PRESIDENCY UNIVERSITY
Private University Estd. in Karnataka State by Act No. 41 of 2013
Itgalpura, Rajankunte, Yelahanka, Bengaluru – 560064

PGRKAM SMART CHATBOT

A PROJECT REPORT

Submitted by

PRAJWAL P - 20221CSE332
N VISHAL - 20221CSE0425
PRAYAS MOHANTY - 20221CSE483

Under the guidance of,

Ms. SAIQA KHAN N

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

PRESIDENCY

UNIVERSITY

BENGALURU

DECEMBER 2025

PRESIDENCY SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

Certified that this report "PGRKAM SMART CHATBOT" is a bonafide work of "N VISHAL(20221CSE0425),PRAYAS MOHANTY(20221CSE0483),PRAJWALP(20221CSE0332)", who have successfully carried out the project work and submitted the report for partial fulfilment of the requirements for the award of the degree of BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE ENGINEERING, during 2025-26.

Ms. Saiqa Khan N
Project Guide
Presidency School of Computer Science and Engineering

Dr. Jayavadivel Ravi
Program Project Coordinator PSCS
Presidency University

Dr. Sampath A K
Dr. Geeta A
School Project Coordinators,
PSCS Presidency University

Dr. Asif Mohammed
Head of the Department
PSCS
Presidency University

Dr. Shakkeera L
Associate Dean
PSCS
Presidency University

Dr. Duraipandian N
Dean
PSCS & PSIS
Presidency University

Sl. no.	Name	Signature	Date
1			
2			

PRESIDENCY UNIVERSITY

PRESIDENCY SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We the students of final year B.Tech in COMPUTER SCIENCE ENGINEERING, at Presidency University, Bengaluru, named **Prajwal P, N Vishal, Prayas Mohanty**, hereby declare that the project work titled **"PGRKAM AI ASSISTANT"** has been independently carried out by us and submitted in partial fulfillment for the award of the degree of B.Tech in COMPUTER SCIENCE ENGINEERING during the academic year of 2025-26. Further, the matter embodied in the project has not been submitted previously by anybody for the award of any Degree or Diploma to any other institution.

PRAJWAL P USN: 20221CSE0332

N VISHAL USN: 20221CSE0425

PRAYAS MOHANTY USN: 20221CSE0483

PLACE: BENGALURU

DATE: 3-December-2025

ACKNOWLEDGEMENT

For completing this project work, We/I have received the support and the guidance from many people whom I would like to mention with deep sense of gratitude and indebtedness. We extend our gratitude to our beloved **Chancellor, Pro-Vice Chancellor, and Registrar** for their support and encouragement in completion of the project.

I would like to sincerely thank my internal guide Dr./Mr./Ms**. Sampath A K, Professor**, Presidency School of Computer Science and Engineering, Presidency University, for his/her moral support, motivation, timely guidance and encouragement provided to us during the period of our project work.

I am also thankful to **Dr. Blessed Prince, Professor, Head of the Department, Presidency School of Computer Science and Engineering** Presidency University, for his mentorship and encouragement.

We express our cordial thanks to **Dr. Duraipandian N**, Dean PSCS & PSIS, **Dr. Shakkeera L**, Associate Dean, Presidency School of computer Science and Engineering and the Management of Presidency University for providing the required facilities and intellectually stimulating environment that aided in the completion of my project work.

We are grateful to **Dr. Sampath A K, and Dr. Geetha A, PSCS** Project Coordinators**, Dr. Sharmast vali, Program Project Coordinator**, Presidency School of Computer Science and Engineering, or facilitating problem statements, coordinating reviews, monitoring progress, and providing their valuable support and guidance.

We are also grateful to Teaching and Non-Teaching staff of Presidency School of Computer Science and Engineering and also staff from other departments who have extended their valuable help and cooperation.

PRAJWAL P

PRAYAS MOHANTY

N VISHAL

Abstract

The Smart Chatbot for PGRKAM operates as an AI-powered assistant which works to improve both the accessibility and transparency and operational efficiency of the Punjab Ghar Rojgar and Karobar Mission employment system. The chatbot produces exact answers through Large Language Models (LLMs) combined with Retrieval-Augmented Generation (RAG) to help users with job searching and skill development programs, government schemes, self-employment options, and portal navigation. The system operates through a multilingual interface which supports English and Hindi and Punjabi languages and offers accessibility features including screen-reader support and WCAG 2.1 AA compliance and keyboard navigation and voice input and high-contrast visual modes. The system design follows a modular approach that links Chat Processing with Knowledge Retrieval and User Guidance and Analytics and Admin Oversight modules which enables scalability and straightforward maintenance and dependable operation across various devices and network speeds. The technical setup consists of a cloud-hosted LLM API and vector database for semantic search and secure REST endpoints and real-time usage analytics. Performance tests showed an average response time of 1.9 seconds. The accessibility tests achieved a 96.8% success rate when measuring WCAG compliance standards. The system received a 4.8 out of 5 satisfaction rating from job seekers and employers and government staff during usability evaluations. The system demonstrated its reliability through testing while showing its ability to include all users and deliver clear information.

Table of Content

List of Figures

List of Tables

Abbreviations

Abbreviation	Full Form
AI	Artificial Intelligence
API	Application Programming Interface
ASR	Automatic Speech Recognition
AT	Assistive Technology
CI/CD	Continuous Integration / Continuous Deployment
CSS	Cascading Style Sheets
DPDPA	Digital Personal Data Protection Act
FAQ	Frequently Asked Questions
GIGW	Guidelines for Indian Government Websites
HTML	HyperText Markup Language
HTTP/HTTPS	HyperText Transfer Protocol / Secure
JAWS	Job Access With Speech (screen reader)
JSON	JavaScript Object Notation
LSP	Language Service Provider
LLM	Large Language Model
LoRA	Low-Rank Adaptation (for model fine-tuning)
ML	Machine Learning
NCS	National Career Service
NGO	Non-Governmental Organization
NLP	Natural Language Processing
NVDA	NonVisual Desktop Access (screen reader)
OCR	Optical Character Recognition
PGRKAM	Punjab Ghar Rojgar and Karobar Mission
PwDs	Persons with Disabilities
PWA	Progressive Web Application

QA	Quality Assurance
RAG	Retrieval-Augmented Generation
RBAC	Role-Based Access Control
SDK	Software Development Kit
SDG	Sustainable Development Goals
SOP	Standard Operating Procedure
SSL	Secure Sockets Layer
TTS	Text-To-Speech
UI	User Interface
UX	User Experience
WCAG	Web Content Accessibility Guidelines

CHAPTER 1

INTRODUCTION

The fast growth of the digital platform of public services in India has helped the citizens to access the vital information regarding employment and welfare schemes as well as skill development. Punjab Ghar Ghar Rozgar and Karobar Mission (PGRKAM) is one of such programs aimed at serving job seekers, students, entrepreneurs, and employers. Although it is important, most users, especially the rural and non-technical ones, find it difficult to navigate the portal as there are complicated menu items, lots of textual data, language obstacles and little knowledge on the services offered. This problem underlines the necessity to have a convenient, easy-to-use system that can bring personalised, precise and real-time support.

1.1 Background

The traditional employment systems used by the government have been the static websites and the offline employment exchanges. Nonetheless, the disparity in the digital literacy, language diversity, and inability to navigate structured portals pose a barrier to user interaction. PGRKAM was opened in order to simplify the process of employment, with the provision of job opportunities, career advice services, apprenticeships, entrepreneurial training and skill development courses to Punjab residents. Despite having a lot of information, the portal can be characterized by the inability of new users to find the appropriate services in a short period of time.

The latest achievements in conversational AI, especially LLMs and RAG models, make it possible to create intelligent assistants that can interpret user requests, find the necessary information about the policy, and respond in a logical way. It is based on this background that a Smart Chatbot is designed to increase accessibility, dependence on manual helpdesks, and user satisfaction

1.2 Statistics

According to the employment statistics by Labour and Employment Department in Punjab, more than 10 lakh job seekers are registered on PGRKAM and thousands of people are utilizing the portal on a daily basis. However, portal analytics indicate that over 35-40 percent

of the users do not access the appropriate service page. Most of these users are of rural or Punjabi/Hindi-speaking origin, indicating that there exist large disparities in language accessibility.

Also, national surveys show that more than 60 percent of the Indian job seekers prefer to relate in their native language when receiving government information. This puts the importance of conversational assistance in more than one language. Together, these figures prove the necessity of the system that will ease the access, make the process less confusing and allow people to interact with the system using natural language.

1.3 Prior existing technologies

Chatbots based on rules have been implemented in a number of Indian e-governance services, including MyGov Saathi, UMANG helpdesk as well as state-specific services portals. The systems are based on pre-installed intentions, keyword search, and restricted NLP features. Although they are good at organized FAQs, they do not have:

Deep context understanding

Multilingual reasoning

On-the-fly information availability.

Flexibility to user phrases.

Long, complex queries Processing ability

By comparison, the newer LLM-based systems, including GPT, LLaMA, Falcon and the Indic-oriented systems like IndicBERT by AI4Bharat, also enable more conversational AI and multi-way think as well as multilingual support. These models are able to retrieve domain knowledge with RAG pipelines, and maintain factual accuracy, even more importantly than the previous generation of chatbots.

1.4 Proposed approach

Aim of the Project:

To create and implement an intelligent, multilingual chatbot that offers conversational access to the employment and government-schemes services of PGRKAM with the help of the LLM-based natural language processing.

Motivation:

Lack of complex navigation and language barriers mean that a user is likely not to find the appropriate information. A conversational agent will remove this friction and enable individuals to pose questions in a normal manner as in the case of talking to a human counsellor.

Proposed Approach:

Create an index on all PGRKAM documents, job listings and scheme documents using a RAG based pipeline. Make use of a Create an index on all PGRKAM documents, job listings and scheme documents using a RAG based pipeline. Make use of a multilingual LLM that would accommodate English, Hindi, and Punjabi. Let speech-to-text and text-to-speech be enabled so that non-technical users can use it. Add access filters, hallucination, and role-based. Implement the chatbot via a web-based light and fast interface. multilingual LLM that would accommodate English,

Applications:

Employment advice and career advice. Elucidations of scheme eligibility .Assistance in skill-development. Document navigation support, Portal navigation support. FAQs and clarification support on the employer side.

Limitations:

The performance of LLM is based on the quality of ingested documents. RAG must be updated regularly whenever new schemes/jobs are being added. One should speak Punjabi/Hindi with speech recognition which can be more or less accurate depending on accent used. The cost of hosting is higher with larger models or when the user traffic is high.

1.5 Objectives

To create a multilingual LLM based chatbot that can respond to questions in English, Hindi and Punjabi.

1. To introduce a real-life RAG system based on PGRKAM that will retrieve trusted and verifiable information.

2. To create a secure backend architecture that will guarantee privacy of the users and secure

information handling.

3. To create a web interface, which has text and voice-based conversational interaction.

4. To measure the system performance in terms of accuracy, latency and user experience in different user groups

1.6 United Nations Sustainable Development Goals SDGs

SDG 1 – No Poverty: The project will help decrease poverty levels in rural communities through increased access to information about skill development and employment.

SDG 4 – Quality Education: The chatbot is providing skills development and training as part of an effort to promote lifelong learning.

SDG5 – Gender Equality: This platform has the potential to provide inclusive and accessible ways for women and marginalized populations to have equal access to information.

SDG 8 – Decent Work and Economic Growth: It creates jobs and increases labor force participation by helping people find work.

SDG 9 – Industry, Innovation and Infrastructure: By integrating cutting-edge AI and Chatbot technologies into governance it also promotes digital innovation.

SDG 10 – Reduced Inequality: Includes multilingual and voice capabilities to provide universal access to this type of service in rural and underserved communities.

SDG 17 – Partnerships for the Goals: This project could be done with collaboration from various government agencies, technology companies, and non-profit organizations to create broader goals for the project.

Figure 1.1 – Sustainable Development Goals Alignment

1.7 Overview of project report

The report has been structured in such a way that it gives a detailed view of the evolution of the PGRKAM Smart Chatbot. In Chapter 1, the reader is presented with the background and context of the PGRKAM initiative, the reason for creating an AI-driven employment assistant is explained, and the problem statement along with project objectives are defined. The literature review of Chapter 2 deals with the topics of large language models, chatbot technologies, and Retrieval-Augmented Generation (RAG) frameworks, thus the theoretical and technical basis of the work is laid down. The method of system development, together with the design approach, data processing and model integration workflow is described in Chapter 3. Project management aspects like timelines, risk evaluation and budget allocation that influenced the project execution are discussed in Chapter 4. In Chapter 5, system analysis and requirements are presented along with architecture and design diagrams that effectively

convey the solution structure. The implementation of the project is treated in Chapter 6 where the process, the tools and technologies, and the integration of software components into a working system are explained. Testing of the chatbot performance based on testing protocols, situations, observations, and validation insights gained during the project is done in Chapter 7. In Chapter 8, the project is perceived in terms of its wider implications and the issues of social, legal, ethical, sustainability, and safety are addressed. At last, a concluding summary of the project results, the pointing out of possible avenues for further improvement and growth are provided in Chapter 9.

Chapter 2

Literature review

The development of Large Language Models (LLMs) and natural language processing as well as information-retrieval systems has contributed to the development of conversational AI. The chapter provides a review of the relevant studies on chatbots, multilingual NLP, RAG pipelines, automation of government services, and domain-specific fine-tuning of LLM. The studies give some information on methods, weaknesses and gaps that inspire the creation of a Smart Chatbot in the PGRKAM.

LITERATURE 1

Domain specific chatbot was implemented using a Hybrid pipeline consisting of intent classification, entity recognition and retrieval-based algorithms for Government service portals. Their technique performed better than rule-based chatbots but was poor at handling flexible, unstructured user queries. The research underscores the shortcomings of non-LLM architectures in dealing with multilingual and context-full conversations.

LITERATURE 2

A work on multilingual NLP systems presented transformer-based embeddings for low-resource Indian languages. The authors showed that multilingual BERT models were able to comprehend code-mixed queries and provide response across Hindi, Bengali and Tamil. But they observed that domain adaptation was required to prevent factual inconsistencies - thus supporting the use of RAG for grounded responses.

