# BiblioAI – An Intelligent Research Mapping and Bibliometric Analytics Platform

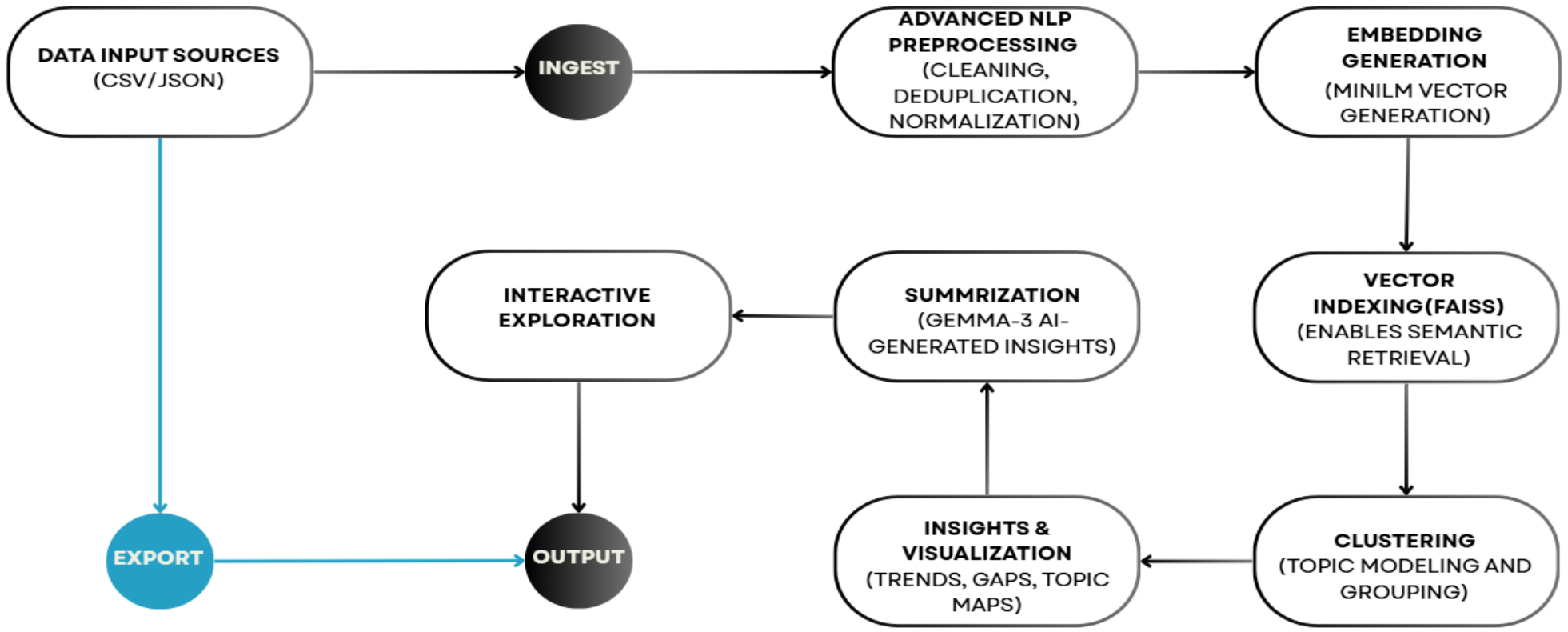
---

- COURSE: AIML403 MAJOR PROJECT
- TEAM MEMBERS:
  DEVANG PATEL (22AIML031)
  HETVI PATEL (22AIML034)

# Outline of the Presentation

- Project Structure and Flow
- Tools & Technologies Used
- Hardware & Software Specifications
- Industry Practices Adopted
- Coding Conventions
- Functional Requirements
- Non-Functional Requirements
- Motivation for Choosing Project
- Learning Outcomes Till Date
- Further Planning

