**FedBridge - Bridging the Skill Gap**

Saim Sheikh, Dat Tri Tat, Khang Chung, Rusheek Ratna Reddy G

*#First-Third Department, First-Third University  
Address Including Country Name*

1first.author@first-third.edu

*\*Second Company  
 Address Including Country Name*

2second.author@second.com

***Abstract*—** *Transition from federal service to non-federal careers presents enormous challenges to terminated government professionals, particularly due to differences in skill terminology, communication expectations, and hiring standards. This paper presents FED BRIDGE, an Internet-based reskilling and career transition platform specifically designed for terminated federal employees. By harnessing AI-driven recommendation software and real-time labor market analytics, FED BRIDGE enables users to map their federal background—i.e., GS grades, security clearances, and policy area specialization—to sector-applicable skills. The platform uses MongoDB to keep structured user data in storage and uses state-of-the-art AI models (OpenAI) to recommend transferable job postings, personalized learning sequences from platforms such as Coursera and LinkedIn Learning and virtual interview simulations with corporate-aligned criticism. Key elements include a Federal Skill Decoder to decode government terminology, an AI Pathway Engine to generate career roadmap plans, and a mentorship network to pair transitioning users with successful private-sector professionals. FEDBRIDGE closes the skill-translation gap and allows laid-off federal workers to adapt and thrive in high-paced corporate environments.*

***Keywords***— ***Federal workforce transition, AI reskilling platform, skill translation, MongoDB schema, interview simulation, government-to-corporate transition, labor market analytics, personalized learning paths, digital career coaching, FEDBRIDGE*.**

1. **Introduction**

The constantly changing employment situation in the United States has witnessed extraordinary changes over the past few years, courtesy of economic pressures, financial constraints, and organizational reorganization that resulted in displacing a massive reservoir of federal workers. Despite their high degree of experience in fields such as procurement, cybersecurity, administration, and policy implementation, the majority of terminated federal workers encounter ongoing barriers to re-employment in the private sector. These are not due to incompetence, but structural mismatches between skills developed in the government and private-sector employment needs. Common issues are under-leveraging security clearances, not being able to describe bureaucratic success in corporate terms, and not knowing about modern technical stacks or agile development methodologies.

To combat this increasingly prevalent problem, we present FEDBRIDGE—a spectacular reskilling and transition platform that makes it easy for retired federal workers to move directly into competitive opportunities in the private sector. FEDBRIDGE employs a hybrid technology framework that combines MongoDB-based profile management with advanced artificial intelligence (AI) functionality powered by large language models such as OpenAI. The platform translates user-provided data, including GS-level title, clearance level, and specialized regulatory knowledge, into portable work qualifications matched to the present labor market demand. In addition, FEDBRIDGE develops skill translation to individualized upskilling pathways via carefully curated online courses and simulated interview environments adjusted to private-sector requirements. On top of this, users are provided with networks of successfully transitioned federal professionals, further amplifying community-driven knowledge transfer. In enabling this process, FEDBRIDGE aims to fill the employment gap, making highly competent, displaced federal workers able to rapidly transition, reskill, and thrive in evolving industry ecosystems..

1. **Technology Stack**

The FEDBRIDGE platform is built on a modular and scalable architecture based on frontend and backend technologies coupled with AI services and a NoSQL database. The core components of the technology stack are as follows:

* **Frontend**: Developed using **React.js**, providing a dynamic, responsive, and user-friendly interface optimized for both desktop and mobile experiences.
* **Backend**: Implemented with **Node.js** and **Express.js**, ensuring efficient API handling and seamless communication between client and server layers.
* **Database**: **MongoDB** is used for managing unstructured user data, such as prior federal roles, security clearances, and skill sets. The document-oriented structure supports flexible schema evolution.
* **AI Integration**: Leveraging **OpenAI GPT-4** **LLMs**, the system performs:
  + Skill translation and role recommendation based on labor market data.
  + Generation of customized learning paths via third-party content providers.
  + Interview question generation and feedback tailored to specific roles and industries.
* **Labor Market Data**: Real-time analytics are obtained through integration with public APIs such as **LinkedIn Jobs API**, **Indeed API**, and **O\*NET** for accurate demand prediction and skill alignment.
* **Course Recommendation Engine**: Connected to APIs from **Coursera**, **Udemy**, and **LinkedIn Learning**, enabling curated learning experiences based on detected skill gaps.  
    
