculty). In future phases, OAuth 2.0 (e.g., Google Sign-In) could be integrated for added convenience.

6.5 Hosting and DevOps

Environment: Local development + GitHub repository
Initial development will occur locally using virtual environments. GitHub will be used for

version control, collaboration, and documentation. A README.md and .env configuration files will support easy setup.

References

[1] EDUC8EU Alliance. (n.d.). Comprehensive database: Data sourced from ESCO, ILO, and O*NET providing up-to-date job profiles, learning pathways, and associated skills. Retrieved May 20, 2025, from https://educ8eu.invest-alliance.eu

[2] Favicontalismatic.com. (n.d.). Predictive analytics tool for educators: Labor market analytics. Retrieved May 20, 2025, from http://favicontalismatic.com

[3] Lumina Foundation. (n.d.). The need to better understand broad labor market trends. Retrieved May 20, 2025, from https://luminafoundation.org

[4] ResearchGate. (n.d.). Job Mining System Architecture [Scientific diagram]. Retrieved May 20, 2025, from https://www.researchgate.net

[5] Syahrial, R. (n.d.). Decoding job descriptions: How NLP unlocks the search for skills. *Medium*. Retrieved May 20, 2025, from https://medium.com

[6] San José State University Career Center. (n.d.). Skills gap analysis [Handout]. Retrieved May 20, 2025, from https://careercenter.sjsu.edu

[7] Antonucci, O. (2012). *Skills gap analysis: All you need to know* [Template]. AIHR. Retrieved May 20, 2025, from https://aihr.com