



Skillpath AI

Technical Documentation

Prepared by:
Anusree Mohanan
Divya Kasa
Sangram Anil Shinde

Table of Contents

Executive Summary	3
Problem Statement	4
Why SkillPath AI?	5
Solution Overview	6
System Architecture	9
Core Features	12
Data Architecture	14
AI/ML Component	17
APIs and Backend	19
Frontend Design	23
Security and Privacy	33
DevOps and Deployment	35
Monitoring and Analytics	37
Challenges and Lessons Learned	39
Design Decisions & Justification	43
Future Scope and Improvements	46
References and Resources	49

Executive Summary

SkillPath AI is an AI-powered personalized learning platform designed to streamline the upskilling journey for students and professionals. In today's fragmented e-learning landscape, learners are often overwhelmed by the number of available courses and the lack of clear guidance tailored to their specific career goals. SkillPath AI addresses this challenge by providing structured, role-specific learning paths based on a user's resume, self-rated skills, and target career aspirations.

The platform leverages modern AI technologies, including Retrieval-Augmented Generation (RAG), large language models (LLMs), and semantic search, to deliver curated course recommendations that evolve with both user progress and market trends. Users can either upload their resume for automated skill gap detection or manually rate their skills to receive a personalized learning roadmap.

SkillPath AI also includes an interactive chatbot interface that allows users to ask career-related questions, get project suggestions, track progress, and revisit previous sessions for long-term career development support.

Built using a modular and scalable architecture, SkillPath AI combines:

- A Streamlit-based frontend for intuitive user interactions
- A FastAPI backend for secure and efficient communication
- A Snowflake data platform enhanced with Cortex for scalable AI workloads and vector-based search

This documentation outlines the system's architecture, key features, data flow, AI components, and future roadmap, offering a comprehensive view of how SkillPath AI transforms fragmented learning into a personalized, intelligent career advancement experience.

Problem Statement

Despite the abundance of online learning platforms, learners often face significant obstacles when trying to upskill effectively:

- **Fragmented Course Ecosystem:** Navigating multiple platforms (e.g., Coursera, edX, Udemy) is time-consuming, leading to confusion and duplicated effort.
- **Lack of Personalization:** Most learning recommendations are generic, failing to consider a learner's existing skills, past experience, or specific career goals.
- **Manual Search Dependency:** Users must search, filter, and evaluate courses manually, resulting in trial-and-error learning that delays progress.
- **No Resume-Driven Guidance:** Few, if any, platforms analyze resumes to generate personalized upskilling paths, making transitions to new roles difficult.
- **Slow Career Progression:** Without a focused and guided roadmap, learners often lose motivation or pursue misaligned paths, prolonging their career transitions.
- These challenges highlight the urgent need for an intelligent, centralized solution that aligns learning with personal skills and evolving job market demands.

Why SkillPath AI?

SkillPath AI is designed to eliminate guesswork from career upskilling. It combines the power of AI with real-time course intelligence to provide a seamless, personalized learning experience. Here's why SkillPath AI stands out:

- **No More Course Hunting**

Skip endless searching—SkillPath AI instantly generates curated learning paths tailored to your resume or rated skills.

- **AI-Powered Roadmaps**

Receive structured, step-by-step plans that progress logically from beginner to advanced levels, tailored to your background and target role.

- **Market-Aligned Learning**

Course recommendations evolve based on user progress and are dynamically aligned with market trends and job role requirements.

- **Faster Career Transitions**

Bridge skill gaps in 3–6 months with focused learning, real-world project suggestions, and industry-relevant certifications.

- **Built-In Career Transition Assistant**

From resume analysis to course recommendations and chatbot guidance, the platform is designed to accelerate career transitions with minimal user friction.

By combining AI-driven insights, structured learning paths, and a conversational assistant, SkillPath AI ensures every learner moves toward their goal with clarity, confidence, and efficiency

Solution Overview

SkillPath AI is a personalized upskilling and career transition platform that leverages artificial intelligence to recommend structured, goal-oriented learning paths. It addresses two core user flows:

- **Career Transition Assistant** – for users uploading a resume and targeting a specific role
- **Learning Path Assistant** – for users rating their skills manually and exploring career-aligned learning paths

These assistants are powered by an AI engine that analyzes user input, detects skill gaps, and returns curated courses organized from beginner to advanced levels—enabling career progression in 3–6 months.

Key Solution Components

4.1 Resume Analysis & Skill Gap Detection

- Users upload their resumes, which are parsed to extract skills using NLP and LLM-based entity recognition.
- A pre-defined skill taxonomy for each target role (e.g., Data Engineer) is matched against extracted resume skills to identify gaps.
- **The system distinguishes between:**
 - Transferable skills
 - Complementary skills
 - Critical gaps to develop

4.2 Role-Specific Learning Path Generation

- Based on detected skill gaps or rated proficiencies, the platform:
 - Recommends beginner-to-advanced courses from Udemy, edX, and Udacity
 - Organizes the courses into a 6-month timeline (3 stages: Beginner, Intermediate, Advanced)
 - Includes project ideas, certification tips, and community suggestions for deeper engagement

4.3 AI-Powered Chat Assistant

- Users can chat with a built-in assistant to:
 - Ask career or skill-related questions
 - Get clarifications on learning paths
 - Resume previous sessions for continuity
- Powered by a fine-tuned GPT model and RAG (Retrieval-Augmented Generation) to provide contextual, relevant answers

4.4 Dynamic User Input Paths

- Resume-based onboarding provides highly automated recommendations based on actual experience
- Self-rated skills onboarding enables flexibility and caters to users with incomplete or unavailable resumes

4.5 Real-World Career Alignment

- SkillPath AI recommendations include practical insights like:
 - Which certifications to pursue (e.g., GCP Professional Data Engineer)
 - What types of portfolio projects to build
 - What communities and roles to target during a transition

Technologies Used

Frontend: Streamlit for a responsive and interactive user interface

Backend: FastAPI to handle routing and communication with core services

Data Platform: Snowflake is used for structured and semi-structured data storage

Transformation Layer: dbt for building modular, version-controlled data models; Snowflake Cortex for embedding generation and semantic enrichment

AI/ML Engine: RAG-based pipeline powered by Snowflake Cortex and large language models for skill extraction, path generation, and contextual chat

Containerization: Docker for consistent deployment and simplified development

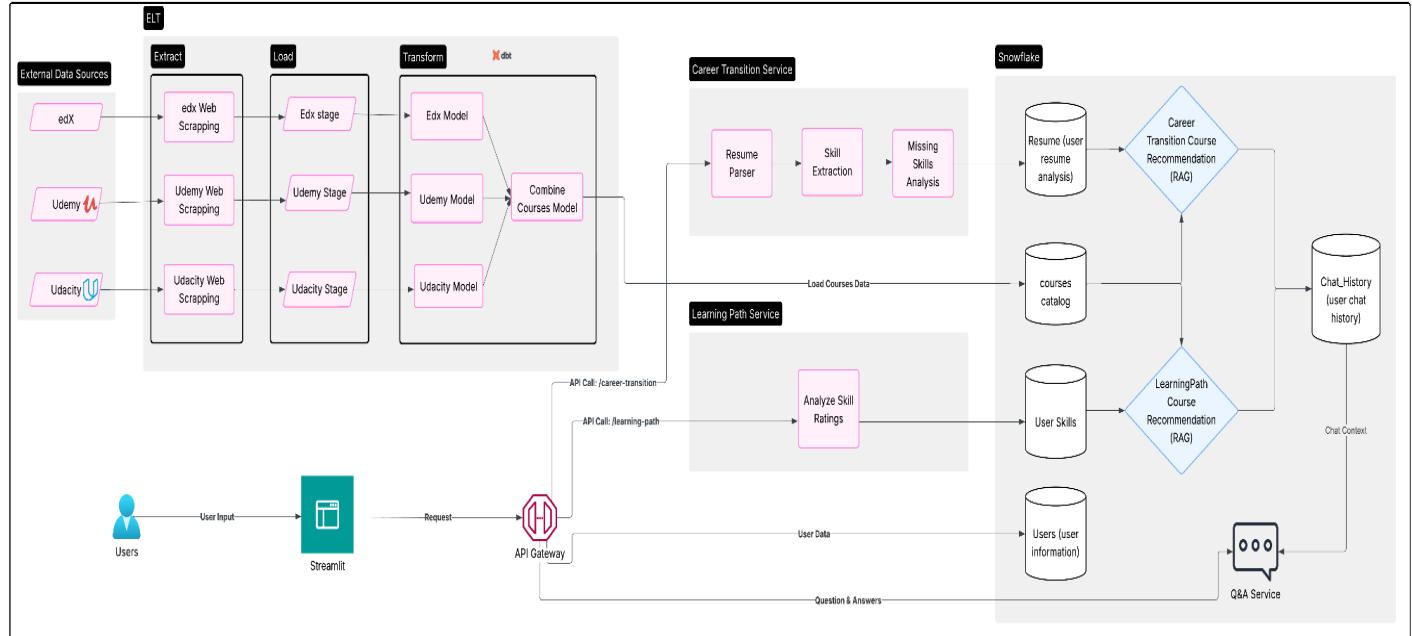
Modular Architecture for Scalability

SkillPath AI is built on a containerized architecture using Docker, enabling consistent deployment across environments and simplifying local development. The modular system allows:

- Seamless integration of new features and services
- Scalable deployment in cloud or on-prem environments
- Rapid onboarding for contributors through Dockerized setup

System Architecture

The architecture of SkillPath AI is designed to be modular, scalable, and data-centric — integrating web scraping, ELT pipelines, AI services, and interactive frontends into a unified platform. The system supports both Career Transition and Personalized Learning Path recommendations, powered by LLMs, Snowflake Cortex, and real-time user inputs.