  A diagram of a software application

  AI-generated content may be incorrect.  
    
  **Fig. 1.** FEDBRIDGE system workflow

This stack was selected to ensure scalability, modularity, and extensibility, supporting future integration with government datasets or additional AI models.

1. **Modules**

The FEDBRIDGE platform is conceived as a suite of interconnected functional modules, each of which is optimized to deal with a key aspect of the federal-to-private career transition. The modules are developed for modularity to enable independent updating and scaling. While independent, all the modules interact through a centralized MongoDB database and are integrated under a unified React-based frontend. Table I provides an overview of the modules, their roles, and the key technologies used.

A. Resume Parsing Module

Resume upload functionality is the first touch point for the users. This module supports file formats of PDF and DOCX. Using AFFINDA API, the system parses structured information from resumes like job titles, work experience, education, certifications, and technical skills. The structured information is encoded in a machine-readable format and stored in the backend database to serve as the foundation for downstream modules like skill analysis, course recommendation, and job-matching.

B. Skill Gap Analyzer

Following resume data parsing, the Skill Gap Analyzer goes through the user's existing competencies and aligns them with a curated dataset of private-industry job requirements. Semantic matching technology and natural language processing enable the system to find gaps in obsolete or missing skills. Each gap found is assigned a relevance score that is weighted based on job demand signals from providers such as LinkedIn and Glassdoor. The ranking feeds the course recommendation engine and personalized career pathways.

C. Course Recommendation Engine

To bridge the identified skill gaps, this module communicates with APIFY for web scraping and Coursera, Udemy, and LinkedIn Learning external APIs. The engine recommends personalized online learning modules that correspond to the user's missing competencies. The recommendations are prioritized based on skill gap severity, course rating, and predicted course length. These courses are stored in the user profile for tracking progress and feedback.

D. Interview Simulation Engine

Perhaps the biggest challenge for transitioning federal employees is mastering private-sector interview styles and communication norms. This module uses Lang Chain and OpenAI's GPT-4 to provide realistic interview simulations, including behavioral and technical questions for desired positions. Participants are given AI-driven coaching to help turn bureaucratic jargon into crisp, results-driven responses familiar in the corporate job market.

E. Job Search Aggregator

The Job Search Aggregator connects to RAPIDAPI to fetch live job listings, ordered by job title, location, and required qualifications. It also fetches salary data from Glassdoor along with job market trends so recommendations are feasible as well as competitive. Job listings are shown within the frontend dashboard, filtered to align with the user's growing skill set.

F. User Profile Manager

This module is the data repository for each individual user. It stores parsed resume information, skill gap metrics, course completion, and interview performance. The backend for profile storage is MongoDB, enabling real-time profile updates and dashboard personalization. The user profile evolves in real time as new information is received, enabling career paths and recommendations to be dynamically modified.

Table I. Core Modules of FEDBRIDGE

|  |  |  |
| --- | --- | --- |
| **Module Name** | **Function** | **Key Technology** |
| Resume Parsing | Extracts structured data from uploaded resumes | AFFINDA API |
| Skill Gap Analyzer | Identifies and scores skill deficiencies | Custom NLP Model + MongoDB |
| Course Recommendation | Suggests online learning paths based on gaps | APIFY + Coursera/Udemy API |
| Interview Simulation | Provides real-time interview practice and coaching | LangChain + GPT-4 |
| Job Search Aggregator | Displays live job listings by location and title | RAPIDAPI + Glassdoor/LinkedIn |
| User Profile Manager | Maintains and updates user data and progress | MongoDB |

1. **Database & Authentication**

Individualized suggestions, progress tracking, and individualized interaction in the FEDBRIDGE platform can be achieved only if supported by a valid and secure data infrastructure. In this section, the database schema implemented at the back-end, authentication techniques utilized, and the rationale for choosing the respective technologies are elaborated.

