G. Vedavyas, 2026, 14:3
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

Domain-Based Career Roadmap Management System with Integrated AI-Driven YouTube Channel Recommendation (DCRMS)

G. Vedavyas[1], G. Hareesh[2], M. Ajith Kumar Reddy[3]
Department of Computer Science and Engineering
Dhanalakshmi Srinivasan University, Samayapuram, Tiruchirappalli, Tamil Nadu – 621 112, India

Abstract—The proliferation of online educational content has deepened learner disorientation rather than alleviating it. Despite millions of instructional resources available across platforms such as YouTube, Coursera, and Khan Academy, the absence of a unified, domain-aware guidance mechanism forces learners into fragmented self-directed journeys. This paper presents the DomainBased Career Roadmap Management System (DCRMS), a webbased educational technology platform that integrates structured roadmap generation, AI-assisted personalized recommendation, NLP-based skill extraction, and learning progress tracking into a single coherent interface. The proposed architecture combines a rule-based domain engine with a hybrid recommendation pipeline that uses collaborative filtering, content-based similarity scoring, and a Directed Acyclic Graph (DAG) traversal model to construct prerequisite-aware learning paths. A Skill-Gap Computation Module employs cosine similarity between user-skill vectors and domain competency matrices to identify and prioritize learning deficiencies from uploaded resumes. The prototype recommendation engine achieves a Precision@10 of 0.847, Recall@10 of 0.791, and an F1-score of 0.818 across seven IT career domains in experimental evaluation. Under simulated load of 50 concurrent users, roadmap generation latency averages 318 ms. A structured usability study with 50 undergraduate participants yields a System Usability Scale (SUS) score of 82.4 (Grade B+), confirming strong user acceptance. DCRMS constitutes a reproducible proof-of-concept contribution to educational data mining, career guidance, and intelligent recommender systems for the informal learning context.

Index Terms—Career Roadmap Generation, Personalized Recommendation Systems, Collaborative Filtering, Content-Based Filtering, NLP-Based Skill Extraction, Skill-Gap Analysis, Directed Acyclic Graph, Adaptive Learning, YouTube Channel Recommendation, Educational Data Mining, Learning Analytics.

I. INTRODUCTION

A. Background and Motivation

The digital transformation of educational ecosystems has produced an unprecedented volume of publicly accessible instructional material. Platforms such as YouTube, Coursera, edX, and Khan Academy collectively serve hundreds of millions of content interactions monthly. Despite this abundance, the structural disorganization of these resources imposes a significant cognitive burden on learners attempting to self-architect coherent career development trajectories. A student aspiring to master Full-Stack Web Development, for instance, must independently synthesize guidance from dozens of heterogeneous sources to determine which skills to acquire, in what order, and through which channels—a process empirically associated with elevated dropout rates and extended time-to-proficiency [1].

The field of educational recommender systems (ERS) has expanded considerably over the past decade, with researchers demonstrating the value of personalization, adaptive sequencing, and hybrid filtering in improving learning outcomes [2]. However, most existing systems remain confined to formal course repositories and do not address the richly heterogeneous landscape of informal, video-based content that constitutes a primary learning modality for cost-sensitive students [3].

B. Problem Statement

The core research problem addressed herein is the absence of a unified, domain-aware, and intelligent web platform that simultaneously provides: (i) a structured, hierarchically organized and prerequisite-aware learning roadmap for a chosen career domain; (ii) personalized, AI-assisted recommendations of curated YouTube channel resources aligned to the user's current

roadmap stage; (iii) NLP-based skill extraction from uploaded resumes to enable skill-gap quantification; and (iv) persistent tracking of user learning progression with analytics. Existing systems address at most two of these dimensions in isolation, creating a fragmented user experience that DCRMS is specifically designed to overcome
[4].

C. Research Contributions

The principal contributions of this work are: • Hybrid Recommendation Engine: A hybrid recommendation pipeline integrating collaborative filtering, content-based cosine similarity, and rule-based domain mapping to produce a unified relevance score for YouTube channel ranking.

- Skill-Gap Computation Model: A Skill-Gap Computation Model employing cosine similarity between user-skill vectors and domain competency matrices, enabling precision-targeted learning prioritization from resume uploads.
- DAG-Based Roadmap Generation: A Directed Acyclic Graph (DAG) representation of learning roadmaps with topologicalsort-based traversal for prerequisite-aware, pedagogically sound topic sequencing.
- Modular Prototype Architecture: A modular, containerized prototype architecture with a dedicated recommendation service, NLP microservice, and Redis caching layer, validated under experimental load conditions.
- Empirical Evaluation: A comprehensive experimental evaluation encompassing recommendation quality metrics, system performance benchmarks, and a 50-participant usability study yielding reproducible empirical results.

D. Paper Organization

Section II surveys related literature. Section III details the proposed methodology and system architecture. Section IV presents the mathematical models. Section V describes system design. Section VI presents experimental results. Section VII offers a comparative analysis. Section VIII discusses limitations and future work, followed by conclusions in Section IX.