5.1 High-Level Components

The platform is divided into five major subsystems:

1. Data Ingestion and ELT Pipeline
2. AI Services (Skill Extraction, Recommendation Engine)
3. Snowflake Data Warehouse and Cortex Services
4. Streamlit Frontend (User Interface)
5. FastAPI Backend (API Gateway + Services)

5.2 Data Sources and Ingestion (Extract → Load → Transform)

External Sources:

Course data is gathered from:

- Udemy (via Kaggle)
- edX and Udacity (via custom web scraping)

ETL Process:

Extract:

Real-time scraping scripts pull fresh course metadata (titles, objectives, levels, skills).

Load:

Data is staged into individual Snowflake tables (edx_stage, udemy_stage, udacity_stage).

Transform (dbt):

Course data is cleaned, standardized, and merged into a combined model (combine_courses_model) using dbt.

5.3 AI Services Layer

a) Career Transition Service

- Resume Parser: Extracts user skills and experience from uploaded resume files.
- Skill Extraction: Identifies key technical skills from resume content.
- Missing Skills Analysis: Compares extracted skills with those required for a target role.
- Output: Feeds into the RAG engine for targeted course recommendations.

b) Learning Path Service

- Analyze Skill Ratings: Collects user-provided ratings (1–5) for a list of LLM-suggested skills.
- Output: Triggers the learning path generation logic with personalized context.

5.4 Snowflake Cortex Integration

Central Data Store:

User data, chat history, and all course-related metadata are stored in Snowflake.

RAG-Based Recommendations:

Two separate Cortex Search + LLM pipelines power:

- Career Transition Recommendations
- Learning Path Recommendations

Cortex Services Used:

- Cortex COMPLETE: Infers missing course skills, prerequisites, and cleans metadata.
- Cortex SEARCH: Performs semantic search to retrieve relevant courses based on user goals or missing skills.

5.5 Chat History and Q&A Service

Chat Context Storage:

All AI interactions are stored in the chat_history table with session-level granularity.

Q&A Layer:

The conversational interface leverages previous user context to provide tailored follow-up responses and support iterative learning.

5.6 Frontend and API Gateway

Streamlit Frontend:

- Users interact with the platform through a Streamlit-based UI.
- Supports resume uploads, role selection, skill rating, and chat-based guidance.

API Gateway (FastAPI):

Handles all requests from the frontend and routes them to appropriate backend services.

Endpoints include:

/career-transition

/learning-path

/chat

/auth

Core Features

SkillPath AI offers a comprehensive suite of features designed to support users through personalized upskilling and career transitions. Each feature is built with a focus on usability, intelligence, and real-world career impact.

6.1 Career Transition Assistant

The Career Transition Assistant is designed for users who want to pivot into a new role. It automates the skill gap analysis and delivers a curated learning strategy based on the user's resume.

- **Resume Upload and Parsing:** Users upload a resume, which is processed using NLP and LLMs to extract relevant technical and soft skills.
- **Target Role Selection:** Users choose a role they wish to transition into, such as "Data Engineer" or "Product Manager."
- **Skill Gap Detection:** The platform compares extracted resume skills with predefined role-specific templates to identify transferable, complementary, and missing skills.
- **Structured Learning Path:** A personalized roadmap is generated, detailing the most relevant courses, certifications, and projects to support the transition within 3–6 months.
- **Career Recommendations:** Users receive guidance on resume adjustments, project portfolios, and strategic networking.

6.2 Learning Path Assistant

The Learning Path Assistant supports users who prefer to self-assess rather than upload a resume. It is well-suited for early-career professionals or those exploring new domains.

- **Target Role Selection:** Users begin by selecting a career goal they want to pursue.
- **Skill Identification via LLM:** An LLM generates a tailored list of essential skills for the selected role based on current market expectations.
- **Skill Self-Rating:** Users rate their proficiency in each skill on a 1–5 scale, providing a personalized baseline.
- **Gap Analysis and Learning Path:** The platform identifies areas of improvement and recommends a time-phased learning path (beginner, intermediate, advanced).
- **Support Resources:** Suggestions include project ideas, certification paths, and role-specific tips.

6.3 AI Chat Assistant

The AI assistant provides conversational support throughout the user's learning journey. It enhances interactivity and clarity by responding to queries in real time.

- Conversational Guidance: Users can ask follow-up questions about recommended courses, next steps, or concepts they don't understand.
- Context Retention: The assistant retains user context across sessions to provide coherent, personalized responses.
- Flexible Interaction: Supports both casual learning exploration and deep-dive technical queries.

6.4 Actionable Skill Recommendations

SkillPath AI offers insights and next steps based on real-time analysis of user progress and industry needs.

- Maps user-rated or resume-extracted skills against industry-defined role requirements.
- Recommends precise learning modules, projects, and communities to accelerate development in high-impact areas.
- Aligns each user's journey with practical career outcomes.

6.5 Session and History Management

To support continuous learning and flexible progress tracking, the platform maintains persistent records of each user session.

- Saves uploaded resumes, role targets, skill ratings, and AI chat interactions.
- Enables users to return, review, or restart their learning journey at any time.
- Provides continuity across devices and sessions.

Data Architecture

7. Data Architecture

SkillPath AI's data architecture is designed for scalability, modularity, and intelligent enrichment. It supports real-time recommendation generation, structured and semi-structured data management, and seamless integration with AI models. The system integrates multiple data sources, processes data using dbt and Snowflake-native AI functions, and prepares enriched datasets for learning path generation.

7.1 Data Sources

SkillPath AI ingests and integrates structured and user-generated data:

Course Data:

- Udemy: Pre-downloaded from Kaggle (25.1 MB)
- edX : Scraped using custom Python scrapers
- Udacity: Scraped using custom Python scrapers

User-Generated Input:

- Uploaded resumes (PDF/Docx format).
- Target role selection and skill rating (via chatbot).
- Chat queries and career-related interactions.

7.2 ETL and Transformation Process

The ETL workflow is built using dbt and Snowflake services to ensure data cleanliness, enrichment, and query optimization:

1. Ingestion:
 - Raw course metadata collected via scraping and dataset downloads.
 - User resumes uploaded and parsed.
2. Transformation (dbt):
 - Initial standardization of raw metadata (titles, objectives, providers).
 - Prepares staging models for AI enrichment.
 - AI-Driven Data Enrichment (Cortex COMPLETE):
 - Uses Snowflake Cortex COMPLETE() to:
 1. Extract key skills from course objectives.
 2. Infer concise technical prerequisites.
 3. Summarize course objectives.
 4. Normalize instructional levels.
 5. Clean noisy text fields.
 6. Automates large-scale metadata cleaning and standardization across all courses.
 - Semantic Search Preparation (Cortex Search):

1. Enriched course data is indexed into a Cortex Search service.
2. Enables semantic retrieval of the most relevant courses during recommendation generation.

7.3 Storage Schema in Snowflake

Structured Tables:

udemy_courses, edx_courses, udacity_courses, user_profiles, chat_history, learning_paths

Semi-Structured Fields:

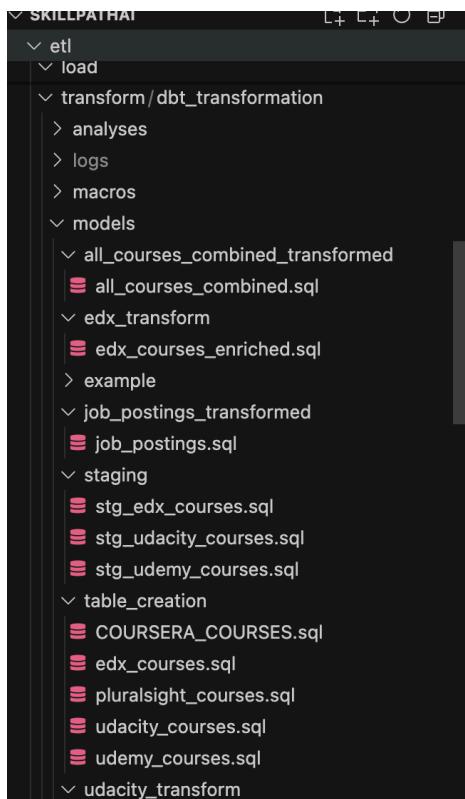
Course fields like OBJECTIVES_SUMMARY, SKILLS, PREREQUISITES, and EMBEDDINGS are stored using VARIANT or JSON types.

Vector Storage:

