

A Project Report on

**RACE Assist – AI Web Chatbot**

Submitted in partial fulfilment for award of degree of

**Master of Technology**

In **Artificial Intelligence**

Submitted by

**Navya T C**

R23MTA09

Under the Guidance of

**Dr. JB Simha**

Chief Mentor, RACE

REVA Academy for Corporate Excellence

**REVA University**

Rukmini Knowledge Park, Kattigenahalli,

Yelahanka, Bangalore – 560064

**September 2025**



# Candidate’s Declaration

I, **Navya T C** hereby declare that I have completed the project work towards the **Master of Technology in Artificial Intelligence** at, REVA University on the topic entitled **RACE Assist – AI Web Chatbot** under the supervision of **Dr. JB Simha, Chief Mentor, RACE.** This report embodies the original work done by me in partial fulfilment of the requirements for the award of the degree for the academic year **2025.**

Place: Bengaluru Name of the Student: Navya T C

Date: Signature of Student

A black and orange text

Description automatically generated

# Acknowledgment of Project Ownership and Usage Rights

I, Navya T C, a student enrolled in the Master of Technology Program and 2025 Year at RACE, hereby acknowledge that any project, including but not limited to software, hardware, research, or other intellectual property created by me during my academic tenure at RACE, is the property of RACE, REVA University.

I understand and agree that RACE has the exclusive rights to use, reproduce, modify, or distribute the projects for academic, research, and further development purposes. This includes the right to monetize, commercialize, or otherwise exploit the projects as deemed fit by RACE.

I acknowledge that I have no objection to RACE, REVA University, using, reproducing, or further developing the projects for the benefit of the institution and its academic community. I further affirm that any commercial or research activities related to the projects conducted by RACE shall not require additional consent or approval from me.

This acknowledgment is made willingly and without any reservations. I am grateful for the education and opportunities provided by RACE, and I recognize the importance of contributing to the academic and research goals of the institution.

Place: Bengaluru Name of the Student: Navya T C

Date: Signature of Student



# Certificate

This is to certify that the project work entitled entitled **RACE Assist – AI Web Chatbot** has been carried out by **Navya T C** with **SRN R23MTA09,** who is a bonafide student of REVA University, is submitting the **first-year** project report in fulfilment for the award of **Master of Technology in Artificial Intelligence** during the academic year 2025. The Project report has been tested for plagiarism and has passed the plagiarism test with the similarity score less than 15%. The project report has been approved as it satisfies the academic requirements in respect of the project work prescribed for the said degree.

Signature of the Guide Signature of the Director

**Dr. JB Simha**

Chief Mentor, RACE

External Viva

Names of the Examiners

Place: Bengaluru

Date:



# Acknowledgement

I would like to express my deep gratitude to my supervisor **Dr. JB Simha, Chief Mentor, RACE** for their continual guidance and support throughout this project. His guidance and vision has been pivotal to make this research possible.

I would like to express my deepest gratitude to everyone who supported me throughout the course of this project. I am truly thankful for the invaluable guidance from my mentors and trainers, whose knowledge and expertise significantly influenced the direction of my work. I would like to extend my sincere appreciation to **Dr. Shinu, Prof. Paramesh G** and **Prof. Kiran Kumar KV**for their unwavering support and encouragement, which played a crucial role in helping me reach this achievement.

I also extend my heartfelt thanks to my classmates for their collaboration and insightful feedback, which greatly enriched my learning journey. My gratitude goes to the program office members for their administrative support and assistance, which was immensely helpful throughout the project.

Furthermore, I am especially grateful to my family and friends, whose constant encouragement and understanding have been a continual source of motivation.

I would like to acknowledge the support provided by the founder and Hon’ble Chancellor, Dr. **P Shayma Raju**, Vice-Chancellor, **Dr. Sanjay Chitnis**, and Registrar, **Dr. K S Narayanaswamy.**

Place: Bengaluru

Date:

Text

Description automatically generated

# Similarity Index Report

This is to certify that this project report titled entitled **RACE Assist – AI Web Chatbot** was scanned for similarity detection. The process and outcome are given below. The plagiarism report is attached in the appendix.

