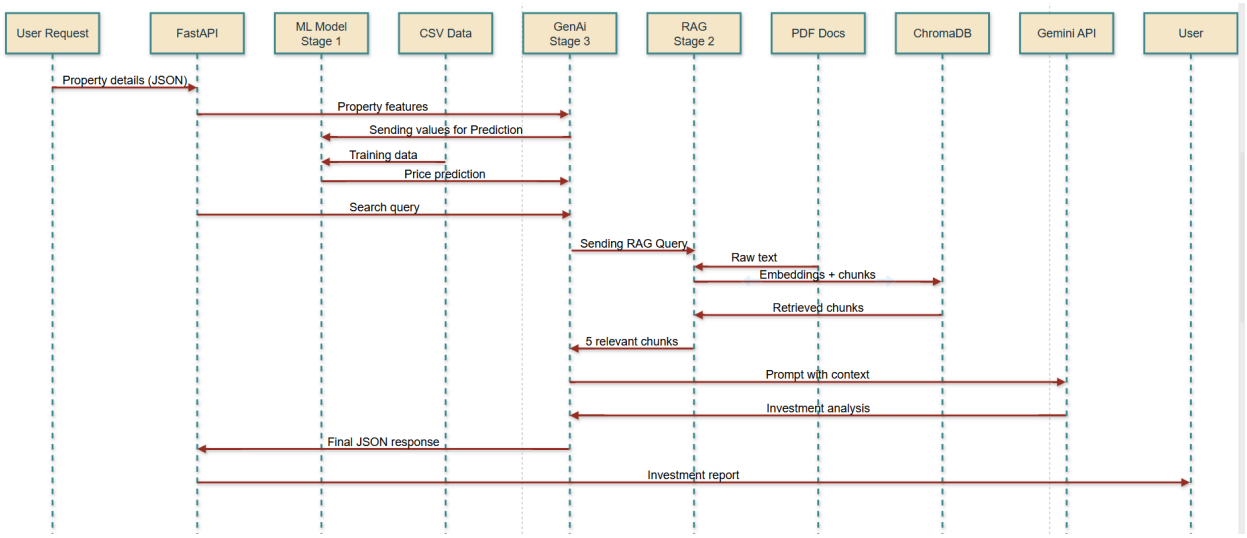
Real Estate GenAI System: Technical Report (Hyderabad)

1. System Workflow (End-to-End)

This section describes how the system operates in two modes: **offline preparation** and **online runtime inference**.



1.1 Offline Workflow (Setup / One-time or periodic)

Step 1: Structured dataset preparation

Step 2: Model training + evaluation

Step 3: RAG document ingestion

1.2 Online Workflow (Per user request)

Step 1: API request intake

Step 2: Valuation inference (Stage 1)

Step 3: Retrieval (Stage 2)

Step 4: Generative reasoning (Stage 3)

Step 5: Response assembly (Stage 5) - focus on output quality over latency

1.3 Key Trade-offs Summary

Decision	Chosen Approach	Alternative	Rationale
ML Model	XGBoost	LightGBM/Deep Learning	Optimal, faster training, native categorical handling
Vector DB	ChromaDB (local)	Pinecone / Weaviate	Simple setup, sufficient for <100K chunks, no external dependencies
LLM Provider	Gemini API	Claude / Local Llama	Free tier available, good quality, fast inference
Chunking	Fixed 800 chars + 100 overlap	Semantic / Sliding window	Balanced approach: fast, predictable, maintains context
Embedding Model	text-embedding-004 (768-dim)	all-MiniLM-L6-v2 (384-dim)	Better quality for longer chunks, still fast enough
API Framework	FastAPI	Flask / Django	Modern async support, auto documentation, type validation

2. Data Assumptions

2.1 Structured dataset (Hyderabad)

- **Dataset:** `Hyderbad_House_price.csv` (used only Hyderabad housing listings - 3k+ rows)
- **Target:** `price(L)` (price in lakhs)

- **Core features used (current baseline):**
  - Numeric: `size_sqft`, `age_yrs`
  - Categorical: `locality`, `property_type`
- **Synthetic feature added (assignment requirement):**
  - `age_yrs` generated randomly between **0 and 5 years** (uniform distribution, rounded to 1 decimal).
  - Rationale: The raw dataset lacks a reliable “property age” field, but age is a meaningful driver of price and rent. This synthetic field satisfies the “feature engineering” expectation while enabling model comparison.
  - Limitation: Because age is synthetic (not observed), it will not reflect real depreciation patterns. In a production setting, this should be replaced with true age (year built / possession year).