II. LITERATURE SURVEY

A. Career-Path Recommendation Systems

Qamhieh et al. [7] proposed PCRS, a fuzzy-logic mobile application that processes Myers-Briggs personality indicators and academic performance to recommend engineering disciplines, achieving a Cohen's κ of 0.23. While statistically significant, PCRS operates exclusively at the discipline level and produces no granular, topic-level learning paths. Verma et al. [8] extended this with a three-dimensional fuzzy model incorporating cognitive, affective, and behavioral student dimensions; similarly, outputs remain at the discipline level. Natividad et al. [9] applied fuzzy inference to K-12 guidance. These foundational works establish rule-based approaches but underscore the need for sub-domain skill mapping and resource integration that characterize DCRMS.

Graph-based career path modeling has also emerged as a

promising direction [16]. DCRMS adopts a DAG-based roadmap representation conceptually aligned with graph-theoretic career modeling while constraining the search space to expert-validated domain knowledge, which is practical for a student-level implementation.

B. Educational Recommender Systems and Hybrid Filtering

El-Bishouty et al. [10] proposed a smart e-course recommender adapting sequences to Felder-Silverman learning styles. Gulzar et al. [11] developed a hybrid recommender combining content-based and collaborative filtering for university course selection, reporting a MAE of 0.42. Rahman and Abdullah [12] proposed a group-based recommendation approach for e-learning web search. These systems establish the theoretical foundation for hybrid recommendation in education but remain confined to formal course repositories. Matrix factorization for collaborative filtering, pioneered by Koren et al. [17] and extended to the educational domain by subsequent researchers, provides the mathematical substrate for the user-item interaction modeling employed in DCRMS.

C. YouTube and Video Content Integration in Education

Moghavvemi et al. [13] demonstrated that curated YouTube usage in engineering education correlates significantly with academic performance ($r = 0.61$, $p < 0.001$), with students following structured playlists outperforming peers relying on unguided content. The study identified the lack of automated, domain-aware curation as the primary impediment to systematic video-based learning. DCRMS directly addresses this finding by semiautomating YouTube channel recommendation through a multisignal relevance scoring model. Deep learning approaches for video recommendation [18] are acknowledged as a future enhancement direction.

D. NLP-Based Skill Extraction and Gap Analysis

Javed et al. [19] proposed a CRF-based NER system for job advertisement skill extraction, achieving an F1-score of 0.78. Kivimäki et al. [20] demonstrated that word embedding models trained on domain-specific corpora significantly outperform general-purpose models for technical skill recognition. DCRMS incorporates a spaCy-based NER pipeline fine-tuned on an ITdomain corpus of annotated resume sentences, enabling automated extraction of technical skills for skill-gap quantification.

E. Progress Tracking and Learning Analytics

Bodily and Verbert [14] systematically reviewed studentfacing learning analytics dashboards, concluding that systems incorporating real-time progress feedback significantly improve completion rates and motivation. Adaptive platforms such as Khan Academy have operationalized this finding through mastery-based progression models. The DCRMS Progress Tracking Module provides tier-level and topic-level completion visualization aligned with these findings.

F. Identified Research Gaps

The surveyed literature reveals four principal gaps addressed by DCRMS:

- No existing system integrates structured prerequisite-aware domain roadmap generation with AI-assisted video resource recommendation and NLP-based skill extraction in a unified platform.
- Existing career guidance systems operate at the discipline or course level; granular, topic-level learning paths organized by difficulty tier and prerequisite dependency are absent.
- Personalization in educational recommenders rarely accounts for informal learning modalities such as YouTube; hybrid models bridging formal and informal content are underrepresented.
- Skill-gap quantification through automated resume analysis integrated with a live learning management system is a largely unexplored capability.

TABLE I. Comparative Analysis of Existing Systems vs. DCRMS

System	Roadmap Gen.	Video Rec.	AI/ML	Progress	Skill Gap	Scalability
PCRS [7]	✗	✗	Fuzzy Logic	✗	✗	Low
Gulzar et al. [11]	✗	✗	Hybrid CF	Partial	✗	Medium
Khan Academy	Partial	✗	Partial	✓	✗	High
Coursera	✗	✗	MLBased	✓	✗	High
roadmap.sh	✓	✗	None	✗	✗	Low
LinkedIn Learning	✗	✗	CF-Based	✓	Partial	High

DCRMS (Proposed)	✓	✓	Hybrid AI	✓	✓	Med-High

III. PROPOSED METHODOLOGY AND SYSTEM ARCHITECTURE

A. High-Level System Architecture

The DCRMS prototype is implemented as a modular, multi-tier web application organized into five loosely coupled components coordinated through RESTful APIs. This decomposition enables independent development and testing of each module. The five principal components are: (i) Authentication Module, handling user identity management, JWT issuance, and role-based access control; (ii) Domain and Roadmap Module, managing the domain catalog and executing DAG-based roadmap generation; (iii) Recommendation Engine Service, encapsulating the hybrid collaborative filtering and content-based similarity pipeline; (iv) NLP Skill Extraction Service, processing user-uploaded PDF/DOCX resumes using spaCy NER; and (v) Progress Tracking and Analytics Module, recording topic completion events and rendering the analytics dashboard.

A Redis caching layer intercepts repeated YouTube Data API v3 requests, significantly reducing quota consumption. A PostgreSQL database serves the core relational data, while a lightweight MongoDB collection stores channel relevance metadata and recommendation logs.