A. Database: MongoDB

FEDBRIDGE keeps semi-structured and unstructured user data in MongoDB, a NoSQL document-oriented database. MongoDB's schema-free data model is best suited for the wide variety of user data like resumes, skills extracted, job search history, course recommendations, and interview feedback without relational joins.

Every user action, like uploading a resume or finishing a course, initiates changes in one or more of the collections in the database. ObjectID is utilized by the system as the ID for every document, and field indexing like user\_id, role, and skills is done for quick retrieval.

Table II: Key MongoDB Collections in FEDBRIDGE

|  |  |  |
| --- | --- | --- |
| **Collection Name** | **Purpose** | **Example Fields** |
| user\_profiles | Stores core user information and preferences | user\_id, email, clearance\_level, GS\_grade |
| parsed\_resumes | Contains extracted resume data | skills, experience, education, job\_titles |
| skill\_gaps | Stores identified gaps between user skills and targets | missing\_skills, gap\_score, target\_role |
| course\_tracks | Tracks recommended and completed courses | course\_id, platform, progress, completed\_at |
| interview\_sessions | Stores Q&A transcripts and coaching feedback | questions, user\_response, AI\_feedback |
| job\_search\_logs | Logs user job searches and applied jobs | job\_title, location, applied, timestamp |

**B. Authentication**

## To provide secure and scalable user authentication, Firebase Authentication is employed in FEDBRIDGE. It accommodates a variety of sign-in methods like traditional email/password credentials, OAuth 2.0 providers like Google and LinkedIn, and anonymous guest access for testing or low-usage applications.

## Firebase Authentication also offers the following main advantages:

## • Token-Based Authorization: JWT is issued upon successful login. The token is safely added to frontend requests and authenticated on backend endpoints for secured resource access.

## • Role-Based Access Control (RBAC): During user registration time, the roles would be customer, manager, or admin. The roles are used to define access privileges in various modules, following secure segregation of functions.

## • Password Security: Firebase automatically enforces industry-standard security measures such as bcrypt hashing and credential salting. This provides strong protection from typical authentication attacks.

## The authentication service is tightly coupled with frontend and backend for single sign-on session management, enforcement of user roles, and data privacy compliances throughout the FEDBRIDGE platform.

## ****V. DEPLOYMENT****

### To facilitate scalability, reliability, and maintainability, FEDBRIDGE is hosted on a modern, containerized, cloud-native architecture. The pipeline for release employs container orchestration, managed cloud services, and automated CI/CD processes to enable frictionless releases and operational efficiency.

### ****A. Deployment Infrastructure****

FEDBRIDGE is deployed on the Google Cloud Platform (GCP) on its managed computer, storage, and networking. The frontend application, written in React.js with Tailwind CSS, is deployed on a GCP Cloud Storage Bucket set up to host static websites. It communicates with a Node.js backend hosted by Google Kubernetes Engine (GKE), which delivers automated containerized service orchestration and load balancing.

All the services are containerized with Docker to facilitate duplication of the development environment, staging environment, and production environment. GKE offers the capacity to auto-scale the system horizontally according to real-time usage to provide low downtime as well as optimal utilization of resources.

The backend talks to MongoDB Atlas, a cloud-based database, to store and fetch user profiles, resume data parsed, course suggestions, and system logs. There's also Firebase Authentication implemented in the deployment, and yes, it does provide secure identity management among all the users.

**B. Continuous Integration and Deployment (CI/CD)**

To automate testing and deployment, **GitHub Actions** is used as the CI/CD pipeline. The system follows a GitOps workflow where code changes pushed to the main branch automatically trigger the following stages:

1. **Build:** Docker containers are built from the latest codebase.
2. **Test:** Test: Unit testing and integration testing are run to verify logic and behavior of services.
3. **Deploy:** Upon successful testing, containers are deployed to GKE nodes and the frontend is uploaded to GCP.

### This approach ensures rapid development cycles, less human error, and rapid rollback on failure at deployment time.

### ****C. Modular Deployment Architecture****

The architecture is built around loosely coupled services interacting via RESTful APIs and shared database endpoints. Authentication service (Firebase) and data access service (MongoDB) are given center service status, with frontend, backend, and AI components running independently in container pods. The modularity makes it easy to replace or scale services individually. A diagram of a software flow

AI-generated content may be incorrect.

