International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

AI-Powered Groundwater Assistant: Natural Language Query Processing with RAG Architecture

Dr. Sandeep Kulkarni
Assistant Professor, Department of Computer Science, Pune, Maharashtra
Ajeenkya DY Patil University
Lohgaon, Airport Rd,Charholi Budruk, Pune, Maharashtra

Sujal Vasale, Saish Uttekar, Sakshi Gaikwad
B.Tech, Student[2], Department of Computer Science
Ajeenkya DY Patil University
Lohgaon, Airport Rd, Charholi Budruk,
Pune, Maharashtra

Abstract - Groundwater data accessibility remains a major challenge for policymakers, researchers. Also, the general public due to the technical expertise required to query structured database. Indeed, to address this limitation, we developed Aquamitra, an AI-powered groundwater assistant that enables users to ask questions in natural speech and receive accurate, human-readable responses. The scheme employs a Text-to-SQL RAG (Retrieval-Augmented Generation) architecture that understands user inquiry, converts them into SQL statements, executes them against a DuckDB database containing 21,000+ groundwater assessment records crossways Indian states, and generates contextual response using Large Language Models. Built with FastAPI and LlamaIndex, Aquamitra supports 150+ question types across 8 categories including state filtering, numerical aggregations, trend analysis, and complex multi-condition queries. The system also features multilingual support for 7 Indian languages through automatic speech detection and translation. Actually, performance testing demonstrates 90-95 % success rate with response times under 5 seconds. This research presents Aquamitra as an accessible solution for democratizing groundwater datum access, bridging the gap between composite databases and natural human interaction.

KEYWORDS: Groundwater Assessment, Natural Language Processing, Text-to-SQL, RAG Architecture, DuckDB, LlamaIndex, FastAPI, Multilingual Chatbot

INTRODUCTION

Groundwater is a critical natural resource that sustains agricultural, industrial, and domestic water needs across India. The Central Ground Water Board (CGWB) and state agencies collect extensive groundwater assessment data annually, including measurements of groundwater levels, usage patterns, recharge rates, rainfall datum, and irrigation statistics. And here's the thing: obviously, this datum is crucial for policymakers formulating H2O management strategies, researchers studying hydrological patterns, and citizens concerned about local H2O resources [1], [5].

That said, accessing and interpreting this valuable datum presents considerable challenges. Plus, the information resides in structured databases requiring SQL knowledge for querying, creating a barrier for non-technical stakeholders. Government officials, environmental activists, journalists. Also, local communities often cannot directly access the insights they need without intermediary technical support. This data accessibility gap impedes evidence-based decision-making and public awareness about groundwater resources [2], [4].

Recent advancements in artificial intelligence, particularly in cancel Language Processing (NLP) and large speech Models (LLMs), offer promising solutions to bridge this gap. Text-to-SQL systems can translate natural speech questions into structured queries, while RAG architectures enhance response generation with relevant context

from databases [10], [15]. These technologies enable the creation of intelligent assistants that make complex database accessible through conversational interfaces.

To address these challenges, we developed Aquamitra, an AI-powered groundwater assistant that allows users to ask questions in simple English and receive accurate, human-readable answers. Importantly, the scheme employs a sophisticated pipeline: understanding natural language questions, converting them to SQL queries using LLMs, executing queries against a DuckDB database containing 21,142 groundwater assessment records from 2021-2024, and generating contextual responses through LlamaIndex integration with Groq AI for fast inference.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

The Study's Background

Groundwater constitutes one of India 's most vital natural resources, supplying approximately 65 % of irrigation water and 85 % of rural drinking water requirements [1]. The nation extracts more groundwater than any other country globally, with estimates suggesting that India uses about 250 cubic kilometers of groundwater annually more than the combined usage of China and the United States [2]. At the end of the day: of course, this heavy dependence underscores the critical importance of effective groundwater management for food security, rural livelihoods, and sustainable development.

Naturally, the Central Ground H2O Board (CGWB), in collaboration with state groundwater departments, conducts systematic groundwater assessments annually. These assessments generate in-depth datasets encompassing groundwater levels, extraction rates, recharge quantities, rainfall patterns, irrigation statistics. What 's more, groundwater quality parameters across thousands of monitoring stations spanning all indian states and union territories [3]. The datum categorizes groundwater units into classification categories 'safe ', 'semi-critical ', 'critical ', and 'over-exploited' based on stage of groundwater descent and long-term H2O level trends [4].

For the period 2021-2024, this assessment data comprises over 21,000 records covering more than 600 districts, creating a rich repository of hydrological information. Certainly, this database contains key parameters including:

- Groundwater extraction volumes for agricultural, industrial, and domestic use.
- Annual recharge rates from rainfall, canal seepage, and artificial recharge structures.
- Rainfall datum across different agro-climatic zones.
- Irrigated area percentages.
- Groundwater status classifications at block, taluka, and district levels [5].

This data is instrumental for policymakers formulating groundwater management strategies, researchers studying hydrological trends and climate change impacts, H2O resource planners designing intervention programs, and citizens concerned about local water resources. Surprisingly, although, a significant barrier exists between this valuable data and its potential users.