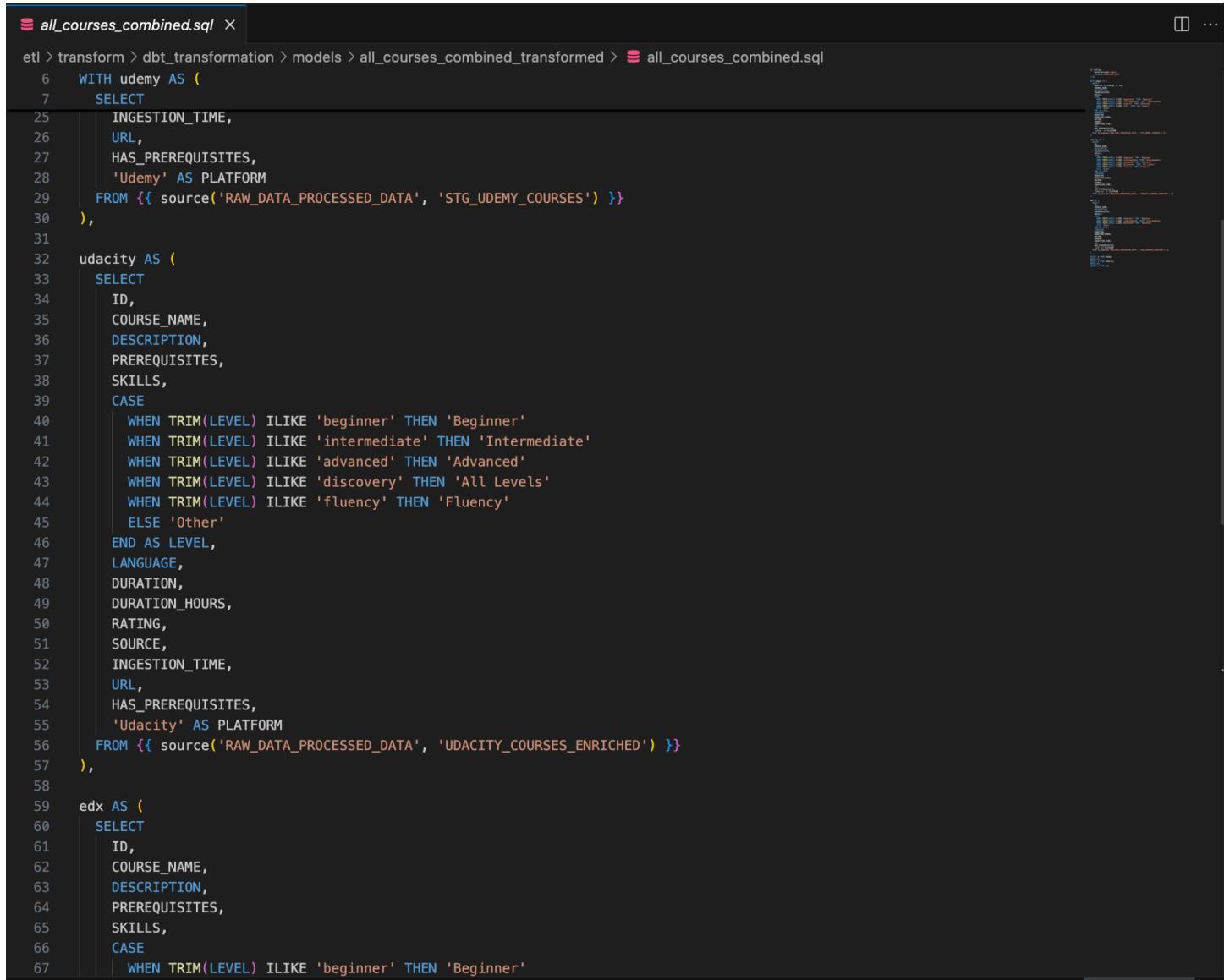
Embedding information is handled internally by Cortex Search for semantic query capabilities

7.4 Automation and Orchestration

dbt Models and Snapshots:



File Structure - dbt_transformation



```

etl > transform > dbt_transformation > models > all_courses_combined_transformed > all_courses_combined.sql
6   WITH udemy AS (
7     SELECT
25       INGESTION_TIME,
26       URL,
27       HAS_PREREQUISITES,
28       'Udemy' AS PLATFORM
29     FROM {{ source('RAW_DATA_PROCESSED_DATA', 'STG_UDEMY_COURSES') }}
30   ),
31
32   udacity AS (
33     SELECT
34       ID,
35       COURSE_NAME,
36       DESCRIPTION,
37       PREREQUISITES,
38       SKILLS,
39       CASE
40         WHEN TRIM(LEVEL) ILIKE 'beginner' THEN 'Beginner'
41         WHEN TRIM(LEVEL) ILIKE 'intermediate' THEN 'Intermediate'
42         WHEN TRIM(LEVEL) ILIKE 'advanced' THEN 'Advanced'
43         WHEN TRIM(LEVEL) ILIKE 'discovery' THEN 'All Levels'
44         WHEN TRIM(LEVEL) ILIKE 'fluency' THEN 'Fluency'
45         ELSE 'Other'
46       END AS LEVEL,
47       LANGUAGE,
48       DURATION,
49       DURATION_HOURS,
50       RATING,
51       SOURCE,
52       INGESTION_TIME,
53       URL,
54       HAS_PREREQUISITES,
55       'Udacity' AS PLATFORM
56     FROM {{ source('RAW_DATA_PROCESSED_DATA', 'UDACITY_COURSES_ENRICHED') }}
57   ),
58
59   edx AS (
60     SELECT
61       ID,
62       COURSE_NAME,
63       DESCRIPTION,
64       PREREQUISITES,
65       SKILLS,
66       CASE
67         WHEN TRIM(LEVEL) ILIKE 'beginner' THEN 'Beginner'

```

all_courses_combined.sql - dbt model for course data compilation

- Maintain modular SQL transformations.
- Capture versioned snapshots of course and skill data evolution over time.

7.5 Data Governance and Observability

- Role-Based Access Control (RBAC) for managing data access.
- Data Quality Checks embedded within dbt pipelines.
- Logging and Monitoring for scrapers, data transformation jobs, and enrichment processes.

AI/ML Component

SkillPath AI integrates multiple AI components to deliver personalized, contextually accurate learning journeys and career advice.

8.1 Large Language Models (LLMs)

- Resume parsing and skill extraction from uploaded documents.
- Dynamic generation of role-specific skill requirements.
- Interactive career guidance and real-time query resolution via chatbot.
- Prompt templates ensure consistent and structured outputs from GPT models.

8.2 Retrieval-Augmented Generation (RAG) using Cortex Search

- Cortex Search powers semantic retrieval over enriched course metadata.
- Retrieves the top matching course entries based on user skill gaps and role preferences.
- Retrieved course context is injected into GPT prompts for more accurate and relevant recommendations.

8.3 Data Cleaning and Skill Inference using Cortex COMPLETE

Cortex COMPLETE is used to:

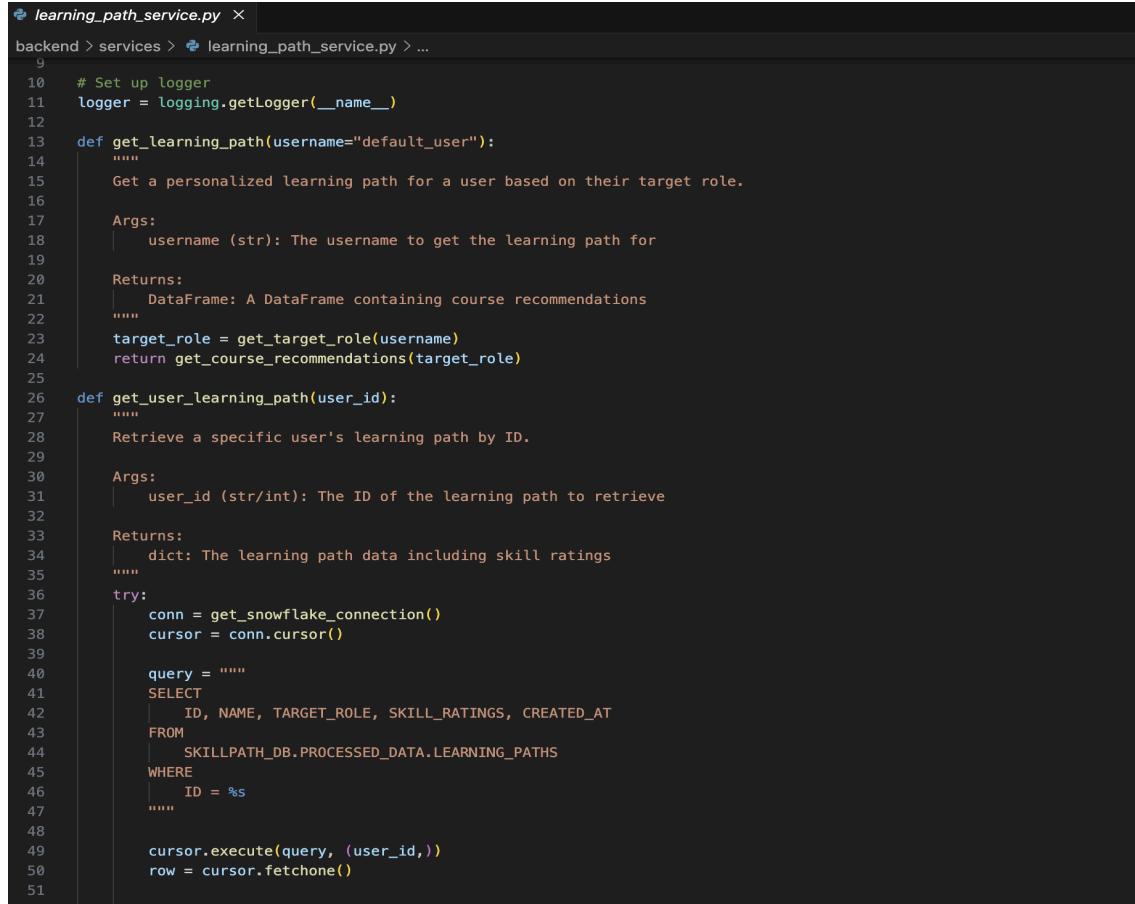
- Extract structured skills from course content
- Infer specific prerequisites based on course objectives and instructional level
- Summarize and clean course descriptions
- Enrichment standardizes the dataset and ensures readiness for AI-driven recommendation

```
WITH base AS (
  SELECT
    *,
    -- Fill in prerequisites only if NULL
    CASE
      WHEN PREREQUISITES IS NULL AND DESCRIPTION IS NOT NULL THEN SNOWFLAKE.CORTEX.COMPLETE(
        'mistral-large2',
        'Extract the specific prerequisites needed to start this course based on the title, description, and instructional level.
        FORMAT GUIDELINES:
        - Return ONLY a concise, comma-separated list of specific prerequisites (e.g., "Basic Python, Statistics, Git")\
        - If no clear prerequisites exist, return "No prerequisites required"\n        - For "Beginner" courses, keep prerequisites minimal or none unless explicitly mentioned\
        - For "Intermediate" courses, include foundational knowledge in the subject area\
        - For "Advanced" courses, include advanced prerequisite skills\
        - Exclude generic terms like "willingness to learn" or "dedication"\n        - Focus on technical skills, knowledge, or experience needed\
        - No sentences, explanations, or bullet points'
        COURSE INFORMATION:
        Title: ' || COURSE_NAME || '\
        Description: ' || DESCRIPTION || '\
        Level: ' || LEVEL
      )
      ELSE PREREQUISITES
    END AS FILLED_PREREQUISITES,
    --
    -- Combine SKILLS and GROW THESE SKILLS
  
```

Cortex query used during dbt transformation for edx courses data.