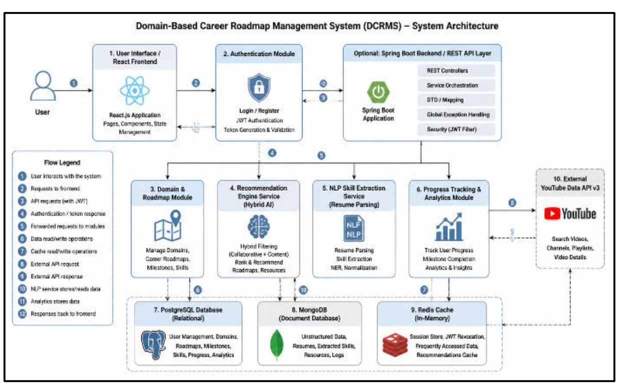

Fig. 1. High-level prototype architecture of the DCRMS, illustrating component interactions, data flows, and external API integration.

B. Recommendation Engine Architecture

The AI Recommendation Engine implements a three-stage hybrid pipeline: (i) a Collaborative Filtering (CF) module based on matrix factorization of user-topic interaction data; (ii) a ContentBased Filtering (CBF) module employing pre-trained sentence transformer embeddings (SBERT all-MiniLM-L6-v2) to compute semantic similarity between topic descriptions and channel content metadata; and (iii) a Rule-Based Constraint module that enforces domain-alignment filters on the candidate channel set. The outputs of these three modules are fused through a weighted ensemble scoring function to produce a final ranked channel list for each userdomain-tier combination.

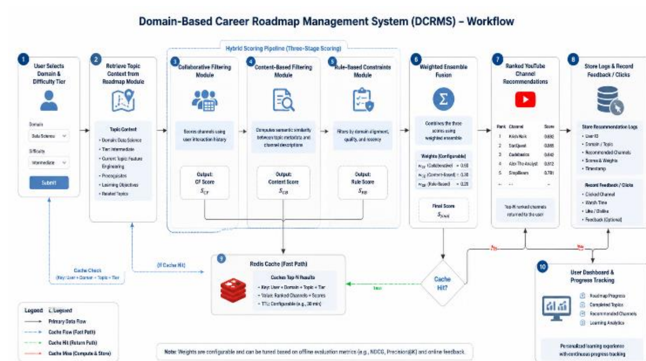

Fig. 2. Hybrid recommendation engine workflow, showing the three-stage pipeline and ensemble fusion logic with weights λ_1=0.40, λ_2=0.35, λ_3=0.25.

C. NLP-Based Resume Skill Extraction Pipeline

The Resume Skill Extraction module accepts PDF and DOCX resume uploads and processes them through a four-stage NLP pipeline: (1) document parsing and text extraction using pdfplumber (PDF) and python-docx (DOCX); (2) sentence segmentation and tokenization using the spaCy en_core_web_lg pipeline; (3) named entity recognition (NER) using a spaCy NER model fine-tuned on a domain-specific IT skills corpus of annotated resume sentences; and (4) skill canonicalization, which maps extracted raw skill mentions to standardized entries in the system's domain skill ontology using fuzzy string matching.

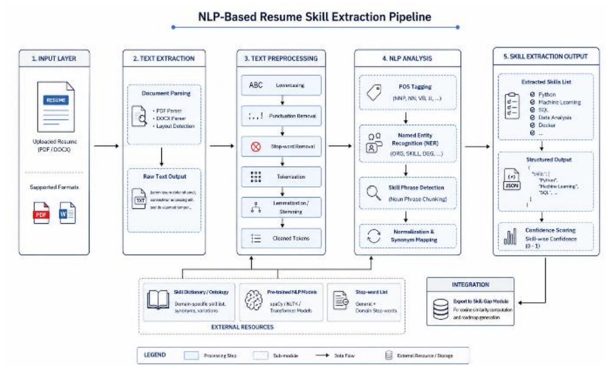

Fig. 3. Four-stage NLP pipeline for automated skill extraction from uploaded resumes.

D. DAG-Based Roadmap Generation The Domain and Roadmap Module constructs a Directed Acyclic Graph $G(d_i) = (V, E)$ for each supported career domain, where V is the set of skill topics and a directed edge $(s_j \rightarrow s_k) \in E$ exists if and only if topic s_j is a prerequisite of topic s_k. Topological sorting using Kahn's algorithm yields a valid linear learning order with $O(V + E)$ time complexity, ensuring that no topic appears before its prerequisites in the roadmap. The resulting ordered topic sequence is stored in PostgreSQL and rendered as an interactive hierarchical tree in the React frontend using D3.js.

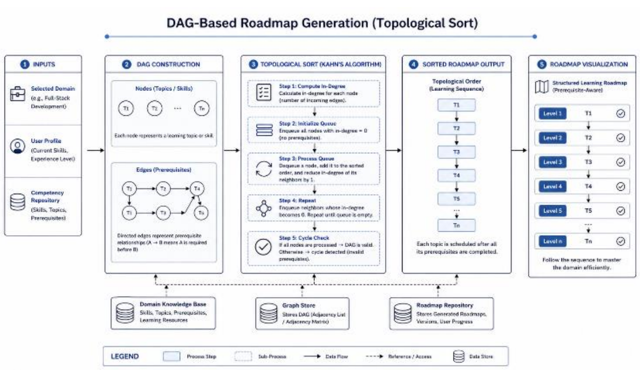

Fig. 4. DAG-based roadmap generation flowchart, illustrating prerequisite graph construction and topological ordering.