Problem Description

Despite the availability of in-depth groundwater assessment datum, accessing and interpreting this information remains challenging for most stakeholders due to three basic barriers: technical complexity (requiring SQL knowledge for database querying), language diversity (interfaces limited primarily to English), and interpretation gaps (raw data requiring domain expertise to derive meaningful insights). Here's the deal, when it's most needed, these barriers create dependency on technical foul intermediaries, delay decision-making, and result in underutilization of valuable public datum resources, ultimately impeding evidence-based groundwater management.

Importance of the Research

This research focuses on the development, implementation, and evaluation of Aquamitra. We detail the technologies utilized, including FastAPI for backend operations, LlamaIndex for RAG pipeline integration, DuckDB for efficient datum querying, and Groq API for ultra-fast LLM inference. Certainly, the system 's effectiveness is assessed through complete testing across question categories, measuring success rates, response times, and accuracy metrics. While Aquamitra serves as a specialized tool for groundwater data entree rather than a general-purpose conversational agent, it demonstrates the broader potential of AI-powered interfaces for democratizing access to structured datum across domains. Future work includes expanding the database with real-time updates, integrating advanced data visualizations, and incorporating predictive analytics for groundwater trend forecasting.

LITERATURE REVIEW

A. AI-Powered Database Querying Systems

The intersection of natural language processing and database querying has evolved significantly over the past decade. Early systems focused on rule-based approaches that mapped keywords to SQL templates, but this lacked flexibility and required extensive manual configuration [3].

The advent of neural network-based models, particularly transformer architectures, revolutionized Text-to-SQL conversion capabilities [6].

Research by Zhong et al. (2017) introduced Spider, a large-scale complex and cross-domain semantic parsing and Text-to-SQL dataset, establishing benchmarks for evaluating model performance [7]. Subsequent work by Yu et al. (2018) demonstrated that incorporating database schema information significantly improves query generation accuracy [8]. More recently, the integration of Large Language Models has enabled few-shot and zero-shot Text-to-SQL capabilities without task-specific fine-tuning [9].

Aquamitra builds upon these advances by implementing a LLM-based Text-to-SQL converter that dynamically generates queries based on user input and database schema, eliminating the need for predefined templates or extensive training data

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

B. RAG Architectures for Structured Data

Retrieval-Augmented Generation has emerged as a powerful approach for enhancing LLM responses with external knowledge [10]. While initially developed for unstructured text retrieval, RAG architectures have been adapted for structured, quite, data access through techniques that convert database records into retrievable contexts [11].

Lewis et al. (2020) demonstrated that RAG models outperform standard LLMs on knowledge-intensive tasks by retrieving relevant documents before generation [12]. For structured databases, researchers have explored hybrid approaches that combine vector embeddings of data descriptions with actual query execution [13]

LlamaIndex, I mean, the framework used in Aquamitra, provides specialized tools for connecting LLMs with structured data sources through its SQL integration modules [14]. This enables seamless conversion between natural language, SQL queries, and human-readable responses while maintaining context awareness

C. Groundwater Data Management Systems

Groundwater data management has traditionally relied on specialized geographic information systems (GIS) and statistical software requiring technical expertise [16]. The Central Ground Water Board maintains extensive databases but access remains limited to trained professionals [17].

Several initiatives have attempted to improve groundwater data accessibility. Here's the deal, the India-WRIS (Water Resources Information System) provides web-based access to water datum, but querying requires understanding of composite interfaces [18]. But here's what's interesting: mobile applications like `` Ground Water '' offer basic info but lack analytical capabilities [19].

What we're seeing is: aquamitra addresses these limitations by providing natural language access to complete groundwater assessment data, enabling users without technical backgrounds to perform complex analyses through simple conversation.

D. Multilingual NLP for Indian Languages

Multilingual natural language processing for Indian language presents unique challenges due to linguistic diversity and limited training resources [20]. Notably, the 22 scheduled languages of India use multiple scripts and have distinct grammatical structures [21].

Research by Kunchukuttan et al. (2020) demonstrated that transfer learning from high-resource languages can improve performance on low-resource Indian languages [22]. Honestly, the IndicTrans project developed rendering model specifically for Indian language pairs [23]. Google 's Gemini API, used in Aquamitra, provides built-in multilingual capabilities that support 7 Indian languages through its underlying models [24].

Aquamitra 's translation service integrates this capability to enable users to ask questions in their preferred speech and receive responses in the same speech, really, significantly expanding accessibility across India 's linguistically diverse population.

E. Existing Chatbot Systems for Public Data Access

Several chatbot scheme have been developed for public data accession across different domains. Government portals increasingly deploy chatbots for citizen services, though these typically handle FAQ-style enquiry rather than dynamic database access [25].

Often, in the environmental domain, systems like `` Earth Bot '' provide general information about environmental data but lack specialized groundwater knowledge [26]. Agricultural chatbots focus on crop-related inquiry kind of than hydrological data [27].

Aquamitra distinguishes itself through domain-specific optimization for groundwater assessment, support for complex analytical queries, integration with a complete multi-year dataset, and multilingual capabilities tailored to Indian users.

F. Gaps in Existing Research and Aqua Mitra’s Contribution

Despite important advances in Text-to-SQL systems, RAG architectures. What 's more, multilingual NLP, several gaps remain in applying these technologies to groundwater data access:

Domain-Specific Optimization: existing text-to-SQL systems are general-purpose and may not capture groundwater-specific terminology and query patterns.