8.4 Personalized Path Generation

- Maps identified skill gaps to curated course selections
- Structures learning paths into beginner, intermediate, and advanced phases
- Aligns each user's journey with certifications, project ideas, and career objectives
- Learning plans are optimized for 3–6 month completion timelines



The screenshot shows a code editor window with the file 'learning_path_service.py' open. The code is written in Python and defines two main functions: `get_learning_path` and `get_user_learning_path`. The `get_learning_path` function takes a `username` parameter and returns a `DataFrame` of course recommendations based on the user's target role. The `get_user_learning_path` function takes a `user_id` parameter and returns a dictionary of learning path data including skill ratings. Both functions use a `get_snowflake_connection` helper to execute SQL queries against a database named `SKILLPATH_DB`.

```
learning_path_service.py
backend > services > learning_path_service.py > ...
9
10 # Set up logger
11 logger = logging.getLogger(__name__)
12
13 def get_learning_path(username="default_user"):
14     """
15         Get a personalized learning path for a user based on their target role.
16
17     Args:
18         username (str): The username to get the learning path for
19
20     Returns:
21         DataFrame: A DataFrame containing course recommendations
22     """
23     target_role = get_target_role(username)
24     return get_course_recommendations(target_role)
25
26 def get_user_learning_path(user_id):
27     """
28         Retrieve a specific user's learning path by ID.
29
30     Args:
31         user_id (str/int): The ID of the learning path to retrieve
32
33     Returns:
34         dict: The learning path data including skill ratings
35     """
36     try:
37         conn = get_snowflake_connection()
38         cursor = conn.cursor()
39
40         query = """
41             SELECT
42                 ID, NAME, TARGET_ROLE, SKILL_RATINGS, CREATED_AT
43             FROM
44                 SKILLPATH_DB.PROCESSED_DATA.LEARNING_PATHS
45             WHERE
46                 ID = %s
47             """
48
49         cursor.execute(query, (user_id,))
50         row = cursor.fetchone()
51     except Exception as e:
52         logger.error(f"Error fetching learning path for user {user_id}: {e}")
53
54         return None
55
56     return {
57         "id": row[0],
58         "name": row[1],
59         "target_role": row[2],
60         "skill_ratings": row[3],
61         "created_at": row[4]
62     }
```

Prompts used for learning path service

APIs and Backend

SkillPath AI's backend is a modular FastAPI application exposing RESTful APIs for authentication, user inputs, skill extraction, learning path generation, and course recommendations. Each functional area is cleanly separated into different routers and service layers, enabling scalable and maintainable development.

9.1 Authentication APIs

The authentication module handles user management and basic security operations.

API Endpoints:

POST /auth/signup — Register a new user account with hashed password storage.

POST /auth/login — Authenticate user credentials and return a token (currently placeholder).

POST /auth/reset-password — Reset the user's password securely.

GET /auth/user-profile — Retrieve the current user's profile information (e.g., username, email, created date).

Key Logic:

User data is stored in Snowflake.

Passwords are hashed securely before saving.

Login validates credentials and issues an access token for session management.

9.2 User Input APIs

The user input module manages everything the user submits to interact with the learning system — resumes, skill ratings, career planning data, chat queries.

API Endpoints:

POST /user-input/resume — Upload a resume file for parsing.

POST /user-input/resume/text — Submit extracted resume text.

POST /user-input/resume/extract — Extract text content from uploaded file using parsing service.

POST /user-input/skills/extract — Extract skills from resume text using an LLM (structured extraction).

POST /user-input/skills/extract-regex — Extract skills using pattern matching (regex-based method).

POST /user-input/skill-ratings/store — Save self-rated skills provided by the user.

POST /user-input/career-transition — Submit user's desired career shift details for planning.

POST /user-input/career-courses — Recommend courses based on career transition plan.

POST /user-input/transition-plan — Generate a step-by-step career transition plan based on skill gaps.

POST /user-input/learning-path — Generate a role-specific personalized learning roadmap.

POST /user-input/save-session-state — Save the user's session and chat history.

POST /user-input/chat-history/clean — Delete old or unnecessary chat sessions.

GET /user-input/chat-history/recent — Retrieve the user's five most recent chat sessions.

POST /user-input/career-question — Enable users to ask career guidance questions to a GPT-powered assistant.

Key Logic:

Resume parsing extracts text from PDF or DOCX files.

Skill extraction uses both regex and LLM-powered techniques.

Career transition plans and learning paths are generated based on gaps between current and required skills.

Chat session state is stored and retrieved from the database for continuity.

9.3 Recommendations APIs

The recommendations module focuses on providing skill suggestions and course recommendations based on user goals and skill assessments.

API Endpoints:

GET /recommendations/skills/top/{role} — Retrieve the top skills expected for a given target role.

POST /recommendations/courses — Recommend best-fit online courses that address the user's missing skills.

Key Logic:

Top skills are fetched dynamically for different roles using Snowflake Cortex models.

Courses are retrieved semantically using Cortex Search to ensure relevance to the user's context and career goals.

9.4 Service Layer Operations

Each API endpoint is connected to a service layer responsible for processing, decision-making, and integrations:

Operation	Description
Authentication	Handles signup, login, password hashing, and user profile retrieval.
Resume Parsing	Extracts raw text from uploaded resume files.
Skill Extraction	Identifies technical skills from resume text using both LLM inference and regex matching.
Skill Gap Matching	Compares current skills with required role-specific skills and identifies gaps.
Career Transition Planning	Maps skill gaps to recommended actions and course plans.
Learning Path Generation	Builds structured, staged learning roadmaps over 3–6 months.
Chatbot Conversations	Manages AI-driven career advice and skill clarification conversations.
Course Recommendations	Retrieves the most relevant course suggestions based on skills needed and Cortex Search rankings.

9.5 Backend Architecture Features

Environment-Aware CORS:

- In Development: Allows all origins for easier testing.
- In Production: Restricts API access to the official frontend URL.

Detailed Logging:

- Every HTTP request and response cycle is logged with processing time, status codes, and payload details.
- Rotating file logs are maintained under logs/api_server.log for operational visibility.

Token-Based Authentication:

- Basic JWT authentication is implemented for secure route access.

Separation of Concerns:

- APIs only handle routing and validation.
- Services encapsulate business logic and database/AI integration.
- Databases and AI models (Snowflake Cortex, GPT) are accessed through standardized service interfaces.

Frontend Design

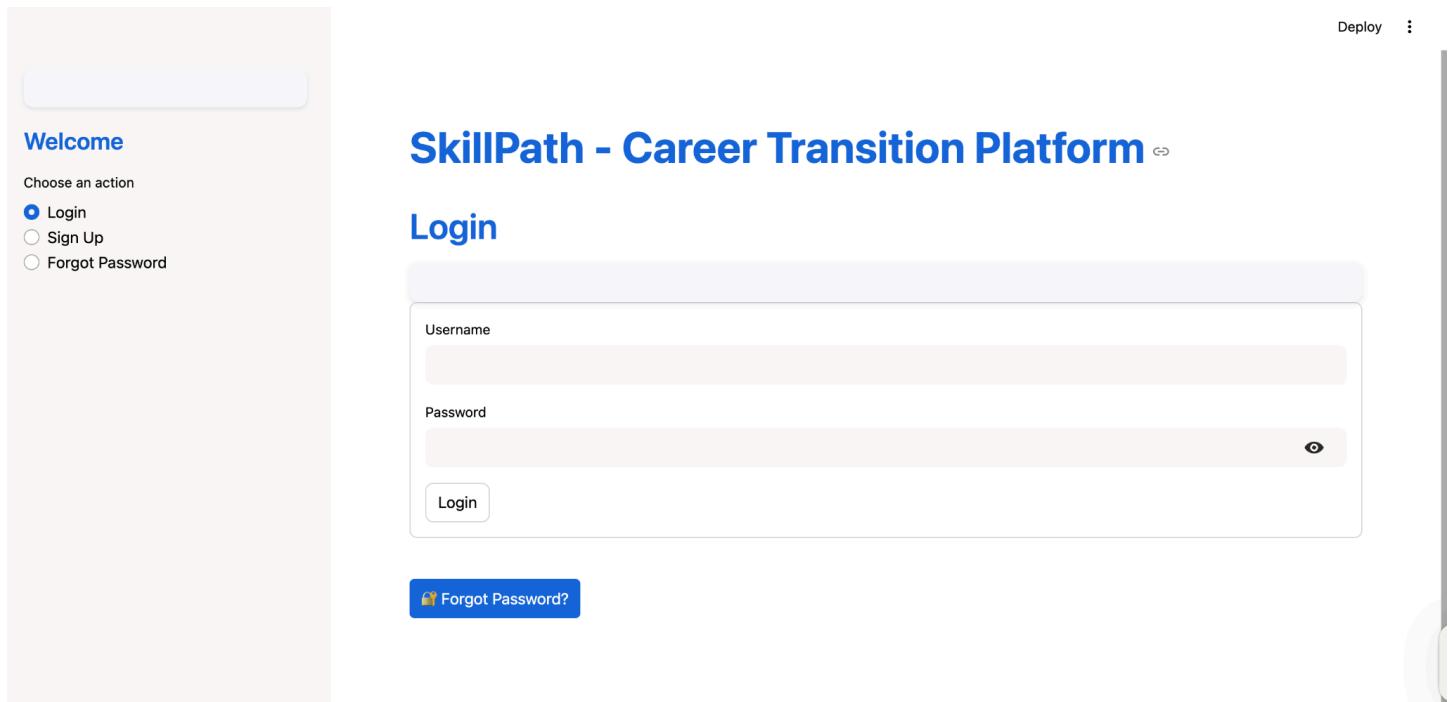
SkillPath AI's frontend is built using Streamlit, designed to offer users a streamlined, intuitive journey toward upskilling or career transition. The user experience is organized into clear stages, providing a step-by-step flow supported by intelligent, AI-driven interactions.