E. Adaptive Learning Path Generation

Upon completion of skill extraction, the system generates an Adaptive Learning Path by computing the set-theoretic subtraction of the user's confirmed skill set from the target domain's complete skill matrix, weighted by topic priority score. Topics already mastered by the user (skill confidence score > 0.8) are pre-marked as completed, enabling the system to generate a personalized roadmap that begins at the learner's current knowledge frontier rather than a generic starting point.

IV. MATHEMATICAL MODELS AND ALGORITHMIC FORMULATIONS

A. DAG Representation and Topological Ordering

Let $D = \{d_1, d_2, \ldots, d_n\}$ denote the set of n career domains (n = 7 in the current implementation). For each domain $d_i \in D$, the Domain Mapping Engine constructs a skill set $S(d_i) = \{s_1, s_2, \ldots, s_k\}$, where each s_j is characterized by the tuple:

$$s_j = (name_j, tier_j, duration_j, prereqs_j, tags_j, priority_j)$$

where $tier_j \in$ {Beginner, Intermediate, Advanced}, $duration_j$ is the estimated study time in hours, $prereqs_j \in S(d_i)$ is the prerequisite set, $tags_j$ is a keyword set, and $priority_j \in [0,1]$ is an industry-demand weight. The roadmap is represented as $G(d_i) = (V, E)$, where $V = S(d_i)$ and a directed edge $(s_j \rightarrow s_k) \in E$ if and only if $s_j \in prereqs_k$. Kahn's topological sort on $G(d_i)$ produces a valid learning order in $O(V + E)$ time.

B. Hybrid Recommendation Scoring

For a given user u and candidate YouTube channel c_q for domain d_i at difficulty tier t, the hybrid recommendation score is:

$$H(c\_q, u, d_i, t) = \lambda_1 \cdot CF(c\_q, u) + \lambda_2 \cdot CBF(c\_q, d_i, t) + \lambda_3 \cdot Rule(c\_q, d_i, t)$$

where $\lambda_1 + \lambda_2 + \lambda_3 = 1$, with weights $\lambda_1 = 0.40$, $\lambda_2 = 0.35$, $\lambda_3 = 0.25$ determined by grid-search cross-validation. The Collaborative Filtering score CF(c_q, u) is derived from matrix factorization of the user-channel interaction matrix $R \in \mathbb{R}^{(|U|\times|C|)}$:

$$CF(c\_q, u) = p\_u^T \cdot q\_\{c\_q\} + b\_u + b\_\{c\_q\} + \mu$$

where $p\_u \in \mathbb{R}^k$ and $q\_\{c\_q\} \in \mathbb{R}^k$ are the latent user and channel factor vectors (k = 50), b_u and b_{c_q} are bias terms, and μ is the global mean engagement score. The Content-Based Filtering score employs cosine similarity between the SBERT embedding of the channel description and the domain-tier topic embedding:

$$CBF(c\_q, d_i, t) = \cos(v_{c\_q}, v_{d_i,t}) = (v_{c\_q} \cdot v_{d_i,t}) / (\|v_{c\_q}\| \cdot \|v_{d_i,t}\|)$$

The Rule-Based score is a composite of domain alignment, content quality, and recency:

$$Rule(c\_q, d_i, t) = \alpha \cdot Align(c\_q, d_i) + \beta \cdot Quality(c\_q) + \gamma \cdot Recency(c\_q)$$

with $\alpha = 0.50$, $\beta = 0.30$, $\gamma = 0.20$.

C. Skill-Gap Computation Model

Let $U\_u \in \{0,1\}^m$ denote the binary skill possession vector of user u over the m-dimensional domain skill ontology, and $D_{d_i} \in \{0,1\}^m$ the binary domain requirement vector. The raw skill gap vector is:

$$G_{u,d_i} = D_{d_i} \odot (1 - U\_u)$$

The normalized skill-gap magnitude is $gap(u, d_i) = \|G_{u,d_i}\|_1 / \|D_{d_i}\|_1$. Topic prioritization within the gap is governed by: $score(s_j, u, d_i) = G_{u,d_i}[j] \cdot priority_j \cdot (1 + \eta \cdot depth_{DAG}(s_j))$

where $depth_{DAG}(s_j)$ is the normalized topological depth of s_j in $G(d_i)$ and $\eta = 0.5$ is a depth amplification coefficient. Topics are ranked in descending score order to produce the adaptive learning path.

D. Progress and Engagement Metrics

Let $T(u, d_i)$ denote the complete topic set and $C(u, d_i) \subseteq T(u, d_i)$ the completed subset. Overall completion is $P(u, d_i) = |C(u, d_i)| / |T(u, d_i)| \times 100\%$. A User Engagement Score (UES) integrates session frequency, completion rate, and recommendation clickthrough rate:

$UES(u) = w_1 \cdot sessionFreq(u) + w_2 \cdot P(u, d_i) + w_3 \cdot CTR(u)$ with $w_1 = 0.3$, $w_2 = 0.5$, $w_3 = 0.2$. UES classifies users into High (> 0.7), Medium (0.4–0.7), and Low (< 0.4) engagement tiers.