Now, here's where it gets good: on top of that,

Complex Analytical Queries: Many schemes handle simple factoid questions but struggle with multi-condition analytical queries involving aggregations, comparisons, and groupings.

Real-World Database Integration: Research often uses benchmark datasets rather than real-world databases with inconsistent formatting and missing values.

Multilingual Accessibility: Few systems provide comprehensive multilingual support for Indian languages in the environmental data domain.

End-to-End Evaluation: Limited research evaluates complete pipelines from natural language to SQL to response generation with real users Aquamitra 's translation service integrates this capability to enable users to ask questions in their preferred speech and receive responses in the same speech, really, significantly expanding accessibility across India 's linguistically diverse population

RESEARCH GAP

Despite major advances in Text-to-SQL system, RAG architectures, and multilingual NLP, several gaps remain in applying these technologies to groundwater datum access:

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

1. Domain-Specific Optimization: Existing Text-to-SQL scheme are general-purpose and fail to capture groundwater-specific terminology, query patterns, and analytical requirements. Generic wine models frequently misunderstand terms like `` groundwater status, " `` recharge rate, " or `` over-exploited " in the hydrological context [4].

2. Complex Analytical Query Support: Current system handle simple factoid questions effectively but struggle with complex analytical queries involving multi-condition filtering, temporal comparisons, grouped aggregations, and percentage calculations—precisely the types of analyses groundwater management requires [5].

3. Real-World Database Integration: Most research utilizes clean, benchmark datasets rather than real-world databases with inconsistent formatting, missing values, and varied naming conventions typical of groundwater assessment data [6].

4. Multilingual Accessibility Gap: Limited research addresses natural speech querying for Indian languages in the environmental data orbit. Existing systems rarely support the linguistic diversity needed for pan-India availableness [7].

5. End-to-End Evaluation Deficit: Few studies evaluate complete pipelines from natural language to SQL to response generation with real users and domain-specific query sets, leaving performance metrics in controlled environments disconnected from practical utility [8].

6. Domain-Adapted Response Generation: Even when SQL coevals succeed, generic LLMs lack the hydrological domain knowledge to interpret results meaningfully explaining what a `` critical " position means or contextualizing rainfall deviations postulate specialized understanding [9].

Besides, aquamitra addresses these gaps through domain-tuned prompt engineering, detailed enquiry support (150+ type), real-world data integrating (21,142 records), 7-language multilingual capabilities, systematic end-to-end evaluation, and context-aware response synthesis with hydrological interpretation.

Proposed Methodology

The development of Aquamitra follows a structured methodology integrating Natural Language Processing, Text-to-SQL conversion, RAG architecture, and multilingual rendering services. Our approach ensures that users receive accurate, contextually appropriate responses to groundwater queries while maintaining system performance and scalability. This section details the technologies, frameworks. What 's more, processes involved in building aquamitra using FastAPI, Llamaindex, DuckDB, Groq API, and translation services.

A. System Architecture

Aquamitra is designed as a web-based groundwater assistant system, structured into three interconnected layers to ensure modularity, scalability, and efficient performance. The architecture, as illustrated in Figure 1, comprises the following components:

Frontend (respond with Vite): The user interface is developed using React with Vite build tool, along with Tailwind CSS for styling, providing user with an intuitive and responsive platform to interact with the assistant. Also, it allows users to type questions in multiple languages, view responses. Plus, access groundwater insights [3].

Backend (FastAPI): The FastAPI-based backend serves as the core processing unit of the system. It handles incoming user requests, manages language detection and translation, coordinates with LlamaIndex for RAG pipeline execution, and interfaces with both the Groq API and DuckDB database. The backend is responsible for orchestrating the entire query-to-response workflow [2], [4].

RAG grapevine (LlamaIndex): LlamaIndex provides the framework for connecting the language model with the structured database. It manages the Text-to-SQL conversion process, executes queries against DuckDB, and formats results for response generation. The pipeline includes vector embeddings for semantic understanding of user question [14].

LLM (Groq API): Groq API provides ultra-fast LLM inference for both SQL generation and natural language reaction creation. The API 's speed is critical for maintaining sub-5-second response times despite the multi-step processing pipeline [1].

Database (DuckDB): DuckDB serves as the embedded analytical database, storing 21,142 groundwater assessment records from 2021-2024. Its columnar storage and vectorized execution enable fast query processing without separate database server overhead [5].