Software Used:

Date of Report Generation:

Similarity Index in %:

Total word count:

Name of the Guide:

Place: Bengaluru Name of the Student: Navya T C

Date: Signature of Student

Verified by:

Signature

Dr. Shinu Abhi,

Director, Corporate Training

# 

# List of Abbreviations

|  |  |  |
| --- | --- | --- |
| **Sl. No** | **Abbreviation** | **Long Form** |
| 1 | AI | Artificial Intelligence |
| 2 | AQAP | Automatic Question-Answer Pair Data Collection |
| 3 | RAG | Retrieval-Augmented Generation |
| 4 | FAISS | Facebook AI Similarity Search |
| 5 | JWT | JSON Wb Token |
| 6 | FAQs | Frequently Asked Questions |
| 7 | UI | User Interface |
| 8 | API | Application Programming Interface |
| 9 | HTML | Hypertext Markup Language |
| 10 | PDF | Portable Document Format |
| 11 | JSON | JavaScript Object Notation |

# List of Figures

|  |  |  |
| --- | --- | --- |
| **No.** | **Name** | **Page No.** |
| Fig.5.1 | Project Methodology | 20 |
| Fig.7.1 | Flowchart: Processing pipeline from user actions to wellness outputs | 27 |
| Fig.8.1 | Sample FER2013 Dataset | 31 |
| Fig.8.2 | Data Pre-processing | 33 |
| Fig. 8.3 | CNN Model | 34 |
| Fig. 8.4 | ResNet50V2 | 35 |
| Fig.9.1 | Session with position emotions | 41 |
| Fig.9.2 | Stress Simulation | 41 |
| Fig.9.3 | Output Predictions of ResNet50V2 Model on test images | 43 |
| Fig.10.1 | ResNet50V2 Confusion Matrix | 46 |
| Fig.10.2 | CNN Confusion Matrix | 47 |
| Fig.10.3 | Learning curve of Custom sequential model | 47 |
| Fig.10.4 | Learning curve of ResNet50V2 model | 48 |
| Fig.10.5 | Learning curve of CNN model | 48 |
| Fig.10.6 | HR Alert Email Output | 59 |
| Fig.10.7 | HR Dashboard of an Employee | 50 |

# List of Tables

|  |  |  |
| --- | --- | --- |
| **No.** | **Name** | **Page No.** |
| Table 6.1 | Fundamental Requirement | 24 |
| Table 9.1 | Designed Test Cases and Expected Outcomes | 38 |
| Table 9.2 | Test Case Results | 39 |
| Table 9.3 | Model Performance Comparison | 42 |
| Table 9.4 | ResNet50V2 Class-Level Performance | 42 |

# Abstract

While institutions of higher education are seeking to broaden their catchment and sustain growth in the students’ base population, intelligent support systems have become important to bridge the gap between potential students and universities. The ***RACE Assist – AI Web Chatbot*** developed at ***Reva Academy for Corporate Excellence, REVA University*** is a futuristic answer-- Power of automated scraping, dynamic data storage and AI chat put to work for the next generation business development and student engagement.

This contrasts with your ‘run-of-the-mill’ portal platform that provide thin FAQ and/or static search and bridges the gap from forked pages to an environment where program specific content streams, retrieves information from official sources in the university, saving each stream in a high performance dedicated database. The platform is powered by a smart router, which immediately processes each query and sends it off to the right program database for context-based answers in real-time AI chat. Such an approach not only creates a trusting and responsive relationship with new and existing students, but also enables business development teams to qualify leads capture them as well as respond to intricate queries and personalized communication at scale.