**Fig. 2** illustrates the full deployment architecture, including service interactions, hosting platforms, and CI/CD flow.

### ****D. Monitoring and Scaling****

## GKE is set up with horizontal pod autoscaling (HPA) to scale the replica count of the container automatically up and down on CPU and memory levels. Google Cloud Monitoring provides performance telemetry, and Google Cloud Logging collects service logs for debugging and audit.

## This deployment strategy makes the system highly available, operationally resilient, and low-latency, thereby enabling FEDBRIDGE to support thousands of users simultaneously using various roles and devices.

## ****V. TESTING****

### A rigorous testing approach was followed to verify that all the modules of the FEDBRIDGE platform perform uniformly, securely, and effectively under different use cases. Here, we present the objectives, steps, and outcomes of unit, integration, and end-to-end testing processes conducted during the development process.

### ****A. Testing Objectives****

The major aims of the test phase were:

* To provide functional correctness of individual components (e.g., resume parsing, skill mapping, course recommendation).
* To ensure seamless data flow and module interactions across the platform.
* To ensure that security elements like authentication and role-based access control (RBAC) had been enforced appropriately.
* To test the credibility of AI-produced content like career path recommendations and interview practice sessions.
* To confirm frontend responsiveness and API consistency under simulated usage conditions.

### ****B. Unit Testing****

The unit tests were established to check whether stand-alone functions and modules work as expected:

* The **Resume Parser** module was tested using sample PDF/DOCX files to ensure consistent data extraction (job titles, skills, education).
* The **Skill Gap Analyzer** logic was tested for correct computation of missing skills and relevance scoring.
* API endpoints for registration, login, and data retrieval were tested using **Postman** and **Mocha/Chai**.
* Frontend components built with **React** were tested using **Jest** and **React Testing Library** to verify UI logic and state updates.
* The **Firebase Auth integration** was tested by simulating token validation and expiration scenarios.

### ****C. Integration Testing****

To ensure that modules communicated effectively with each other, a series of integration tests were conducted:

### Confirmed that resumes uploaded were parsed and saved properly in MongoDB Atlas, and that the parsed information-initiated skill gap analysis and course generation processes.

### Verified that the AI-driven interview questions were drawn from saved user profile information.

### Artificial multi-step processes such as:

### Login → Resume Upload → Skills Analysis → Course Suggestions → Practice Interview

### Confirmed database updates and user state persistence following every significant interaction.

### ****D. End-to-End Testing (E2E)****

End-to-end testing was done with complete user flows to make sure real usage would be seamless and faultless. Test cases were:

* **New user sign-up and login, checking Firebase token issuance and secured route protection.**
* **Upload of resumé and verifying that data extracted was displayed on the user dashboard.**
* **Obtaining the tailored course suggestions and ensuring completion progress updates were mirrored in the backend.**
* **Participation in AI-led interview sessions and review of the feedback stored in the user profile.**

**These workflows were tested using Cypress for automation and manual testing via the deployed frontend and Postman.**

**E. Performance Testing**

Though not the focus, some rudimentary performance testing was conducted to gauge system responsiveness under normal use:

* Resume parsing and course recommendation average API response time was under 500ms.
* MongoDB read/write operations were benchmarked on large profile data using Indexing and ObjectID-based retrieval.
* Firebase login response time averaged <200ms, even under concurrent requests.
* Frontend page loading time was under 2.5 seconds, satisfying Google Lighthouse standards.

**F. Results Summary**

The outcome of the core test cases run during the development process is shown in Table III. All the tests were designed to guarantee a particular module or process of the FEDBRIDGE system and were run using a mix of automated and manual testing tools.  
  