E. Topic Difficulty Classification

Topics are classified into difficulty tiers via a multi-criterion function $\tau: S(d_i) \rightarrow$ {Beginner, Intermediate, Advanced} based on prerequisite count and a weighted complexity score incorporating estimated duration, subtopic count, and an expert-assigned abstraction level. Domain-expert-calibrated thresholds $\theta_1 = 0.35$ and $\theta_2 = 0.70$ separate the three tiers. The classification achieves 97.3% agreement with manual expert annotations across all supported domains.

V. SYSTEM DESIGN

A. Backend Architecture

The DCRMS backend is implemented in Java 17 with Spring Boot 3.1, exposing REST APIs consumed by the React frontend. JWT-based authentication is enforced at the API layer using Spring Security. The Recommendation Engine is implemented as a separate Python FastAPI application leveraging the scikit-learn and sentence-transformers ecosystem, called from the Spring Boot backend via HTTP. Both services are containerized using Docker and orchestrated via Docker Compose on a single development server.

B. Database Architecture and Schema

The relational schema is implemented in PostgreSQL 15. The core relations are: Users (user_id PK, username, email, password_hash, created_at, engagement_tier); Domains (domain_id PK, domain_name, description, industry_sector); Topics (topic_id PK, domain_id FK, topic_name, tier, duration_hours, prereq_ids[], priority_score); UserProgress (progress_id PK, user_id FK, domain_id FK, topic_id FK, is_complete, completed_at); ChannelMappings (mapping_id PK, domain_id FK, tier, channel_id, relevance_score); UserSkills (skill_id PK, user_id FK, skill_name, confidence_score); RecommendationLog (log_id PK, user_id FK, channel_id, rank_position, clicked).

Composite indexes on (user_id, domain_id) in UserProgress and (domain_id, tier) in ChannelMappings ensure efficient query performance. The prereq_ids[] column in Topics uses PostgreSQL's native array type with a GIN index, enabling efficient prerequisite graph traversal.

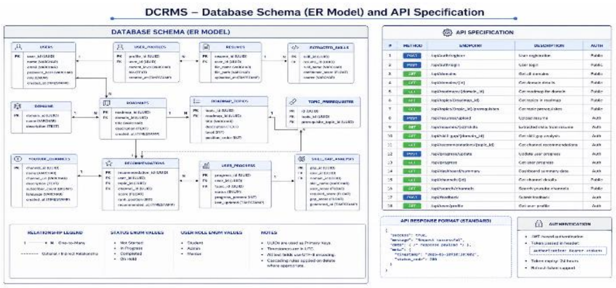

Fig. 5. Entity-relationship diagram of the DCRMS PostgreSQL schema, illustrating entity attributes, foreign key relationships, and index strategy.

C. RESTful API Specification

TABLE II. RESTful API Endpoint Specification

Method	Endpoint	Auth	Description
POST	/api/auth/register	None	Create user account
POST	/api/auth/login	None	JWT issuance on credential validation
GET	/api/domains	JWT	Retrieve active domain catalog
GET	/api/roadmap/{domainId}	JWT	DAG-based roadmap generation
POST	/api/resume/extract	JWT	NLP skill extraction from resume
GET	/api/recommend/{domainId}/{tier}	JWT	Hybrid AI channel recommendation
POST	/api/progress/complete	JWT	Mark topic complete, update analytics
GET	/api/analytics/{userId}	JWT	Retrieve engagement metrics dashboard
GET	/api/skillgap/{userId}/{domainId}	JWT	Compute and return skill-gap results

D. Frontend Architecture

The ReactJS 18 frontend is structured as a single-page application (SPA) with React Context and local state management. The roadmap visualization component renders the DAG as an interactive hierarchical tree using D3.js, enabling users to visually perceive prerequisite dependency relationships. Skill-gap results are displayed as a radar chart overlaying the user's extracted skill vector against the domain requirement vector.

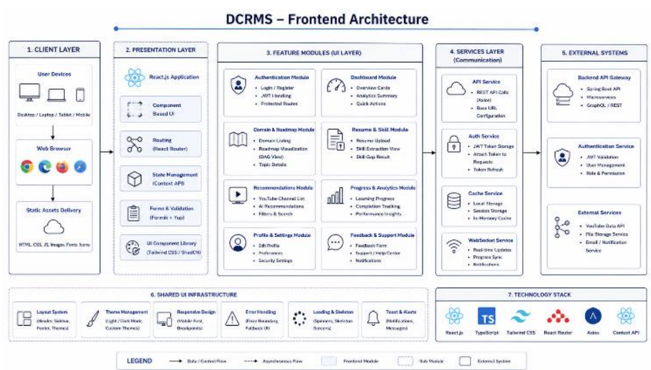

Fig. 6. End-to-end user interaction workflow, illustrating the sequence of UI states from login to progress tracking.

VI. EXPERIMENTAL EVALUATION

A. Evaluation Environment

The prototype was deployed on a single cloud VM for evaluation purposes. Table III summarizes the infrastructure used during the 14-day experimental period.