With intelligent AI-driven orchestration, context-sensitive retrieval and a smooth chat interface, ***RACE Assist – AI Web Chatbot*** makes sure that ***RACE, REVA University*** sits right at the top of tech-powered forward-looking academia. The platform functions as a strategic lever, driving customer acquisition and continued engagement while positioning REVA RACE as a market leader in digital education services.

In the end, ***RACE Assist – AI Web Chatbot*** creates a new standard for in-academic support; automating today’s and tomorrow’s data-oriented processes to propose an innovative business model which can transition a classical educational institution into its own intelligence provider.

**Table of Contents**

[Candidate’s Declaration 1](#_Toc165638414)

[Acknowledgment of Project Ownership and Usage Rights 2](#_Toc165638415)

[Certificate 3](#_Toc165638416)

[Acknowledgement 4](#_Toc165638417)

[Similarity Index Report 5](#_Toc165638418)

[List of Abbreviations 6](#_Toc165638419)

[List of Figures 6](#_Toc165638420)

[List of Tables 8](#_Toc165638421)

[Abstract 9](#_Toc165638422)

[Chapter 1: Introduction](#_Toc165638423) 11

[Chapter 2: Literature Review](#_Toc165638424) 14

[Chapter 3: Problem Statement 18](#_Toc165638425)

[Chapter 4: Objectives of the Study 19](#_Toc165638426)

[Chapter 5: Project Methodology 20](#_Toc165638427)

[Chapter 6: Resource Requirement Specification 24](#_Toc165638428)

[Chapter 7: Software Design 26](#_Toc165638429)

[Chapter 8: Implementation 31](#_Toc165638430)

[Chapter 9: Testing and validation 37](#_Toc165638431)

[Chapter 10: Analysis and Results 45](#_Toc165638432)

[Chapter 11: Conclusions and Future Scope 51](#_Toc165638433)

[Bibliography 53](#_Toc165638434)

[Appendix 54](#_Toc165638435)

[Plagiarism Report 55](#_Toc165638436)

[Paper Publications in a Journal/Conference Presented/White Paper 56](#_Toc165638437)

[Certificate for the Conference Presentation 57](#_Toc165638438)

[Github Link 57](#_Toc165638439)

# Chapter 1: Introduction

In today's higher education marketplace, it is a tough job to reach, engage and enrol dream students. Enrollment Growth: To meet enrollment targets, business development teams need to deliver more than great program quality; it’s now also about getting the right answer – fast!- to increasingly better informed student inquiries. Conventional support tools like emails, static websites and FAQ pages were built for content publishing, not for real-time, personalized responses in sales.

With these in mind, a disruptive digital solution, called ***RACE Assist – AI Web Chatbot*** at ***RACE, REVA University*** was developed. At its heart, the platform utilizes an automatic web scraping and intelligent anthology process to dynamically curate and keep program documentation current pulled from the university website itself. You might have been asking about one of the other 129 (as of this writing) ABA-approved law schools). All information we provide is localized and customized in real-time, just for each program's data, baby—that's how we like it.

The innovation strategy of RACE-AI is the introduction of a smart real-time AI chat interface, based on a smart query routing engine. This type of router is capable of answering users' requests on the fly, by interpreting those requests and forwarding them instantaneously to the right program database, thus returning reliable answers with a great degree of persuasiveness. For the business team, that means every chat is a chance to educate, excite and enroll another student -- adding revenue and brand to his or her institution.

In conclusion, the AI-friendly platform of REVA RACE places the university at a vanguard stage in respect to digital interaction. Leveraging intelligent automation, adaptive knowledge management and customer-centered workflow, RACE-AI not only gives business development colleagues but also the overall student body a new level of active AI powered career support.

# Chapter 2: Literature Review

**2.1 The Information Void in Choosing College Programs**

Visitors to university websites often find themselves in a state of confusion about their future, from finding the best course to ascertaining fees and comparing programs. Research also indicates that static web pages and human‐written FAQs seldom offer the follow up to the personalization or immediacy desired by students, in order for them to make a fully informed decision.