  
Table III. Test Case Summary for FEDBRIDGE Modules

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Modules Covered** | **Status** | **Tools Used** |
| Resume Upload & Data Extraction | Resume Parser, MongoDB | Pass | Postman, AFFINDA |
| Skill Gap Analysis with Missing Skills | Skill Analyzer, Course Recommender | Pass | Mocha/Chai |
| Role-Based Access Enforcement | Firebase Auth, Backend APIs | Pass | Manual, JWT Validation |
| Interview Simulation with AI Feedback | GPT-4, LangChain, Profile Manager | Pass | Manual Testing |
| Full Workflow (Login → Job Apply) | All Frontend + Backend Modules | Pass | Cypress + Manual |
| Concurrent Login + Resume Upload | Auth, Resume, MongoDB | Pass | JMeter (light test) |

## ****VI. CONCLUSION****

In this paper, FEDBRIDGE, an AI-powered, cloud-native platform, was introduced to help displaced federal workers bridge into private-sector careers. Through its combination of resume parsing, skill gap analysis, course recommendation, and AI-powered interview preparation, FEDBRIDGE tackles the distinct issues of federal professionals—outdated language, security clearance translation, and bureaucratic communication patterns.

The architecture leverages modern technologies including React.js, Node.js, Firebase Authentication, MongoDB Atlas, and container orchestration with Google Kubernetes Engine (GKE). With OpenAI GPT-4 and LangChain, context-aware AI response for job matching and mock interviews are enabled. Through rigorous testing—unit, integration, and end-to-end verification—the platform has demonstrated resilience, stability, and usability in critical functionalities.

FEDBRIDGE effectively bridges the gap in language and skills between government roles and industry requirements, thereby increasing the employability and versatility of its users in a competitive labor market.  
  
A graph with orange bars

AI-generated content may be incorrect.

**Fig. 3** User interactions across platform modules during pilot testing

**VII. FUTURE WORK**

Although the initial rollout of FEDBRIDGE is fulfilling its main goals, several aspects have been pinpointed for potential future development:

* Federated Learning for Privacy-Protection Profiles: Use decentralized training models to enable personalized recommendations without uploading private resume information to central servers.
* Improved Analytics Dashboard: Provide an admin dashboard for platform administrators to see user success rates, skill progress indicators, and course completion heatmaps.
* Voice and Video Interview Simulation: Incorporate speech recognition and live voice feedback into the interview module for enhanced realism and accessibility.
* Multilingual Support: Enable resume parsing, recommendation, and interview simulation in languages other than English to support a more diverse group of users.
* Integration with Federal Workforce Systems: Interface with current government job websites (e.g., USAJobs.gov) for easy data import and displaced worker job tracking.

These enhancements will also continue to make the platform more pertinent, scalable, and inclusive for its intended group.

**References**

[1] P. Cappelli, "Skill Gaps, Skill Shortages, and Skill Mismatches," ILR Review, vol. 68, no. 2, pp. 251–290, 2015.

[2] R. Bollinger et al., "Leveraging Information Technology to Bridge the Health Workforce Gap," Bulletin of the World Health Organization, vol. 91, no. 11, pp. 890–894, 2013.

[3] A. Foroughi, "Supply Chain Workforce Training: Addressing the Digital Skills Gap," Higher Education, Skills and Work-Based Learning, vol. 10, no. 4, pp. 601–612, 2020.

[4] B. Hoadley et al., "Addressing the Specialist Workforce CAMH Skill-gap: Providing and Evaluating ICAMH Training in Fiji and Papua New Guinea," Australasian Psychiatry, vol. 27, no. 6, pp. 579–583, 2019.

[5] "Technician Training Bridges the Skill Gap," Industrial and Commercial Training, vol. 38, no. 7, pp. 370–373, 2006.

[6] "Training Fills the Recruitment Skill Gap," Industrial and Commercial Training, vol. 39, no. 2, pp. 98–101, 2007.

[7] T. M. Nesari, "Skill Development and Ayurveda Sector: Bridging the Gap of Workforce in India," International Journal of Ayurveda Research, vol. 14, no. 1, pp. 45–50, 2023.

[8] D. C. Stamps and C. Lovetro, "Building Community Partnerships Through Targeted Workforce Development," Nursing Management, vol. 51, no. 10, pp. 38–44, 2020.

[9] K. W. Smith, T. E. Gibbs, and O. Khan, "Workforce in Training," Delaware Journal of Public Health, vol. 8, no. 3, pp. 36–39, 2022.

[10] O. Adeyinka, "The Skills Gap in AI-Driven IT Auditing: Implications for Workforce Training and Development," World Journal of Research and Review, vol. 19, no. 2, pp. 20–25, 2024.