10.1 User Interface Structure and Functional Flows

1. Authentication and Access

Login Page:

Allows users to securely authenticate using a username and password. Proper error handling is in place for invalid credentials or empty fields.



Signup Page:

New users can create an account by providing basic information such as username, email, and password.

SkillPath - Career Transition Platform

Welcome

Choose an action

- Login
- Sign Up
- Forgot Password

Sign Up

Full Name

Username

Email

Password

Confirm Password

Sign Up



Forgot Password Page:

Users who have forgotten their password can request a reset. Upon entering their registered email, reset instructions are sent. Success and error notifications are clearly displayed.

Welcome

Choose an action

- Login
- Sign Up
- Forgot Password

Forgot Password

Username

New Password

Confirm Password

Reset Password



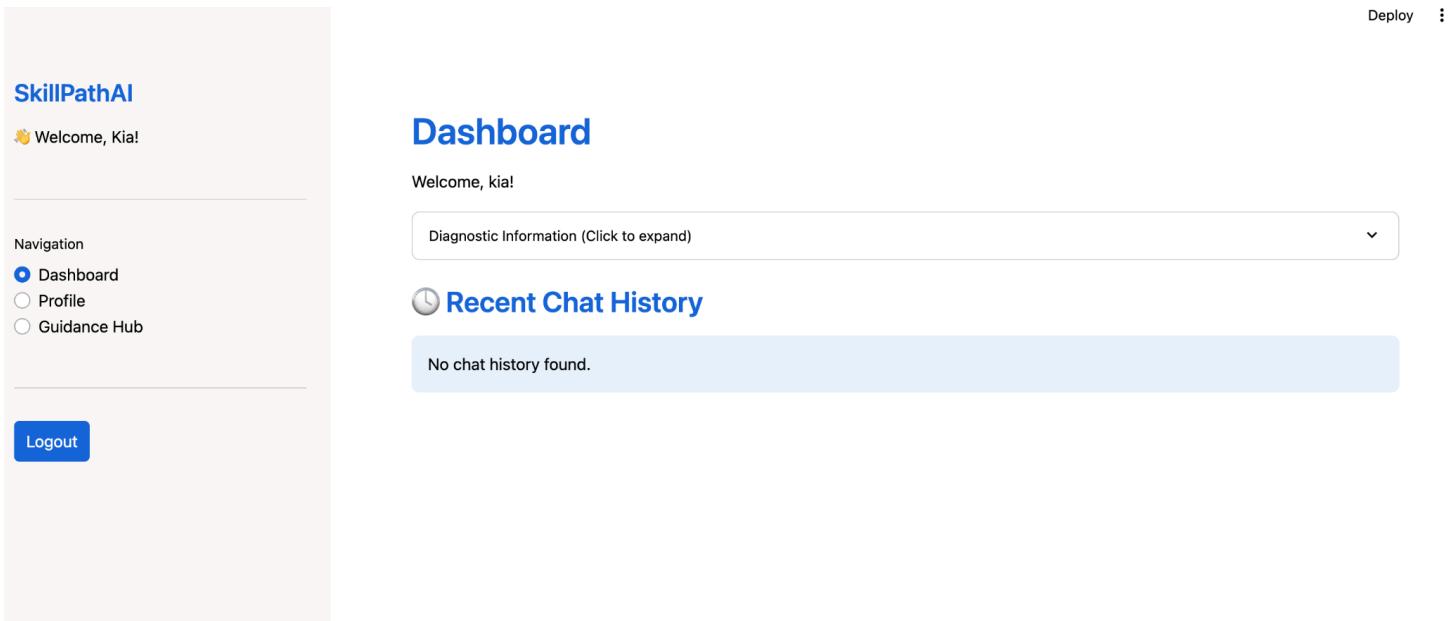
Authentication State Management:

User login sessions are securely managed, ensuring that unauthorized users are redirected to the login page.

2. Dashboard

After successful login, users land on a personalized Dashboard:

- Displays a welcome message with the user's name.
- Presents a Recent Chats section showing previous learning or career planning conversations.
- Each chat card includes:
 - Chat title
 - Timestamp
 - Target role
 - A preview of the last interaction
 - A button to reopen and continue the chat session



A side navigation bar provides access to different sections:

- Dashboard
- Profile
- Guidance Hub
- Logout

The layout emphasizes a clean, card-based design with intuitive navigation.

3. Guidance Hub

The Guidance Hub acts as a central decision point where users choose their upskilling journey:

1. Personalized Learning Path:

Users who want to learn new skills based on a target role can start this flow.

2. Career Transition Assistant:

Users looking to pivot careers can analyze their resume and get a transition plan.

The screenshot shows the SkillPathAI platform's Guidance Hub section. On the left, there's a sidebar with the 'SkillPathAI' logo, a 'Welcome, Kia!' message, and a navigation menu with 'Dashboard', 'Profile', and 'Guidance Hub' options, where 'Guidance Hub' is selected. A 'Logout' button is at the bottom. The main area has a title 'Guidance Hub' with a compass icon. It says 'Choose the tool that best suits your current goals:' followed by two options: 'Personalized Learning Path' and 'Career Transition Assistant'. Each option has a brief description and a blue 'Go to [Tool Name]' button. The 'Deploy' button is in the top right corner of the main content area.

Each option is presented clearly with titles, brief descriptions, and action buttons leading to the respective workflows.

4. Personalized Learning Path Flow

For users who select Personalized Learning Path:

1. Target Role Selection:

Users select the role they aspire to transition into (e.g., Data Analyst, Cloud Engineer).

2. Skill Self-Assessment via Chat:

The chatbot asks users to rate their proficiency in key skills (scale 1–5) relevant to the selected role.

3. Skill Gap Detection:

The system analyzes rated skills to identify areas of improvement.

4. Learning Path Generation:

A personalized roadmap is generated that organizes recommended courses into Beginner, Intermediate, and Advanced phases.

5. Actionable Recommendations:

Includes certifications, project ideas, and communities to join, with a recommended timeline of 3–6 months for completion.

SkillPathAI

👋 Welcome, Kia!

Navigation

- Dashboard
- Profile
- Guidance Hub

[Logout](#)**Chat Controls**[↻ Restart Analysis](#)[☒ End Chat](#)[⬅ Back to Guidance Hub](#)**🎯 Personalized Learning Path**

Diagnostic Information (Click to expand)

🤖 🌟 Hello! I'll help you create a personalized learning path. What's your name?

🤖 Kia

🤖 Nice to meet you, Kia! What career would you like to pursue? (e.g., Data Engineer, Data Scientist, Software Engineer)

🤖 Devops Engineer

🤖 Great! For a Devops Engineer role, these are the top skills you'll need:

- Python
- Kubernetes
- Terraform
- CI/CD

... more

User adding the target role, he or she wanted to pursue

SkillPathAI

👋 Welcome, Kia!

Navigation

- Dashboard
- Profile
- Guidance Hub

[Logout](#)**Chat Controls**[↻ Restart Analysis](#)[☒ End Chat](#)

🤖 How would you rate your proficiency in CI/CD on a scale of 1-5?

1: No experience 2: Basic knowledge 3: Intermediate 4: Advanced 5: Expert

🤖 1

🤖 How would you rate your proficiency in Cloud Security on a scale of 1-5?

1: No experience 2: Basic knowledge 3: Intermediate 4: Advanced 5: Expert

🤖 2

🤖 Thanks for the ratings! Now generating your personalized learning path...

🤖 Your Current Skills Assessment ➡

- Python: ★★★☆☆ (3/5)
- Kubernetes: ★☆☆☆☆ (1/5)
- Terraform: ★☆☆☆☆ (1/5)
- CI/CD: ★☆☆☆☆ (1/5)
- Cloud Security: ★★☆☆☆ (2/5)

Skill Rating for the target role

DevOps - The Complete Guide

SkillPathAI

Welcome, Kia!

Navigation

- Dashboard
- Profile
- Guidance Hub

[Logout](#)

Chat Controls

[Restart Analysis](#)

[End Chat](#)

Platform: Udemy | Level: All Levels

What you'll learn: Learn AWS Cloud infrastructure tool, hugely popular DevOps Engineer, the basics of DevOps Skill., Learn Chef DevOps tool that is gaining a lot of attention lately, Learn Docker and Deploy application anywhere anytime without VM or Cloud.

Key skills:

- AWS, DevOps, Chef, Docker

[Enroll in this course](#)

Get into DevOps: The Masterclass

Platform: Udemy | Level: All Levels

What you'll learn:

- You will learn how to use the most popular DevOps tools, such as: Docker,
- Ansible, Terraform, Jenkins, Chef, Puppet, Kubernetes, You will learn all about
- DevOps concepts such as Continuous Integration, Continuous Delivery and
- Continuous Deployment, You will learn the fundamentals and differences between
- the most popular cloud platforms: AWS Amazon Web Services, Google Cloud
- Platform, Microsoft Azure and Openstack

Key skills:

Course recommendation according to skill gap of the user for the target role

5. Career Transition Assistant Flow

For users who select Career Transition Assistant:

1. Resume Upload:

Users upload their resume (PDF or DOCX format).

2. Skill Extraction:

The system extracts technical and domain-specific skills from the resume using AI.

3. Career Gap Analysis:

Based on the user's chosen target role, missing critical skills are identified.

4. Learning Plan and Strategy:

Suggested learning paths are generated along with strategic advice:

- Resume improvement tips
- Project ideas to strengthen the profile
- Intermediate job roles to target during the transition

- Estimated completion timelines (typically 3–6 months)

SkillPathAI

Welcome, Kia!

Navigation

- Dashboard
- Profile
- Guidance Hub**

Logout

Chat Controls

- Restart Analysis
- End Chat

Career Transition Assistant

Diagnostic Information (Click to expand)

👋 Hello! I'll help you transition to a new career based on your resume. What's your name?

Kia

Nice to meet you, Kia! Please upload your resume so I can analyze your current skills.

Upload your resume (PDF/DOCX):

Drag and drop file here
Limit 200MB per file • PDF, DOCX

Browse files

Anusree Mohanan Resume.pdf 129.8KB

Analyzing your resume...