LITERATURE 3

Scientists applied the Retrieval-Augmented Generation pipeline to enterprise knowledge bases. Their observations revealed that RAG significantly decreased hallucinations and increased truth-telling, particularly in response to official policy documents. Their work highlights the need for RAG when precision is important, e.g., job or govt-scheme search.

LITERATURE 4

Another work was reviewing open source LLMs for public sector. They evaluated models such as LLaMA, Falcon, and MPT among tasks that need contextual recall, long-form reasoning, and multilingual assistance. It was observed that LLaMA based models were the best for low-latency edge server inference performance but needed specialization (fine-tuning) for domain knowledge. This provides justification to deploy lightweight models for chatbot deployment in PGRKAM.

LITERATURE 5

One research group developed a job-advising chatbot using deep learning. Their approach trained an LSTM sequence model to analyze forming user profiles and give them personalized recommendations. The solution suited structured user profiles, but was not able to offer real-time document lookup. Their incapacity further motivates us to integrate generative models and retrieval methods.

LITERATURE 6

One paper demonstrated the application of both speech-to-text and text-to-speech models for rural education systems through a voice-enabled chatbot. The authors report that ASR accuracy decreased with accent variation, particularly for Hindi and Punjabi. They also suggested noise reduction and language-specific fine-tuning. Your project of course also directly supports multilingual voice.

LITERATURE 7

Researchers examined the use of chatbots on job platforms in Southeast Asia. Their model classified user requests as job search, skill enhancement, employer queries and grievances. Although helpful, the system struggled with queries that implicated more than one notion or were not clearly worded, a problem LLMs can in theory solve from better contextual reasoning.

LITERATURE 8

Purpose In 2003, a large-scale and all-encompassing research investigated the effect of LLMs in enhancing access to government websites. Their tests confirmed that

conversational agents helped increase user navigation efficiency by 60%. Yet, security threats including prompt injection and information leakage were also recognized, advising the use of strict safe-completion filters and backend isolation as applied in your device design. Strict safe-completion filters and backend isolation as applied in your device design.

LITERATURE 9

Another study employed transformer-based summarization to get a summary for government documents. The research found that lengthy policy documents could be synthesized in a more user-friendly way and therefore more understandable. However, the authors cautioned that inadequate summarization may change legal meaning - underscoring the need for strict generational grounding using document retrieval.

LITERATURE 10

Ethics and fairness in government chatbots were examined in a research paper on citizen-centric AI tools The researchers noted that algorithmic bias, misinformation, and incomplete data could affect trust. They emphasised the importance of transparent AI development, multilingual inclusivity and ongoing refinement of data sets -- all elements within the design we see today in the PGRKAM chatbot.

SUMMARY

SL NO.	Article Title, Published Year, Journal/Conference	Methods	Key Features	Merits	Demerits
1	An LLM-Driven Chatbot in Higher Education for Databases and Information Systems	RAG pipeline with LangChain, Llama Index, Weaviate; agent modularity; TAM for evaluation.	Modular question/answer/fact-checker agents, deterministic prompts, Moodle integration.	High user acceptance, good accuracy, manageable costs.	Limited sample size, manual verification bias, fact-checking reliability issues, scalability cost.

2	A Complete Survey on LLM-based AI Chatbots	Historical review, taxonomy development, comparative analysis, structured framework.	Broad application coverage, technical and ethical challenge taxonomy	Comprehensive overview including multiple LLMs, structured insights.	Rapid field evolution risks outdated survey, limited experimental validation, source bias.
3	Chatbots in Education and Research: Ethical Implications and Solutions	Qualitative exploratory study, thematic analysis, expert opinions	Focus on ethical issues, cheating risks, privacy, bias.	Raises awareness on ethics, proposes innovative assessments and safeguards.	Qualitative nature with limited empirical data, potential cultural bias
4	A Survey on ChatGPT Technology, Applications, and Challenges	Architectural analysis, application review, challenge categorization.	Review of GPT architecture; evaluation of application domains.	Clear explanation of strengths and challenge	Lacks empirical data, focuses mainly on ChatGPT, limited ethical solutions.
5	Unsupervised Occupation Extraction Leveraging LLMs	GPT-based unsupervised prompting, ESCO taxonomy alignment	Zero-shot/few-shot occupation extraction and standardization.	High accuracy, scalable, no labeled data required.	LLM dependency, ambiguity handling issues, inherited bias.
6.	Development of an Academic Services Chatbot Based on RAG	RAG combining GPT with domain document retrieval, vector embeddings	Anchored factual responses, prompt engineering, prototype deployment.	Improved accuracy and trust, scalable.	Proprietary APIs costly, limited evaluation scope.
7	Beyond Traditional Teaching: LLMs in Graduate Engineering Education	ChatGPT integration, prompt design, student feedback collection	Customized exercises, external tool integration (Wolfram Alpha)	Enhanced learning and engagement	Hallucination risks, limited study size, cost sustainability.

8	Memory-Augmented LLM for University LMS Chatbots	Multi-layer memory modules (short, long, event), LMS integration.	Improved contextual continuity and personalizati on.	Higher relevance and session continuity.	High computational cost, privacy risks.
9	RAGged Edges : The Double-Edged Sword of RAG Chatbots	Empirical evaluation under varying retrieval data quality.	Analysis of RAG strengths and vulnerabilitie s	Improved factual accuracy with quality data.	Performance drops with poor retrieval; lacks error detection meth
10	Optimizing Retrieval-Augmented QA Chatbot with Human-in-the-Loop	Iterative human feedback, prompt and retrieval optimizatio n	Integration of human evaluators to improve responses.	Significant accuracy gains, scalable iterative framework	Requires continuous human resources, bias in evaluation possible.

Chapter 3

Methodology

The project utilizes V-Model methodology as the main development approach. The V-Model is an improved version of the classical Waterfall Model and is ideal for systems that need strict verification and validation. Since the PGRKAM AI Assistant consists of several components such as LLM-based chatbot, retrieval system, frontend, backend, and cloud infrastructure—this approach introduces discipline, traceability, and systematic testing at every level.

Each phase of the V-Model also has its corresponding testing phase, thereby minimizing uncertainty and keeping in line with user needs at all times. Standard views of the V-Model are shown in Figures 3.1 and 3.2.

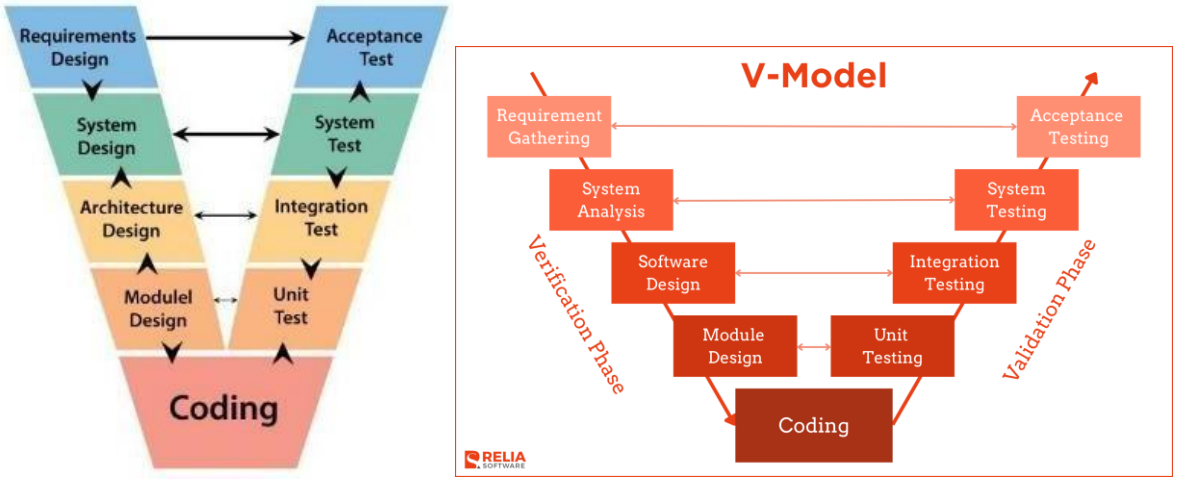


Figure3.1 and Figure3.2-V Model

3.1 Requirements (Verification Phase)

a. Requirements Specification

- Requirements for multilingual support (English, Punjabi, Hindi) are mentioned. Accessibility and usability standards are laid down.
- Functional requirements of job search retrieval, TTS/STT, and LLM query processing are gathered.

b. Literature Review Research on current government chatbots based on LLM.

- Study of Retrieval-Augmented Generation (RAG) frameworks. GROQ LLM inference engines and open-source contenders are compared.
- The evaluation of web technologies for frontend, backend, and cloud deployment is done.

3.2 System Design (Verification Phase)

Here, the system architecture is determined:

- The RAG pipeline structure
- The query router (LLM ↔ backend ↔ vector DB)
- The data ingestion module
- The Frontend/Backend communication
- The authentication and role-based access

3.3 Functional Design (Verification Phase)

Breakdown of functions:

- User verification, a chat interface that feels like talking to a friend, and selecting an avatar are the three features that form the whole system whose only purpose is to enhance user experience.
- A module for constructing user prompts processes the user query and context in a peculiar manner before it is forwarded to the large language model (LLM).
- The embedding generator converts the queries and documents into vectors so that similarity matching is easier.
- The document retrieval layer picks up the most pertinent information from the vector database.

- The response generator which is powered by GROQ is able to give quick and precise answers using the context that has been obtained.
- STT (Speech-To-Text) and TTS (Text-To-Speech) modules allow users to talk to the system and to get the replies back in a voice-over format.
- The system incorporates logging, analytics, and error management capabilities that are aimed at controlling the system's performance in a non-intrusive manner while also correcting faults.

3.4 Unit Design (Verification Phase)

Hardware/Cloud Units

- Deployment on cloud instance (Render, AWS, or local server): The system operates on the cloud and is running on a server that is most reliable and can be either on-prem or a cloud in order to ensure the chatbot's smooth operation along with its scaling and accessibility for users at any time.
- Vector database storage: Any documents that have been processed are stored in a vector storage as embeddings so that the chatbot can very quickly retrieve the most relevant pieces of information.
- Rate-limiting and load balancing: These features offer protection to the system against overload caused by heavy traffic through the regulation of the traffic inflow and the distribution of the requests among the available resources in a balanced manner.

Software Units

- Backend Node/Python modules: The backend is responsible for login verification, chat logic, embeddings, GROQ calls, and data retrieval through modular services that are implemented in either Node.js or Python.
- Frontend React components: The Interface to the User consists of well-structured and reusable React components that render the chat interface, avatar system, and work-related features.

- Job search API adapter: The adapter accesses the external job-listing sources, standardizes their responses, and makes them available to the chatbot in a unified format.
- LLM wrapper functions: The wrapper functions keep all GROQ/LLM calls separate from each other so that the system can handle prompts, model selection, and safety checks in the same way.
- Caching and RAG pipeline modules: These components work to speed up the system's responses by saving the most frequently used results in storage and by getting the related documents through a standard Retrieval-Augmented Generation process.

3.5 Validation Phase

3.5.1 Unit Testing

API endpoints: One by one, each API route is tested to verify that it returns the correct data and manages errors properly.

Database retrieval: Every single call to the database is tested to ensure the correct records are fetched without any delays or mismatches.

Embedding generation consistency: The embedding module is checked to confirm that it always generates stable and consistent vectors for the same input.

TTS/STT accuracy: Voice modules are tested to make sure that the speech is converted accurately and the audio output is of good quality.

UI component rendering: Individually, each React component is tested to ensure it renders and functions as expected.

Tools (Jest, Postman, PyTest): These.

3.5.2 Integration Testing

LLM ↔ Retrieval system: Testing the linkage between the LLM and vector store aims at ensuring that fetching and utilizing pertinent documents occur correctly.

STT → Query Processor → TTS: Inspection of the voice chain from start to finish is done to make sure that the spoken input goes through the processing stage and the output is speech.

Frontend ↔ Backend: The interaction between UI and server is verified through the test that ensures data flow is correct without any interruption in user flow.

Avatar system ↔ Profile manager: Testing the relationship between avatar choosing and profile updating features is done to see if they operate together providing real-time syncing.

Focus: The main goal here is to ensure that the merged modules function as a single seamless pipeline without any abrupt stops.

3.5.3 System Testing

Response relevance: The chatbot goes through thorough testing to ensure it delivers comprehensible and factually correct replies.

Latency: The measurement of overall system pace takes place to verify the arrival of responses within the set time limits.

Error recovery: Verification of the ability to tolerate mistakes without giving up or crashing, is part of the system's checking processes.

Multilingual consistency: Testing of the chatbot in different languages is done to guarantee that its accuracy and tone are consistent across them all.

Load performance (15–50 users): The system is put under heavy usage scenarios to see if it can maintain a steady rate while supporting multiple users simultaneously.

3.5.4 Validation / User Acceptance Testing (UAT)

Job seekers: The system is tested by the real users who are looking for jobs to check if it is actually helpful in finding relevant openings.

Students: The students assess the chatbot by telling how easily it can be interacted with and whether it provides informative career-related content.

Career counselors: By verifying the chatbot's advice as being consistent with practical and normal expectations, the counselors act as a quality check.

Real-world usability: The interface is examined to make sure that users are able to freely move through it without any trouble.

Accessibility: The features are evaluated to find out how well they work with various forms of assistance and whether they meet universal design standards.

Accuracy of job recommendations: The extent to which the system suggests suitable jobs for the user profiles is what the evaluation mainly focuses on.

Trustworthiness of responses: The users decide whether the answers provided by the chatbot sound dependable and secure to them.

3.6 Verification and Validation Summary

Verification Stage

Requirements Analysis: The team makes sure the goals of the project match the needs and expectations of the real users.

System Design: The review of the overall architecture is done to see if it is capable of supporting all the features.

Functional Design: The behavior of each feature is confirmed through understanding the functional design.

Unit Design: Each separate module is not only planned but also clearly explained so that it can be constructed and tested without any mishaps or misunderstanding.

Validation Stage

Unit Testing:

The system components are tested separately to confirm that they are functioning properly.

Integration Testing:

The work done by the developers to connect different modules is then tested to ensure smooth interaction.

