



*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: 01  
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: 08 - 02 - 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>

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Overview . . . . .	2
1.2	Motivation . . . . .	2
1.3	Problem Domain . . . . .	2
1.4	Objectives . . . . .	3
1.5	Scope of the Project . . . . .	3
1.6	Conclusion . . . . .	4
<b>2</b>	<b>System Features</b>	<b>5</b>
2.1	Functional Requirements . . . . .	5
2.2	Non-Functional Requirements . . . . .	7
<b>3</b>	<b>Model Selection</b>	<b>9</b>
3.1	Model Selection . . . . .	9
3.1.1	Selected Model: Agile Model . . . . .	10
3.1.2	Application to the Project . . . . .	10
<b>4</b>	<b>Discussion</b>	<b>11</b>

# Chapter 1

## Introduction

### 1.1 Overview

The rapid expansion of the job market and the increasing volume of job postings have made it challenging for job seekers to find suitable opportunities efficiently. Traditional job-matching platforms rely on basic keyword-based filtering, which often leads to irrelevant recommendations, making the job search process time-consuming and ineffective. To address these inefficiencies, this project introduces the **AI-Powered Next-Gen Job Recommendation System (NGJRS)**, leveraging Natural Language Processing (NLP) and Machine Learning (ML) techniques to provide intelligent, personalized job recommendations.

### 1.2 Motivation

The traditional job-seeking process is inefficient, as candidates often have difficulty finding opportunities that align with their skills and aspirations. Many job portals overlook the nuances of job descriptions and resumes, resulting in mismatches. Additionally, recruiters struggle to identify suitable candidates amid the flood of applications, many of which do not meet job requirements.

### 1.3 Problem Domain

Traditional job-matching systems have several shortcomings:

- **Inefficiency in Job Matching:** Existing platforms rely on simple keyword-based filtering, which often results in inaccurate or irrelevant recommendations.
- **Challenges in Niche Skill Matching:** Specialized skills are often overlooked due to the lack of contextual understanding in job descriptions and resumes.
- **Lack of Context-Aware Recommendations:** Traditional systems fail to interpret user intent, career aspirations, and skill progression, leading to poor job suggestions.

- **Scalability Issues:** Current job platforms struggle to handle large-scale data efficiently while maintaining recommendation accuracy.
- **Employer Recruitment Challenges:** Employers face difficulties in shortlisting candidates efficiently due to a lack of AI-driven filtering mechanisms.

## 1.4 Objectives

The primary 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.
- **Ensure Scalability and Real-Time Performance:** Develop a system capable of handling large volumes of job listings and user profiles while maintaining fast response times.
- **Integrate External Job Data Sources:** Aggregate job listings from various external job boards to provide users with up-to-date employment opportunities.

## 1.5 Scope of the Project

The AI-Powered Next-Gen Job Recommendation System is designed to support multiple stakeholders, including job seekers, employers, and administrators. The key features and scope of the system include:

- **Job Seekers:**
  - Create and manage profiles, including educational background, skills, and work experience.
  - Upload and parse resumes for automated job recommendations.
  - Receive AI-powered personalized job suggestions based on profile data and career preferences.
  - Apply for jobs through the system and track application statuses.
- **Employers:**

- Post job listings with detailed descriptions and required qualifications.
  - Receive AI-generated candidate suggestions for open positions.
  - Shortlist candidates based on automated scoring and ranking systems.
- **System Capabilities:**
    - Leverage NLP models like BERT for job-description and resume matching.
    - Implement real-time job search and filtering based on location, salary, skills, and job type.
    - Provide an admin dashboard to manage user activity, job postings, and system performance.
    - Ensure data privacy and security compliance, including encryption and access control mechanisms.

## 1.6 Conclusion

This project aims to transform job matching by utilizing advanced AI and NLP technologies to provide intelligent, accurate, and personalized job recommendations. By tackling major recruitment challenges, the AI-Powered Next-Gen Job Recommendation System will enhance job discovery, improve hiring efficiency, and create a more dynamic employment ecosystem.

# Chapter 2

## System Features

### 2.1 Functional Requirements

#### User Registration and Authentication (URA)

- The system must allow job seekers and employers to create accounts.
- It must provide authentication features for login and password recovery.
- The system should support OAuth for social media sign-ups (e.g., Google, LinkedIn).