Uploading Resume by the user

SkillPathAI

Welcome, Kia!

Navigation

- Dashboard
- Profile
- Guidance Hub**

Logout

Chat Controls

- Restart Analysis
- End Chat

Deploy :

• Financial Modeling
• Supply Chain Management
• Inventory Management
• Forecasting
• Predictive Analytics
• Business Intelligence
• Data Storytelling
• Research
• Academic Writing

What career would you like to transition to?

Devops Engineer

Thanks! I'll now analyze the skill gaps between your current profile and the requirements for a Devops Engineer role. This will take a moment...

I've analyzed your path to becoming a Devops Engineer!

Your Skills Assessment for Devops Engineer

Skill assessment of the user based on the target role and the resume given

SkillPathAI

👋 Welcome, Kia!

Navigation

- Dashboard
- Profile
- Guidance Hub

[Logout](#)**Chat Controls**[↻ Restart Analysis](#)[☒ End Chat](#)

Your Devops Engineer Career Transition Path

Based on your background and the skills you need to develop, I've curated this learning path to help you transition into a **Devops Engineer** role successfully. This plan focuses on:

- Building on your existing expertise in Data Engineering
- Developing critical skills in Kubernetes, Docker, Ansible
- Creating a strategic approach to your career transition

The courses below are organized into a structured learning path you can complete in 3-6 months.

Beginner Level (Month 1-2) ↗

Docker, DevOps and Kubernetes for Absolute Beginner [2022]

Platform: Udemy | Level: All Levels

What you'll learn: You will get to learn Docker Architecture., You will learn DevOps tools like docker and kubernetes easily., You will learn Images and containers

Key skills:

Course recommendation from Career Transition Platform

SkillPathAI

👋 Welcome, Kia!



Career Transition Strategy for Devops Engineer

Follow this step-by-step roadmap to maximize your chances of a successful career transition:

1. **Build transferable skills** - Focus first on Kubernetes, Docker, Ansible through the recommended courses above
2. **Create a transition portfolio** - Develop 2-3 projects that showcase your new Devops Engineer skills while leveraging your existing expertise in Data Engineering
3. **Bridge your experience** - Reframe your resume to highlight how your background in Python is relevant to Devops Engineer positions
4. **Network strategically** - Connect with current Devops Engineers on LinkedIn and professional communities to understand the industry's needs
5. **Target transitional roles** - Look for hybrid positions that value both your existing expertise and your new skills

With dedicated effort on this plan, most career changers can successfully transition within 3-6 months.

Do you have any questions about your career transition plan?

Ask a question about your career transition plan or type 'restart' to begin again

**Chat Controls**[↻ Restart Analysis](#)[☒ End Chat](#)

Career transition Strategies

6. AI Chat Assistant

- The GPT-powered chatbot is embedded across the platform to provide conversational support:
- Users can ask questions about learning paths, required skills, certifications, or career strategies.
- Chat history is stored, allowing users to continue past conversations seamlessly.
- Multi-turn context is preserved to maintain personalized assistance throughout the journey.

The screenshot shows the SkillPathAI dashboard. On the left, there's a sidebar with the brand name "SkillPathAI" and a greeting "Welcome, Kia!". It includes a navigation menu with "Dashboard" (selected), "Profile", and "Guidance Hub", and a "Logout" button. The main content area is titled "Dashboard" and features a "Recent Chat History" section. This section lists two previous chats: "Learning Path Chat" and "Career Transition Chat", each with a timestamp and role information. Below this, there's a large preview of a new chat session titled "Next Steps After Completing Your Learning Path". It includes a brief description starting with "Once you've completed the c...*", an "Open Chat" button, and a small icon of a briefcase. To the right, another preview for the "Career Transition Chat" is shown with a similar structure. At the top right of the dashboard, there are "Deploy" and three-dot buttons.

Chat History and Chat preview- Dashboard Page

The screenshot shows the SkillPathAI application. At the top right, there are 'Deploy' and three-dot menu icons. A message bar at the top center asks, 'Do you have any questions about your learning path or career next steps?'. Below this, a user asks, 'can you explain me the last course given by you?'. The AI responds with, 'The last course I mentioned is: Get into DevOps: The Masterclass'. It describes the course as available on Udemy for all levels, with a breakdown of what to expect. It includes a 'Course Description' section and a list of tools learned: Docker, Ansible, Terraform, Jenkins, Chef, Puppet, and Kubernetes. It also mentions DevOps concepts like CI/CD. On the left, a sidebar titled 'SkillPathAI' shows a 'Welcome, Kia!' message, a navigation menu with 'Dashboard', 'Profile', and 'Guidance Hub' (selected), and 'Logout' and 'Chat Controls' buttons.

Navigation

- Dashboard
- Profile
- Guidance Hub

[Logout](#)

Chat Controls

[Restart Analysis](#)

[End Chat](#)

can you explain me the last course given by you?

The last course I mentioned is:
Get into DevOps: The Masterclass

This course is available on Udemy and is designed for all levels, from beginners to advanced learners. Here's a breakdown of what you can expect to learn:

Course Description:

In this masterclass, you'll learn how to use the most popular DevOps tools, such as:

- Docker
- Ansible
- Terraform
- Jenkins
- Chef
- Puppet
- Kubernetes

You'll also learn about DevOps concepts, including:

- Continuous Integration
- Continuous Delivery
- Continuous Deployment

Continuing the previous chat

10.2 User Experience (UX) and Visual Styling

Color Palette:

The interface uses shades of Blue and White for a calm, professional, and clean aesthetic.

Minimalist Layout: Information is presented step-by-step to reduce cognitive load.

Each screen focuses on a single major action (e.g., upload resume, select role, rate skills).

Feedback Mechanisms: Real-time error and success messages guide users throughout the process.

Accessibility: High contrast between text and background, Readable font sizes, Clear call-to-action buttons

Responsiveness: The application is optimized for desktop users, with basic responsive behavior for tablets.

Security and Privacy

SkillPath AI implements foundational security measures focused on authentication, user data protection, and operational logging. While advanced security features are planned, the platform already ensures encrypted password storage, controlled session management, and detailed API activity tracking.

11.1 Authentication and Password Protection

1. Account Registration and Login:

Users can register and log in using a username and password through secure FastAPI endpoints.

2. Password Hashing:

- All user passwords are securely hashed using the bcrypt algorithm before being stored in the Snowflake database.
- Plain-text passwords are never stored or transmitted.

3. Password Reset:

A "Forgot Password" feature allows users to reset their credentials securely. New passwords are re-hashed before updating the user record.

11.2 User Data and Session Management

1. Session State Tracking:

User session states (user ID, username, login status) are managed in the frontend via `st.session_state` to personalize and protect user access.

2. Token Handling:

Basic token generation is implemented after login, though full JWT expiry or validation features are yet to be enforced.

3. User Information Stored:

User profiles are stored securely in Snowflake, containing `user_id`, `name`, `username`, `email`, `hashed_password`, and `created_at`.

11.3 Data Transmission and Storage

1. Communication Security:

Development communication occurs over HTTP, while production deployments are configured for HTTPS to ensure encrypted communication.

2. Database Security:

- User information and session history are stored in Snowflake using standard access protections.

- No advanced database-level encryption or masking is currently implemented.

11.4 Logging and Monitoring

API Request and Response Logging:

- Every incoming API request is logged with details such as:
- HTTP method
- URL endpoint
- Request headers
- Request body (where appropriate)

Responses are logged with:

- Status codes
- Processing times
- Any server errors or exceptions

Performance Monitoring:

- Request timing is captured for every endpoint call.
- Slow responses and server issues are automatically flagged for review.

Log Storage:

- Logs are stored systematically in local log files (logs/api_server.log).
- Log files are rotated periodically to avoid excessive storage buildup.

Security Audit:

Basic auditability is available by tracing API usage, request patterns, and anomalies through the logs.

11.5 Privacy Practices

1. Minimal Data Collection:

Only essential user information and resume/skill-related data is collected to generate personalized learning paths and recommendations.

2. No Third-Party Tracking:

The platform does not use third-party analytics, cookies, advertising SDKs, or user behavior tracking.

3. User Control:

Planned features include allowing users to delete their resume data, skill ratings, and chat history upon request.

DevOps and Deployment

SkillPath AI adopts a containerized deployment strategy using Docker, ensuring consistent environments for development, testing, and future production scaling. The platform architecture is lightweight, modular, and cloud-ready for easy migration to cloud services like AWS, Azure, or GCP when needed.

12.1 Containerization with Docker

1. Docker-Based Environment:

- The entire application stack — frontend (Streamlit) and backend (FastAPI) — is containerized using Docker.
- Containers encapsulate all required dependencies, ensuring that the application runs identically across different machines and environments.

2. Docker Compose Orchestration:

Docker Compose is used to manage multiple services:

- Streamlit app container
- FastAPI backend container
- Services are defined declaratively in docker-compose.yml, allowing one-command orchestration (docker-compose up).

12.2 Local Development Workflow

1. Building Services:

- Images are built locally using Dockerfiles for frontend and backend components.
- Compose orchestration ensures that services can communicate internally (e.g., frontend calling backend APIs).

2. Running Locally:

- Developers can spin up the complete environment with a single command.
- Port mappings expose the Streamlit app (default: localhost:8501) and the FastAPI backend (default: localhost:8000).

3. Environment Variables:

- Secrets such as Snowflake connection settings, API keys, and frontend URLs are injected at runtime using .env files.
- Different .env configurations are supported for development and production environments.

12.3 Cloud Deployment Readiness

1. Cloud-Ready Architecture:

- The Dockerized application is portable to cloud providers like AWS ECS, Azure Container Apps, or Google Cloud Run.
- Minimal adjustments are needed to deploy containers on managed cloud services.

2. Database Connectivity:

- The system is designed to securely connect to Snowflake from both local and cloud deployments.

- Secrets and credentials can be managed using services like AWS Secrets Manager or Azure Key Vault during cloud scaling.