2.2 Cleaning and preparation

- **Duplicates removed** to reduce repeated listings.
- Numeric fields (`price(L)`, `size_sqft`, `age_yrs`) coerced to numeric; non-parsable entries treated as missing.
- Rows with missing required fields dropped.
- **Outlier handling:** IQR-based filtering on `price(L)` and `size_sqft`.
  - Rationale: Listings often contain extreme/incorrect values that distort learning.
  - Trade-off: Outlier filtering reduces noise but may remove valid luxury inventory.

2.3 Data split and evaluation setup

- Train/test split: **80/20**, fixed random seed.
- Metrics computed on holdout set:
  - **RMSE**: penalizes large errors (important for high-priced properties)
  - **MAE**: interpretable average absolute error in lakhs

3. Model Architecture

3.1 Feature engineering & preprocessing

- **Numerical scaling:** StandardScaler on `size_sqft` and `age_yrs`
- **Categorical encoding:** OneHotEncoder on `locality` and `property_type`
  - `handle_unknown="ignore"` ensures safe inference even when new localities appear

This preprocessing enables fair comparison across model families (linear vs. tree-based).

3.2 Models compared (at least two)

Model A: Linear Regression (baseline linear model)

- Why used:
  - Establishes a simple, interpretable baseline for tabular price prediction
  - Provides a clear reference point to judge whether more complex models truly add value
  - Works well when the relationship between predictors (e.g., area) and price is approximately linear
  - Fast to train and easy to explain (important for small datasets + academic evaluation)
- Best suited when:
  - Relationships are mostly linear and interactions are weak
  - Dataset is small and noisy
  - We want to detect whether model improvements are due to genuine structure or overfitting

Model B: XGBoost Regressor (non-linear gradient boosting)

- Why used:
  - Learns non-linear relationships and feature interactions (e.g., locality × property_type × size)
  - Typically performs strongly on structured tabular data
- Best suited when:
  - Real estate pricing is driven by complex patterns (which is common)
  - There are interactions and non-linear effects

3.3 Which model is "better" and why

```
Model Comparison:
  XGBoost:           RMSE=20.71L, Confidence=0.79
  Linear Regression: RMSE=25.4L, Confidence=0.75
  Best Model:        XGBoost
```

For Hyderabad housing data, **XGBoost is expected to outperform Linear** on RMSE/MAE because:

- Price is rarely purely linear in square footage across localities
- Locality effects vary by property type (interaction)
- Boosted trees capture "threshold" behaviors (e.g., price jumps above certain sqft ranges)

However, **Linear Regression remains valuable**:

- If dataset size is small or noisy, simpler models can generalize better
- Linear Regression provides interpretability and baseline sanity-checking

Decision rule used:

- Pick the model with lower **RMSE and MAE** on the held-out test set.
- Use Linear Regression as the baseline; use XGBoost if it improves metrics meaningfully without unstable behavior.

3.4 Uncertainty / confidence estimate

- A simple confidence proxy is derived from prediction error scale (RMSE).
- Since the dataset is small, the focus is on **effective, reasonable output** rather than complex probabilistic uncertainty.
- Production-ready alternative: conformal prediction intervals or quantile regression.

4. RAG Design (Stage 2)

4.1 Inputs (Unstructured text)

The RAG system ingests:

- **Legal PDFs**: Telangana RERA regulations, registration rules, compliance documents
- **Market/news text**: announcements, infrastructure updates (metro extensions, roads), analyst notes, local market commentary
  Allowed formats: **PDF, TXT, Markdown**

4.2 Chunking strategy

- Chunking: fixed-size chunks (approx. **800 characters**) with **overlap ~100 characters**
- Rationale:
  - Fixed chunk sizes simplify implementation and make retrieval consistent
  - Overlap prevents losing key statements at chunk boundaries
- Trade-off:
  - Larger chunks = more context but may dilute relevance
  - Smaller chunks = higher precision but higher risk of missing connected clauses across sections

4.3 Embedding strategy

- Embeddings generated using the **Gemini embedding model** (`text-embedding-004`).
- Rationale:
  - Produces semantic vector representations aligned with the Gemini reasoning model family

    - Good for matching regulatory / market queries even when phrased differently

4.4 Vector database choice: ChromaDB

ChromaDB is used as the vector store, persisted locally.

Why ChromaDB (for this assignment + dataset scale):

- Free and runs locally
- Simple persistent setup with minimal infra overhead
- Works well for small/medium document collections typical in assignments

Why not Pinecone (here):

- External managed service → cost + account setup overhead
- Overkill for small scale (few documents, thousands of chunks)

Scale assumption:

- With a small dataset and small document corpus, retrieval latency is not the bottleneck.
- Focus is **retrieval quality + clarity** rather than micro-optimizing speed.