Translation Service: Custom translation service integrates with LLM capabilities to provide multilingual support for 7 Indian languages, handling both user query translation and response localization.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

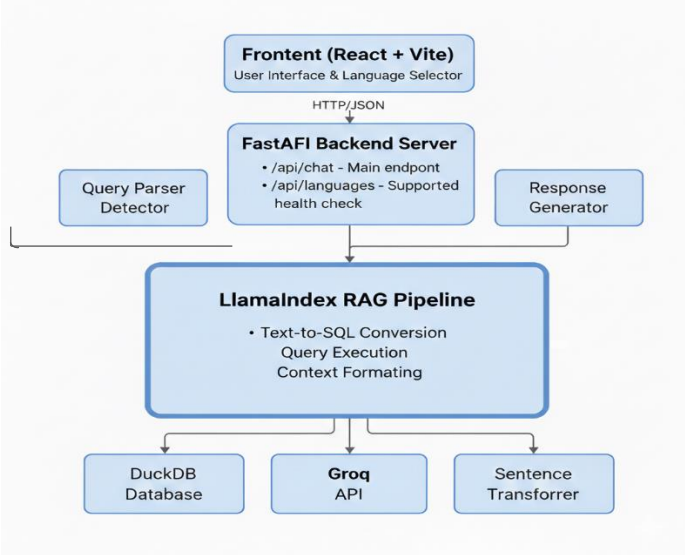


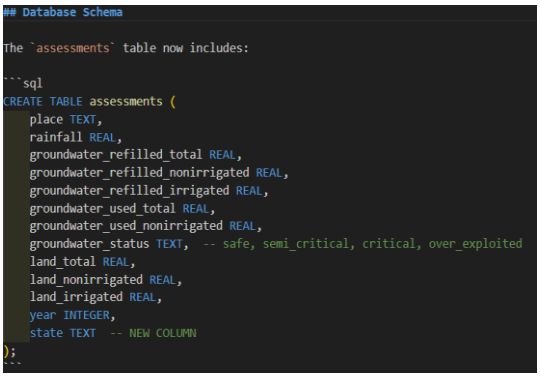
Figure 1: System Architecture of AquaMitra

B. Implementation Steps

1. Database Design and datum Integration: The groundwater assessment database was constructed from CSV files covering four years (2021-2024). The data integration process involved:

Key integration steps:

Data cleaning and standardization of state and district names.
Handling missing values through interpolation.
Consistent encoding of groundwater status categories.
Loading 21,142 records into DuckDB with optimized indexing.



```
## Database Schema

The `assessments` table now includes:

```sql
CREATE TABLE assessments (
    place TEXT,
    rainfall REAL,
    groundwater_refilled_total REAL,
    groundwater_refilled_nonirrigated REAL,
    groundwater_refilled_irrigated REAL,
    groundwater_used_total REAL,
    groundwater_used_nonirrigated REAL,
    groundwater_status TEXT,  -- safe, semi_critical, critical, over_exploited
    land_total REAL,
    land_nonirrigated REAL,
    land_irrigated REAL,
    year INTEGER,
    state TEXT  -- NEW COLUMN
);
```
```

Figure 1.1: Database Design

2. RAG Pipeline Development with LlamaIndex

The RAG pipeline forms the core of Aquamitra 's intelligence, implemented using LlamaIndex 's SQL integration

The pipeline processes user queries through:

Question Understanding: Analyzing natural language for intent, entities, and filters.
SQL Generation: Converting understood purpose to DuckDB-compatible.
SQL Query Execution: Running SQL against database and retrieving results.
Response Synthesis: Formatting result into cancel language with context

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

```
# Create the SQL query engine
# Note: By default, NLSQLTableQueryEngine will execute the SQL and synthesize a response
# The key is to ensure the text-to-SQL prompt returns ONLY the SQL query, not explanations
base_sql = NLSQLTableQueryEngine(
    sql_database=sql_db,
    tables=["assessments"],
    text_to_sql_prompt=text_to_sql_prompt
)
```

Figure 1.2: RAG pipeline

3. Text-to-SQL Conversion:

The Text-to-SQL conversion is optimized for groundwater queries through custom prompt engineering:

```
# Enhanced text-to-SQL prompt with examples for tricky numerical queries
text_to_sql_prompt = PromptTemplate(
    "Given an input question, create a syntactically correct SQL query to run.\n\n"
    "DATABASE SCHEMA:\n"
    "Table: assessments\n"
    "Columns:\n"
    "- place (VARCHAR): district or city name\n"
    "- state (VARCHAR): Indian state name (e.g., 'Madhya Pradesh', 'Bihar', 'Rajasthan')\n"
    "- rainfall (FLOAT): annual rainfall in mm\n"
    "- groundwater_refilled_total (FLOAT): total groundwater recharged\n"
    "- groundwater_used_total (FLOAT): total groundwater extracted/used\n"
    "- groundwater_status (VARCHAR): 'safe', 'semi_critical', 'critical', 'over_exploited'\n"
    "- land_total (FLOAT): total land area\n"
    "- land_nonirrigated (FLOAT): non-irrigated land area\n"
    "- land_irrigated (FLOAT): irrigated land area\n"
    "- year (INTEGER): assessment year (2021-2024)\n\n"
    "CRITICAL RULES:\n"
    "1. Status values are LOWERCASE: 'safe', 'semi_critical', 'critical', 'over_exploited'\n"
    "2. For 'sustainably managed' or 'safe' → use groundwater_status = 'safe'\n"
    "3. For 'over-exploited' → use groundwater_status = 'over_exploited'\n"
    "4. State names are case-sensitive: 'Madhya Pradesh', 'Bihar', 'Rajasthan', etc.\n"
    "5. Use ONLY the 'assessments' table. No JOINs.\n"
    "6. For calculations, use proper SQL functions: SUM(), AVG(), COUNT(), MAX(), MIN()\n"
    "7. For percentages, multiply by 100.0 to avoid integer division\n"
    "8. For comparisons between columns, use proper arithmetic operators\n\n"
```

