



Prototype 2 – Software Development & Demonstration Report

Program: Computer Programming

Course: Community Sponsored Project

Project Title: AI-Powered Job Description Optimization & Smart Builder

Client: Guhuza – AI-driven staffing platform (www.guhuza.com)

Submission Date: Week 7 – October 2025

1. Objective

The objective of Prototype 2 is to transform the proposal from Prototype 1 into a **fully functional and demonstrable software system**.

This phase evaluates the team's ability to:

- Develop the complete working application,
- Integrate front-end, back-end, database, and AI components,
- Test and debug the system for performance and reliability,
- Deploy and demonstrate the software to stakeholders effectively.

2. Project Overview

The **AI-Powered Job Description Optimization & Smart Builder** helps employers create **high-quality, inclusive, and structured job descriptions** using artificial intelligence.

Core Workflows

1. Optimize Existing Job Description

- Employers upload an existing JD (DOCX, PDF, or TXT).
- The AI analyzes and scores it, providing categorized suggestions: *Critical*, *Recommended*, and *Nice-to-Have*.
- Employers can apply suggestions, skip them, and preview the improved JD.

2. Smart Builder (AI Generation)

- Employers without a JD can input basic details (title, location, employment type).
- The AI automatically generates a professional, inclusive, and ready-to-post job description.

Key Benefits

- Saves time and improves JD quality and structure.
 - Reduces bias through AI-based language analysis.
 - Enhances candidate matching accuracy for recruitment.
 - Provides hands-on learning experience in AI and full-stack development.
-

3. System Development Summary

Frontend Development

- Framework: **Next.js (React + TypeScript)**
- Styling: **Tailwind CSS**
- Designed responsive and accessible pages for JD Upload, Smart Builder, Suggestions, and JD Preview.

- Focused on usability and WCAG 2.1 accessibility compliance.

Backend Development

- Environment: **Node.js + Express.js**
- Handles data flow between frontend, database, and AI API.
- Implements routes for uploading, parsing, and analyzing JDs, and saving results.

Database

- Managed using **MySQL** and **Prisma ORM**.
- Stores job descriptions, analysis results, and timestamps for tracking history.

AI Integration

- Connected to **Guhuza AI Service API** (<https://api-ghz-demo-v2.azurewebsites.net/api/v2/ai/chat>).
- Powered by GPT-based models to analyze and optimize job descriptions.
- Uses prompt engineering and token optimization for efficient performance.

Deployment

- **Frontend:** Deployed on **Vercel**.
- **Backend:** Deployed on **Guhuza server** and tested locally for performance.
- Used environment variables for secure API key management.

Setup Instructions:

1. Backend:

```
cd server  
npm install  
npm start
```

2. Frontend:

```
cd client  
npm install  
npm run dev
```

Code Repository:

The complete source code, commit history, and documentation for this project are available at: <https://github.com/manishdonna/prototype2>

Please refer to the README.md in the repository for detailed setup, features, and known issues.

4. Testing & Debugging

Testing was conducted by **Manish Thapa Magar** and **Dhayanand Sankar Mani** to validate all core functionalities.

Test Case	Description	Expected Result	Actual Result	Status
Upload valid JD	AI returns score + suggestions	Works as expected	Works	✓
Upload empty JD	Displays error message	Works	Works	✓
Smart Builder	Generates complete JD	Works	Works	✓
API delay	Shows loading indicator	Works	Works	✓
Database save	Stores JD data successfully	Works	Works	✓
Export JD	Downloads JD correctly	Works	Works	✓

Bug Fixes

- Fixed CORS and data parsing issues.
- Optimized backend for faster AI response time.
- Implemented error validation for invalid or empty uploads.

5. Testing Evidence & Screenshots

Below are screenshots demonstrating successful testing and validation of system functionality.