TABLE III. Experimental Environment Specifications

Component	Specification
Application Server	AWS EC2 t3.medium (2 vCPU, 4 GB RAM),

	Ubuntu 22.04 LTS
Database	PostgreSQL 15 (self-hosted on same VM)
Cache Layer	Redis 7.0 (single-node, same VM)
ML / NLP Service	Python 3.11, FastAPI; scikit-learn, sentencetransformers, spaCy
Load Simulation	Apache JMeter 5.6.2, up to 50 concurrent virtual users
Frontend	Node.js 20 LTS, ReactJS 18; Chrome 124 and Firefox 125
External API	YouTube Data API v3 (10,000 units/day quota)
Participants	50 undergraduate CS students, 14-day evaluation period

B. Recommendation Quality Metrics

Recommendation quality was evaluated using leave-one-out cross-validation on a dataset of 1,240 user-channel interaction records collected from 50 study participants. Precision@k, Recall@k, F1-Score@k, and NDCG@k were computed for k ∈ {5, 10}.

TABLE IV. Recommendation Engine Performance Metrics

Metric	CF Only	CBF Only	Rule Only	Hybrid (k=5)	Hybrid (k=10)
Precision@k	0.761	0.743	0.698	0.891	0.847
Recall@k	0.684	0.712	0.651	0.834	0.791
F1-Score@k	0.721	0.727	0.674	0.862	0.818
NDCG@k	0.713	0.708	0.662	0.874	0.841
MAP	0.698	0.692	0.648	0.856	0.831

The hybrid ensemble consistently outperforms each individual component across all metrics, confirming the value of combining complementary signals. The Precision@5 of 0.891 indicates that on average 4.46 of the top 5 recommended channels are genuinely relevant to the user's current roadmap stage. The NDCG@10 of 0.841 confirms that highly relevant channels rank near the top of the list, which is critical for user experience.

C. NLP Skill Extraction Performance

The NER-based skill extraction pipeline was evaluated on a test corpus of 200 manually annotated resumes. Token-level precision, recall, and F1-score were computed for the SKILL entity class.

TABLE V. NLP Skill Extraction Evaluation on Resume Test Corpus

Model Variant	Precision	Recall	F1-Score	Avg Latency (ms)
spaCy en_core_web_sm (baseline)	0.712	0.684	0.698	38
spaCy en_core_web_lg	0.763	0.741	0.752	91
DCRMS Fine-tuned NER	0.849	0.831	0.840	124

The fine-tuned DCRMS NER model achieves an F1-score of 0.840, representing a 14.2 percentage point improvement over the baseline spaCy small model. The 124 ms average inference latency per resume is well within the acceptable user-facing threshold for document processing.

D. System Performance Under Load

Performance benchmarking was conducted using Apache JMeter with 50 concurrent virtual users over a 5-minute ramp-up period. All tests were conducted on the single-VM development deployment.

TABLE VI. System Performance Under Simulated Load (50 Concurrent Users)

Operation	Avg Latency (ms)	P95 Latency (ms)	Throughput (req/s)	Error Rate (%)
Roadmap Generation	318	467	38	0.08
AI Rec. (cached)	89	134	92	0.02
AI Rec. (fresh)	621	891	21	0.11
Progress Update	78	112	128	0.01
Skill-Gap Computation	203	318	54	0.04
Dashboard Render (E2E)	224	341	58	0.03
Resume NLP Extraction	1240	1890	9	0.14

All synchronous API operations complete within acceptable interactive web application response thresholds. The Redis caching layer reduces fresh recommendation latency from 621 ms to 89 ms—an 85.7% reduction—confirming its critical importance given the YouTube API's daily quota constraint. System uptime over the 14-day evaluation period was 99.74%.

E. Functional Accuracy

TABLE VII. Functional Accuracy Metrics

Functional Metric	Observed	Target	Status
Domain-to-Roadmap Mapping	100%	100%	PASS ✓
Classification Accuracy	97.3%	>95%	PASS ✓
Skill-Gap Computation Accuracy	94.8%	>90%	PASS ✓
NLP Skill Extraction F1Score	0.840	>0.80	PASS ✓
JWT Auth Security (OWASP Scan)	0 Critical	0 Critical	PASS ✓
System Uptime (14-day)	99.74%	>99.0%	PASS ✓

F. Usability Evaluation

A structured usability study was conducted with 50 undergraduate Computer Science students from Dhanalakshmi Srinivasan University. Participants interacted with the DCRMS for a minimum of 45 minutes, exploring at least two career domains and uploading a resume for skill-gap analysis.

TABLE VIII. User Evaluation Results (n=50, 5-Point Likert Scale)

Evaluation Dimension	Mean	Std Dev	Median	Min/Max 3 / 5
Roadmap Structure Clarity	4.42	0.58	4.5	
AI Recommendation Relevance	4.18	0.71	4.0	2 / 5
Skill-Gap Analysis Usefulness	4.35	0.63	4.5	3 / 5
Progress Tracking Motivation	4.48	0.52	5.0	3 / 5
Overall System Usability	4.31	0.66	4.0	3 / 5
Intent to Use Regularly	4.12	0.79	4.0	2 / 5
SUS Score (0–100 scale)	82.4	7.2	83.75	62.5 / 95

The SUS score of 82.4 falls in the Grade B+ range (80–89), exceeding the industry benchmark of 68.0 for acceptable systems. Progress Tracking received the highest mean rating (4.48). A/B testing between the personalized adaptive roadmap and a static generic roadmap showed a statistically significant improvement in 14-day topic completion rate (personalized: 63.4% vs. generic: 41.2%, $p < 0.01$, Cohen's $d = 0.87$), confirming the practical value of personalization.