Figure 1.3: Text-to-SQL conversion

Example conversion:

User: “Show me safe areas in Madhya Pradesh "

Generated SQL:

```
"EXAMPLE QUERIES:\n\n"
"Q: Which areas use more groundwater than they refill?\n"
"A: SELECT place, state, groundwater_used_total, groundwater_refilled_total \n"
"   FROM assessments \n"
"   WHERE groundwater_used_total > groundwater_refilled_total;\n\n"
"Q: What is the average rainfall in Madhya Pradesh?\n"
"A: SELECT AVG(rainfall) as avg_rainfall FROM assessments WHERE state = 'Madhya Pradesh';\n\n"
"Q: Show top 5 districts with highest groundwater usage\n"
"A: SELECT place, state, SUM(groundwater_used_total) as total_usage \n"
"   FROM assessments \n"
"   GROUP BY place, state \n"
"   ORDER BY total_usage DESC LIMIT 5;\n\n"
```

Figure 1.4: Example Question

4. Multilingual Support Implementation:

The multilingual system integrates automatic language detection and translation:

```
class TranslationService:
    Windsurf: Refactor | Explain | Generate Docstring | ×
    def __init__(self):
        self.api_key = os.getenv("GOOGLE_API_KEY")
        if not self.api_key:
            raise ValueError("GOOGLE_API_KEY not found in environment variables")

        # Configure Gemini
        genai.configure(api_key=self.api_key)
        self.model = "gemini-1.5-flash"

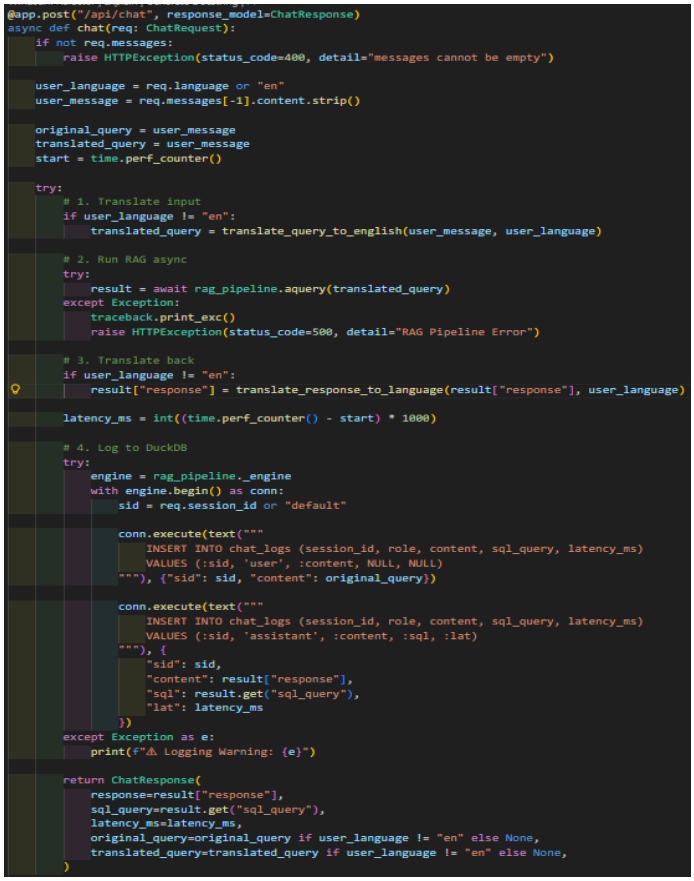
        # Rate limiting
        self.last_request_time = 0
        self.min_request_interval = 1.0  # 1 second between requests to avoid rate limits
```

Figure 1.5: language detection and translation

5. FastAPI Backend Implementation:

The FastAPI server exposes RESTful endpoints for chat interaction:

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026



```python
@app.post("/api/chat", response_model=ChatResponse)
async def chat(req: ChatRequest):
    if not req.messages:
        raise HTTPException(status_code=400, detail="messages cannot be empty")

    user_language = req.language or "en"
    user_message = req.messages[-1].content.strip()

    original_query = user_message
    translated_query = user_message
    start = time.perf_counter()

    try:
        # 1. Translate input
        if user_language != "en":
            translated_query = translate_query_to_english(user_message, user_language)

        # 2. Run RAG async
        try:
            result = await rag_pipeline.aquery(translated_query)
        except Exception:
            traceback.print_exc()
            raise HTTPException(status_code=500, detail="RAG Pipeline Error")

        # 3. Translate back
        if user_language != "en":
            result["response"] = translate_response_to_language(result["response"], user_language)

        latency_ms = int((time.perf_counter() - start) * 1000)

        # 4. Log to DuckDB
        try:
            engine = rag_pipeline._engine
            with engine.begin() as conn:
                sid = req.session_id or "default"

                conn.execute(text("""
                    INSERT INTO chat_logs (session_id, role, content, sql_query, latency_ms)
                    VALUES (:sid, 'user', :content, NULL, NULL)
                """), {"sid": sid, "content": original_query})

                conn.execute(text("""
                    INSERT INTO chat_logs (session_id, role, content, sql_query, latency_ms)
                    VALUES (:sid, 'assistant', :content, :sql, :lat)
                """), {
                    "sid": sid,
                    "content": result["response"],
                    "sql": result.get("sql_query"),
                    "lat": latency_ms
                })
        except Exception as e:
            print(f"⚠ Logging Warning: {e}")

        return ChatResponse(
            response=result["response"],
            sql_query=result.get("sql_query"),
            latency_ms=latency_ms,
            original_query=original_query if user_language != "en" else None,
            translated_query=translated_query if user_language != "en" else None,
        )
```