12.4 Current Deployment Approach

1. Manual Deployment:
 - No CI/CD pipelines are configured yet.
 - Code updates and container rebuilds are performed manually during development iterations.
2. Source Code Management:
 - Version control is maintained through Git, ensuring traceability and rollback capability for both backend and frontend changes.
 - Branching strategy supports feature development and bug fixing separately from main deployment branches.

Monitoring and Analytics

SkillPath AI incorporates foundational monitoring and logging systems to track user interactions, API performance, and AI output quality. While full production-grade monitoring tools are planned for future releases, the current setup ensures basic system observability and performance diagnostics.

13.1 Usage Analytics

1. Session Activity Tracking:
 - Tracks major user actions, including:
 - Resume uploads
 - Skill ratings via the chatbot
 - Learning path generation
 - Career transition plan generation
 - Helps assess user engagement across different platform features.
2. Chat Interaction Logging:
 - Records user conversations with the AI assistant.
 - Allows review of the number of chats initiated, continued, or abandoned.
 - Enables the analysis of user intent and common learning or career transition questions.
 - Learning Path Flow Completion:
 - Tracks how many users who start rating skills or uploading resumes complete the learning path or career transition generation.
 - Helps identify bottlenecks or friction points in user flows.

13.2 Performance Monitoring

1. API Request and Response Logging:
 - Every backend API request is logged, capturing:
 - HTTP method and URL
 - Request and response headers
 - Status codes
 - Request body (where appropriate)
 - Response time (latency)
 - Error and Exception Logging:
 - Logs server-side exceptions and errors with detailed tracebacks.
 - Helps detect backend bugs, integration issues, or system downtime early.
2. Processing Time Metrics:
 - Measures time taken for critical operations:
 - Resume parsing
 - Skill extraction
 - Learning path generation
 - Chatbot responses
 - Helps optimize slow endpoints and resource-heavy services.
3. Log File Management:

- Logs are saved systematically under logs/api_server.log.
- Rotating file strategies are applied to prevent excessive file growth.

13.3 AI Output Auditing

1. Prompt and Response Storage:

Captures structured logs for:

- Prompts sent to Cortex COMPLETE
- Prompts sent to GPT models
- Final AI responses generated for skills, recommendations, and career advice

2. Quality Control:

Periodic audits are conducted to:

- Detect hallucinations (incorrect or fabricated AI responses)
- Identify vague or off-topic answers
- Refine and improve prompt templates over time

3. Skill Extraction Validation:

- Skill lists extracted using Cortex COMPLETE are reviewed to ensure they remain aligned with course objectives and industry standards.

Challenges and Lessons Learned

Building SkillPath AI involved navigating real-world technical, data, and user experience challenges. Each obstacle provided critical insights that helped refine the platform's architecture, AI workflows, and user interactions.

14.1 Data Collection and Quality Challenges

Building SkillPath AI involved navigating real-world technical, data, and AI system integration challenges. Each obstacle presented opportunities to rethink design, improve workflows, and build a more resilient and scalable platform.

14.1 Data Collection and Quality Challenges

a) Initial Approach

Early datasets were sourced from Kaggle, covering platforms like Udemy, edX, Udacity, Coursera, and Pluralsight. These were used to set up the initial Snowflake staging tables and ELT pipelines.

b) Challenges Faced

- Missing Skills: Course metadata often lacked explicit skills being taught, making skill extraction incomplete.
- Vague Descriptions: Objective summaries were broad or overly promotional.
- Outdated Metadata: Many courses were obsolete, renamed, or no longer available.
- Incorrect Instructional Levels: Some courses were misclassified (e.g., labeled "Beginner" but required prior knowledge).
- API Rate Limits: Some official course APIs (Coursera, Pluralsight) were either paid, rate-limited, or deprecated.
- Web Scraping Complexity: Course pages had inconsistent structures across platforms, requiring platform-specific scrapers.

The screenshot shows the Udemy Instructor API Documentation (v1.0). At the top, there is a navigation bar with the Udemy logo and the word "Developers". Below the navigation bar, there is a sidebar titled "Overview" which lists various API methods. The main content area is titled "Udemy Instructor API Documentation (v1.0)". It contains a green box with a checkmark icon and text about the creation of a supported set of APIs for instructors. Below this, there is a section titled "Getting Started" with a brief introduction to the API's features and organization. At the bottom of the page, there is a "Creating an API Client" section with some instructions and links.

Udemy API Documentation - Clearly inviting instructors only.

```

571
572 | SELECT * FROM PLURALSIGHT_COURSES LIMIT 5;
573
574
575
576

```

↳ Results ~ Chart

	<u>A</u> LINK	<u>A</u> TYPE	<u>A</u> NAME	<u>A</u> INSTRUCTURE	<u>A</u> LEVEL	<u>A</u> DURATION	<u>L</u> DATE CREATED	# RATING
1	https://www.pluralsight.com/courses	courses	Oracle Database 12c Fundamentals	by Tim Warner	Beginner	3h 42m	0021-10-01	357.0
2	https://www.pluralsight.com/courses	courses	Oracle Database 12c Disaster Recovery	by Tim Warner	Intermediate	3h 26m	0021-10-01	65.0
3	https://www.pluralsight.com/courses	courses	Oracle Database 12c: Installation and Licensing	by Tim Warner	Beginner	2h 47m	0021-10-01	152.0
4	https://www.pluralsight.com/courses	courses	Microsoft Azure Solutions Architect: Designing	by John Savill	Advanced	1h 10m	0020-11-04	35.0
5	https://www.pluralsight.com/courses	courses	Microsoft Azure Solutions Architect: Implementing	by Aaron Sampson	Advanced	1h 17m	0020-11-17	35.0

Inconsistent results after staging Pluralsight data from Kaggle

c) Solutions Adopted

Udemy: Retained the Kaggle dataset but enhanced it using Cortex COMPLETE to infer missing skills, prerequisites, and better objectives.

edX & Udacity: Built custom scrapers to collect real-time metadata directly from their official platforms.

Coursera & Pluralsight: Excluded due to limited access, scraping restrictions, and inconsistent data formats.

Transformation Layer: Used dbt to standardize objectives, levels, and unify datasets into a clean combined model.

LLM-Based Enrichment: Cortex COMPLETE was used to infer structured skills, level tagging, and prerequisite logic from unstructured course text.

```

219
220 | SELECT
221 |   ID,
222 |   COURSE_NAME,
223 |   DESCRIPTION,
224 |   LEVEL,
225 |   SNOWFLAKE.CORETEX.COMPLETE(
226 |     'mistral-large2',
227 |     'Given the following course title, description, and level, extract the most relevant technical or professional skills. \
228 |     If the description lacks detail, use the title to infer skills. \
229 |     Return only a concise, comma-separated list of skills (e.g., Python, SQL, Agile, Data Analysis). \
230 |     Do not include full sentences or general goals. \
231 |     Course Info: || COURSE_NAME || . || DESCRIPTION || . Level: ' || COALESCE(LEVEL, 'Not Specified')
232 |     ) AS INFERRED_SKILLS
233 |   FROM SKILLPATH_DB.RAW_DATA.STG_UDACITY_COURSES
234 |   WHERE (SKILLS IS NULL OR TRIM(SKILLS) = '')
235 |     AND DESCRIPTION IS NOT NULL;
236
237

```

↳ Results ~ Chart

	<u>A</u> ID	<u>A</u> COURSE_NAME	<u>A</u> DESCRIPTION	<u>A</u> LEVEL	<u>A</u> INFERRED_SKILLS
1	bba42c0b-ee3e-4adf-b1cd-7f29bccf5a	Introduction to Javascript Courses	In this course, you will learn the basics of the JavaScript programming language.	Beginner	JavaScript, Data Types, Conditionals, Loops, Functions
2	e85636f3-7d56-4410-a15a-9b80bd23	Preparing and Modeling Data with Python	This course is a crucial step in Python's Data Science ecosystem.	Fluency	Data Modeling, Data Cleaning, Data Transformation, Pandas
3	e10c7da0-331a-4641-9245-5774db8b	Building Generative AI Solutions	This lesson covers vector databases and generative AI models.	Advanced	Vector Databases, AI, Long-term Memory, Retrieval Models
4	0b69f057-5672-4a63-83a1-07b35f7f9	Embedded Systems and Robotics	This course will teach you the basics of embedded systems.	Fluency	Embedded Systems, Electrical Circuits, Digital Logic, Microcontrollers
5	7e1f395d-e493-4d46-a2d9-3d1d33e6	Contrastive Language-Image Processing	In this lesson, you will learn about contrastive learning.	Fluency	CLIP, Machine Learning, Natural Language Processing
6	5e285ab9-ab28-4989-b4af-071d42ae7	UX Fundamentals & Design Research	Learn the fundamentals of User Experience design.	Beginner	UX Design, Design Research, User Experience, User-Centred Design

14.2 AI and RAG Pipeline Challenges

a) Initial Direction

RAG implementation began with evaluating external vector databases like Pinecone and FAISS for semantic retrieval.

b) Challenges Faced

Infrastructure Overhead: Required managing separate storage, embedding APIs, and metadata sync.

Embedding Complexity: Needed to integrate OpenAI/HuggingFace embeddings, manage versioning, and update vector indexes.

Latency + Cost: Introduced increased latency and operational cost.

Prompt Tuning: Fine-tuning prompt templates for skill extraction and course matching was iterative, and required multiple rounds of optimization for consistency.

c) Solutions Adopted

- Cortex Search: Switched to Snowflake Cortex Search for in-warehouse semantic search with minimal setup.
- Native to Snowflake tables
- No external embedding service needed
- Simple SQL interface
- Prompt Design: Prompts for Cortex COMPLETE were modularized, role-aware, and constrained with formatting rules to ensure high-accuracy outputs (e.g., concise skill lists, no hallucinated text).