**2.2 AI Chatbots for immediate support**

New studies demonstrate AI chatbots offer interactive support, automatically pulling program comparisons, fee data and career highlights from the institution's databases. They excel relative to classical navigational devices by leading the users through identifying routines, answering complicated questions and customizing the answers according to a user’s particular curiosities.

**2.3 Retrieval-Augmented Generation for Document Querying**

Today's RAG-based educational bots can generate text based on several documents, providing its users with factually correct and contextually relevant answers about academic programs, admission requirement or financial mechanisms. Scraping Document and Combined RAG Scraping the document and combining with RAG has been shown to result in vastly improved user satisfaction and engagement as evident from literature.

**2.4 Automatic Question-Answer Pair Data Collection (AQAP)**

Automated Web scraping keeps the foundation for AI platforms up to date — collecting new programs, fees changes, scholarships and admissions criteria in near real time. By slicing and dicing scraped content by program, colleges are able to provide specific insights to ever website visitor.

**2.5 Research Gap**

The main gap I found indicated by recent reading is the absence of one platform that combines: instant AI chatbot support (24/7), continuous data collection per program, and context based query routing. RACE-AI directly tackles this by providing students with real time, fully precise, and tailored answers to every doubt point – so that they never lose a chance of engagement or conversion.

# Chapter 3: Problem Statement

Despite an expanding portfolio of academic programs and extensive promotional efforts by ***RACE, REVA University***, prospective students visiting the official portal frequently encounter significant challenges in finding the right course, understanding fee structures, comparing program strengths, and accessing up-to-date information. The current website model is fragmented, with details distributed across static pages and downloadable brochures, resulting in information overload, ambiguity, and missed opportunities for engagement.

Students face difficulty in:

1. Identifying which program aligns best with their background, goals, and career aspirations.
2. Gathering complete, clear details about fees, eligibility, intake cycles, and outcomes.
3. Receiving immediate answers to personalized or complex queries.

Business development and admissions staff are similarly hindered by the absence of a real-time, interactive platform for responding to queries, understanding prospect needs, and guiding users through decision-making.

These issues collectively lead to lost leads, low engagement, and slow conversions—creating a critical need for an intelligent, centralized solution that delivers instant, context-aware information and drives student satisfaction and recruitment outcomes. This highlighted the urgent need for a smart, one-stop conversational assistant that could:

1. Seamlessly integrate with brochures, FAQs, and program documents.
2. Enable semantic search and AI-powered query resolution in natural language.
3. Support real-time analytics for administrators and faculty.
4. Enhance collaboration and transparency between students, staff, and industry.

***RACE, REVA University*** *therefore required an* ***AI Assistant*** *capable of overcoming the limitations of static portals by delivering intelligent, real-time, and scalable academic support—fostering better engagement, operational efficiency, and stronger industry connections.*

# Chapter 4: Objectives of the Study

The background of the problem in relation to potential students and business development team of ***RACE, REVA University*** is what encouraged this research. Visitors to the academic portal have a hard time comparing programs, understanding fee structures or finding career data, and BD staff need better ways to engage and nurture high-potential prospects. In response to these pain points, we aim a following at in rectifying such shortcomings.

**1. Resolve Information Fragmentation**

Organizational: Build out and integrate pre-scraped website information and program brochures into partitioned databases, allowing users to find fully-decisive answers to every question about their journey for the highest-quality results.

**2. Enable Instant, Contextual Support**

Provide AI-powered, instant, on-demand support for mission-critical queries: such as program selection, admissions details, fee information, and career options with smart dispatching and conversational chatbot interface.

**3. Facilitate Effective Prospect Engagement**

Assist the business development team with ideas and suggestions of how to follow up, reach out, and engage students. The AI chat logs and interactions analytics offer precisely targeted communications, to improve chances of recruitment.