4.5 Retrieval (Top-K)

- Query embedding is computed from the user/property question
- Top-K (e.g., 5) most relevant chunks retrieved
- **Why Top-K = 5:**
  We use **K=5** to get *enough* evidence for Stage 3 (regulatory + market + risks) without overwhelming Gemini. With a small document set, the best matches usually appear in the top few results, going lower risks missing key context, and going higher adds noise/duplicates that can reduce answer quality.
- Each chunk returns:
    - text
    - source document name
    - chunk index
    - distance score

4.6 Retrieval quality discussion (precision vs latency trade-off)

Since the dataset and corpora are small:

- We prioritize **precision and relevance** over latency
- We tune:
    - chunk size + overlap
    - Top-K value

    - query formulation (adding locality + “Telangana RERA” + “infrastructure” keywords)

5. Prompt Engineering (Stage 3)

5.1 Inputs to the LLM (Gemini)

The reasoning layer consumes:

1. **Property details** (city, locality, sqft, property_type, age)
2. **ML prediction output** (predicted price, RMSE, MAE, confidence proxy)
3. **Retrieved Top-K chunks** (regulatory + market evidence)
4. **Investor profile context** (risk tolerance, time horizon)

5.2 Prompt structure

The prompt is designed to:

- Provide explicit evidence blocks (“SOURCE 1…SOURCE 4”)
- Force the model to answer specific investment questions:
    - rental yield reasoning
    - RERA compliance concerns
    - infrastructure effects
    - legal risks and documentation checks
- Output strictly in **JSON** so the API layer can consume results

5.3 Hallucination safeguards

- Instruction: analysis must use retrieved context; if context is missing, explicitly state limitations
- Output constrained to a schema:
    - investment_view rating
    - summary
    - drivers
    - risks
    - assumptions
    - sources used
- If JSON parsing fails, the system returns a controlled fallback response

5.4 Explicit limitations communicated

- “RAG does not guarantee completeness of legal review”
- “Synthetic age feature may not reflect true depreciation”
- “Recommendations are informational, not financial/legal advice”

6. Evaluation and Limitations

6.1 ML evaluation (Stage 1)

- Metrics:
    - **RMSE**: penalizes large errors → important for expensive properties
    - **MAE**: interpretable average error in lakhs
- Goal:
    - Provide reliable baseline prediction for investor context and narrative grounding
- Small dataset approach:
    - Prefer stable metrics and clear baselines rather than aggressive tuning

6.2 RAG evaluation (Stage 2)

Recommended evaluation (lightweight but credible):

- Manual relevance judging for ~20 representative queries
- Metrics:
    - Track latency (but not a primary concern due to small scale)

6.3 End-to-end evaluation (Stage 3)

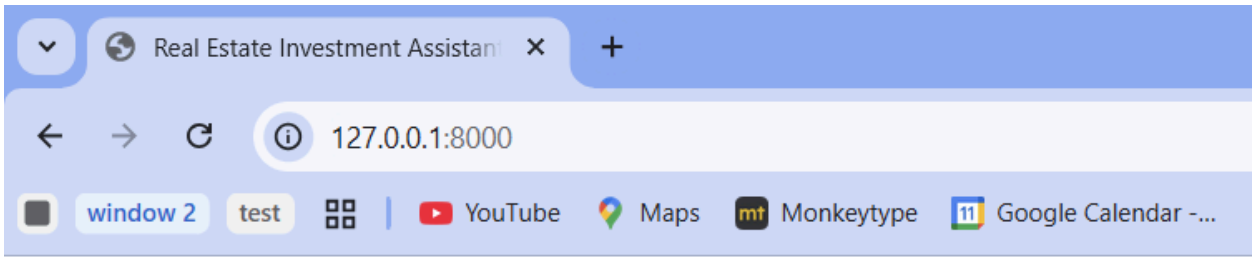
- JSON validity rate (did it produce schema-correct JSON?)
- Citation presence (did it reference sources?)
- Human review: does narrative align with retrieved evidence?

6.4 Key limitations

- **Synthetic age_yrs** limits realism of age-related interpretation
- Dataset may not include critical drivers like:
    - BHK, amenities, floor, facing, parking, furnishing, approvals
- RAG corpus quality controls output quality:
    - If legal PDFs are incomplete/outdated, analysis may miss constraints
- This system provides an *investment-style narrative*, not legal clearance

7. Example Outputs

1.



