



*Green University of Bangladesh*

*Department of Computer Science and Engineering (CSE)  
Semester: (Spring, Year: 2025), B.Sc. in CSE (Day)*

---

# **AI-Powered Next-Gen Job Recommendation System Using NLP and Machine Learning**

---

*Lab Report: 05  
Course Title: Integrated Design Project II  
Course Code: CSE - 406  
Section: 213 - D6*

## Students Details

<b>Name</b>	<b>ID</b>	<b>Role</b>
Irteja Mahmud	213902016	Leader
Pankaj Mahanta	213902002	Member-1
Nazmul Hasan	213902003	Member-2

*Submission Date: 12 - 04 - 25  
Course Teacher's Name: Babe Sultana*

[For teachers use only: **Don't write anything inside this box**]

<u><b>Lab Project Status</b></u>	
<b>Marks:</b>	<b>Signature:</b>
<b>Comments:</b>	<b>Date:</b>

# 1 Introduction

The AI-Powered Next-Gen Job Recommendation System (NGJRS) leverages Natural Language Processing (NLP) and Machine Learning (ML) to enhance job matching. It provides personalized job recommendations by analyzing user profiles, resumes, and job listings. The system offers real-time job search, automated CV parsing, employer-candidate matching, and a user-friendly interface. By integrating external job boards and continuously learning from user interactions, NGJRS ensures accurate and efficient job recommendations, making the hiring process faster and more effective for both job seekers and employers.

This project bridges the gap between candidates and recruiters by offering a data-driven solution that minimizes human bias and inefficiencies in the job market. The design prioritizes scalability and modularity, enabling future integration with mobile platforms and third-party APIs.

## 2 Objectives

The major objectives of this project are:

- **Develop an AI-Based Job Recommendation System:** Implement an intelligent recommendation engine that utilizes NLP and machine learning to enhance job matching accuracy.
- **Improve Candidate-Job Suitability:** Use BERT embeddings and semantic similarity techniques to ensure that job recommendations align with candidate skills and preferences.
- **Optimize the Recruitment Process:** Provide recruiters with a ranked list of potential candidates based on job requirements, reducing manual filtering efforts.
- **Enhance User Experience:** Design an intuitive user interface for job seekers and employers, ensuring ease of use and seamless navigation.
- **Integrate External Job Data Sources:** Aggregate job listings from various external job boards to provide users with up-to-date employment opportunities.
- **Implement Role-Based Access Control:** Ensure secure access and functionality based on user roles such as job seeker, recruiter, and admin.
- **Enable Resume Parsing and Skill Extraction:** Automatically extract key skills and experience from uploaded resumes to improve recommendation precision.

These objectives are aimed at streamlining the end-to-end recruitment workflow, improving hiring outcomes, and providing a robust foundation for further AI enhancements in future phases.

### 3 System Development Tools and Rationale

#### a. Identification of Tools

The development of the AI-Powered Next-Gen Job Recommendation System involved a wide array of tools and technologies, spanning frontend and backend development, machine learning, and database management. The integration of these tools allowed for the successful construction of a full-stack, data-driven web application. The following tools were utilized:

- HTML, CSS (Tailwind), JavaScript – for frontend structure, styling, and interactivity.
- PHP – for backend logic and server-side scripting.
- MySQL – for relational data storage and retrieval.
- Python (pandas, scikit-learn) – for machine learning-based job recommendation.
- AJAX – for seamless dynamic updates without page reloads.
- Visual Studio Code (VS Code) – primary code editor and IDE.
- XAMPP – for hosting and running the local server (Apache, MySQL, PHP).
- phpMyAdmin – for managing the MySQL database through a graphical interface.

#### b. Description and Justification of Each Tool

- **HTML, CSS (Tailwind), JavaScript:** These technologies formed the backbone of the user interface. HTML provided the structure, CSS (using Tailwind) ensured rapid styling using utility classes, and JavaScript added dynamic interactivity such as sidebar filters, form validation, and modal popups. Tailwind was selected over traditional CSS frameworks for its flexibility and minimalist design.
- **PHP:** Used for backend processing such as handling login, registration, job posting, profile setup, and session management. Its compatibility with XAMPP and phpMyAdmin made local testing efficient. PHP's mature ecosystem also allowed rapid development using built-in methods.
- **MySQL:** A structured, scalable relational database management system (RDBMS) used to store user information, resumes, job postings, applications, and bookmarks. Its schema-based architecture made it ideal for managing complex relationships across different entities in the system.
- **Python (pandas, scikit-learn):** Python scripts were used for the recommendation engine. Using TF-IDF vectorization from scikit-learn and cosine similarity, the system compared job descriptions to user profiles to deliver relevant job suggestions. Pandas helped in data manipulation, reading from CSVs, and preparing data for model input/output.

- **AJAX:** Enabled asynchronous communication between the frontend and backend. It was used particularly for job filtering, job pagination, saving bookmarks, and updating dynamic content without full page reloads, improving user experience and responsiveness.
- **Visual Studio Code (VS Code):** Chosen as the primary development environment for its IntelliSense support, built-in terminal, Git integration, debugging features, and extension support for both PHP and Python.
- **XAMPP:** Provided a local server environment with support for Apache (web server), MySQL (database), and PHP (scripting). It allowed the team to simulate real-world hosting scenarios during development and testing phases.
- **phpMyAdmin:** Simplified database management by offering a GUI to interact with the MySQL database. Allowed the team to inspect tables, run queries, manage foreign key relationships, and back up data during development.

These tools were selected based on ease of integration, performance, community support, and suitability for a university-level full-stack AI project. Together, they enabled the team to build a feature-rich, scalable, and maintainable web platform.

## 4 Conclusion & Discussion

The AI-Powered Next-Gen Job Recommendation System (NGJRS) successfully demonstrates the application of artificial intelligence in improving digital recruitment workflows. By combining Natural Language Processing (NLP), machine learning, and real-time web technologies, the system achieves intelligent job matching while simplifying the user experience for both job seekers and employers.

The system architecture was divided into modules that handle resume upload, registration, job posting, recommendation logic, profile completion, and search filtering. Each module was rigorously tested using a structured test case strategy. Over 10 core modules were validated with corresponding screenshots and actual UI results to ensure functional correctness and robustness.

This project also served as a practical demonstration of how different technologies—frontend tools (HTML/CSS/JS), backend logic (PHP), databases (MySQL), and machine learning (Python)—can work cohesively in one ecosystem. The recommendation engine, in particular, proved to be accurate and flexible, adapting to real job metadata and skill keywords.

Through collaborative effort, structured testing, and modular implementation, the system met its objectives of improving job-candidate matching and reducing recruiter workload. Additionally, role-based navigation and dynamic UI rendering ensured that the platform remains user-centric and secure.

### Future Improvements:

- Integrate OAuth 2.0 (Google/LinkedIn) for streamlined and secure login.

- Add resume parsing using spaCy or other NLP models to automatically extract skills and experience.
- Extend the recommendation engine with deep learning models (e.g., BERT-based embeddings).
- Build interactive dashboards for recruiters to track application analytics.

In summary, NGJRS offers a scalable foundation for AI-driven hiring systems and provides valuable insights into how machine learning and web development can be used to address real-world employment challenges.