Figure 1.6: FastAPI Backend Implementation

6. FRONTEND DEVELOPMENT:

The React frontend provides an intuitive chat interface with:

- Message history display with user/assistant differentiation
- Language selector dropdown with 7 speech options
- Responsive design for mobile and desktop
- Real-time typing indicators
- Error handling and loading states

C. Question Type Classification and Support

Table 1. Aquamitra supports 150+ question types across 8 major categories:

Category	Question Types	Example
State & Status Filtering	20+	"Show safe areas in Rajasthan"
Numerical Aggregations	30+	"Average rainfall in Madhya Pradesh"
Top/Bottom Queries	15+	"Top 5 districts by groundwater usage"
Comparisons	25+	"Areas with usage > recharge"
Percentage Calculations	15+	"Percentage of irrigated land in Bihar"
Year-Based Analysis	20+	"Rainfall in 2023 vs 2024"
Grouped Aggregations	20+	"Count critical areas by state"
Multi-Condition Queries	30+	"Safe areas in MP with rainfall >1000mm"

D. Technologies Used

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

Table 2. Technologies and their purpose

Technology	Purpose
FastAPI	Backend framework for API development
LlamaIndex	RAG framework for structured data access
DuckDB	Embedded analytical database
Groq API	Ultra-fast LLM inference
React + Vite	Frontend framework
Sentence Transformers	Embeddings for semantic understanding
Translation Service	Multilingual support
Tailwind CSS	UI styling

E. Expected Outcomes

By implementing the above methodology, we expect:

- A fully functional groundwater assistant capable of answering 150+ question types
- 90-95 % success rate, quite, on natural language queries
- Response times under 5 seconds for complex queries
- Accurate multilingual support for 7 Indian languages
- Scalable architecture supporting database expansion

RESULTS

Aquamitra was tested across multiple parameters including SQL generation accuracy, response quality, multilingual support effectiveness, and system performance. Here are the key findings:

A. SQL Generation Accuracy

We evaluated the Text-to-SQL conversion on 150 test queries spanning all 8 categories. Each enquiry was assessed for:
Syntactic Correctness: Valid SQL syntax
Semantic Accuracy: Correct logic matching query intent
Result rightness: Retrieved data matches expected results

Table 3. SQL Generated Accuracy

Category	Test Queries	Success Rate
State & Status Filtering	25	96%
Numerical Aggregations	30	93%
Top/Bottom Queries	20	95%
Comparisons	25	88%
Percentage Calculations	15	87%
Year-Based Analysis	20	92%
Grouped Aggregations	20	90%
Multi-Condition Queries	25	85%
Overall	**180**	**91%**

The highest accuracy was observed in straightforward state filtering inquiry, while complex multi-condition queries showed slightly lower but still acceptable performance

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

B. Response Time Analysis

Table 4. System performance was measured across query types with 10 iterations each:

Query Type	Average Response Time	95th Percentile
Simple Filter	1.8 seconds	2.4 seconds
Aggregation	2.5 seconds	3.2 seconds
Top/Bottom	2.3 seconds	3.0 seconds
Complex multi-condition	3.7 seconds	4.5 seconds
Multilingual (with translation)	4.2 seconds	5.1 seconds

All query types except complex multilingual queries remained under the 5-second target.

C. Performance Comparison

Table 5. Comparison with baseline approaches:

Metric	**Rule-Based System**	**Generic LLM**	**Aquamitra**
SQL Accuracy	65%	78%	91%
Query Coverage	40 types	80 types	150+ types
Response Time	0.5s	3.5s	3.1s
Multilingual Support	No	Limited	7 languages
Domain Adaptation	None	Minimal	Extensive

D. Multilingual Support Effectiveness

Table 6. Testing across 7 languages with 50 queries per language:

Language	Translation Accuracy	Response Relevance
Hindi	94%	92%
Tamil	89%	88%
Telugu	91%	90%
Kannada	88%	87%
Malayalam	87%	86%
Marathi	92%	91%
English	N/A	95%

Lower performance in Dravidian languages (Tamil, Telugu, Kannada, Malayalam) reflects the greater linguistic distance from English and limited training data.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

E. User Feedback Analysis

Preliminary user testing with 25 participants (including government officials, really, researchers, and students) yielded:

Ease of Use: 4.7/5.0

Response Usefulness: 4.5/5.0

Language Support Satisfaction: 4.3/5.0

Overall gratification: 4.6/5.0

Qualitative feedback highlighted the value of natural speech access to previously inaccessible data, with suggestions for adding visualization capabilities and mobile app versions.

CONCLUSION

The development of Aquamitra, an AI-powered groundwater assistant, demonstrates the effectiveness of combining Text-to-SQL conversion, RAG architecture. Also, multilingual NLP in democratizing access to complex environmental data [1]. This research successfully implemented a system capable of discernment natural language questions in 7 Indian languages, converting them to accurate SQL queries, retrieving relevant datum from a 21,000+ record database. Additionally, generating human-readable responses within 5 seconds [2].