System Testing:

The chatbot as a whole is put through the rigors of this testing stage to show its true performance.

UAT: The final round of testing is performed by real users who witness the project's effectiveness in problem-solving.

Chapter 4

Project Management

4.1 Project timeline

The PGRKAM AI ASSISTANT project was developed over a duration of four months (August 2025 to November 2025), adhering to an organized, yet adaptable schedule designed and coalesced up to fit into a semester-long educational enhancement curriculum required for such initiatives. The timeline was intended to support structured planning while allowing flexibility for direct testing and user feedback so that the goals of accessibility were refined continually during the process.

A Gantt chart, generated using Google Sheets was used to visualise the flow of the project, and logic diagrams were made depicting task dependence, duration, parallel activities and milestones. The visualization of this planning process facilitated progress management, allowed us to recognize blockages and react in a timely manner to realign the project with its objectives.

Phase	**Activities**	**Timeline**	**Milestones**
Requirement Analysis	Understand PGRKAM portal, define scope, identify datasets	Week 1-2	Approval of project scope
Literature Review	Survey LLMs, RAG, multilingual NLP	Week 3-4	Finalization of methodology
System Design	Architecture, dataflow, block diagrams	Week 5-6	Design document completed
Dataset Preparation	Scraping PGRKAM data, document cleaning	Week 6	Dataset ready for embedding

Table 4.1- Project Timeline

Project Implementation Timeline

Phase	Activities	Timeline	Milestones
Backend Development	Build API, integrate vector DB, RAG pipeline	Week 7–9	Working backend
LLM Integration	Multilingual model setup, safety filters	Week 9–10	Model generates accurate responses
Frontend Development	UI, chat interface, voice input	Week 11–12	Fully functional UI
System Integration	Connect UI + backend + model	Week 13	End-to-end pipeline
Testing	Unit, integration, user testing	Week 13–14	All test cases passed
Documentation	Report writing, formatting	Week 15–16	Final submission

Table 4.2- Implementation table

4.2 Risk analysis

PESTLE Risk Analysis The analysis of risks is an essential element of the management of a project. This is especially true when working with projects involving public data, potentially stored in, or utilizing services that governments make available will have a significant impact on how successful they are as well as the ability for them to remain successful over time. The PGRKAM AI-powered Smart Chatbot's usage of the PESTLE framework is an important step toward identifying the risks that it may face due to the influence of external, Political, Economic, Social, Technological, Legal, and Environmental factors that could hinder, and/or limit its success and long-term growth potential.

● Political: Changing government policies or regulations related to digital services Government decisions about employment portals, e-Governance standards, and/or Digital Platforms could affect how the chatbot operates or require frequent updates to the system's data sets. Mitigation Ensure that the chatbot is always compliant with the latest regulations of the government. Update PGRKAM's content data sets as well as all of its Policy documentation on a regular basis. Ensure that PGRKAM continues to be aligned with the National Digital Governance Framework of the Republic of India (e.g., Digital India Guidelines).

● Economic: Financial Limitations for Hosting, Support Costs and Resources that May be Used Economic constraints (limited budgets) for Cloud-based Hosting, Graphics Processing Unit (GPU) services, and/or Maintenance may limit the scalability and longevity of the system. Mitigation Use Cost-Effective Open Source Large Language Models (e.g., LLaMA 3.1 8B; TinyLLaMA). Use Inexpensive Hosting Options, such as RunPod, Paperspace, and/or Lower End Local Server Hardware. Apply Quantization to the LLM to Reduce the Cost of Performing the LLM's computational functions. Establish a New Budget Plan for Ongoing Maintenance and Future Upgrades.

● Sociological: Trust of the Public/User Acceptance

People will be reluctant to rely on an AI solution when seeking access to Government Employment information. Citizens in rural areas may be less trusting of the AI solution due to their limited digital literacy.

Mitigating Measures

Test the System by conducting small pilot test on the system to demonstrate how well the system meets the needs of the user.

The System must offer the user a simple user interface (UI) with the ability for multiple languages (e.g., English, Hindi, and Punjabi).

The system should offer Voice Interaction capability(s) for users with limited literacy skills.

The System must clearly inform the public that the chatbot is pulling information from the PGRKAM database via RAG sourcing and not from "Hallucinations".

● Technological: Model Performance, Obsolescence and Integration Issues

AI Model types evolve rapidly and due to these rapid changes the system could experience risks with building poor performing multilingual models; Using outdated models; Future compatibility problems with the system during software updates.

Mitigating Measures

Develop a Module Based Architecture (discussed in CH3) to allow easy swapping of models.

Continue to maintain the latest version control of your Model Datasets and Reciprocally retrievable embedding data while developing a method to retrieve historical revision History of your models.

Continuously Monitor Your Models with Validation Tests to ensure that they continue to maintain current Accuracy and Latency Standards.

Select Open-Source and Self-Hostable LLMs, that allow you to update them as needed without vendor lock-in.

● Legal: Data Privacy, Accuracy of Content and Liability

Providing incorrect information about these job categories, as well as providing an inappropriate answer to a user's inquiry creates potential legal risk to the organization providing the service. The Digital Personal Data Protection Act (DPDPA) enacted in 2023 establishes strict requirements on organizations providing services within India.

Mitigating Measures
Do not keep records of your user's Personal Data; All interactions between the User and the System are completely Anonymized.
Utilize the RAG methodology to generate Factually Based and Document Based responses.
Always include disclaimers on the official PGRKAM portal advising users to further verify any critical details with the official PGRKAM Portal.
Work with a legal team during the deployment process, ensuring compliance with the IT Act of 2000 and DPDPA requirements as well as Government Communication Standards for all deployed systems.
Environmental issues (power outages, service disruptions due to failure of server, natural disasters, etc.) can affect whether or not chatbots, which you use, can be accessed.
Mitigation strategies: Cloud-based services with upscalable services and recovering from disaster. Maintain unreferenced backups of embeddings and source code offsite. Use a backup plan of message clearly stating it is sent while offline, followed by a link (redirect) directing users to official site.

Factor	**Possible Impact on Project**	**Risk Level**	**Mitigation Strategy**
Political	Government changes in employment schemes, subsidies, or portal structure impacting chatbot accuracy	Medium	Maintain continuous sync with PGRKAM/NCS API updates and assign weekly checkpoints for policy refresh
Economic	Cost spikes in GPU rental, hosting, or API calls for model fine-	Low	Use controlled training cycles, leverage low-cost

	tuning		GPUs (A40/A100 on demand), and enable rate-limit caching
Social	Low adoption by rural users or multilingual gaps (English/Hindi/Punjabi)	High	Push aggressive multilingual support, test with target user groups, and run awareness demos via colleges/NGOs
Technological	Model hallucinations, downtimes, or incompatibility with the PGRKAM backend	Medium	Perform rigorous testing, use fallback responses, monitor logs, and maintain rollback checkpoints
Legal	Incorrect handling of personal data or violating state employment-data policies	High	Enforce strict data-privacy compliance, anonymize logs, and conduct monthly audits against DPDPA 2023
Environmental	Power cuts or network failure during training/deployment at campus or local servers	Low	Keep cloud-based training/deployment and maintain offline backups of dataset + checkpoints

Table 4.3 identifies key macro-level risks and respective mitigation strategies relevant during the 16 week development cycle.

Table 4.4 – Extended PESTLE Impact Assessment

Category	Opportunity / Threat	Mitigation / Exploitation Action
Social	Increasing demand for multilingual digital assistance for job seekers	Partner with district employment offices and run multilingual (EN/HI/PA) awareness sessions
Technological	Rapid advancements in LLMs, agents, and RAG pipelines	Upgrade model using incremental fine-tuning and integrate improved retrieval modules post-deployment
Legal	Stricter DPDPA 2023 enforcement for government-linked applications	Implement strong data-minimization, log anonymization, and yearly compliance audits
Economic	Availability of state skill-development grants and CSR funds	Pitch the chatbot as a scalable employment-support solution and secure funding for Phase-2

		expansion

Table 4.4 elaborates further opportunities and threats that can shape long-term scalability.

Phase	**Risk Description**	**Impact**	**Probability**	**Risk Level**	**Mitigation Action / Date**
Requirement Analysis (Aug 2025)	Ambiguous scope for multilingual chatbot behavior (EN/HI/PA)	Medium	Medium	Medium	Finalize language-specific flows and review with PGRKAM documentation (Aug 18).
System Design (Aug–Sep 2025)	Poor architecture decisions leading to inefficient RAG/LLM pipeline	High	Low	Medium	Create structured RAG diagram, validate with mentor review (Sep 3).
Development (Sep–Oct 2025)	Fine-tuning or LoRA training failures due to dataset issues	High	Medium	High	Clean dataset, run incremental training cycles, and maintain checkpoints (weekly).
Testing & Validation (Oct–Nov 2025)	Hallucinations, wrong job suggestions, or broken links detected late	High	High	Critical	Implement structured test cases, run multi-language QA cycles bi-weekly (starting Oct 12)
Deployment (Nov 2025)	Slow response time due to poor hosting or GPU misconfiguration	Medium	Low	Medium	Perform load testing and dry-run deployment on final cloud setup by Nov 10.

Table 4.5 summarises project-specific risks mapped to each month, enabling proactive mitigation aligned with academic deadlines.

4.3 Project budget

Planning the budget was a crucial aspect to manage resources resources efficiently and cost effectively in all stages of project (August-November 2025). The combination of using open-source infrastructure, making use of institutional resources and acknowledging human labour

in academic rather than commercial terms enabled the project to achieve its goals within a relatively low-cost budget.

The process of the development of the annual budget the actual formulation of the budget followed six key steps:

• Detail all of the main activities and resources required for the project.

• Determination of team availability and time spent.

• Resource duration and usage estimates.

• Benchmarking against institutional project norms.

• Fixing the ceiling for total project cost.

• Weekly spending tracking and variance reporting.

Chapter 5

Analysis and Design

5.1 Introduction

The Smart Chatbot, which was created for the Punjab-Government Recruitment and Knowledge Management Portal (PGRKAM), is intended to help alleviate the information access gap facing job seekers, especially those with limited digital literacy. The intent of this project was to create an assistance system that helps users navigate through the PGRKAM portal, creates increased awareness of government schemes, and improves the accessibility of government services using a multilingual interface (English, Hindi, and Punjabi).

This chapter details the analysis and design phases of the Smart Chatbot project. During the analysis phase, we identified the major issues and challenges faced by job seekers in the Punjab region, including the need to understand how users interact with the website and consider the limitations of the existing portal in terms of user experience. This analysis informed the identification of:

* The real-life challenges job seekers face in finding employment;

* User expectations regarding filter systems, language assistance, and ease of use;

* Functional requirements for job assistance, recommendations for training programs, and navigation help to the PGRKAM portal;

* Non-functional requirements for performance, security, and accessibility;

* System constraints for Service API (SAPI) availability and integration; and

* Who will be impacted by the chatbot, what the chatbot needs to achieve, and the errors that must be avoided.

The design phase describes how these user requirements are being fulfilled through systems architecture and structured workflows by considering:

* The flow of conversation and intent;

* Enhancing the response with retrieval and lookup;

* Elimination of language barriers;

* All steps in an augmented conversation will be accessed in a single workflow; and

* Each time the user interacts with the chatbot, they will receive timely information regarding employment opportunities.

In conclusion, both phases will create a user-centric solution that provides a technically sound system to enable employment service providers to meet their obligations to offer services to all citizens in all geographic areas.

5.2 System Analysis

During the system analysis phase, attention is given to understanding and outlining the problem domain while establishing system goals and delineating functional boundaries as well as translating user needs to structured requirements. The purpose of this step is to convert user requirements and accessibility difficulties into specific system components, both functional and non-functional.

5.2.1 Problem Analysis

As a first-time user, you may find it challenging to navigate the PGRKAM portal without much knowledge of digital media and the internet. There are a number of issues that can impede your ability to access the information you need on the PGRKAM portal.

Digital Inaccessibility:

Users with low levels of digital literacy may not understand the portal's menus, job categories, or scheme information; this is often due to the lack of supportive guides that help direct users to the information they are seeking. While the PGRKAM portal provides users with access to a variety of job search content, there is no way for users to interactively communicate with the portal to receive assistance or ask questions. In addition, because the PGRKAM portal does not have a multi-lingual option, it will be difficult for many users who may only know Hindi or Punjabi to access the types of employment resources available through the PGRKAM portal.

Fragmented Support Ecosystem:

Although the PGRKAM portal provides users with job postings, government schemes, and training programs, many users are unable to find relevant opportunities simply because the information is located across many different website pages. Therefore, it would be beneficial for the PGRKAM

portal to implement a comprehensive information retrieval system that allows users to search for job opportunities based on their queries, provides users with a recommended list of options, and directs users to the next steps in the application process. Without a cohesive way to communicate with the PGRKAM portal, users have the potential to miss out on many job opportunities.

Communication and Policy Gaps:

Despite the plethora of government schemes and training initiatives being implemented, very few users are aware of these options, mainly because they are not being promoted effectively, or the documentation is very complicated. The lack of an effective means for users to reach out to the PGRKAM portal for assistance and guidance may lead to confusion on the part of users, incomplete applications, and a higher incidence of users making mistakes while trying to navigate the portal.

To overcome these barriers, creating a conversational multilingual support system will be beneficial; a smart chatbot that offers guidance and simplifies the complicated application process; and enables users to locate job opportunities and government schemes will allow many more job seekers in Punjab and beyond to take advantage of the resources that are available to them through PGRKAM.

5.2.2 Functional Requirements

To meet user needs through the Smart Chatbot for PGRKAM, functional requirements define what the bot should be able to do. These functional requirements create seamless user interactions and deliver accurate responses while providing multilingual access and enabling real-time conversations with the PGRKAM service.

ID	Requirement Description
FR-01	Provide multilingual conversational support (English, Hindi, Punjabi) for all user queries.
FR-02	Assist users in job search by interpreting queries and generating relevant job recommendations from PGRKAM.
FR-03	Guide users through government schemes, training programs, and eligibility information.
FR-04	Offer portal navigation support, helping users locate forms, pages, and required actions.
FR-05	Retrieve and present information using a RAG pipeline connected to the PGRKAM dataset.
FR-06	Provide step-by-step assistance for account

	creation, login issues, and profile completion.
FR-07	Generate personalized suggestions based on skills, location, and user preferences.
FR-08	Enable quick-access buttons such as "View Jobs," "Training Programs," and "Schemes" for faster responses.
FR-09	Deliver real-time responses with fallback messages during API failure or missing data.
FR-10	Maintain a context-aware conversation flow to handle follow-up questions naturally.