# System Diagram



# Tools and Technology

**Python & Pandas**
Industry-standard for data manipulation and analysis with extensive ecosystem support

**MiniLM Embeddings**
Lightweight yet powerful semantic understanding for efficient research document mapping

**FAISS Vector Search**
Facebook's library enables ultra-fast similarity search for millions of documents

**Gemma-3 Summarization**
State-of-the-art LLM generates coherent, context-aware research summaries

**Streamlit UI**
Rapid development of interactive web interface with minimal frontend complexity

# Hardware and Software

## HARDWARE REQUIREMENTS

- Processor: Intel Core i5/i7 or AMD Ryzen 5/7 (minimum 4 cores)
- RAM: 16 GB minimum, 32 GB recommended for large datasets
- Storage: 256 GB SSD for fast data access
- GPU: (NVIDIA GTX/RTX series accelerates embedding generation)

## SOFTWARE STACK

- OS: Windows 10/11, Linux Ubuntu 20.04+, or macOS 12+
- Python: Version 3.9 or higher
- Libraries: NumPy, SciPy, Scikit-learn, Transformers
- IDE: VS Code, PyCharm, or Jupyter Notebook

# Industry Best Practices

**Modular Coding**
Separate modules for data ingestion, preprocessing, embedding generation, and visualization ensure maintainability and code reusability

**Version Control**
Git-based workflow with regular commits, descriptive messages, and feature branching for collaborative development and change tracking

**Comprehensive Documentation**
In-line comments, API documentation, and project README explaining architecture, setup instructions, and usage guidelines

**Testing & Traceability**
Unit tests for critical components, input validation checks, and requirement traceability matrix linking features to specifications

# Coding Standards Followed

### Naming Conventions

- snake_case for variables and functions
- PascalCase for classes
- UPPER_CASE for constants
- Descriptive, meaningful names

### Modular Structure

- Separate scripts for each component
- Clear function responsibilities
- Import management
- Configuration files for parameters

### Code Readability

- Consistent indentation (4 spaces)
- Meaningful comments
- Docstrings for public functions
- PEP 8 compliance

# Functional Requirements

- **Dataset Upload**
  Support CSV and JSON formats
- **Data Processing**
  Cleaning and validation pipeline
- **Semantic Retrieval**
  Context-aware search using vectors
- **Clustering**
  Automatic topic identification

- **AI Summaries**
  Concise research document insights
- **Trend Analysis**
  Research gap detection

# Non-functional Requirements

- **Usability**
  Intuitive interface requiring minimal training
- **Scalability**
  Handles 10,000+ documents efficiently
- **Reliability**
  Robust error handling and data validation
- **Performance**
  Fast response times for search and analysis

# Project Motivation

- Driven by the potential of Artificial Intelligence to transform academic research, we aimed to develop a solution that enhances insight extraction and knowledge discovery from vast scientific repositories

**AI-Driven Research Intelligence**

- This project provided an invaluable opportunity to apply theoretical knowledge, gain practical experience with cutting-edge AI technologies, and develop skills in data processing, machine learning, and system

**Hands-On Learning Experience**

- Beyond academic exercise, a core motivation was to build a tangible tool that could genuinely assist researchers, making complex information more accessible and

**Impactful Real-World Application**

# Learning outcome

**Semantic Embedding Generation**

Learned to generate dense vector embeddings from bibliometric datasets using **MiniLM** for semantic understanding of research papers.

**FAISS Vector Indexing & Retrieval**

Gained hands-on experience in building and optimizing **FAISS indexes** for efficient similarity-based research paper retrieval.

**Clustering for Topic Discovery**

Implemented embedding-space clustering to organize papers into thematic research groups for topic-wise exploration.

**AI-based Summarization**

Integrated **Gemma-3** to generate meaningful cluster-wise summaries while ensuring outputs remain interpretable and source-linked.

**Trend & Gap Analysis Foundations**

Started trend and gap analysis by examining publication patterns, keyword relationships, and identifying under-explored research areas.

# Future Planning

- **Enhance Core Functionalities**
  Refine trend detection, optimize for larger datasets, and improve data visualizations.
- **Implement Advanced Analytics**
  Develop citation network and co-authorship mapping for deeper research insights.
- **Explore AI-Powered Innovations**
  Integrate predictive forecasting and explore AI-driven hypothesis generation.

*THANK YOU*