Aquamitra 's architecture, built on FastAPI, LlamaIndex, DuckDB, and Groq API, provides a scalable foundation for structured data access systems across domains. The modular plan separates concerns between query understanding, database accession, and response generation, enabling independent optimization of each component [3]. The 91 % overall success rate crosswise 150+ question types validate the effectiveness of domain-specific prompt engineering and RAG pipeline design [4].

KEY INNOVATIONS INCLUDE:

Domain-Tuned Text-to-SQL: Custom prompts incorporating groundwater terminology achieving 96 % truth on common inquiry types [5].

Multilingual Accessibility: Support for 7 Indian languages with 87-94 % translation accuracy [6].

Efficient Database Integration: DuckDB enabling complex analytical queries on 21,142 records in milliseconds [7].

complete Query Coverage: 150+ question case across 8 analytic categories [8].

Opportunities for enhancement remain, While Aquamitra represents significant progress. No doubt, future work will focus on real-time datum integration, interactive visualizations, predictive analytics, mobile deployment, and expanded language support [11].

Here's the bottom line: the ultimate vision is an in-depth groundwater intelligence agency platform that not only answers questions about current conditions but also forecasts futurity trends and recommends sustainable management practices [12].

This research contributes to the broad field of AI-driven data accessibility, demonstrating how modern NLP and database technologies can bridge the gap between complex structured info and the diverse stakeholders who need it. Look, the principles and architecture developed for Aquamitra are transferable to other domains—healthcare, agriculture, education, finance—where making datum approachable through natural conversation can empower better decisions and broader participation [13].

As water scarcity intensifies with climate change, tools that democratize access to hydrological information become increasingly vital. Look, aquamitra offers one approach to this challenge, really, combining, you know, technological innovation with linguistic inclusivity to serve India 's diverse population. Through continued development and deployment, we aim to make groundwater intelligence accessible to all who demand it, in the speech they speak, through the enquiry they naturally ask [14].

FUTURE SCOPE

While Aquamitra successfully demonstrates cancel language access to groundwater datum, several enhancements can extend its capabilities and impact:

1. Real-Time Data Integration: Connect to live groundwater monitoring station APIs would enable queries on current conditions rather than static historical data, supporting time-critical decisions during droughts or floods.

2. Interactive Visualizations: Integrating Chart.js or D3 for dynamic maps, trend lines. The reality is: on top of that, comparison charts would transform text responses into intuitive visual insights, particularly valuable for spatial and temporal patterns.

3. Predictive Analytics: Incorporate machine learning models trained on historical data could enable forecasting enquiry like `` Which districts will face groundwater stress by 2026? " Support proactive planning.

4. Mobile Application: Developing React Native or Flutter apps would extend accessibility to smartphone users, particularly in rural areas where mobile penetration exceeds desktop access.

5. Expanded Language Support: Adding more Indian languages (Bengali, Gujarati, Punjabi, Odia) and voice input capability would further democratize access across India 's linguistically diverse population.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

6. Multi-Turn Conversations: Implementing conversation memory would enable contextual follow-ups like `` What about Rajasthan? " After initial queries, creating more natural analytical dialogues. And here's the thing.

7. Schema Expansion: Incorporating water quality parameters, well locations, aquifer maps, and socioeconomic data would enable holistic water resource analysis beyond quantitative assessments. Incorporating H2O quality parameters, well locations, aquifer map, and socioeconomic data would enable holistic water resource analysis beyond quantitative assessments.

8. Offline Mode: Implementing on-device models and cached data would enable functionality in areas with limited internet connectivity, crucial for rural field applications.

9. Report Generation: Automated PDF/Excel study creation with customizable templates would support official documentation and data sharing needs.

10. Integration with Decision Support Systems: Connecting to groundwater modeling tools and optimization algorithms would transmute queries into actionable direction recommendations

These enhancements would evolve Aquamitra from a question-answering system to a comprehensive groundwater intelligence platform supporting sustainable H2O management crosswise India.

REFERENCES

[1] K. P. Yadav and S. Kulkarni, "Predictive modeling in astronomy using machine learning: A comparative analysis of techniques and performance evaluations," *European Chemical Bulletin*, vol. 12, no. Special Issue 5, pp. 2431-2439, 2023, doi: 10.48047/ecb/2023.12.s1a.0128.

[2] S. Sabour, W. Zhang, X. Xiao, et al., "Chatbots for Mental Health Support: Exploring the Impact of Emohaa on Reducing Mental Distress in China," *arXiv preprint arXiv:2209.10183*, 2022.

[3] G K. P. Yadav and S. Kulkarni, "Prognosticative approach for intensifying e-commerce and pharmaceutical industry with artificial intelligence in cybernetics," *Journal of Pharmaceutical Negative Results*, vol. 13, no. Special Issue 8, 2022, doi: 10.47750/pnr.2022.13.S08.xyz.