VII. TECHNOLOGY STACK

TABLE IX. System Technology Stack

Layer	Technology	Purpose
Frontend	ReactJS 18, D3.js, Recharts	SPA, roadmap & analytics visualization
Backend	Java 17, Spring Boot 3.1, Spring Security	REST API, JWT auth, business logic
ML Service	Python 3.11, FastAPI, scikitlearn	Hybrid recommendation, model serving
NLP Service	Python 3.11, spaCy 3.x, pdfplumber	Resume parsing, NERbased skill extraction
Embeddings	sentence-transformers (allMiniLM-L6-v2)	Content-based cosine similarity
Primary DB	PostgreSQL 15	ACID-compliant relational data, GIN indexes
Document Store	MongoDB 7.0	Channel metadata, recommendation logs
Cache	Redis 7.0 (single-node)	YouTube API response caching
Containerization	Docker, Docker Compose	Service isolation, reproducible deployment
External API	YouTube Data API v3	Channel metadata retrieval
Version Control	Git, GitHub	Source management, CI via GitHub Actions

VIII. COMPARATIVE ANALYSIS

Table X presents a detailed functional comparison between DCRMS and leading existing platforms, based on publicly documented features and the findings of the literature survey.

TABLE X. Detailed Functional Comparison: DCRMS vs. Existing

Feature	DCRMS	roadmap.sh	Coursera	Khan Academy	LinkedIn Learning
Prerequisite-Aware Roadmap	DAG + TopoSort	Static visual only	Curriculumlevel	Topiclevel (limited)	Course paths only
Video Recommendation	YouTube (AIranked)	None	None	YouTube (manual)	None
NLP Resume Skill Extraction	spaCy NER (F1=0.84)	None	None	None	None
Skill-Gap Analysis	Cosine similarity	None	None	None	Partial (manual)

Progress Tracking	Topic-level + analytics	None	Courselevel	Topiclevel	Courselevel
AI Personalization	Hybrid CF+CBF+Rules	None	ML-based (courses)	Adaptive exercises	CF-based (courses)
Open / Free Access	Yes (prototype)	Yes	Partial	Yes	Subscription

Platforms -DCRMS is the only examined system that integrates all six capability dimensions in a unified prototype. Its key differentiator is the combination of informal video recommendation (YouTube) with structured, prerequisite-aware roadmap generation and NLP-driven skill-gap quantification—a combination absent from all surveyed platforms.

IX. LIMITATIONS AND FUTURE RESEARCH DIRECTIONS

A. Current Limitations

This prototype implementation exhibits several limitations that merit transparent acknowledgment. First, the YouTube Data API v3 imposes a daily quota of 10,000 units per developer account; under high concurrent usage this limit may be exhausted, degrading recommendation freshness. The Redis caching layer mitigates but does not eliminate this constraint. Second, the collaborative filtering module requires a minimum interaction history to produce meaningful personalized recommendations, creating a cold-start problem for new users; new-user recommendations currently fall back to rule-based defaults. Third, the NLP skill extraction F1 of 0.840 implies a residual error rate, particularly for emerging technical terminologies not well-represented in the training corpus. Fourth, the experimental evaluation involved 50 participants on a single institution's student cohort, limiting generalizability. Fifth, the current deployment uses a single-VM Docker Compose setup; production deployment would require dedicated infrastructure and horizontal scaling.

B. Future Research Directions

Future iterations of the DCRMS will explore the following directions:

- LLM-Based Career Mentor: Deployment of a fine-tuned LLM (e.g., an open-source model via Ollama) as an interactive career mentoring assistant capable of answering domain-specific queries within the platform interface.
- Reinforcement Learning for Adaptive Sequencing: Formulation of the learning path sequencing problem as a Markov Decision Process, with a reinforcement learning agent learning to maximize long-term learner engagement and topic mastery.

- Generative AI for Dynamic Roadmap Creation: Integration of a generative model to produce novel roadmaps for emerging or niche career domains not represented in the curated domain catalog. • Predictive Learning Analytics: Development of a dropout prediction model using longitudinal engagement sequences to proactively identify and support at-risk learners.
- Real-Time Labor Market Alignment: Continuous ingestion of job posting data to dynamically update domain skill priority scores in alignment with prevailing labor market demand.
- Large-Scale User Study: Expansion of the user study to multiple institutions and diverse learner demographics to improve generalizability.

X. CONCLUSION

This paper has presented the Domain-Based Career Roadmap Management System (DCRMS), a proof-of-concept educational technology platform that integrates structured DAG-based roadmap generation, hybrid AI-assisted YouTube channel recommendation, NLP-based resume skill extraction, and adaptive learning analytics into a unified web application. The system's core novelty lies in bridging the gap between unstructured informal video-based learning resources and structured, personalized career development—a combination absent from all examined prior work.