# Chapter 5: Project Methodology

This study outlines the methodology and reasoning applied to achieve the defined objectives of the ***RACE Assist – AI Web Chatbot***. The approach integrates conversational AI design, data handling, architectural considerations, and security measures—each decision aligned with delivering a sustainable, scalable, and user-centric solution.

#### **5.1 Business Understanding**

1. **Students & Prospects**: Require instant, reliable answers to program-related queries, brochure details, and admission FAQs without needing to browse multiple static pages.
2. **Faculty & Staff**: Need to minimize repetitive question handling and gain visibility into the types of queries students ask for better academic guidance and outreach.
3. **Administrators**: Require real-time analytics on user interactions, query trends, and engagement statistics to inform strategic planning and decision-making.

#### **5.2 Data Collection and Management**

1. **Primary Data Content**: Program brochures, FAQs, course details, and admission policies.
2. **Supplementary Data**: Uploaded documents (PDF brochures), query logs, and chat transcripts for training and evaluation.
3. **Data Storage**: PostgreSQL with pgvector/FAISS indexing for semantic retrieval and structured schema for metadata.
4. **Data Validation**: Schema enforcement and error handling during ingestion to ensure clean, reliable records.

#### **5.3 AI Integration and Conversational Search**

1. **Semantic Search**: FAISS/pgvector embedding to enable context-aware query responses, outperforming basic keyword search.
2. **PDF Analysis & Extraction**: Automated parsing of brochures with metadata extraction to provide precise answers from documents.
3. **Conversational Assistant**: Gemini-powered Chabot designed for natural interactions, capable of handling FAQs, brochure queries, and cross-document reasoning.

#### **5.4 Security, Analytics, and Administration**

1. **Authentication**: JWT tokens for secure access, role-based privileges, and Supabase Row-Level Security for granular data protection.
2. **Admin Dashboard**: Real-time analytics on queries, user engagement, and content access with charts, logs, and usage reports.
3. **Monitoring & Validation**: Continuous query logging, schema-level validation, and regular audits for reliability and compliance.

#### **5.5 Analytical Reasoning and Benchmarking**

1. **Performance Metrics**: Latency, response accuracy, and semantic search efficiency compared with baseline keyword methods.
2. **Stakeholder Feedback**: Iterative pilot testing with students, faculty, and staff to refine chatbot responses and interface usability.
3. **Technology Rationale**: Chosen stack (PostgreSQL, FAISS/pgvector, and Gemini AI) ensures scalability, conversational intelligence, and secure integration into institutional workflows.

This methodology combines conversational AI, structured data management, semantic retrieval, and strong security practices to deliver a scalable solution. Each step ensures alignment with institutional needs while improving accessibility, discoverability, and user experience across ***RACE, REVA University***.

# Chapter 6: Resource Requirement Specification

Chapter 7: Software Design

The ***RACE Assist*** system is designed to unify web data extraction, brochure intelligence, semantic vector search, and AI conversational support into a modular, scalable platform. It enables students and faculty to query program-specific information in real-time, ensuring accuracy and accessibility of academic data.

## 7.1 System Architecture

**Fig. 7.1 High-Level Architecture of the RACE-AI System**  
This figure illustrates a modular AI-driven academic information retrieval system. The process begins with user queries through a web interface, followed by document pre-processing and knowledge base construction. Semantic vector search powers retrieval and context is integrated with the generative AI engine to provide natural language answers.

### 1. User Interface Layer

The User Interface is the entry point for students, educators, or administrators. It provides an intuitive web application for interaction.

1. Chat interface built using Flask for real-time Q&A.
2. Category filtering for academic program-specific queries.
3. Responsive design for accessibility across devices.
4. Supports session-based query handling and interactive experience.

### 2. Document Processing & Knowledge Base Layer

This layer collects, pre-processes, and organizes academic program information into structured semantic databases.