Property details

City: Hyderabad
Locality: Madhapur
Property type: Villa
Size (sqft): 1200
Age (years): 2

User investment context

is this good investment?

Analyze Investment

Investment explanation

View: POOR

Summary: This property is considered a poor investment today, not due to identified negative factors, but because of insufficient evidence in the retrieved documents to assess crucial investment metrics. Critical information on rental yield, price appreciation potential, local infrastructure, and broader legal risks is unavailable, preventing a comprehensive and informed investment decision. While the property likely falls under Telangana RERA regulations, a full assessment of its investment potential cannot be made.

Drivers

- Existence of a regulatory framework (Telangana RERA Rules, 2017) which applies to projects approved after January 1, 2017, providing some market oversight.

Risks

- Insufficient evidence in retrieved documents to assess rental yield for passive income potential (3%+ is considered good for the Indian market).
- Insufficient evidence in retrieved documents to assess price appreciation potential or broader market trends.
- Lack of information in retrieved documents regarding local infrastructure (metro connectivity, roads) and upcoming market developments in Madhapur.
- Insufficient evidence in retrieved documents to evaluate broader legal risks such as specific documentation issues, additional approvals beyond RERA, or encumbrances.
- Uncertainty regarding the specific project's RERA registration and compliance status.

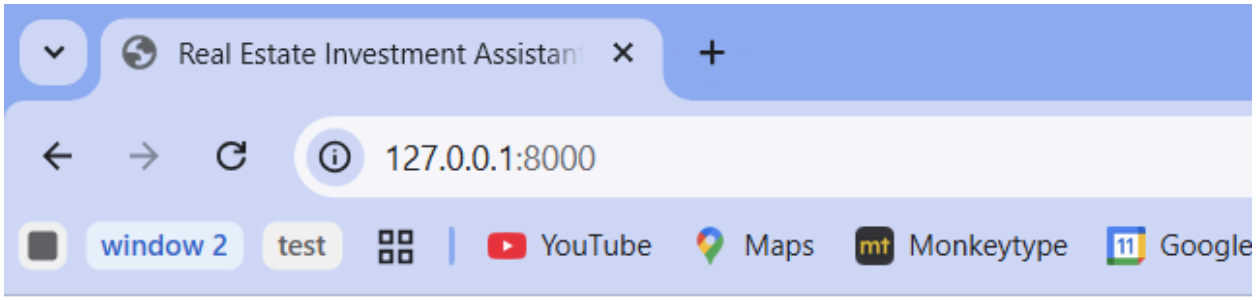
Assumptions

- The property's age (2 years) implies that its building permissions were likely approved after January 1, 2017, making it subject to the Telangana State Real Estate (Regulation and Development) Rules, 2017.
- The ML model's predicted price of 112.96 Lakhs is accepted as a current market valuation point, but its underlying methodology and market context are not detailed in the RAG chunks.
- The investment decision relies solely on the information provided in the RAG chunks and property details.

Model prediction

Predicted price: 112.96 L
RMSE: 20.71
Confidence: 0.79

2.



Property details

City: Hyderabad
Locality: Kompally
Property type: Apartment
Size (sqft): 2000
Age (years): 1

User investment context

Is this good investment?

Analyze Investment

Investment explanation

View: AVERAGE

Summary: No, a definitive 'good investment' assessment cannot be made at this time due to significant gaps in critical market and financial data. While the existing RERA framework in Telangana provides a regulatory structure for real estate projects, essential details like rental income, specific market trends, and upcoming infrastructure developments are unavailable, preventing a comprehensive evaluation.

Drivers

- The Telangana State Real Estate (Regulation and Development) Rules, 2017, provide a regulatory framework for real estate projects, aiming to enhance transparency and consumer protection (RR1, chunk 1, 2, 5).
- The property's age (1 year) indicates it falls under the purview of these Telangana RERA Rules, as they are applicable to projects with building permissions approved on or after January 1, 2017 (RR1, chunk 3).

Risks

- Insufficient evidence in retrieved documents to assess local rental income, making it impossible to calculate the rental yield or determine its suitability for passive income.
- Insufficient evidence in retrieved documents regarding specific market trends for Kompally, Hyderabad, or upcoming infrastructure developments (e.g., Metro connectivity, road projects) to evaluate future price appreciation potential.
- Insufficient evidence in retrieved documents on local floor/construction limits or zoning regulations applicable to this property.
- Potential legal risks exist if the promoter of this specific project has not fully complied with RERA registration requirements or has made changes to approved plans without strict compliance as per the Act (RR1, chunk 3, 4).