#### Profile Management (PM)

- Job seekers must be able to create and update their profiles with relevant information (education, experience, skills, etc.).
- Employers should be able to post and update job listings.
- The system must allow both job seekers and employers to upload documents such as resumes and job descriptions.

#### CV Generation and Upload (CV GU)

- The system must provide a feature for job seekers to generate a CV based on the profile information entered.
- Users should also be able to upload their existing CVs.
- Based on the uploaded or generated CV, the system must analyze the content and recommend relevant jobs to the user using NLP techniques.

#### Job Search and Filtering (JSF)

- Job seekers must be able to search for jobs using filters like location, job type, salary range, skills, etc.

- Employers should be able to search for candidates using filters like qualifications, experience, and location.

### **Job Recommendation (JR)**

- The system must provide personalized job recommendations for job seekers based on their profiles and CVs.
- It should suggest potential candidates to employers based on the job listing criteria.

### **Notification System (NS)**

- The system must send email notifications to job seekers for new job recommendations or when they are shortlisted by employers.
- Employers should receive notifications when new candidates matching their job postings are available.

### **Admin Dashboard (AD)**

- An admin panel must be available to manage user accounts, job listings, and profiles.
- The system must allow admins to view system statistics such as the number of active users, job postings, and applications.
- Admins should have access to system logs for monitoring performance and security incidents.

### **Job Listings Aggregation (JLA)**

- The system must integrate external APIs to provide real-time job updates, ensuring that job seekers have access to the latest job postings.
- The job listings must be categorized for easy filtering based on industries, job types, and locations, offering a seamless browsing experience for users.

## 2.2 Non-Functional Requirements

### Performance Requirements

- The system must handle up to 1000 concurrent users without a significant drop in response times. (Connected to URA, AD)
- Job searches and job recommendations should be processed and delivered within a few seconds of the user query. (Connected to JR, JSF)
- The backend must efficiently handle CV analysis and job classification using NLP, ensuring minimal latency during data processing. (Connected to CV GU, AD)

### Safety Requirements

- The system must prevent data loss or corruption in case of system crashes or network failures by implementing regular data backups. (Connected to PM, AD)
- Any actions that modify or delete sensitive information such as user profiles, CVs, or job listings must require additional authentication. (Connected to URA, PM)
- The system should include fail-safe mechanisms, including system monitoring and error logging. (Connected to AD)

### Security Requirements

- All sensitive user data, including CVs, profiles, and login credentials, must be encrypted both at rest and in transit. (Connected to CV GU, URA)
- The system should implement role-based access control (RBAC) to restrict access to sensitive information based on user roles. (Connected to AD, PM)
- Regular security audits must be conducted to detect vulnerabilities. (Connected to URA, AD)

### Software Quality Attributes

- **Adaptability:** The system must be adaptable to future updates. (Connected to JSF, JLA)
- **Usability:** The user interface should be intuitive. (Connected to CV GU, JSF)
- **Reliability:** The system should maintain 90% uptime. (Connected to all functional requirements)
- **Scalability:** The system must be able to scale horizontally. (Connected to JSF, JLA, AD)



- **Interoperability:** The system should be compatible with external platforms like LinkedIn and GitHub. (Connected to JLA, URA)
- **Maintainability:** The codebase must be modular and well-documented. (Connected to all functional requirements)

### **Interoperability Requirements**

- The system should support integration with third-party APIs for authentication. (Connected to URA, AD)
- It should allow importing user data such as CVs from external job portals or personal websites. (Connected to CV GU, PM)

### **Ethical Requirements**

- The AI algorithms used for job recommendations must be free of biases. (Connected to JR)
- Transparency in how job recommendations are made is crucial to build trust with users. (Connected to JR, AD)

### **Legislative Requirements**

- The system must adhere to employment laws in the countries or regions where it operates. (Connected to PM, JLA, AD)

### **Privacy Requirements**

- Users must be given control over the visibility of their data. (Connected to PM, URA)
- The system must include clear consent options for collecting, storing, and using user data. (Connected to URA, AD)

# Chapter 3

## Model Selection

### 3.1 Model Selection

#### Comparison Matrix with Different Models

The following table provides a refined comparison of various SDLC models based on key criteria that are crucial for the success of the **AI-Powered Next-Gen Job Recommendation System**. The weightage of each criterion has been adjusted to reflect its significance in our project.