Figure 1: JD Upload and AI Analysis Result

The screenshot shows the Guhuza Job Description Optimizer interface. At the top, there is a navigation bar with the logo 'Guhuza Job Description Optimizer', a '+ New Job' button, a 'My Jobs' button, a user profile icon, the email 'magarmanish798@gmail.com', and a 'Logout' button. A green notification box on the right side states 'File "Forensic-Accountant-Job-Description-Template.docx" analyzed successfully!' with a checkmark icon.

The main area is divided into two sections: 'Manual Entry' and 'Analysis Results'.

Manual Entry: This section contains a text input field with tabs for 'Paste Text' and 'Upload File'. Below the input field is a dashed box with a cloud icon and the text 'Click to upload or drag and drop'. It also specifies file types: 'TXT, PDF, or DOCX (max 5MB)'.

Analysis Results: This section displays the 'Overall Quality Score' as '72' in large orange digits, with a small orange flame icon and the word 'Good' below it. The 'Category Breakdown' table shows five categories with their scores:

Category	Score
Job Title	8
Role Summary	6
Reporting Structure	7
Responsibilities	8
Qualifications	7

Below the table, each category has a corresponding numerical value: Job Title (5), Role Summary (5), Reporting Structure (7), Responsibilities (8), and Qualifications (9).

Manual Entry

Upload or paste an existing job description for AI analysis

[Paste Text](#) [Upload File](#)

Job Description Text

Looking for employee

21 characters

[Analyze Job Description](#)

Analysis Results

25

Overall Quality Score

 Needs Improvement

Figure 2: Smart Builder – AI-Generated Job Description

Smart Builder

Fill in the details below and let AI generate a professional job description

Job Title *

cook

Location

Toronto

Work Condition

On-site

Employment Type

Full-time

Key Responsibilities (Optional)

Strong,Cheerful

Minimum Requirements (Optional)

2 years in hospitality

Preferred Skills (Optional)

no special skills required

Additional Notes (Optional)

Any other details you'd like to include...

Generate Job Description

Analysis Results

85

Overall Quality Score

🌟 Excellent!

Category Breakdown



ⓘ Matchability Insights

Title Clarity:	good	Skills Coverage:	good
Location Specificity:	good	Seniority Level:	clear

AI Suggestions

🟡 Recommended (1)

✓ Accept All

✗ Skip All

Add more details about the kitchen environment

💡 This helps candidates understand the work environment better.

general

Figure 3: Database Record Verification

The screenshot shows the 'My Job Postings' section of the Guhuza platform. At the top, there are summary statistics: Total Jobs (1), Active (1), Drafts (0), and Avg Score (85). Below these are search and filter options, including a search bar and buttons for 'All', 'Active', 'Drafts', and view modes. A single job listing is displayed for a 'cook' position in Toronto, On-site, Full-time, generated on Oct 14, 2025, with an AI Generated score of 85.

Figure 4: Error Handling for Empty File Upload

The screenshot shows the 'Manual Entry' page for the Smart Builder. A red error message box at the top right states 'File appears to be empty or unreadable'. Below it, there's a file upload area with a placeholder text 'Click to upload or drag and drop' and a note about supported file types (TXT, PDF, or DOCX, max 5MB). The footer contains links for 'About Guhuza', 'Quick Links' (Main Website, Social Media), and 'Features' (AI Job Description Analysis, Smart Builder, Quality Scoring, Real-time Suggestions).

Figure 5: Error Handling for empty field in Smart Builder

Smart Builder

Fill in the details below and let AI generate a professional job description

Job Title *

e.g., Senior Software Engineer

Location

e.g., New York, NY or Remote

! Please fill out this field.

Select work condition

Employment Type

Select employment type

Key Responsibilities (Optional)

Describe the main duties and responsibilities...

Minimum Requirements (Optional)

Required skills, education, experience...

Preferred Skills (Optional)

Nice-to-have skills and qualifications...

Additional Notes (Optional)

Any other details you'd like to include...