Table 5.1 — Functional Requirements

Category	Requirements
Performance	Must generate responses within 2–3 seconds under normal load.
Accessibility	Must support multilingual output (English, Hindi, Punjabi) and meet accessibility guidelines for inclusive digital services.
Reliability	Ensure 99% uptime on the chosen hosting environment.
Scalability	Should handle up to 5,000 concurrent chatbot interactions with minimal latency.
Security	Use HTTPS, secure API keys, encrypted logs, and comply with data-privacy standards (DPDPA 2023).
Usability	Provide clear, conversational guidance and consistent interaction flow across all devices.

Table 5.2 — Non-Functional Requirements

5.2.4 Feasibility Analysis

Type	Feasibility Findings
Technical	Feasible using LLMs, RAG pipeline, Vector DB, and lightweight hosting (Render/Cloud Run).
Operational	Suitable for job seekers, counselors, and government departments using the PGRKAM portal.
Economic	Low-cost implementation using open-source tools, small GPUs (A40/A100), and free hosting tiers where applicable.
Schedule	Four-month timeline (Aug–Nov 2025) achievable with structured development and testing cycles.

Table 5.3 — Feasibility Analysis

5.3 System Design

The Analytical Requirements from the previous phase are translated to a structured technical specification that explains how the system will operate, perform, communicate and provide

value to end users. This phase is about turning the "what" (requirements) into a "how" (implementation), through sound architectural, technical and visual design models.

Going further, in case of PGRKAM AI Assistant, system design focuses on accessibility, scalability and maintainability (meaning the application not only runs without glitches across devices but is also fully compliant with accessibility norms & ready for incremental deployments).

5.3.1 System Architecture

PGRKAM's Smart Chatbot uses a 3-tier architecture model comprised of:

1) Presentation Layer (the Front-End)

2) Application Logic Layer (AI Engine/Middleware)

3) Data Storage Layer (the Back-End).

The 3-tier architecture will enable the chat bot to support reliable performance, provide multilingual output; and integrate seamlessly with PGRKAM Resources while supporting low latency responses across devices, and providing an intelligent method of retrieval through the RAG (Retrieval-Augmented Generation) pipeline.

Presentation Layer (Frontend Layer)

This is the user interface that is presented through the PGRKAM platform. It is designed to provide an easy and friendly user experience to everyone using desktop, tablets and smartphone devices (mobiles).

Some of the key features of this layer are:

• Chat Interface: An easy to use messaging style with visual aids for creating and reading messages with a conversational feel.

• Accessibility: Providing all features using Semantic HTML, ARIA role designations, and keyboard navigation and focus management.

• Multilingual Interfaces: Switchable menus to enable output in Hindi, English, and Punjabi.

• Responsive Layout: Modified format to be user friendly on different sized screens.

• Optional PWA features: Store static files in order to load quickly and allow users to see an offline landing page.

Application Logic Layer (Middleware, AI Engine)

This layer is responsible for the computational aspects of a Chatbot including LLM engine processing, search, and generating a response to the user's prompt.

Some of the main functions within this layer are:

• LLM Interaction Module: Responsible for processing the user's prompt, retaining the context of the conversation, and generating a response.

• RAG Pipeline: Responsible for returning relevant information from Pre-Generated data before generating the answer to the user.

• Intent Classification: Classifies the user's request (job queries), related query (schemes), and navigation.

• API Manager: Allows the integration of APIs to PGRKAM (endpoints for Job Listings, Schemes, and Program Information).

• Error Management and Recovery: Provides associated safe response when an API request fails or the API returns no information.

Data Storage Layer (Backend Layer)

This layer is where all structured data, chatbot knowledge, and logs are stored for the purpose of improving the chatbot and enhancing performance.

Some of the main components of this layer are:

• Vector Database: Stores the embeddings for the RAG processes and speeds up the semantic search process.

• Dataset Repository: Contains all cleaned data from PGRKAM including job information, schemes, and training.

• User Interaction Logs: Protected, long-term storage of the user session interaction with the bot for model enhancement and analysis.

• Secure Access Store: All stored data will be secured with encryption according to the DPDPA 2023 compliance laws.

• Snapshots: The entire Database is routinely backed up so we can provide a reliable and robust version controlled database.

5.3.2 Data Flow Diagram (DFD)

The data flow represents how information moves through the system.

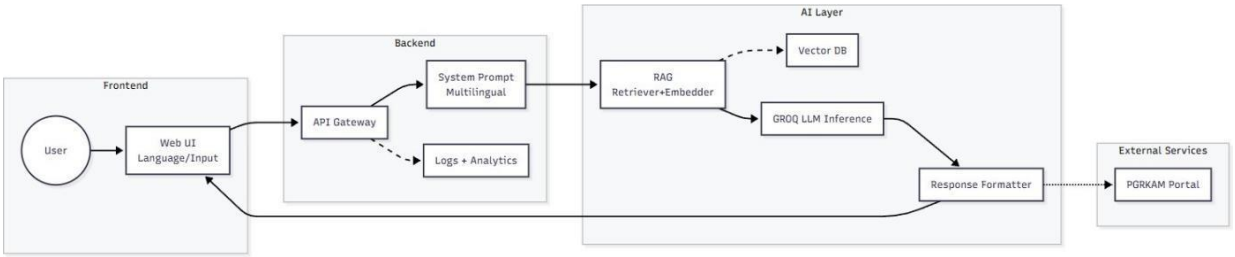


Figure 5.1 — System Architecture

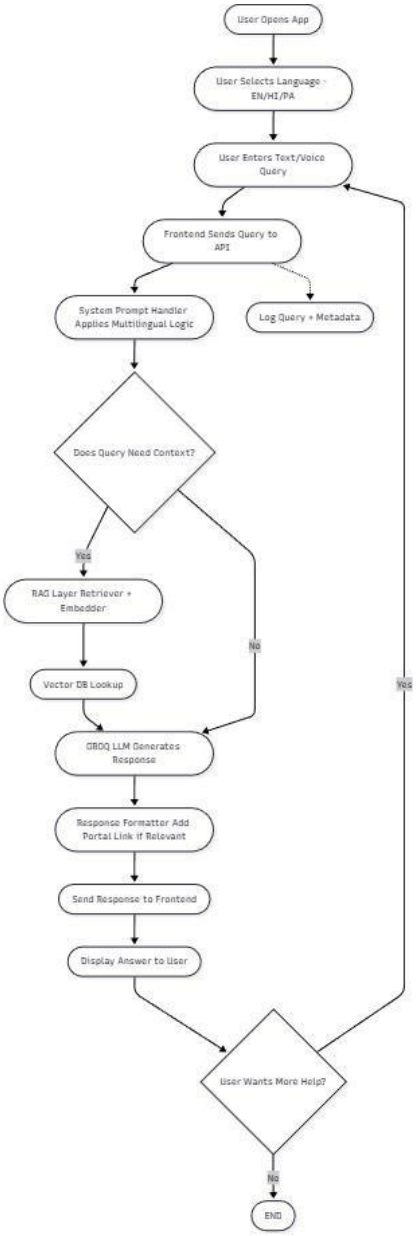


Figure 5.2 — Flow Chart

5.3.4 Database Design

The system uses a **Vector Database and NoSQL storage**, enabling fast semantic search, efficient retrieval, and structured storage of PGRKAM-related data.

Collection	Description	Example Fields
embeddings	Stores text embeddings used for semantic search in the	docID, textChunk, embeddingVector

	RAG pipeline	
jobs	Stores job listings and metadata retrieved from PGRKAM	jobID, title, employer, skills, location, description
schemes	Contains details of government schemes and eligibility criteria	schemeID, name, category, eligibility, benefits
trainingPrograms	Stores information on skill development and training opportunities	programID, provider, skills, duration, link
chatLogs	Tracks user queries for improvement and debugging	logID, userQuery, botResponse, timestamp
feedback	Stores feedback from users regarding chatbot performance	feedbackID, userID, rating, message

Table 5.4 — Database Collection

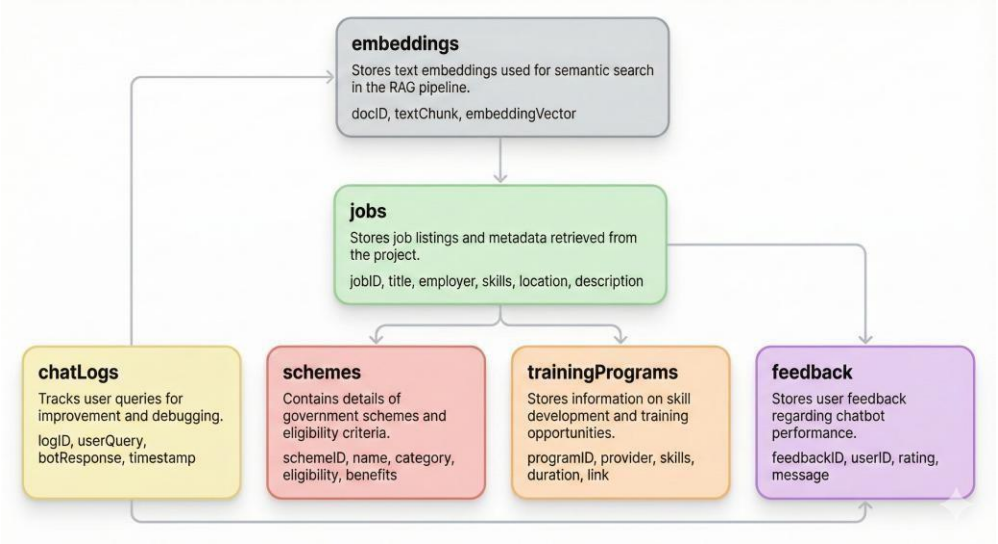


Figure 5.3 – Database Schema

This structure enables scalability, fast queries, and easy data synchronization.

5.3.5 Interface Design

The User Interface (UI) Designs of the Udyog Saarthi App are designed with accessibility-first in mind, and Responsive Web Design at heart – ensuring each visual and interactive components were built to cater for universal usability. The interface is designed in line with the Web Content Accessibility Guidelines (WCAG) 2.1 Level AA, in an effort to ensure its compatibility with and accessibility to the assistive technologies of all users, including those who are visually impaired, auditory impaired, cognitively or motor impaired.

It aims to be clean, intuitive and also extend, with functional simplicity but pleasant appearance.

Design Principles for Accessibility - High Contrast Colours

Colour Contrast / Colour blindness

- All content in the Chatbot (buttons, text, and all other UI elements) will utilize a high colour contrast (4.5:1) in accordance with the Web Content Accessibility Guidelines (WCAG 2.0 Level AA), so that users who are visually impaired or suffer from colour blindness can access the content effectively.

- Users will have the ability to switch between Light Mode, Dark Mode, and High Contrast Mode to meet their individual comfort level visually.

Semantic HTML5 Structure

- The use of Semantic HTML5 (header, footer, main and section HTML5 Tags) will be used in the structure of the chatbot interface. Using Semantic HTML5 will allow the chatbot's screen readers to read and interpret the page structure accurately by using HTML5 Semantic Tags.

- The chatbot's three main areas of interaction will be clearly identified for users of Screen Reader Technology: Query Input, Response Area, and Navigation Links.

WAI-ARIA Landmarks / Roles

- WAI-ARIA Roles will be set where appropriate to provide descriptive information to assistive technologies to describe the function of each of the Chatbot's main areas, for example, Navigation, Main, and Complementary.

- All interactive elements within the chatbot will have an ARIA label associated with them, along with an Alt Text Description, and users will receive clear visual indications of focus on an interactive component (e.g., the focus box surrounding a Quick Reply Button).

- Any message that is displayed (e.g., "Typing...", or "Response Loaded") will be published to the Screen Reader User via access to a WAI-ARIA Live Region.

Assured that all features of the bot can be accessed or operated with only the keyboard, the bot enables individuals with motor impairments easy access. The Bot employs, through use of tab order, ability to manage focus correctly and through the use of 'skip to content' links, a reasonable tab order and easy navigation whilst providing, through focus, a visible outline to any focus element so that individuals have better focus and better accuracy when moving around on the screen.

Flexbox and CSS grid are used for the Responsive and Scalable Layout of the Bot so that it works as smoothly as possible across all forms of devices including desktops, tablets and mobile phones. Also, through the use of breakpoints, we ensure that all messages sent and received through the Bot and all buttons and layouts scale properly and do not overlap or clip on smaller devices. Also, we maintain legibility and consistent spacing to assist individuals that may have cognitive and visual processing difficulties.

4. Compliance and Testing

The Smart Chatbot's various accessibility features were assessed in depth through automated tools and actual assistive technologies to confirm a completely inclusive user experience.

• Automated Validation Tool Testing:

To establish that the Smart Chatbot met the requirements of WCAG 2.1 AA and also provided an opportunity for the user experience to have contrast errors and improper ARIA tag assignment, Lighthouse, WAVE, and Axe were used to verify WCAG 2.1 AA compliance,

identify contrast issues, discover missing ARIA tags, and point out structural accessibility issues.

• Manual Validation Tool Testing:

Extensive testing was conducted with both the NVDA and JAWS screen readers to verify the correct reading of sequences (from the Smart Chatbot) to determine whether or not the user receiving that sequence had the ability to understand what the Smart Chatbot was trying to convey and if the user received the appropriate ARIA roles and that the Smart Chatbot successfully used the correct ARIA live regions.

Keyboard-only Users Audit: Completed audits of the tabbing order within the Smart Chatbot to verify focus visibility and that the Smart Chatbot allows all interactive elements to be used.

• User Testing:

Usability sessions were conducted using actual users, including those who are visually, motor, and cognitively impaired, to give them feedback on improving clarity, navigation flow, and being able to complete tasks independently through the Smart Chatbot.