Assumptions

- The ML model's predicted price of 109.68 L for the 2000 sqft apartment in Kompally, Hyderabad, is a reasonable estimate.
- Further due diligence, including verification of RERA registration for this specific project, all legal documents, and a site visit, would be conducted for a complete investment decision.
- The property is structurally sound and free from immediate physical defects, as no information to the contrary was provided.

Model prediction

Predicted price: 109.68 L
RMSE: 20.71
Confidence: 0.79

The ‘rating reflects disciplined uncertainty, not weak modeling, demonstrating that the system correctly avoids overconfidence when market and legal evidence is missing.

8. Future Enhancements

8.1 Functional Improvements

Enhancement	Description	Priority
Rental Yield Prediction	Add ML model to estimate rent and calculate ROI/yield	High
Better Uncertainty Estimates	Use quantile regression or bootstrap intervals instead of simple confidence scores	Medium
Advanced RAG Retrieval	Add metadata filtering, query classification, and reranking for better precision	Medium
Document Management	Admin API to upload/version documents with timestamps	Low
Evaluation Suite	Track Precision@k and relevance metrics on fixed test queries	Medium

8.2 Scale & Production Requirements

If expanding beyond prototype:

1. **Scalability:** Migrate ChromaDB → Pinecone/Weaviate for >100K documents
2. **Reliability:** Add retries, fallbacks, and strict LLM output validation
3. **Security:** Use secret manager for API keys, avoid logging sensitive data
4. **Monitoring:** Track latency per stage, failure rates, and RAG quality metrics
5. **Model Ops:** Automated retraining triggers (monthly or on data drift detection)

8.3 Agentic Architecture (Recommended next step)

Introduce a modular, agent-based orchestration layer so each capability is isolated, testable, and extensible:

- **Valuation Agent**
  Responsible for running the ML pipeline (Linear Regression + XGBoost), selecting the best model using RMSE/MAE, and returning prediction + confidence estimate.

- **Market Intelligence Agent**
  Responsible for market/news document retrieval (RAG queries), identifying locality-specific signals (infrastructure announcements, demand drivers), and producing evidence-backed bullet insights.

- **Risk & Compliance Agent**
  Responsible for retrieving legal/regulatory content (RERA rules, approvals, compliance checks) and producing risk flags with strict evidence rules (no invention; “Insufficient evidence…” if unsupported).

- **Narrative Agent**
  Responsible for composing the final structured narrative (drivers/risks/assumptions) using only the outputs of the other agents and their citations.

MCP / A2A communication pattern (conceptual):

- Agents communicate via **structured message passing** (JSON objects), not raw text.
- Each agent produces an output schema with:
  `claims`, `evidence`, `confidence`, `errors`.
- The Narrative Agent acts as a **coordinator** that merges outputs and enforces guardrails.

Error handling (agent level):

- If Market/Risk agent retrieval returns zero chunks → degrade gracefully with explicit “Insufficient evidence…”
- If Gemini response fails JSON parsing → fallback response with error flag and raw snippet stored for debugging
- If a model artifact is missing → auto-train once or return a controlled “model unavailable” response

Extensibility:

- Add future agents without touching existing ones (e.g., Rental Yield Agent, Comparable Sales Agent, Fraud Detection Agent, Document Freshness Agent).

8.4 Out of Scope

Explicitly excluded to maintain focus on core functionality:

- ❌ **Live data feeds** - No real-time listing ingestion
- ❌ **Financial underwriting** - No mortgage calculations, IRR, NPV modeling
- ❌ **Legal verification** - Cannot certify titles or validate RERA registration
- ❌ **Multi-tenancy** - No user authentication or role-based access
- ❌ **Production vector DB** - No sharding, replication, or managed infrastructure
- ❌ **Auto-updates** - No document freshness detection or version tracking
- ❌ **MLOps pipeline** - No CI/CD for models or automated drift monitoring
- ❌ **Human review workflow** - No approval queues for recommendations
- ❌ **Multi-city support** - Scoped to Hyderabad/Telangana only

Scope Justification: These were excluded to prioritize correctness, evidence-based reasoning, and delivering a working end-to-end prototype within assignment constraints

9. Conclusion (Why this design fits the assignment)

This solution meets the rubric by delivering:

- Structured **price prediction** with RMSE/MAE
- Comparison of **two models** (baseline + stronger non-linear model)
- **RAG pipeline** with chunking, embedding, vector DB choice, and precision/latency trade-offs
- **Generative reasoning layer** with grounded prompts, guardrails, and explicit limitations
  And because the data scale is small, the system intentionally prioritizes **effective output quality** and evidence-grounded reasoning over latency optimization.