1. **Web Data Scraping** – Simulated browser scraping using BeautifulSoup + Requests to fetch HTML data from official program pages.
2. **PDF Brochure Extraction** – Uses PyPDF2 to extract structured text from program brochures.
3. **Unified Data Pipeline** – Normalizes both HTML and PDF text, splits it into semantic chunks, and generates embeddings using Sentence Transformers.
4. **Program-Specific Vector Databases** – FAISS indices store embeddings and text chunks for each program, enabling modular and scalable search.
5. **Category Mapping** – JSON-based mapping links each program name to its dedicated FAISS index for efficient routing of queries.

### 3. Semantic Search & Retrieval Layer

This layer ensures that queries are resolved with accurate, context-aware results.

1. Converts user queries into embeddings (MiniLM).
2. Routes queries to the appropriate FAISS index based on program selection.
3. Retrieves top-matching context chunks for precise, category-wise results.
4. Ensures fallback scraping for missing or outdated context.

### 4. Generative AI & Response Layer

The intelligent reasoning core of the system that integrates retrieval with generative AI.

1. Sends user query + retrieved context to Google Gemini API.
2. Produces coherent, natural language answers with references to academic content.
3. Ensures consistency, accuracy, and personalization of responses.
4. Returns results seamlessly to the UI.

### 5. Backend Logic & API Layer

This layer connects all system components and manages the communication between UI, knowledge base, and AI.

1. Flask REST APIs handle JSON/HTML output, session management, and vector database calls.
2. Flask powers the interactive dashboard and live chat UI.
3. Ensures modularity for easy upgrades (new programs, updated brochures, or alternative AI models).

## 7.2 Low-Level Design

The low-level design details the execution workflow, showing how user queries move through different components to generate intelligent responses.

**Fig. 7.2 Flowchart**  
This figure outlines the end-to-end query resolution pipeline, starting from a student query to the final AI-generated answer.

### Step-by-Step Execution Flow

1. **User Query Submission**

* A student selects a program or submits a general query via the web interface.
* The query can be free-text or category-specific (e.g., MBA, MS, Certification).

1. **Program & Data Routing**

* System checks program selection.
* Routes query to the respective FAISS vector database.
* If required, triggers fresh scraping of program pages to update context.

1. **Data Processing & Context Retrieval**
   * Query embedding generated using MiniLM.
   * FAISS vector search retrieves the most relevant text chunks (from brochures or scraped pages).
   * Retrieved context structured into JSON format.
2. **AI Evaluation & Response Generation**
   * User query + retrieved context sent to Google Gemini API.
   * Gemini integrates context into a natural, conversational answer.
   * Feedback includes specific program details, course structures, or admission guidelines.
3. **Response Delivery**
   * The AI-generated response is displayed in the chat UI.
   * Option for session-based saving, PDF export of answers, or program-wise FAQ compilation.

Chapter 8: Implementation

As described in the previous chapter, the design specifications and modular architecture directly shaped the coding and integration of the **RACE-AI Support Platform**. This chapter documents the actual implementation process, detailing how individual modules were developed, integrated, and validated. The emphasis is on demonstrating functionality and alignment with the system’s academic, business, and technical goals rather than reproducing raw code.

## 8.1 Environment Setup

The RACE-AI system was implemented in **Python 3.10+**, leveraging its rich ecosystem for text processing, semantic search, and web application development. The environment was created using **Anaconda/virtualenv** for package isolation, with dependencies listed in a requirements.txt file to enable reproducibility.

1. **Frameworks & Tools:**
2. **Streamlit** – Interactive live chat interface for end-users.
3. **Flask** – REST API backend for programmatic access and integration.
4. **Data Processing Libraries:**
   1. **BeautifulSoup + Requests** – Scraping structured HTML content.
   2. **PyPDF2** – Extracting text from official brochures.
   3. **Sentence Transformers (all-MiniLM-L6-v2)** – Generating embeddings for semantic search.
   4. **FAISS** – Vector database for high-speed similarity search.
5. **Generative AI Integration:**
6. **Google Gemini API** – Provides context-aware conversational responses.
7. **Storage & Metadata:**
8. **Pickle/JSON** – For storing vector index metadata and brochure-to-program mappings.
9. The application was tested on **Windows 10/11** and is deployable to **Streamlit Cloud, Heroku, or any Flask-compatible server** for academic and business access.