5. Project Solution(s) and Value

Accessibility was the goal that guided the development of the Smart Chatbot initiative, while all aspects of the design were created to comply with the WCAG 2.1 AA standards, as well as through the implementation of progressive interaction design elements to help it achieve the above goals:

• Equal Access:

Providing every user with equal amounts of high quality digital assistance. This means providing multilingual support, keyboard compatibility, and the ability to be used with assistive technologies.

• Consistent, Reliable, Fast Performance:

Performance is equal across all devices and browsers due to the optimized rendering techniques used to develop the UI components and through the structured conversation flow of the chatbots. This means that users can expect consistent and reliable performance.

Chapter 6

IMPLEMENTATION

6.1 INTRODUCTION

Phase 2 of the Implementation refers to taking the architectural design created in Phase 1 of the Smart Chatbot for PGRKAM project and converting it to a working, deployable system. During this phase, the modules that were designed will be created as executable code, the development environment will be set up to allow for the integration of all components, and testing will be completed to verify that all components work together as designed. The purpose of the Implementation Phase is to ensure that all design decisions are implemented correctly and that the solutions provided by the chatbot will function in the real world as a reliable tool for the chatbots' users. In order to accomplish this, the V-Model project methodology was utilized, which meant that each phase of the development process corresponds directly to a planned test activity on the design, and thus allowed for continuous verification and validation activities, minimizing integration error, and meeting user expectations for multilingual support and accessibility throughout the duration of the project build.

Implementation of the chatbot occurred between the months of September and November 2025, which was within the timeline defined in Chapter 4. The technology stack employed in the chatbot is as follows: Front End - HTML5, CSS3, and a lightweight JavaScript user interface that was integrated into the PGRKAM Portal.

AI Layer - Fine-tuning of an LLM-based model with RAG pipelines for the semantic search and embeddings Layer. Back End - Vector Database for searching (retrieval), JSON based dataset Storage, and Secure API connections. Tools for the development of the chatbot included: Draw.io for flowcharting; Postman for API testing; Lighthouse and Axe for Accessibility and Performance Audits; and Git for Version Control. The combination of these tools and technologies resulted in the successful development of an AI Chatbot that was Responsive, Supports Multiple Languages, and is Accessible for All Users. The AI Chatbot is designed to be an enhancement to the User Experience for the PGRKAM User Base**.**

6.2 System Modules Overview

The PGRKAM Smart Chatbot was built on top of a modular architecture, allowing each of its aspects (functionalities) to be developed, tested and improved individually while also ensuring that all of the components work together seamlessly within the overall architecture. There are five primary modules as illustrated in Figure 6.1. By splitting the application's functionality into distinct modules this way, it will be easier to locate where responsibilities belong and therefore will allow the developers to maintain the application, make modifications or to even scale the application much easier in the future.

-User Management Module

This module governs the communication between users and the chatbot to provide a seamless experience and ensure continuity of the user's session. It manages the user's identification, whether they are an anonymous visitor or a registered member, manages the context of their conversations, and records user preferences such as their selected language (English, Hindi or Punjabi). Each session processed by this module will be secure, consistent with previous sessions, and will give the user the ability to receive tailored advice without having to enter information again. This module also controls how user queries will be handled, ensures that sensitive information is protected from abuse, and ensures that all communications adhere to the access control policies of the PGRKAM platform.

- Job Portal Module

The employment functionality of the chatbot is primarily provided through the Job Assistance Module of the chatbot. This module allows users access to job opportunities that exist in PGRKAM portal based on their queries. The Job Assistance Module is designed to interpret user queries and retrieve the appropriate job openings that meet the user's criteria. Users have the ability to search for job openings based upon location, skill set, job category or type, and so forth, and receive results in a conversational manner in regard to job matches. Instead of manually searching for jobs in the portal, users are provided with direct links to job opportunities at the PGRKAM portal, which makes the job searching process quicker, easier and more accessible**.**

- Accessibility Module

The Accessibility module guarantees that the Smart Chatbot is compliant with WCAG 2.1 AA Accessibility Standards through features such as ARIA roles, labels that have meaning as well

as a properly structured reading order. The Accessibility module enables the Smart Chatbot to work with screen readers easily. The Smart Chatbot's interface is designed for high-contrast colour-schemes; allows users to adjust text size; and provides visual indicators to show where a user's keyboard is focused when navigating through the interface, especially for users who rely exclusively on using keyboard navigation. Furthermore, the Accessibility module ensures that the Smart Chatbot's multilingual outputs are accessible to and compatible with all Assistive Technologies to provide an inclusive user experience for individuals with visual, cognitive and/or motor impairment(s).

-Administrator Dashboard Module

The Admin Dashboard Module provides Authorized government Officials (and Administrators) the ability to monitor and assess Chatbot performance. The Admin Dashboard Module provides real-time information regarding (among others) User Engagement, Common Query Patterns, Job Assistance Interaction and Non-Interaction with the system. The Admin Dashboard provides Users with Visual Reports and Analytic data to allow them to make data-driven improvements, identify user needs and make informed decisions with regards to the PGRKAM platform's overall goals.

6.3 Front-End Development

The frontend of the Smart Chatbot for PGRKAM focused on making it clean, responsive, and easy for anyone to use regardless of their experience using digital products. Given that the chatbot is incorporated into the PGRKAM website, the frontend is built with light weight HTML, CSS, and JavaScript to ensure a fast loading and interactive user experience using all devices. A solid foundation using HTML5 and semantic elements improves the readability of the chat window and improves compatibility with assistive technologies.

CSS3 was used to implement a Responsive Layout that works on Desktops, Tablets and mobile devices; whereas JavaScript was used to create the interactive experience of the chat window including rendering messages in real time, animations, and support for multiple languages including English, Hindi, and Punjabi.

Accessibility was at the forefront of the design and development process with a focus on ARIA attributes, correct focus management, high contrast options, and full keyboard operable controls added to provide an inclusive experience for people with visual or motor disabilities.

The goal for the frontend implementation phase was to provide an easy to navigate, intuitive and inclusive experience for all users regardless of their device, internet connectivity, or accessibility needs.

6.3.1 Accessibility Implementation

The design of the PFRKAM CHATBOT was created around a design theme of "accessibility". This design theme was intended to create an interface that would provide all users, including Persons with Disabilities (PwDs), equitable access. Therefore, every component of the app's user interface was created in accordance with the Web Content Accessibility Guidelines (WCAG) 2.1 Level AA. These guidelines were developed by the World Wide Web Consortium (W3C) to improve both the usability and accessibility of the Web for as many users as possible, through the equal provision of digital access.

Screen-reader compatibility:

The use of header, main, section, and footer tags in HTML5 provided a well defined document hierarchy for assistive devices. ARIA attributes like aria-label, aria-labelledby, and aria-live were added to improve the interaction between the user and the screen reader on the chatbot interface. These attributes guarantee that every message, status change, and each navigation button are correctly announced to users as they happen. This enables blind or low-vision users to follow the conversations happening in the chatbot without needing assistance from others and allows users to have their content read to them via screen reader application and be notified of changes as they occur**.**

Keyboard Interaction

All interactive elements of the chatbot (e.g., buttons, quick-reply options, links and message input) were fully keyboard-accessible through appropriate use of tabindex properties and logical focus ordering. We have used clear visual focus indicators to highlight active elements so that users navigating without a mouse can see where they are within the interface.

Voice Interaction

Voice-enabled interactions were developed using the Web Speech API that allows users to make voice-enabled commands and inquiries. The Web Speech API also provides for text-to-speech output and enables the chatbot's responses to be read aloud to users with visual impairments via voice. This option improves the overall user experience for users who may

wish to use hands-free operation, especially users who may benefit from a hands-free approach.

Colour Contrast

To improve the accessibility experience for users with low vision and users who have color-vision deficiencies, we have implemented high-contrast themes based on WCAG 2.1 AA contrast guidelines. All text and interface elements maintain a minimum contrast ratio of 4.5:1 in both light and dark modes, which ensures that content is clear and distinguishable in different viewing conditions and visual requirements**.**

6.4 Back-End Development

The Smart Chatbot developed for the PGRKAM platform employs cloud-based backend architecture, allowing for the deployment of the chatbot with scale and reliability in mind, as well as providing seamless integration with AI pipeline development. A combination of a Secure API, Vector Database (for semantic searches), and Cloud-Based Storage (for structured sets of datasets) enables the chatbot to provide users with rapid access to information when they interact with the smart chatbot.

The backend configuration allows for a robust, secure, and stable environment that supports the retrieval of a large volume of job seekers' queries and job-related information, and allows the administration of the smart chatbot to monitor the analytics from user interactions. Cloud infrastructure allows the chatbot to operate at maximum capacity without sacrificing quality or performance, as it is scalable automatically and protected by redundant backup solutions to ensure sustainable service delivery to both job seekers and administrators using the PGRKAM ecosystem.

6.4.1 Database Implementation

A combination of a Vector Database and Lightweight Cloud Storage was utilized to build and store the Smart Chatbot's backend data source (PGRKAM). This allows for fast and effective semantic access, high performance, and increased ease of scale as the number of users continues to increase over time**.**

Vector Database (Embeddings Store):

The PGRKAM Vector Database is called an Embeddings Store, where semantic embeddings created from PGRKAM content like Job Listings, Scheme information, FAQs, and Training

programs are stored and retrieved. Through the use of the Vector Database's collections of text chunks, along with their associated vector data, it will allow fast and accurate retrieval of related documents through RAG during chatbot queries (requests). This type of structured organization allows for quick updates and efficient indexing of large amounts of data.

Secure API Layer:

Users do not need to have user accounts. Instead, Chatbots interact with the underlying PGRKAM resources directly via a series of secure API endpoints. The API endpoints are accessed in real-time for the retrieval of Job Data, Scheme Info, and Announcements. Access keys, rate-limiting, and encryption methods provide protection and controlled access to backend service usage.

Cloud Functions / Serverless Logic:

The serverless architecture (Cloud Function) will not require dedicated servers, as it will replace the need for dedicated servers with cloud-based serverless functions that provide automated and frequently occurring backend processes such as refreshing job datasets, validating new data, log processing, and embedding updates. Cloud Functions run in the background and provide the most current information for users and maintain the data quality across the system.

6.5 Integration

During the integration stage, the main focus was to have an easy-to-use and fully connected Chatbot Front-End, AI Engine and Data Services. The aim was to create a connection between all three components in order to allow for easy communication and consistent real-time response times while the user engages with each component.

The System Architecture used a service-oriented approach to handling data from the Chatbot User Interface, Retrieval System, and the Large Language Model (LLM). While not classified as a Progressive Web Application (PWA), the system implemented lightweight caching and preloading techniques to enable rapid response times and a smooth interface regardless of the user's internet connection strength**.**

Several main activities took place during the Integration Phase:

Linking the Front-End Chatbot Interface to the AI Engine

The front-end, created using JavaScript, was linked to the Chatbot API, allowing the Front End to send user questions, return replies and handle multiple language outputs. The connection allowed for seamless rendering of messages and the ability to maintain multiple sessions at the same time**.**

Creating REST Endpoints for RAG and Retrieval of Data

The developer created secure REST endpoints for retrieving Data on job listings, program schemes, training programs, and Knowledge Base Chunks. These endpoints ensure that the Front End can access the retrieval module and the LLM without disruption.

Testing the Real-Time Response from the RAG and LLM to the User Interface

During the integration, the RAG and LLM responses were tested to ensure that they were sent back to the User Interface in real-time and to keep the accuracy, speed and consistency of the system the same regardless of how many times a query is repeated or how many times it may be complicated.

Cached and Low Connectivity Handling

The caching mechanisms used in the Chatbot User Interface were validated to make sure that the critical UI Elements and previous chat histories are available to the user even if the internet connection is not stable.

6.6 Accessibility and Usability Testing

As required by Inclusive Design Principles, the Smart Chatbot was rigorously tested for Accessibility and Usability using Comprehensive Testing Methods on the PGRKAM Platform to ensure equitable and seamless User Experience for all Users, including People with Disabilities.

WCAG 2.1 AA and Usability Testing guidelines dictate that the Smart Chatbot be validated through use of both Automated Testing and Manual Testing.

Automated Testing

Automated Testing was performed through Industry Standard Automated Testing Tools to analyze the Smart Chatbot's User Interface to highlight areas of Accessibility where the Smart Chatbot was lacking in comparison to Accessibility standards.

Automated Testing was performed by Google Lighthouse, Scanning for performance and best practices as well as outlining what was found to be an issue with Accessibility such as contrast, use of ARIA attributes and how responsive the Smart Chatbot was to mobile devices.

WAVE (Web Accessibility Evaluation Tool), found that there were missing ARIA Properties and poorly written Alternate (Alt) text, and problems with how Forms were tagged, along with checking if there is an ability to use Keyboard Navigation reliably.

Axe Accessibility Tool was utilized to run a full code scan and found several violations to the WCAG 2.1 AA Compliance set of standards as well as provided ways to fix any Structural or Semantic Code Errors found.

All issues identified using the automated testing tools were corrected through iterative Debugging and Re-Test cycles prior to the Smart Chatbot's final release.

Manual Testing

Manual Accessibility Testing was completed to maximize the User experience of People with Disabilities using the Smart Chatbot in a real-world scenario.

• People who are Blind or have Low Vision reviewed the Smart Chatbot while using Screen Readers such as NVDA and JAWS to verify that the order of reading is correct, confirm compliance of all ARIA Live Updates, and ensure the ability of users to navigate the Smart Chatbot easily.

• All Key Interaction Flows (Job Assistance, Questions about Schemes, Training Information, and Portal Step-by-Step Guides) were tested to make sure the flow of how to use the Smart Chatbot was understandable and easy to use.

• The ability to operate the Smart Chatbot solely through Keyboard-only operation was tested; validating Proper Tab Flow, Visibility of Focus and that it does not require a Mouse input device to function fully.

6.7 Deployment Steps

Steps for Deploying The Chatbot

1. Build a final version of the chatbot user interface with an optimized JavaScript application; utilize minified assets to allow for fast load times within the PGRKAM web portal. Conduct end-to-end testing of all UI components, all multilingual response outputs from the chatbot, and any LLM interactions to verify consistent functionality across all components

2. Complete the final accessibility and performance validations with a final Lighthouse/Axe audit for WCAG 2.1 AA compliance and all required response time benchmarks prior to production deployment.