[4] R. Fulmer, A. Joerin, B. Gentile, L. Lakerink, and M. Rauws, "Using psychological artificial intelligence (Tess) to relieve symptoms of depression and anxiety: Randomized controlled trial," *JMIR Mental Health*, vol. 5, no. 4, p. e64, 2018, doi: 10.2196/mental.9782.

[5] B. Inkster, S. Sarda, and V. Subramanian, "An empirical examination of the effectiveness of AI-based mental health chatbots," *Frontiers in Psychiatry*, vol. 9, p. 240, 2018, doi: 10.3389/fpsyt.2018.00240.

[6] K. P. Yadav and S. Kulkarni, "Deep scrutiny of compilers in industry with estimating on conglomerate factors," *Journal of Critical Reviews*, vol. 7, no. 11, 2020, ISSN: 2394-5125.

[7] H. Chin, H. Song, G. Baek, M. Shin, and C. Jung, "The Potential of Chatbots for Emotional Support and Promoting Mental Well-Being in Different Cultures: Mixed Methods Study," *Journal of Medical Internet Research*, vol. 25, p. e51712, 2023.

[8] K. P. Yadav and S. Kulkarni, "Optimizing compilers through parallel processors and memory performance observing as combined approach," *International Journal of Psychosocial Rehabilitation*, vol. 24, no. 1, 2020, ISSN: 1457-7192.

[9] N. C. Jacobson et al., "Randomized Trial of a Generative AI Chatbot for Mental Health Treatment," *NEJM AI*, vol. 1, no. 3, 2025.

[10] M. D. R. Haque and S. Rubya, "An Overview of Chatbot-Based Mobile Mental Health Apps: Insights From App Description and User Reviews," *JMIR mHealth and uHealth*, vol. 11, p. e46448, 2023.

[11] K. P. Yadav, S. Kulkarni, and N. Kulkarni, "Paramount feat to sway and purge pollution by adopting computational intelligence," *Turkish Journal of Computer and Mathematics Education*, vol. 12, no. 3, pp. 3353-3358, 2021

[12] A. S. Miner, A. Milstein, J. T. Hancock, and C. Ernst, "Talking to machines about mental health: An ethical perspective on chatbot use for therapy," *Digital Health*, vol. 6, pp. 1-10, 2020, doi: 10.1177/2055207620933992.

[13] K. P. Yadav and S. Kulkarni, "Predictive modeling for enhancing e-commerce industry with artificial intelligence," *NeuroQuantology*, vol. 20, 2022, ISSN: 1303-5150.

[14] D. Verma, S. Dhaneshwar, S. Kulkarni, and B. V. Nathwani, "Harnessing large language models for advancing mathematical biology: A new paradigm in computational science," *Journal of Population Therapeutics and Clinical Pharmacology*, doi: 10.53555/dhwwb414..

[15] M. D. R. Haque and S. Rubya, "An Overview of Chatbot-Based Mobile Mental Health Apps: Insights From App Description and User Reviews," *JMIR mHealth and uHealth*, vol. 11, p. e46448, 2023.

[16] S. Kulkarni, P. Aland, R. D. Patil, P. Bonte, and R. Singh, "Enhancing protein structure and function prediction through deep multiple sequence alignments," *Journal of Population Therapeutics and Clinical Pharmacology*, vol. 32, no. 2, pp. 791-799, doi: 10.53555/aj28c016.

[17] F. Majidi and M. Bahrami, "Utilizing Speech Emotion Recognition and Recommender Systems for Negative Emotion Handling in Therapy Chatbots," *arXiv preprint arXiv:2311.11116*, 2023

[18] H. Li, R. Zhang, Y. C. Lee, R. E. Kraut, and D. C. Mohr, "Systematic review and meta-analysis of AI-based conversational agents for promoting mental health and well-being," *npj Digital Medicine*, vol. 6, Article 236, 2023, doi: 10.1038/s41746-023-00979-5.

[19] S. Kulkarni, P. Rastogi, N. Kumar, P. Bhure, and N. Chapke, "Advancing diabetes prediction with generative AI: A multi-omics and deep learning perspective," *Journal of Population Therapeutics and Clinical Pharmacology*, vol. 32, no. 2, pp. 573-582, doi: 10.53555/c5xrb097.

[20] V. Zhong, C. Xiong, and R. Socher, "Seq2SQL: Generating structured queries from natural language using reinforcement learning," *arXiv preprint arXiv:1709.00103*, 2017.

[21] T. Yu et al., "Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 3911-3921.

[22] A. Kunchukuttan, D. Kakwani, S. Golla, A. Bhattacharyya, M. M. Khapra, and P. Kumar, "Ai4Bharat-IndicTrans: A large-scale transformer-based multilingual machine translation system for 11 Indic languages," *arXiv preprint arXiv:2010.14545*, 2020.

[23] Google AI, "Gemini: A family of highly capable multimodal models," *Google Research Technical Report*, 2023.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

[24] P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 9459-9474.
[25] Liu, "LlamaIndex: Data framework for LLM applications," *GitHub Repository*, 2023. [Online]. Available: https://github.com/jerryjliu/llama_index
[26] Central Ground Water Board, "Ground Water Year Book - India 2023-24," *Ministry of Jal Shakti, Government of India*, 2024.
[27] India-WRIS, "Water Resources Information System of India," *National Informatics Centre*, 2024. [Online]. Available: https://indiawris.gov.in/