AI-Based Domestic Relocation Assistant for Türkiye

Harun Hüdai Tan
TED University
Dept of Software Engineering
Ankara Turkey
harunhudai.tan@tedu.edu.tr

Baran Erol
TED University
Dept of Software Engineering
Ankara Turkey
baran.erol@tedu.edu.tr

Hüseyin Yiğit Erten
TED University
Dept of Computer Engineering
Ankara Turkey
huseyinyigit.erten@tedu.edu.tr

Ahmet Arda Bağdatlı
TED University
Dept of Software Engineering
Ankara Turkey
ahmetarda.bagdatli@tedu.edu.tr

Kasım Murat Karakaya
TED University
Dept of Computer Engineering
Ankara, Türkiye
murat.karakaya@tedu.edu.tr

2025 10th International Conference on Computer Science and Engineering (UBMK) | 979-8-3315-9975-1/25/$31.00 ©2025 IEEE | DOI: 10.1109/UBMK67458.2025.11206933

***Abstract*—The decision to relocate to a new city presents significant challenges for individuals who must research multiple aspects of cost of living before making informed choices. This paper presents an Artificial Intelligence-based domestic relocation assistant system specifically designed for domestic relocation within Turkey that retrieves comprehensive information across key living expense categories and generates personalized analysis reports based on user preferences. The system uses Google Gemini AI's API integrated with Python and a modular multi-agent data collection architecture. By integrating multiple grounding techniques including Retrieval Augmented Generation (RAG), internet search, web crawling and specialized system prompts, the system retrieves relevant information across multiple domains: real estate, market prices, education costs, and transportation expenses. The collected data is processed by Google Gemini to generate comprehensive advice reports that enable faster and more informed decision-making for domestic relocation within Turkey.**

Keywords — Artificial Intelligence; Retrieval Augmented Generation; System Prompts.

I. INTRODUCTION

The decision to relocate within a country presents complex challenges that require comprehensive analysis of multiple economic factors. In Turkey, this challenge is particularly significant given the large scale of internal migration increase across the nation. According to Turkish Statistical Institute (TÜİK), a total of 3,450,953 people migrated between provinces in Turkey in 2023 [1]. This is a significant increase considering the internal migration rate increased from 3,27% in 2022 to 4,04% in 2023.

The scale of internal migration in Turkey demonstrates the need for an accessible comprehensive cost of living information. Traditional approaches to gathering cost of living information for domestic relocation are time-consuming and fail to provide personalized information and recommendations for people who want to migrate within Turkey.

The rapid advancement of Artificial Intelligence, particularly in natural language processing and automated report generation, presents an opportunity to change how individuals access and analyze location-based cost of living information. This study proposes an AI-Based Domestic Relocation Assistant system that leverages multiple specialized data collection techniques and Gemini AI's [2] Natural Language Processing capabilities to generate comprehensive reports based on users' cost of living preferences helping them to make better decisions before moving from one city to another within Turkey.

The main technical contributions of this work include a modular multi-agent architecture that enables domain-specific data collection and processing; integration of multiple grounding techniques including RAG, real-time web scraping, and structured database queries; and a template-based report generation system that synthesizes heterogeneous data sources into personalized recommendations.

II. RELATED WORK

A. Traditional Cost of Living Analysis Systems

Traditional cost of living applications such as Numbeo [3] and Expatistan [4] provide crowdsourced pricing information across different cities which rely on only static information that does not have accurate price information for provinces in Turkey. These traditional calculators allow city-to-city comparisons, but they do not provide personalized advice. In contrast, modern AI systems provided with relevant information coming from web sources and personal preferences can make recommendations to an individual's budget and preferences.

B. Retrieval-Augmented Generation Systems

Retrieval Augmented Generation has emerged as a promising solution for enhancing Large Language Models by incorporating knowledge from external databases. Lewis et al. [5] introduced the RAG framework, which retrieves relevant information chunks based on semantic similarity, thereby addressing challenges like hallucination and outdated knowledge in Large Language Models. This approach has proven particularly effective in domains requiring up-to-date information, making it highly relevant for cost-of-living analysis where current pricing data is crucial.

In another study Gao et al. [6] provided a comprehensive survey of RAG systems, demonstrating their effectiveness across various domains by incorporating external knowledge sources. Their analysis showed that RAG architectures significantly improve accuracy and reduce hallucinations in large language models, particularly for knowledge-intensive applications requiring up-to-date information. The survey highlighted RAG's ability to handle dynamic data sources and

979-8-3315-9975-1/25/$31.00 ©2025 IEEE

domain-specific information retrieval, making it highly suitable for applications like cost-of-living analysis where current pricing data and location-specific information are essential.

C. Prompt Engineering

The effectiveness of AI-powered applications heavily depends on the quality and structure of prompts used to guide the language model. Recent research in prompt engineering has identified several key techniques that significantly improve the quality and consistency of generated responses from Large Language Models.

System prompts play a crucial role in establishing the AI's role, response format, and analytical approach. Based on the comprehensive survey by Schullhoff et al. [7], effective prompt engineering for Large Language Models requires careful consideration of several key principles. Role-based prompting is fundamental to our system architecture, where each specialized agent is assigned a distinct expertise domain with clear responsibilities, as the survey demonstrates that assigning specific roles to LLMs creates more focused and reliable outputs for domain-specific tasks.