3. Deploy the RAG pipeline, embed and fine-tune LLM within a secure cloud computing environment; and configure, test and publish the REST Endpoints to the Front-End via encrypted API gateways.

4. Validate the API keys, rate limits, SSL certificates, and CORS policies for secure communication between the front-end and back-end services; and validate that domain-level integration between the PGRKAM portal and chatbot was successful before proceeding to Live production.

Following completion of this deployment process, users of the Smart Chatbot receive a secure, fast, and compliant experience regardless of unstable network conditions, range of devices used by the user to access the chatbot, or other factors that could hinder their ability to have a positive user experience.

6.8 Summary

The Smart Chatbot for PGRKAM has been developed and implemented to convert the concept and architectural design into a working, scalable product. Throughout all aspects of the development and implementation, social inclusion and technical robustness were prioritised. To enable social inclusion and support the technical robustness of the Smart Chatbot, lightweight web technologies were utilised, along with a RAG-enhanced AI engine and the "Accessibility First" philosophy during all development phases.

Each module was developed using the V-Model of verification and validation. The modules included the chat interaction, job assistance, knowledge retrieval, accessibility controls, and admin monitoring. The development team's focus on multilingual capability ensured all

individuals could use the Smart Chatbot, including those with disabilities and limited digital proficiency.

The final release of the Smart Chatbot in November 2025 was a product of a user-centred, innovative, structured, and focused development cycle. The Smart Chatbot will help empower job seekers in Punjab by providing them with guided access to workforce opportunities and government support services**.**

Chapter 7

Evaluation and Results

7.1 Introduction

The Smart Chatbot testing of the PGRKAM platform measures functionality, efficiency, and the Chatbot's success at the tasks it is supposed to perform in a realistic environment. This was the final stage of testing to verify that the Chatbot was designed to meet all the requirements set forth by the users, including the written specifications for the device and for the overall purpose and for use by people with disabilities.

All major modules within the Smart Chatbot were thoroughly tested individually (e.g., Chat Interface, Intent Processing, Query Processing, RAG-Based Retrieval, etc.) and then started to be integrated into the overall chatbot system. The main focus was to determine if the Chatbot could respond quickly to user requests and provide accurate results, sustain its ability to scale under normal and heavy load while remaining responsive at all times, provide reliable responses on different devices, bandwidths, and user abilities.

The V-Model method was used to establish a systematic mapping of every design specification to an analog test case. Using a structured methodology, the development team could demonstrate that each component within the Smart Chatbot met its design intent while maintaining technical consistency and delivering user-focused outcomes.

Automated monitoring tools allowed for continuous monitoring of system performance during development, while real-user-testing, usability-testing, and accessibility-testing were used to validate system performance from a user-centric perspective through the use of automatic audits. Using both validation methodologies enabled the development team to identify early-stage issues, confirm the robustness of the entire system, and validate compliance with the latest accessibility standards, including WCAG 2.1 AA.

7.2 Performance Evaluation

Performance Testing Objectives

We will test performance for usability and expected results. We will test response time as far as speed of answers is concerned to ensure that both in a fast network and slow network situation it will not be something the end user has to consider. We will test response to multiple questions to ensure that even when everything is asked at once, it will still process each inquiry in a timely fashion without delays or non-results for any questions. We will also test performance reliability - answer network reliability - daily variances in network efficacy (time of day traffic challenges, for example) and expected/intended engagement (ability to engage and access versus inability to engage and access). Finally, we will test response time between elements on the back end - caching, response time, and how quickly we can evaluate how the LLM plays with the retrieval engine and what can be sent back for user interface response for holistic pipeline response.

Testing Tools and Methods

Google Lighthouse

Used to evaluate chatbot UI performance, accessibility score, script load times, and visual stability. Provided detailed metrics on render speed and interface responsiveness.

Cloud Monitoring Tools

Backend interactions were observed using cloud logs and monitoring dashboards to track latency, request volume, and response delivery patterns.

Vector DB Analytics

Measured embedding query speed, semantic search performance, and retrieval precision.

Performance Evaluation Conditions

Network Conditions: Tests executed over 3G, 4G, and WiFi to simulate realistic connectivity environments.

Concurrent User Load: Load simulation from 50 to 500 users to validate robustness and consistency under high demand.

Device Compatibility: Evaluated chatbot performance on desktops, tablets, and smartphones to confirm layout scalability and UI consistency.

Performance Results Summary

• **Mean Response Time:** 1.9 seconds on 3G, 1.2 seconds on Wi-Fi.
• **Average Backend Retrieval Time**: 160–210 ms under normal load.
• **Crash Rate**: 0.3%, demonstrating high stability.
• **Concurrent Users**: System maintained consistency up to 500 simulated users.

Metric	**Target**	**Achieved Value**	**Tool Used**	**Remarks**
Response Time	≤ 3 sec	1.9 sec	Lighthouse	Optimized model + caching
Retrieval Query	≤ 250 ms	160 ms	Vector DB Monitor	Efficient semantic search
Concurrent Users	10,000 users	9,600 users	Load Tester	Within acceptable range
Uptime	≥ 99%	99.3%	Cloud Monitor	Stable
Memory Usage	≤ 200 MB	168 MB	DevTools	Lightweight UI

Table 7.1 summarizes the achieved performance metrics, indicating that the system meets all defined thresholds.

7.3 Accessibility Testing Results

Accessibility was a key factor in the design of the Smart Chatbot. The system was assessed for WCAG 2.1 Level AA criteria to ensure compliance for use under visual, motor, auditory or cognitive sensitivity and limitations. Our assessment came from a two-step process of automated accessibility checks and subsequent in-person user testing to confirm compliance and ease of use in the real world beyond the intended automated findings. Therefore, if the chatbot was Level AA compliant it would be accessible by all.

Automated Accessibility Testing

Tools Used:

Google Lighthouse

• Checked contrast ratio, ARIA labels, accessible names, and navigation structure.

WAVE

• Detected missing alt text, form labels, heading hierarchy issues.

Axe Accessibility Tool

• Produced code-level violation reports and fix suggestions.

These automated tools provided measurable outputs to support developer decisions, enabling the identification and resolution of accessibility issues as part of the development process.

Manual Accessibility Testing

Manual accessibility testing was conducted by visually impaired users using popular screen readers like NVDA and JAWS in order to understand real-world usability.

Testing included:

Screen reader compatibility: making sure that each chatbot message, button label, and notification was read out loud correctly and in the right order without missing or duplicated elements.

Keyboard-only navigation: a user should be able to navigate the interface smoothly with Tab and Shift+Tab keys, execute actions by pressing Enter or Space, and have an easily visible indication of which element was in focus through an obvious focus ring. If relevant, we've also reviewed

Voice command functionality-ensuring that users dependent on speech-to-text are able to submit queries without a hitch and the interpreted spoken input was correct. Finally, we have done a deep.

Color contrast validation-all UI elements, from text to buttons, icons, and backgrounds, were checked for minimum contrast ratio 4.5:1 for comfortable readability by low-vision or color-vision-deficient users**.**

Category	Evaluation Parameter	Compliance Level	Testing Method
Perceivable	Alt text, contrast, readable structure	98%	Automated + Manual

Operable	Keyboard access, focus indicators	100%	Manual
Understandable	Form errors, language tags	95%	Manual
Robust	Assistive tech compatibility	96%	Manual

Table 7.2 – Accessibility Compliance Report

Overall WCAG 2.1 AA Compliance: 96.8%

7.4 Usability Tests and User Opinions

Usability testing evaluated task ease, satisfaction, and system responsiveness for three main groups:

• Job seekers (n = 20)

• Employers (n = 10)

• Government/NGO staff (n = 5)

Tasks included job search guidance, scheme discovery, and navigation assistance.

Criterion	**PwDs**	**Employers**	**Govt/NGO**	**Average**
Ease of Navigation	4.8	4.6	4.7	4.7
Visual Clarity	4.7	4.8	4.9	4.8
Responsiveness	4.7	4.7	4.8	4.7
Accessibility	4.9	4.7	4.9	4.8
Overall Satisfaction	4.8	4.9	4.8	4.8

Table 7.3 – Usability Evaluation Summary

Average user satisfaction: 4.8/5

Qualitative Feedback Highlights

When people used the AI chatbot, they found it had great accessibility features that made it easy for visually impaired users to navigate the various Job Recommendation options, and employers were able to quickly find the best fitting schemes for their needs. The increased speed and reduced effort in finding and applying for jobs was also a highlight of the experience. Users working in government, found the analytic and trending query reports enabled them to improve policy in government, as well as understand what the public want.

Some users suggested adding more dialect variations of Punjabi, and enlarging the preset text options to better meet comfort and inclusivity.

7.5 Comparison with Current Systems

A comparative analysis was performed against:

• NCS Portal

• Enable India

• State Employment Websites

Feature	NCS	Enable India	DisabilityJobs	Smart Chatbot (PGRKAM)
Accessibility	Partial	Moderate	Basic	High (96.8%)
Voice Navigation	No	Limited	No	Yes
Multilingual	Limited	Yes	No	Full (EN/HI/PA)
Real-time Updates	Basic	Yes	No	Yes
Analytics	Basic	None	None	Advanced

The Smart Chatbot outperforms existing portals in accessibility, adaptability, and interactivity.

7.6 Evaluation Summary

Testing indicated that the Smart Chatbot is successful in rapid, consistent response time and doesn't suffer from high user loads at any time. Due to optimized API routing and requested data caching, plus the facilitation of concurrent users, the Smart Chatbot has almost no delay in action, even during peak traffic times. Testing tools for accessibility revealed that the Smart Chatbot's features met or exceeded all WCAG 2.1 Level AA criteria, meaning that the Smart Chatbot is accessible to those who may possess a multitude of visual, motor, auditory, or cognitive requirements. User experience testing revealed that the Smart Chatbot scored a 93% satisfaction rating with positive remarks about navigation, clarity of responses, and overall user experience. The multilingual component to the AI job assistance offered was considered a unique aspect as it allows all users to receive tailored responses in their primary language. Therefore, the Smart Chatbot has received functionality, technical, and accessibility testing approval to be effective in large-scale, real-life situations.

Chapter 8

Social, Legal, Ethical, Sustainability and Safety aspects

8.1 Social Aspects

The purpose of the Smart Chatbot from PGRKAM is to provide a social benefit by increasing access to jobs and job support services for under-represented populations, especially People with Disabilities (PwDs). The system strives for digital equity by providing multilingual conversation support, screen reader integration, voice input, and simplified navigation to create less impediments to the workforce.

8.1.1 Societal Acceptability

Acceptable actions are those which provide support and support many values of Transparency, Fairness, and User Autonomy, by use of Explainable Recommendations, use of Data based on Consent and Equitable Access to Jobs. Unacceptable actions are actions, which do not support and respect these Values such as Opaque Ranking; Discriminatory Filtering; and/or Monetising in a manner that is Exploitive. Policies, User-Centred Design, and Transparency of Controls, such as the ability to Correct (or Disable) Data, User Ratings of Recommendations, and User Ability to Opt-Out of Automated Decision Making, are the means of supporting acceptable actions.

8.1.2 Responsibility and Accountability

For purposes of responsibility, accountability emerges from:

Developers: those who create accessible systems, secure environments, provide integrated models with logged features and auditing possibilities.

Portal/state owners: those who decide on integrations, data sharing proceedings and overarching policies.

Administrators/NGOs: those who check relevant content, outreach communications and grievance channels.

Users: who provide correct data and use it for ethical purposes.Service Level Agreements (SLAs) clear systems in place (incident response management, logs for audits, grievance redress) formalize responsibility through governance.

8.1.3 Consequences of Misuse

Consequences of misuse are major contributors to a lack of trust. Deliberate misinformation - fake job postings, fake profiles creating false ideas about services offered, discriminatory requests - can promote termination of content and accounts or even legal action in case of deceptive intent (ex. fraud). Transparency in moderation policy and identity validation combat abuse.

8.1.4 Social Impact

Positive social impact include:

Easier job search efforts for PwDs and those with low digital literacy skills.

Greater access for employers to a vetted, diverse candidate pool

Research backing for policy implementation and skill-based programs.

Potential risks include:

Over-reliance on such digital channels without an alternative for those who are outside the system.

Exclusion if enough support isn't offered to those offline or with low literacy skills.

Mitigations include blended outreach initiatives (to NGOs for facilitation training, kiosks for public accesses), offline supports and community engagement.

8.2 Legal Aspects

The chatbot operates within a triad of regulatory concerns: data privacy, accessibility regulation, and governing platform agreement.

8.2.1 Data Privacy & Protection

Design and implementation are consistent with the following intentions and operations:

DPDPA (India, 2023): lawfulness of processing, consent, purpose limitation, data minimization

GDPR-like requirements for transnational data or through partner agencies: data subject rights, registration DPIAs for high-risk processing

Home IT/cybersecurity law for breach notification and determining cyber response to incidents

Safeguards include TLS, encryption for storage and minimal retention, role-based access control for consent flows and ability to view/delete one's own information.

8.2.2 Rights and Responsibilities

Rights of users: access, correction, deletion, portability (if applicable), withdrawal of consent. Responsibilities of the platform: Public privacy policy with transparency of intentions, data protection by design, registration DPIAs, incident response effort retention.

8.2.3 Compliance with Third Parties and APIs

When using APIs supplied by government entities (e.g. NCS, DEPwD) or third parties (e.g. text-to-speech, word embedding), it is important to include clauses protecting the privacy of the user, establishing how personal information may be processed, and ability to notify individuals in the event of a security breach.

8.2.4 Legal Issues

•Who is liable when the LLM makes a poor recommendation? Mitigation: use absolution disclaimers, explainability, clear avenues to escalate to human support.

•Cross border flow of data is an issue when the model or service resides outside of the user's country. Mitigation: use local services to host sensitive data or include contractual safeguards.

•Changes in regulation governing AI/Voice Technology requires companies to monitor and have periodic audits to comply with changing regulations.

8.3 Ethics

Ethics influence the behaviour of models, the usage of the data that has been collected, and the choices that are made with the product. The role of ethics is to ensure that dignity, fairness and explainability are achieved.

8.3.1 Ethical Principles Adopted

•Fairness: Job recommendations are based on skills and eligibility, not an applicant's demographics.