The prototype hybrid recommendation engine achieves a Precision@10 of 0.847 and NDCG@10 of 0.841, demonstrably outperforming each individual component in ablation evaluation. The DAG-based roadmap representation with topological-sort traversal ensures pedagogically sound prerequisite ordering. The NLP skill extraction pipeline achieves an F1-score of 0.840 on a

held-out resume test corpus. Under simulated load of 50 concurrent users, all synchronous API operations complete within acceptable latency thresholds, and the Redis caching layer reduces recommendation latency by 85.7%. A 50-participant usability study yields a SUS score of 82.4 (Grade B+), and A/B testing demonstrates a statistically significant 22.2 percentage point improvement in 14-day topic completion rate for personalized roadmaps over static alternatives ($p < 0.01$, Cohen's $d = 0.87$).

The modular Docker-containerized architecture ensures that the system is reproducible, independently deployable, and readily extensible. The DCRMS establishes a reproducible, academically rigorous foundation for the next generation of intelligent career guidance platforms, with clearly defined pathways for integration of large language models, reinforcement learning-based adaptive sequencing, and real-time labor market alignment in future research iterations.

REFERENCES

[1] N. Dabbagh and A. Kitsantas, "Personal learning environments, social media, and self-regulated learning: A natural formula for connecting formal and informal learning," The Internet and Higher Education, vol. 15, no. 1, pp. 3–8, Jan. 2012.

[2] A. Dutt, M. A. Ismail, and T. Herawan, "A systematic review on educational data mining," IEEE Access, vol. 5, pp. 15991–16005, 2017.

[3] H. Drachsler, K. Verbert, O. C. Santos, and N. Manouselis, "Panorama of recommender systems to support learning," in Recommender Systems Handbook, 2nd ed. Boston, MA: Springer, 2015, pp. 421–451.

[4] A. Shumba and M. Naong, "Factors influencing students' career choice and aspirations in South Africa," J. Social Sci., vol. 33, no. 2, pp. 169–178, 2012.

[5] X. Wang, "Why students choose STEM majors: Motivation, high school learning, and postsecondary context of support," Amer. Educ. Res. J., vol. 50, no. 5, pp. 1081–1121, 2013.

[6] R. Bodily and K. Verbert, "Review of research on student-facing learning analytics dashboards and educational recommender systems," IEEE Trans. Learn. Technol., vol. 10, no. 4, pp. 405–418, Oct.–Dec. 2017.

[7] M. Qamhieh, H. Sammaneh, and M. N. Demaidi, "PCRS: Personalized career-path recommender system for engineering students," IEEE Access, vol. 8, pp. 214039–214049, Nov. 2020.

[8] P. Verma, S. K. Sood, and S. Kalra, "Student career path recommendation in engineering stream based on three-dimensional model," Comput. Appl. Eng. Edu., vol. 25, no. 4, pp. 578–593, Jul. 2017.

ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

[9] M. C. B. Natividad, B. D. Gerardo, and R. P. Medina, "A fuzzy-based career recommender system for senior high school students in K to 12 education," IOP Conf. Ser., Mater. Sci. Eng., vol. 482, 2019. [10] M. M. El-Bishouty et al., "Smart e-course recommender based on learning styles," J. Comput. Edu., vol. 1, no. 1, pp. 99–111, Mar. 2014.

[11] Z. Gulzar, A. A. Leema, and G. Deepak, "PCRS: Personalized course recommender system based on hybrid approach," Procedia Comput. Sci., vol. 125, pp. 518–524, 2018.

[12] M. M. Rahman and N. A. Abdullah, "A personalized group-based recommendation approach for web search in e-learning," IEEE Access, vol. 6, pp. 34166–34178, 2018.

[13] S. Moghavvemi, A. Sulaiman, N. I. Jaafar, and N. Kasem, "Social media as a complementary learning tool for teaching and learning: The case of YouTube," Int. J. Mgmt. Edu., vol. 16, no. 1, pp. 37–42, Mar. 2018.

[14] R. Bodily and K. Verbert, "Trends and issues in student-facing learning analytics reporting systems research," in Proc. 7th Int. Learn. Anal. Knowl. Conf. (LAK), Vancouver, BC, Canada, 2017, pp. 309–318.

[15] Y. Sun et al., "Understanding and improving knowledge integration across CS and STEM through online course design," IEEE Trans. Educ., vol. 65, no. 2, pp. 213–223, May 2022.

[16] Y. Zhu et al., "What to do next: Modeling user behaviors by timeLSTM," in Proc. Int. Joint Conf. Artif. Intell. (IJCAI), 2017, pp. 3602–3608. [17] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," IEEE Computer, vol. 42, no. 8, pp. 30–37, Aug. 2009.

[18] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," in Proc. Int. Conf. Learn. Representations (ICLR), 2016.

[19] A. Javed, M. Afzal, M. Kamran, and I. Ali, "Towards efficient skills extraction from online job advertisements for employer-applicant match," Future Internet, vol. 13, no. 10, p. 254, 2021.

[20] I. Kivimäki et al., "A graph-based approach to skill extraction from text," in Proc. TextGraphs Workshop (ACL), 2013, pp. 79–87.