## 8.2 Leveraging a Unified Knowledge Base

**Fig. 8.1: Knowledge Base Construction**

The foundation of the system lies in creating a structured and query-ready academic knowledge base.

1. **Code Reference:** process\_brochures.py
2. **Implementation Steps:**
   1. Scraping structured HTML program pages using BeautifulSoup.
   2. Extracting brochure text via PyPDF2.
   3. Applying text chunking and embedding generation.
   4. Creating metadata for each program and storing FAISS vector databases separately.
3. **Output:**
4. Dedicated FAISS vector DB for each program.
5. Metadata stored in JSON/Pickle for modular query routing.

* **Implementation Callout:**  
  (Insert console screenshot/code snippet of scraping, PDF extraction, and FAISS creation here).

## 8.3 Delivering Real-Time, AI-Powered Guidance

**Fig. 8.2: AI Conversational Pipeline**

This module connects the knowledge base to the conversational AI, enabling students to retrieve context-specific academic insights.

1. **Code Reference:** app.py (Streamlit), flask\_app.py (Flask API).
2. **Implementation Steps:**
   1. User queries routed to relevant program via brochure\_mapping.json.
   2. Query embedding generated using MiniLM.
   3. Semantic search performed within the program’s FAISS index.
   4. Retrieved text chunks assembled as context.
   5. Query + context forwarded to Google Gemini API.
   6. Gemini returns natural, program-specific answers.
3. **Output:**

A **live chat interface** providing instant, context-aware responses.

1. **Implementation Callout:**  
   (Insert screenshot/code snippet of query → FAISS search → Gemini response pipeline and Streamlit chat interface).

## 8.4 Supporting Business Development and Engagement

**Fig. 8.3: Engagement and Analytics Module**

Beyond academic use, the platform also supports business development by capturing and analysing interaction data.

1. **Code Reference:** Integrated within app.py and flask\_app.py.
2. **Implementation Steps:**
   1. Chat sessions logged for user interaction tracking.
   2. Optional features implemented for **prospect flagging** (e.g., frequent inquiries, intent detection).
   3. Session data analysed to identify high-interest programs or student concerns.
3. **Output:**
   1. Actionable insights on user behaviour.
   2. Support for faculty/BD teams in identifying prospects and improving follow-up.

**Implementation Callout:**  
(Insert screenshot/code snippet showing session logging, analytics summary, or prospect-flagging logic).

## 8.5 Evidence of Working Prototype

1. **Knowledge Base Pipeline:** Successful generation of multiple FAISS indices, each mapped to academic programs.
2. **Live Chat Application:** Real-time question answering tested across MBA, MS, and Certification programs.
3. **Analytics Logging:** Sample logs demonstrate interaction capture and potential for business insights.

The implementation of the RACE-AI system successfully demonstrates the integration of document intelligence, semantic vector search, and AI-driven guidance into a cohesive platform. It not only improves academic query resolution but also enhances engagement tracking for institutional growth. The modular design ensures scalability, enabling easy integration of new programs or AI models in the future.

Chapter 9: Testing and validation

Chapter 10: Analysis and Results

Chapter 11: Conclusions and Future Scope

# Bibliography

[1] G. S. Kumar, J. Cheriyan, N. Aparna, and J. Swathy, “Unleashing Facial Expression Recognition for Stress Detection Using Deep CNN Model.,” in *Procedia Computer Science*, Elsevier B.V., 2025, pp. 306–315. doi: 10.1016/j.procs.2025.03.332.