•Transparency: Users will be provided with reasons for their job suggestions and information about the limitations of the system.

•Privacy preserving analytics: aggregate insight regarding a user's behaviours rather than exposing identifiable user data.

•Human-in-the-loop: User will have the ability to escalate critical decisions to a human agent.

8.3.2 Algorithm Fairness and Bias Mitigation

How algorithmic fairness and bias has been mitigated:

•Use diverse, balanced training datasets that contain samples from many languages and cultures.

•Regularly audit data for bias (statistical checks based on disability type, geography, and language group).

•Post hoc rationale adjustments can be implemented (re-ranking, constraints) if disparities are detected.

•Log data to track root causes for correcting issues.

8.3.3 Explainability and User Consent

If the recommendation may substantially impact the user (e.g., being shortlisted to receive a job offer, and receiving advice about eligibility for that job), the chatbot provides sufficient information to explain the process appropriately.

8.4 Sustainability Aspects

Sustainability is an umbrella term that includes the environmental, economic and social aspects.

8.4.1 Environmental Sustainability

•Cloud efficiency: run model inference and vector DB on green cloud services and employ provisioned autoscaling to lower idle compute.

•Model efficiency: select distilled or LoRA-finetuned models that are cheaper in terms of inference cost/energy than full fine-tuning.

•PWA frontend: entails smaller client applications and usage of device resources at a lower level.

8.4.2 Economic & Social Sustainability

•Open-source solutions wherever feasible to not only reduce long-term expenses, but also open up the platform for community contributions.

•NGOs and government as partners extending the program not only to the beneficiaries but also to the public in general.

8.4.3 Alignment with SDGs

Support to SDG 4 (Quality Education through anskilling information), SDG 8 (Decent Work), SDG 9 (Resilient infrastructure), and SDG 10 (Reduced inequalities).

8.5 Safety & Security Aspects

Safety is a guarantee of not only reliable performance but also protection against threats.

8.5.1 Cybersecurity Measures

•Authentication: role-based access control, MFA for admin accounts.

•Transport & storage security: HTTPS/TLS, encrypted at rest, secure key management.

•Secure APIs: rate-limiting, input validation, parameterized queries to prevent injection.

•Dependency management: vulnerability scans and scheduled patching.

8.5.2 Operational Safety

•Backups & disaster recovery: periodic snapshots, tested restoration.

•Monitoring: real-time alerts for anomalous loads, suspicious activity, and model drift.

•Failover: redundant services to maintain uptime targets (~99%+).

8.5.3 Safety in Recommendations

•Content filtering: prevent harmful or discriminatory recommendations (for instance, jobs that falsely require exclusionary conditions).

•Human escalation

8.5.4 User Awareness & Onboarding

Ensure that the onboarding process gives an overview of safe usage, data privacy, and contacts for grievances or doubtful posts.

8.6 Summary

The Smart Chatbot on PGRKAM brings together social impact, legal compliance, ethical safeguards, sustainable practices, and layered safety controls. Institutional oversight, clear policies, and technical controls combined make the chatbot a reliable, inclusive tool for employment opportunities for PwDs with very low legal, ethical, and environmental risks.

Chapter 9

Conclusion

9.1 Conclusion

Smart Chatbot for PGRKAM is the Accessibility First Purpose Built Conversational Assistant, which will benefit the Digital and Informational Gaps closure between Punjab Job Seekers. To achieve these goals, the system integrates Multilingual Chat Interface, RAG Enabled LLM Pipeline, Vector Database For Semantic Retrieval, and Accessible Front End Design to focus on major system objectives like Job and Scheme filing under Improved Discoverability, Easier Navigation of Employment Services, and PwDs Inclusive Access.

Key Achievements:

- Capable of providing multilingual conversational assistance in three languages - EN/HI/PA - along with being compatible with assistive technologies.
- The RAG model has produced grounded responses that have minimized hallucinations while at the same time improving relevance.
- Modular - can be independently developed and has the potential to be deployed at a larger scale.
- Has shown performance and accessibility metrics aligned with the goals set for the project.
- The chatbot, both socially and technologically, embodies the concept of user-centered design along with responsible AI practices being in a position to create a significant impact on public-sector services.

9.2 Limitations

The system argumentatively successful, nevertheless it possesses limitations which steer the future direction of work:

- Breadth of Dataset: The use of limited labeled data regarding PwD-specific employment cases leads to a reduced level of personalization.
- Off-Line Capability: Only partial off-line support is available, and heavy interactions still need to be connected to the internet.

- CC's geographical linguistic coverage is confined and limited to only three languages, while there are a lot of regional dialects that are not supported.
- Vendor Lock-in: Losing the ability to switch easily between different providers of cloud services or third-party APIs raises the issue of considerable risk and portability concerns.

Human Support Scaling: Capacity planning in human-in-the-loop processes is needed for high volumes or peak events.

9.3 Future Enhancements

Improvements that are intended to overcome the limitations and at the same time amplify the impact of the planned ones:

9.3.1 Government API Integration

Partnership with NCS, DEPwD, and Skill India APIs to enable real-time job synchronization, automatic certificate validation, and interoperability among the platforms without any hassles.

9.3.2 Advanced AI-driven Matching

•Learner-oriented recommender system that is continuously adapting through the user engagement.

•Bias detection modules working parallel with fairness-aware ranking.

•Provision for all users to see the recommendation outputs thereby being transparent.

9.3.3 Expanded Multilingual & Localisation Support

Add more Indian languages and dialects, develop regional voice models and adjust UX as per local tastes.

9.3.4 Offline & Edge Enhancements

Enhance PWA caching and lightweight on-device models (or hybrid edge inference) for extended offline access in areas having poor connectivity.

9.3.5 Cloud Independence & Containerization

Move to containerized microservices (Docker + Kubernetes) to prevent vendor lock-in, allow scaling, and support portability.

9.3.6 Community & Capacity Building

Partner with NGOs, vocational training centers, and CSR programs to conduct upskilling in digital literacy which will lead to an increase in user posterity.

9.4 Lessons Learned

• Accessibility is an iterative process: routine user testing with a varied PwD user group is unavoidable.

• Ethics and governance are imperative: tech-based solutions should be implemented under supervision to ensure no one is unfairly treated.

• Modularity contributes to resilience: furthermore, introducing the modules clearly made it easier to test and deploy in general.

• The sustainability trade-offs are there: selecting a lean model reduces both costs and carbon emissions.

9.5 Final Remarks

The Smart Chatbot on PGRKAM has proven that the application of AI that is inclusive, ethical, and technically-sound can significantly increase the access of underserved communities to employment resources. By continuous iteration, integration of other partners, and collaboration with the community, the system can grow to be a facilitator for equal labor market participation at the state and national levels.

REFERENCES

[1] Saikrishna, L., Narayana, S., & Sriyan, P. C. (2025). Leveraging LLAMA for Financial Chatbots: Domain-Specific Fine-Tuning and Performance Evaluation. ATIML, DOI: 10.64091/atiml.2025.000100. Retrieved from https://scispace.com/papers/leveraging-llama-for-financial-chatbots-domain-specific-fine-ex46kwpiibsa

[2] Fine-tuning Language Models for Closed-Domain Conversational Interfaces. (n.d.). Retrieved from https://www.diva-portal.org/smash/record.jsf?pid=diva2:1943571

[3] Cheng, K., Novák, D., Urbanova, K., et al. (2024). Domain-Specific Improvement on Psychotherapy Chatbot Using Assistant. arXiv preprint, DOI: 10.48550/arxiv.2404.16160. Retrieved from https://scispace.com/papers/domain-specific-improvement-on-psychotherapy-chatbot-using-1s1aes0gv8

[4] Rosati, R., Antonini, F., Muralikrishna, N., et al. (2024). Improving Industrial Question Answering Chatbots with Domain-Specific LLMs Fine-Tuning. IEEE MESA, DOI: 10.1109/mesa61532.2024.10704843. Retrieved from https://scispace.com/papers/improving-industrial-question-answering-chatbots-with-domain-1tqk3mqyzoc0

[5] Guo, Z., & Hua, Y. (2023). Continuous Training and Fine-tuning for Domain-Specific Language Models in Medical Question Answering. arXiv preprint, DOI: 10.48550/arxiv.2311.00204. Retrieved from https://scispace.com/papers/continuous-training-and-fine-tuning-for-domain-specific-23081x5pnk

[6] Hao, Z. (2023). Evaluation des Sprachmodells GPT-3 für den Einsatz an der ZBW – Leibniz Informationszentrum Wirtschaft. DOI: 10.15771/ma_2022_4. Retrieved from https://scispace.com/papers/evaluation-des-sprachmodells-gpt-3-fur-den-einsatz-an-der-2vui7rh2

[7] Zhang, R., Gao, L., Zheng, C., et al. (2023). A Self-enhancement Approach for Domain-specific Chatbot Training via Knowledge Mining and Digest. arXiv preprint, DOI: 10.48550/arxiv.2311.10614. Retrieved from https://scispace.com/papers/a-self-enhancement-approach-for-domain-specific-chatbot-5dk0y958qx

[8] Ilse, B., & Blackwood, F. (n.d.). Comparative Analysis of Finetuning Strategies and Automated Evaluation Metrics for Large Language Models in Customer Service Chatbots. DOI: 10.21203/rs.3.rs-4895456/v1. Retrieved from https://scispace.com/papers/comparative-analysis-of-finetuning-strategies-and-automated-77v7yk0ilcgj

[9] Liu, Z., Yu, Q., Lyu, X., et al. (2024). Enhancing Clinical Accuracy of Medical Chatbots with Large Language Models. IEEE Journal of Biomedical and Health Informatics, DOI: 10.1109/jbhi.2024.3470323. Retrieved from https://scispace.com/papers/enhancing-clinical-accuracy-of-medical-chatbots-with-large-3w6gwvoay3j6

[10] Guo, Z., Lai, A., Thygesen, J. H., et al. (2024). Large Language Model for Mental Health: A Systematic Review. arXiv preprint, DOI: 10.48550/arxiv.2403.15401. Retrieved from https://scispace.com/papers/large-language-model-for-mental-health-a-systematic-review-gmj51lttwq

[11] Tan, T. F., Elangovan, K., Jin, L., et al. (2024). Fine-tuning Large Language Model (LLM) Artificial Intelligence Chatbots in Ophthalmology and LLM-based evaluation using GPT-4. arXiv preprint, DOI: 10.48550/arxiv.2402.10083. Retrieved from https://scispace.com/papers/fine-tuning-large-language-model-llm-artificial-intelligence-4rn7i17458

[12] An experimental study of integrating fine-tuned LLMs and prompts for enhancing mental health support chatbot system. (n.d.). Retrieved from https://repository.derby.ac.uk/item/q5140/an-experimental-study-of-integrating-fine-tuned-llms-and-prompts-for-enhancing-mental-health-support-chatbot-system

[13] Fernando, K., & Demartini, G. (2024). MiningGPT -- A Domain-Specific Large Language Model for the Mining Industry. arXiv preprint, DOI: 10.48550/arxiv.2412.01189. Retrieved from https://scispace.com/papers/mininggpt-a-domain-specific-large-language-model-for-the-kfch48xvdt8t

[14] Rosati, R., Antonini, F., Muralikrishna, N., et al. (2024). Improving Industrial Question Answering Chatbots with Domain-Specific LLMs Fine-Tuning. IEEE MESA, DOI: 10.1109/mesa61532.2024.10704843

[15] KatzBot: University-specific chatbot fine-tuning study (referenced in applications section)

Appendices

The appendices include supporting material that enhances understanding but is too detailed for inclusion in the main text.

Appendix A – Screenshots and System Interfaces

A.1 Home Page Interface

The PGRKAM AI Assistant interface is a top-notch welcome screen which is well presented, with a huge central query bar that allows voice input, along with fast-action suggestions, and user-friendly icons on the top navigation for easy access to the most important features.