Table 3.1: Updated Comparison Matrix of SDLC Models for NGJRS.

Priority	Criteria	Waterfall	V-Shape	Iterative	Spiral	Agile	Prototype
10	User-Centric Design	No	No	Yes	Yes	<b>Yes</b>	Yes
10	User Feedback Integration	No	No	Yes	Yes	<b>Yes</b>	Yes
10	Flexibility for Changes	No	Yes	Yes	Yes	<b>Yes</b>	No
9	Risk Management	No	No	No	Yes	No	No
9	Adaptability to New Technologies	No	No	Yes	Yes	<b>Yes</b>	Yes
8	Integration with Machine Learning	No	No	Yes	Yes	<b>Yes</b>	No
8	Time to Market	No	No	Yes	No	<b>Yes</b>	Yes
8	Performance	No	Yes	Yes	Yes	<b>Yes</b>	No
8	Scalability	No	No	Yes	Yes	<b>Yes</b>	Yes
7	Security and Dependability	No	Yes	No	No	<b>Yes</b>	No
7	Complexity of Development	No	Yes	Yes	Yes	<b>Yes</b>	Yes
6	Time-Consuming	Yes	Yes	No	Yes	No	No
<b>Total</b>	<b>Overall Score</b>	20	57	95	101	<b>105</b>	82

### 3.1.1 Selected Model: Agile Model

Based on the analysis in Table 3.1, the **Agile model** scored the highest (105), making it the best choice for this project. Agile's strengths in **flexibility, user feedback integration, scalability, and adaptability to new technologies** align perfectly with the needs of an AI-driven job recommendation system.

Agile's **iterative approach** allows continuous improvements and rapid adaptation to changing requirements, essential for integrating evolving machine learning models. It also enhances **collaboration among developers, data scientists, and UX designers**, ensuring a seamless user experience.

Moreover, Agile ensures **high performance and user-centric design**, addressing real-time job recommendations efficiently. With its structured yet adaptable methodology, Agile minimizes risks, accelerates deployment, and maintains high-quality standards, making it the ideal SDLC model for our project.

### 3.1.2 Application to the Project

In the context of the AI-driven job recommendation system, the Agile model allows for frequent updates to the job matching algorithms, based on real-time data and evolving user requirements. The iterative nature of Agile also ensures that the system can scale as more data is added, and new technologies, such as advanced machine learning models, are incorporated. Moreover, Agile's adaptability makes it the perfect fit for handling the complexity of integrating Natural Language Processing (NLP) models and user feedback, ensuring the system is constantly optimized for accuracy and relevance.

# Chapter 4

## Discussion

The *AI-Powered Next-Gen Job Recommendation System Using NLP and Machine Learning* enhances job matching through advanced NLP techniques and machine learning algorithms. Key functionalities include user registration, profile management, CV parsing, job search, real-time job recommendations, and candidate suggestions for employers. The system also integrates external job boards, real-time notifications, and an admin dashboard to ensure seamless management. Given the dynamic nature of AI-based applications, selecting a flexible and scalable SDLC model was crucial. A comparison matrix (Table 3.1) evaluated models based on flexibility, user feedback integration, scalability, adaptability to new technologies, and security. Agile scored the highest (105), slightly outperforming Spiral (101). Agile was chosen over Spiral due to its iterative development, rapid user feedback, lower development costs, scalability, and faster deployment. Unlike Spiral, which requires extensive documentation and risk analysis, Agile enables continuous updates and improvements essential for AI-driven systems. Its adaptability ensures efficient integration of evolving NLP models and machine learning techniques, making it the most suitable SDLC model for this project. Furthermore, Agile's incremental approach allows for early detection of potential issues, reducing risks and improving system stability. The continuous delivery mechanism ensures faster feature rollouts, keeping the system up-to-date with industry trends. This makes Agile the ideal choice for a job recommendation system that must evolve to meet user needs and market demands efficiently.

# References

- [1] GeeksforGeeks. Software development life cycle (sdlc), 2023. Accessed: 2024-10-17.
- [2] IEEE Computer Society. Software requirements specifications, 2023. Accessed: 2024-10-05.
- [3] Javatpoint. Agile model, 2023. Accessed: 2024-10-17.

[1] [2] [3]