[2] H. P. Chandika, B. Soumya, B. N. E. Reddy, and B. M. S. SaiManideep, “Real-Time Stress Detection and Analysisusing Facial Emotion Recognition,” *IJARCCE*, vol. 13, no. 3, Mar. 2024, doi: 10.17148/ijarcce.2024.13324.

[3] M. A. H. Akhand, S. Roy, N. Siddique, M. A. S. Kamal, and T. Shimamura, “Facial emotion recognition using transfer learning in the deep CNN,” *Electronics (Switzerland)*, vol. 10, no. 9, May 2021, doi: 10.3390/electronics10091036.

[4] T. Kumar Arora *et al.*, “Optimal Facial Feature Based Emotional Recognition Using Deep Learning Algorithm,” *Comput Intell Neurosci*, vol. 2022, 2022, doi: 10.1155/2022/8379202.

[5] J. Zhang, H. Yin, J. Zhang, G. Yang, J. Qin, and L. He, “Real-time mental stress detection using multimodality expressions with a deep learning framework,” *Front Neurosci*, vol. 16, Aug. 2022, doi: 10.3389/fnins.2022.947168.

[6] H. P. Chandika, B. Soumya, B. N. E. Reddy, and B. M. S. SaiManideep, “Real-Time Stress Detection and Analysisusing Facial Emotion Recognition,” *IJARCCE*, vol. 13, no. 3, Mar. 2024, doi: 10.17148/ijarcce.2024.13324.

[7] K. Singh, S. K. Chawla, G. Singh, and P. Soni, “Stress Detection using Machine Learning Techniques: A review,” in *Proceedings - IEEE 2023 5th International Conference on Advances in Computing, Communication Control and Networking, ICAC3N 2023*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 255–260. doi: 10.1109/ICAC3N60023.2023.10541696.

[8] R. Li and Z. Liu, “Stress detection using deep neural networks,” *BMC Med Inform Decis Mak*, vol. 20, Dec. 2020, doi: 10.1186/s12911-020-01299-4.

[9] S. F. Ghahfarokhi, “A review of the effectiveness of stress management skills training on academic vitality and psychological well-being of college students,” 2015.

[10] J. A. Ballesteros, G. M. Ramírez V, F. Moreira, A. Solano, and C. A. Pelaez, “Facial emotion recognition through artificial intelligence,” *Front Comput Sci*, vol. 6, 2024, doi: 10.3389/fcomp.2024.1359471.

[11] T. Kopalidis, V. Solachidis, N. Vretos, and P. Daras, “Advances in Facial Expression Recognition: A Survey of Methods, Benchmarks, Models, and Datasets,” *Information (Switzerland)*, vol. 15, no. 3, Mar. 2024, doi: 10.3390/info15030135.

[12] Z. Y. Huang *et al.*, “A study on computer vision for facial emotion recognition,” *Sci Rep*, vol. 13, no. 1, Dec. 2023, doi: 10.1038/s41598-023-35446-4.

[13] A. Pandey, A. Gupta, and R. Shyam, “FACIAL EMOTION DETECTION AND RECOGNITION,” *International Journal of Engineering Applied Sciences and Technology*, vol. 7, no. 1, pp. 176–179, May 2022, doi: 10.33564/IJEAST.2022.v07i01.027.

[14] D. Bhagat, A. Vakil, R. K. Gupta, and A. Kumar, “Facial Emotion Recognition (FER) using Convolutional Neural Network (CNN),” in *Procedia Computer Science*, Elsevier B.V., 2024, pp. 2079–2089. doi: 10.1016/j.procs.2024.04.197.

[15] “FER-2013 dataset,” https://www.kaggle.com/datasets/msambare/fer2013.

# Appendix

## Plagiarism Report[[1]](#footnote-1)

**Plagiarism Report** with below 15% Similarly index to be attached in the annexure. The title page and last pages with the similarity index report are attached.

1. Turnitn report to be attached from the University. [↑](#footnote-ref-1)