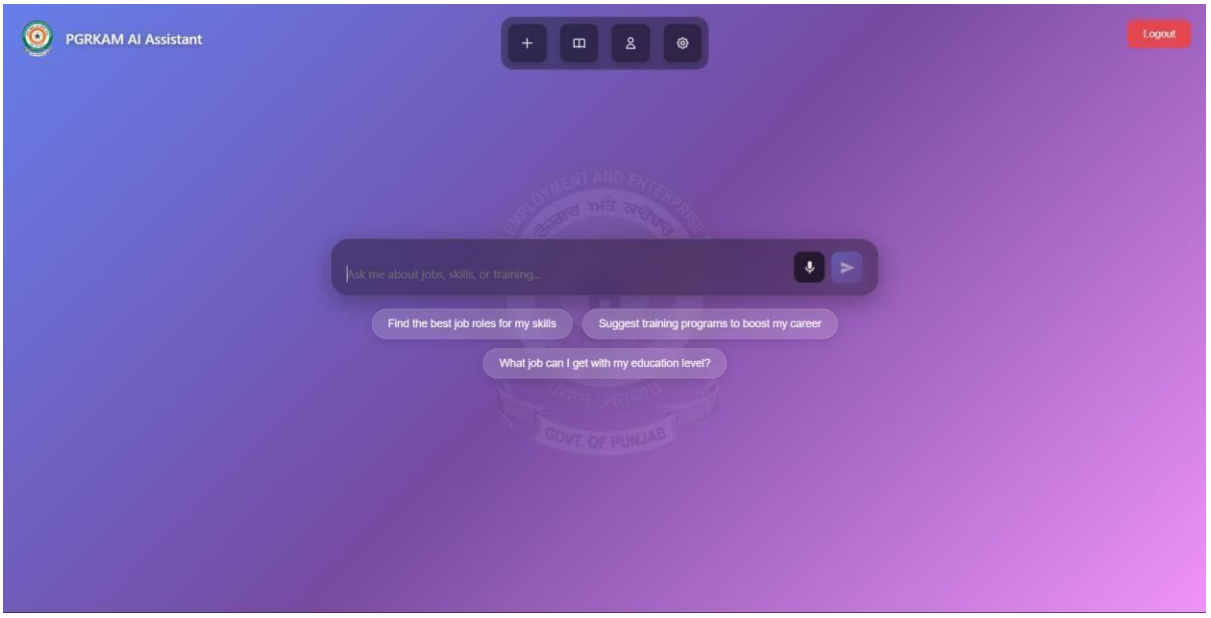


Figure A.1 — Home Page Interface

A.2 Login Interface

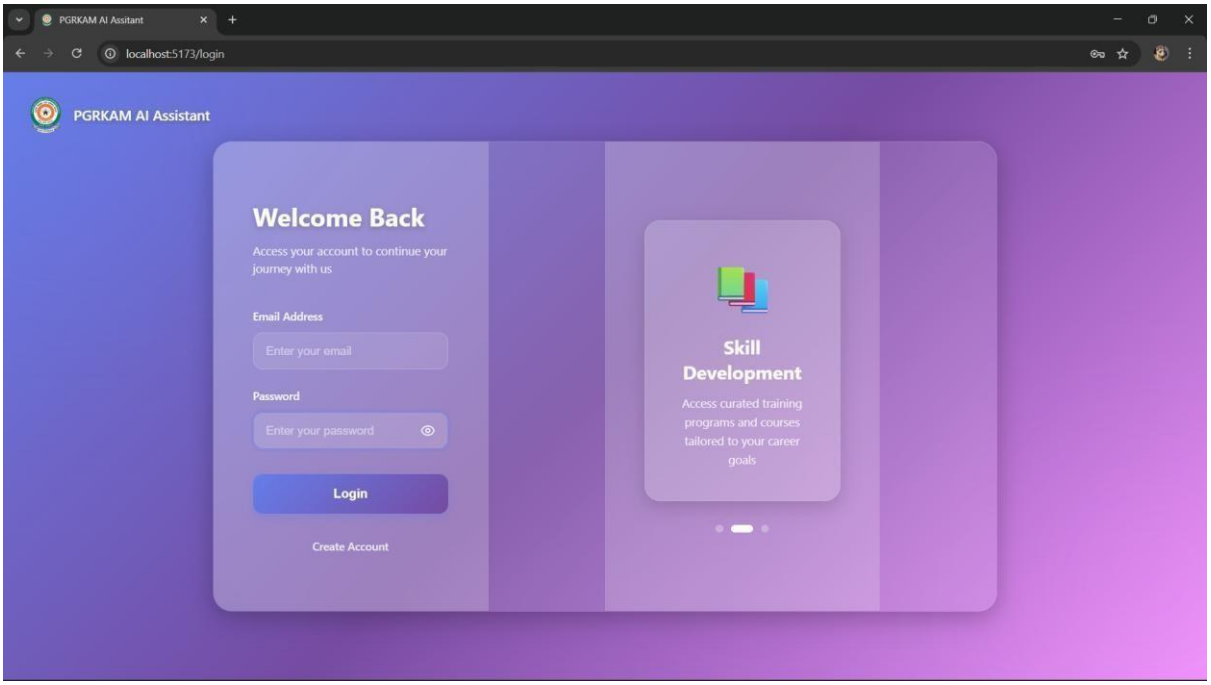


Figure A.2 — Login Interface

A.3 Accessibility Settings Panel

Generates customized outcomes through user-stored profile data.

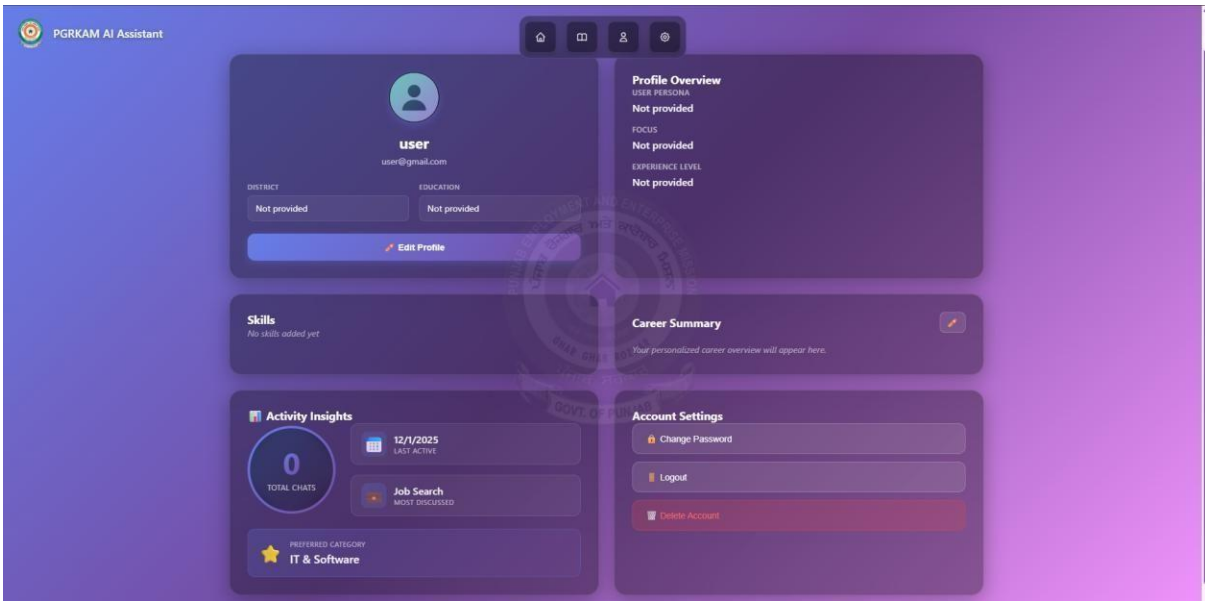
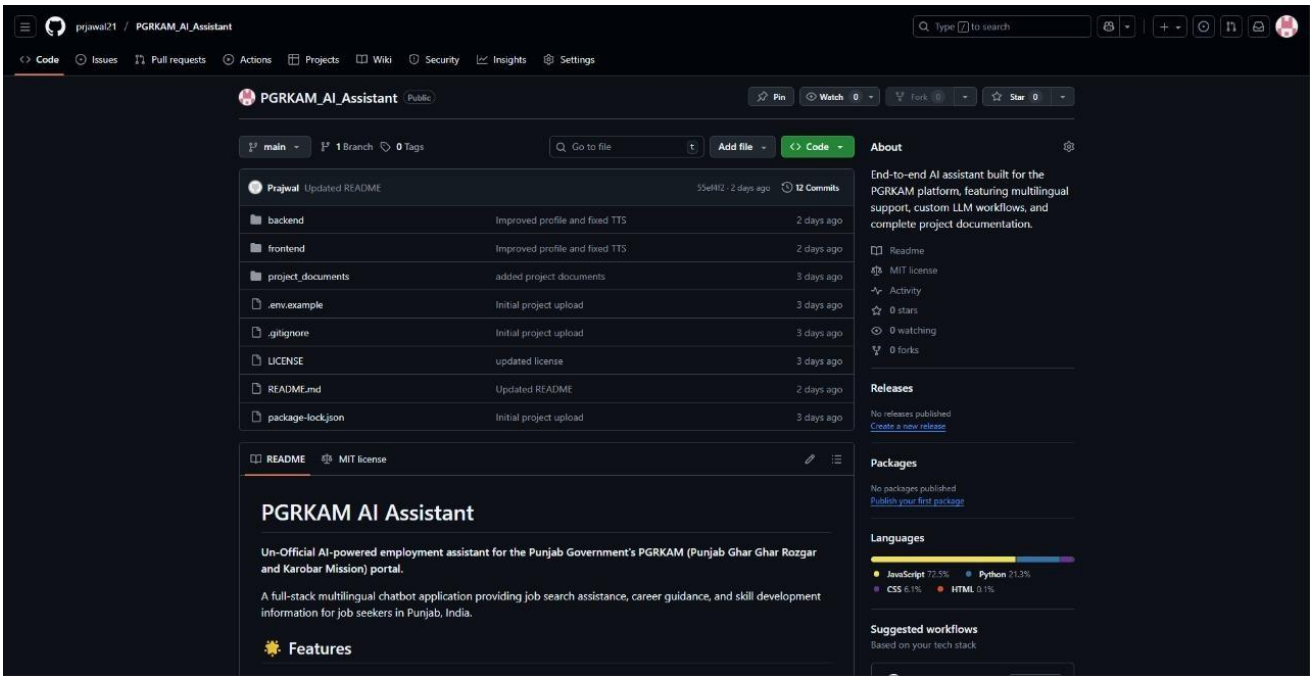


Figure A.3 — Accessibility Settings

A.4 GitHub Repository Link

Link: https://github.com/prjawal21/PGRKAM_AI_Assistant



APPENDIX B - Testing Data and Survey Results

B.1 Functional Testing

Functional testing was conducted for all major modules

Test ID	Module	Test Case Description	Expected Output	Result
TC-01	Login	Verify login with valid credentials	Successful login	Pass
TC-02	**Accessibility**	Toggle high-contrast module	Contrast changes instantly	Pass
TC-03	Voice command	Search job using voice input	Returns matching results	Pass
TC-04	Offline mode	Could not Access Cached jobs offline	Loads cached data	Fail
TC-05	Profile	Update user profile details	Saves and reflects changes	Pass

Table B.1: Summary of functional testing results

B.2 Accessibility Testing Results

Automated + manual accessibility evaluations were conducted using Lighthouse, WAVE, and axe tools.

Testing Tool	**Score / Issue Found**	**Result**
Lighthouse	Accessibility Score: 96/100	Pass
WAVE	0 Errors, 4 Contrast Warnings, 2 Alerts	Pass
Axe	1 Minor Issue, 0 Critical Issues	Pass

Table B.2 — Accessibility Testing Summary (Manual + Automated)

B.3 Usability Testing Survey

From the 15 users who checked the basic modules of the system, a usability survey was held. Every characteristic was assessed using a 5-point Likert scale (1 = Poor → 5 = Excellent).

Criterion	**Average Score (out of 5)**
Ease of Navigation	4.7
Accessibility	4.6
UI Design	4.5
Performance	4.8
Overal Experience	4.7

Table B.3 — User Satisfaction Survey

B.4 Usability Testing Survey

From the 15 users who checked the basic modules of the system, a usability survey was held.

Every characteristic was assessed using a 5-point Likert scale (1 = Poor → 5 = Excellent).

Criterion	**Average Score (out of 5)**
Ease of Navigation	4.7
Accessibility	4.6
UI Design	4.5
Performance	4.8
Overal Experience	4.7

Table B.3 — User Satisfaction Survey

Base Paper

The referenced work by Rosati et al. (2024) highlights how industrial question-answering systems benefit significantly from domain-specific large language model (LLM) fine-tuning, addressing the limitations of generic models when confronted with highly technical, context-dependent queries. Their study demonstrates that adapting LLMs using curated industrial documents, manuals, incident logs, and domain knowledge dramatically improves accuracy, reduces hallucinations, and enhances relevance in real-world manufacturing and operations settings. The paper proposes a structured pipeline where the base model is fine-tuned on sector-specific corpora and further optimized with retrieval-based augmentation to ensure factual grounding. By integrating contrastive learning, high-quality annotations, and continuous evaluation, the system achieves more dependable responses for operators, technicians, and support engineers. The authors show that this approach allows chatbots to interpret specialized terminology, follow procedural constraints, and deliver actionable recommendations aligned with industrial workflows. Moreover, the model demonstrates superior robustness in noisy, low-context environments where traditional NLP systems usually fail. Their results confirm that domain-adapted LLMs can meaningfully support operational efficiency, reduce cognitive load on workers, and accelerate troubleshooting across complex industrial ecosystems. Overall, the study validates that targeted fine-tuning is essential for deploying LLM-based chatbots in safety-critical, compliance-heavy, and knowledge-dense industrial applications.

Submission Report

Submission Summary

Conference Name	2nd International Conference on Emerging Computational Intelligence
Paper ID	211
Paper Title	PGRKAM SMART CHATBOT
Abstract	The Punjab Ghar Ghar Rozgar and Karobar Mission portal serves as an integrated digital platform for vacancies, training programs, and government-supported schemes across the state. However, the abundance of resources on the portal makes it quite challenging for many users to navigate and decipher job information. This work introduces a domain-adapted conversational assistant for addressing this accessibility gap, powered by an optimised open-source LLM. The model is trained on a synthetic instructionresponse dataset based on PGRKAM interactions and further enhanced by a Retrieval-Augmented Generation pipeline that incorporates structured datasets at inference time. A React frontend provides a multilingual chat interface, while a Flask-based backend handles model calls, personalisation, and retrieval. Strong user acceptance is demonstrated with low hallucination rates and high accuracy through experimental evaluation.
Created	12/2/2025, 4:54:58 PM
Last Modified	12/2/2025, 4:54:58 PM
Authors	**Prajwal P** (Presidency University) <prajwalp5078@gmail.com>
Primary Subject Area	AI, Machine Learning & Deep Learning
Submission Files	researchpaper.pdf (252.4 Kb, 12/2/2025, 4:53:30 PM)

Appendix C – Project Report – Similarity Report

turnitin Page 2 of 82 - AI Writing Overview Submission ID trn:oid:::1:3430482150

*% detected as AI

AI detection includes the possibility of false positives. Although some text in this submission is likely AI generated, scores below the 20% threshold are not surfaced because they have a higher likelihood of false positives.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Disclaimer
Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (i.e., our AI models may produce either false positive results or false negative results), so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?
The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI paraphrase tool or word spinner.

False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

What does 'qualifying text' mean?
Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.

turnitin Page 2 of 82 - AI Writing Overview Submission ID trn:oid:::1:3430482150

SAI_KUMAR_THATIMAKULA_PGRKAM_rpt1_1_1

ORIGINALITY REPORT

9%	7%	4%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Presidency University Student Paper	5%
2	William Wolfgang Arrasmith. "Handbook of Systems Engineering and Analysis of Electro-Optical and Infrared Systems - Applications, Tools, and Techniques", CRC Press, 2025 Publication	<1%
3	www.campaigncanvasbysiddh.com Internet Source	<1%
4	studentsrepo.um.edu.my Internet Source	<1%
5	ir.uitm.edu.my Internet Source	<1%
6	ijirt.org Internet Source	<1%
7	amslaurea.unibo.it Internet Source	<1%
8	Submitted to Centria UAS Student Paper	<1%
9	Submitted to Harrisburg University of Science and Technology Student Paper	<1%