14.3 User Experience (UX) Challenges

a) Skill Rating Flow

Initial plan involved form-based sliders for users to rate themselves on multiple skills.

b) Issues Faced

- Sliders were unintuitive for rating large skill sets.
- Users dropped off when faced with long, form-like interactions.

c) Solution

- Replaced with a chat-based skill rating system.
- GPT-powered assistant asks skill-by-skill (e.g., “Rate your experience with SQL from 1–5”), making the process more interactive, less overwhelming, and easier to complete.

14.4 Logging and Debugging Challenges

a) Early Limitations

- Lack of real-time error tracking and internal observability made debugging difficult.
- No automated alerts or centralized logs in early development.

b) Solution

- Implemented structured API logging:
- Request URL, method, payload, status code, and latency
- Server-side error logging with stack traces
- Logs stored using rotation under logs/api_server.log
- Used logs to improve performance, identify endpoint bottlenecks, and detect bad inputs.

14.5 Lessons Learned

Data Comes First:

Good AI recommendations are impossible without clean, reliable data. Data quality must be handled before model tuning.

Don't Overengineer Early:

Switching from external vector DBs to Cortex Search removed friction and boosted development velocity.

Build for Modularity:

Using dbt, modular scrapers, and prompt blocks made it easier to adjust logic across services.

Conversational UX Works:

Chat-based flows kept users more engaged than traditional forms — especially in skill assessment.

Logs Are Non-Negotiable:

Early logging investments accelerated debugging, validation, and model output tracing.

Design Decisions & Justification

Throughout the development of SkillPath AI, deliberate design decisions were made to optimize scalability, data quality, AI integration, and user experience. These decisions directly impacted the platform's performance, maintainability, and effectiveness.

15.1 Using Snowflake Cortex Search Instead of Traditional Vector Databases

Challenge:

- Retrieval-Augmented Generation (RAG) typically requires a vector database to store and retrieve semantic embeddings.
- Traditional setups (FAISS, Pinecone) introduce complexity — requiring external vector storage, embedding pipelines, syncing metadata, and managing separate APIs.

Decision:

Use Snowflake Cortex Search instead of a traditional vector database.

Justification:

1. Unified Platform:

Cortex Search operates directly over Snowflake tables, eliminating the need for separate infrastructure.

2. No Embedding Management:

Embeddings are automatically generated using Snowflake's built-in models (snowflake-arctic-embed).

3. Simple Querying:

Semantic search is performed using familiar SQL syntax without complex APIs.

4. Enterprise-Grade Security and Scaling:

Cortex Search benefits from Snowflake's security model, access controls, and auto-scaling features.

5. Development Speed:

Faster iteration cycles without needing to integrate, maintain, and troubleshoot a separate vector store.

15.2 Using LLMs (Cortex COMPLETE) for Data Cleaning and Enrichment

Challenge:

Course datasets from Kaggle lacked consistency: missing skills, vague objectives, and unclear prerequisites.

Decision:

Use Snowflake Cortex COMPLETE for AI-driven data cleaning, skill extraction, and prerequisite inference.

Justification:

- Automated Skill Extraction:
- Inferred missing skills directly from course titles and objective summaries.

Prerequisite Inference:

Standardized prerequisites for beginner, intermediate, and advanced level courses.

Objective Summarization:

Transformed vague marketing texts into concise, skills-focused descriptions.

Quality Control:

Structured prompt templates ensured high consistency and domain relevance in LLM outputs.

15.3 Realigning Data Sources Based on Data Quality

Challenge:

Kaggle course datasets were outdated or incomplete for some platforms (especially edX, Udacity, Coursera, Pluralsight).

Decision:

- Retain Udemy Kaggle dataset (with LLM enrichment).
- Build custom web scrapers for edX and Udacity to collect updated, reliable course metadata.
- Drop Coursera and Pluralsight datasets to avoid compromising quality.

Justification:

Data Accuracy:

Ensured that learning paths are built on accurate, up-to-date course information.

Focused Effort:

Investing in fewer but higher-quality sources led to better downstream AI performance.

15.4 Shifting from Form-Based Skill Rating to Chat-Based Skill Rating

Challenge:

Traditional form sliders for skill rating proved tedious and disengaging for users when dealing with large skill sets.

Decision:

Implement a chat-based skill rating flow where the AI assistant interactively asks users to rate one skill at a time.

Justification:

User Engagement:

Conversational flows are more natural, less overwhelming, and easier to complete.

Personalization:

Ratings can dynamically adapt based on follow-up questions or context.

Consistency:

Provides a human-like assessment experience integrated seamlessly with AI guidance.

15.5 Lightweight DevOps and Dockerized Local Deployment

Challenge:

At early-stage development, setting up full CI/CD pipelines and cloud orchestration would slow down iteration.

Decision:

Use Docker and Docker Compose for local development orchestration without full cloud deployment initially.

Justification:

Rapid Prototyping:

Developers can bring up frontend and backend services with one command.

Environment Consistency:

Ensures that services behave identically across machines.

Cloud-Ready:

Containers are portable to AWS, Azure, or GCP when needed without major rewrites.

15.6 Summary

Strategic decisions — including native semantic search via Cortex, LLM-driven data enrichment, realignment of data sources, chat-based UX flows, and Dockerized deployment — enabled SkillPath AI to deliver a scalable, maintainable, and high-quality personalized learning platform.

Future Scope and Improvements

While SkillPath AI delivers a robust personalized learning and career transition experience today, several improvements are planned to enhance system capabilities, user experience, operational efficiency, and AI performance. The roadmap focuses on scaling the platform both technically and functionally as user adoption grows.

16.1 AI/ML Enhancements

Fine-Tuning and Model Personalization:

- Fine-tune LLMs (such as GPT) on education-specific datasets to improve the relevance of skill suggestions and career advice.
- Implement user-specific personalization layers based on learning history and preferences.

Dynamic Course Recommendation Updates:

- Enable periodic re-ranking of recommended courses based on:
- User progress
- Industry trend changes
- New skill demands
- Multi-Turn Conversation Memory Expansion:
- Improve the chatbot's ability to retain context across multiple sessions and learning journeys.
- Implement long-term memory for recalling user interests, goals, and past queries.

16.2 Data Expansion and Quality

Add More Learning Platforms:

Integrate scraping or API-based ingestion for additional platforms such as:

- LinkedIn Learning
- Skillshare
- FutureLearn
- Google Career Certificates (if feasible)

Advanced Skill Taxonomy Mapping:

- Build a structured skill graph to relate foundational, intermediate, and advanced skills hierarchically.
- Help users visualize progression paths more clearly.

Continuous Dataset Refresh:

Schedule automatic scraping and enrichment pipelines to keep course metadata up to date.

16.3 Platform Features

User Progress Tracking:

- Track completed courses, earned certifications, and project submissions.
- Allow users to set goals, deadlines, and reminders within the app.

Gamification and Motivation Tools:

Introduce badges, streaks, or points for course completions and milestones.

Expanded Chat Capabilities:

Support career exploration questions beyond learning paths — e.g., interview preparation, portfolio advice, etc.

Resume and Portfolio Builder:

Generate updated resumes automatically after learning path completion, showcasing newly acquired skills and certifications.

16.4 DevOps and Deployment Upgrades

CI/CD Automation:

Implement automated build, test, and deployment pipelines using GitHub Actions, Jenkins, or GitLab CI.

Cloud Deployment:

- Deploy frontend and backend services on managed cloud services (e.g., AWS ECS, Azure App Service).
- Migrate Snowflake connection layers to cloud-native secure networking.

Real-Time Monitoring and Alerts:

- Integrate centralized monitoring (e.g., Prometheus + Grafana, AWS CloudWatch).
- Set up automated alerts for API downtime, scraping failures, or LLM service disruptions.

Scalable Load Management:

Container orchestration using Kubernetes (K8s) for future high-traffic scalability.

16.5 Security and Compliance

Full JWT Authentication and Authorization:

Implement token expiration, refresh tokens, and fine-grained route protection.

User Data Deletion and Export:

Allow users to request deletion or download of all their personal data (resume, skill ratings, chats).

GDPR and Data Privacy Compliance:

Align with GDPR and CCPA principles for international user base expansion.

16.6 Strategic Expansion

Career Path Intelligence Layer:

Build predictive models suggesting ideal next roles, based on skill trajectories and market demand.

Partnerships with Learning Platforms:

Collaborate with course providers to offer discounts, certifications, or exclusive project-based content.

Corporate Upskilling Programs:

Extend SkillPath AI to companies looking to build internal employee learning and career mobility programs.

References and Resources

The development of SkillPath AI leveraged a variety of technical resources, datasets, documentation, and tools. Below is a list of key references used throughout the project.

17.1 Official Documentation and Technical References

- [Snowflake Official Documentation](#)
 - [Snowflake Cortex Search Documentation](#)
 - [FastAPI Documentation](#)
 - [Docker Documentation](#)
 - [OpenAI \(ChatGPT\) Documentation](#)
 - [Lucidchart](#) — Diagram creation and architecture visualization
 - [Cursor AI](#) — AI coding assistant
 - Claude AI — AI research reference
 - [Perplexity AI](#) — Research and quick search assistant
-

17.2 Course Datasets and Sources

- **Udemy Dataset:**
[Udemy Courses Meta-Data \(Kaggle\)](#)
- **edX Dataset:**
[edX Courses Dataset \(Kaggle\)](#)
- **Coursera and Online MOOCs:**
[Online MOOCs Dataset \(Kaggle\)](#)
- **Udacity Dataset:**
[Udacity Courses \(Kaggle - within Coursera Dataset\)](#)

- **Official Websites for Scraping/Reference:**

- [edX Official Course Catalog](#)
 - [Udacity Official Course Catalog](#)
-

17.3 Snowflake Marketplace

- **Snowflake Data Marketplace:**

Utilized to explore external datasets and AI services available for potential future integrations.

17.4 Research Papers and Industry Resources

- **Retrieval-Augmented Generation (RAG) in LLMs** — Research articles exploring semantic search + LLM synergy.
- **AI in Education and Skill Development** — Research papers discussing the impact of AI-driven personalized learning.