Generate Job Description

6. Technologies Used

Category	Tools / Technologies
Frontend	React (Next.js), TypeScript, Tailwind CSS, Shadcn/UI
Backend	Node.js, Express.js
Database	MySQL + Prisma ORM
AI Integration	Guhuza AI API (GPT-based)

Category	Tools / Technologies
Hosting	Vercel (Frontend), Guhuza Server (Backend)
Version Control	GitHub
Testing Tools	Postman, Browser Console, Manual QA

7. Team Members & Roles

Name	Role	Responsibilities
Manish Thapa Magar	Project Leader & Backend Developer	Project coordination, Node.js development, API integration, testing
Sudip Bhattacharai	Lead Frontend Developer	React/Next.js development, UI implementation, AI integration testing
Biplove Nepali	UI/UX Designer	User interface design, wireframing, user experience planning
Niroj Shrestha	Database Administrator	MySQL database design, schema management, data modeling
Ahil Krishna Prasad Sindhu	AI Specialist	Prompt engineering, GPT integration research, scoring logic
Pradip Adhikari	System Architect	System architecture design, documentation, API planning
Dhayanand Sankar Mani	QA & Deployment Engineer	Testing strategies, quality assurance, deployment management

8. Challenges & Solutions

Challenge	Description	Solution Implemented
API Integration	Authentication and async call management	Used environment variables and async/await handling

Challenge	Description	Solution Implemented
Response Time	AI response delay	Added loading indicators, reduced token size
File Parsing	Issues with large JDs	Implemented text normalization before AI submission
Deployment Setup	Environment configuration	Corrected .env setup and build variables
Team Coordination	Managing parallel development	Weekly meetings and GitHub version control

Stakeholder Feedback Implementation:

- Based on feedback from Prototype 1, we improved the Smart Builder interface for better readability.
- AI response time was optimized by reducing token size and implementing async handling.
- Suggestions for clearer error messages and guidance during JD uploads were incorporated.

9. Presentation Overview

The Prototype 2 presentation will include:

- 1. Introduction & Objectives**
- 2. Problem Statement and Solution Overview**
- 3. System Architecture and Data Flow**
- 4. Live Demonstration: JD Upload → AI Analysis → Smart Builder → Export**
- 5. Challenges and Solutions**
- 6. Feedback Integration & Team Roles**

10. Conclusion

The **AI-Powered Job Description Optimization & Smart Builder** successfully implements all major functionalities outlined in Prototype 1.

It demonstrates the ability to design, build, and integrate an AI-driven application with database connectivity and real-time feedback.

This prototype highlights strong teamwork, technical proficiency, and practical application of artificial intelligence in recruitment systems.

11. References

1. **Guhuza Project Brief (2025).** *AI-Powered Job Description Optimization & Smart Builder.*
2. **Guhuza AI Service Documentation (2025).** Internal project file: *api_documentation.docx*.
3. **OpenAI API Documentation (2025).** Retrieved from <https://platform.openai.com/docs>
4. **Claude AI by Anthropic (2025).** *Claude 3 Model Overview & API Documentation.* Retrieved from <https://www.anthropic.com/clause>
5. **React Documentation (2025).** *React Developer Guide.* <https://react.dev/learn>
6. **Next.js Framework (2025).** *Next.js Documentation.* <https://nextjs.org/docs>
7. **Tailwind CSS (2024).** *Utility-First CSS Framework Documentation.* <https://tailwindcss.com/docs>
8. **Prisma ORM (2025).** *Database Integration for Node.js & TypeScript.* <https://www.prisma.io/docs>
9. **Node.js Documentation (2025).** *Node.js API Reference.* <https://nodejs.org/en/docs>
10. **Postman Learning Center (2025).** *API Testing Best Practices.* <https://learning.postman.com>
11. **WCAG 2.1 Accessibility Guidelines (W3C, 2018).** <https://www.w3.org/WAI/standards-guidelines/wcag/>
12. **Canva Design School (2024).** *Design Principles for Effective UI/UX.*

13. ChatGPT (2025). AI Prompt Engineering & Integration Examples for Developers.

End of Report