Few-shot prompting represents another critical technique for prompt engineering, improving model performance by providing a limited number of examples that demonstrate the desired task completion format and guide the model's responses. The survey [7] identifies several key design decisions for effective few-shot prompting, including exemplar quantity, ordering and quality considerations that significantly impact model behavior. Research shows that increasing the quantity of exemplars generally improves model performance, particularly in larger models, though benefits may diminish beyond a certain threshold. The order of exemplars affects model behavior substantially, with the survey [7] demonstrating that exemplar sequence can cause accuracy to vary from sub -50% to 90%+ on some tasks. These few-shot prompting principles are particularly relevant for cost-of-living analysis systems where consistent output formatting, accurate numerical reasoning, and domain-specific knowledge application are essential for generating reliable financial recommendations.

III. Material and Methodology

A. User Information Gathering

To enable the AI-Based Domestic Relocation Assistant system to operate more efficiently, users will be asked to provide specific information through a form. As detailed in Table I, the system collects comprehensive information across multiple cost of living domains to ensure accurate analysis. Users specify their housing preferences including preferred type of housing and number of rooms, along with current rent and utility expenses for electricity, gas, water, and internet services.

Vehicle ownership information is gathered, including fuel type, tank capacity, and monthly fuel fill count to calculate monthly fuel costs. For users pursuing higher education, the system collects current tuition costs, target university and department preferences to provide specific tuition cost comparison.

Transportation preferences encompass current public transportation pass expenses for cost comparison analysis between current and target cities public transportation prices. The system also captures other expenses such as lifestyle and entertainment covering gym memberships, healthcare expenses, dining, and subscription services. Finally, detailed grocery shopping preferences including dietary needs and product categories are collected for comprehensive market price analysis. This structured data collection approach, allows the system to tailor its research across all cost-of-living categories to match users' specific needs and preferences.

TABLE I
USER DATA REQUIREMENTS

Data Field	Details
Location Information	Current and target location (city, district) for cost comparison analysis between locations.
Personal Details	Age, family size, and monthly net income to determine budget capacity and household affordability.
Housing Preferences	Current rent, utility bills (electricity, gas, water, internet), preferred housing type, and number of rooms for housing cost comparison in target city.
Vehicle Information	Vehicle ownership status, fuel type, tank capacity, and monthly fuel fill count for calculating fuel expenses in current and target city.
Public Transportation Expenses	Current public transportation pass expenses and for transportation cost comparison analysis.
Education Information	Current tuition costs, target university and department preferences for specific university expense calculations.
Grocery List	Shopping preferences, dietary needs, and product categories for accurate monthly grocery cost estimation and price comparisons.
Other Expenses	Monthly budget for gym, healthcare, dining, entertainment, and subscriptions for lifestyle cost analysis.

B. Data Collection and Integration Strategy

The system will utilize various tools for data collection, with the most important being the Serper API [8] for Google Search and Crawl4AI [9] for web scraping processes applied as Fig. 2 web scraping process [10] to get relevant information. These tools will be primarily used for real estate and market research. Information related to education, transportation, and utilities will be sourced from pre-collected, static datasets. Among these, educational data will be stored in a RAG knowledge base that is used, as illustrated in Fig. 1, the traditional RAG architecture [11], which follows a "Retrieve-Read" framework and stores documents as chunks in a vector database, specifically designed for the education agent, while the remaining structured data such as transportation and utility costs will be stored in a database system due to their formatted nature. The system will integrate and use all this information to generate more accurate and relevant reports, better aligning with user preferences and needs.

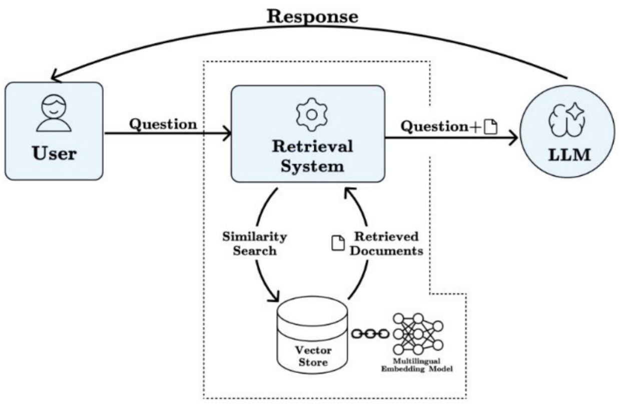

Fig. 1. Traditional RAG Pipeline [11]

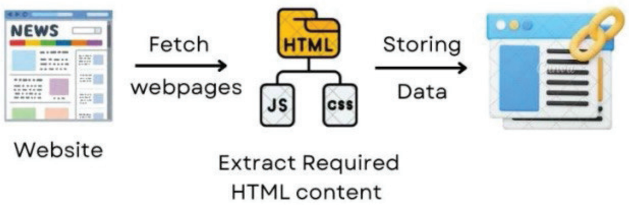

Fig. 2. Web Scraping Process [10]

TABLE II
RAG PARAMETERS

Parameter	Value
Chunk Size	800
Chunk Overlap	100
Tokens per Chunk	128
Embedding Model	distiluse-base-multilingual-cased-v1

Table II outlines key parameters used in RAG system, which are critical for how information is segmented and processed. The chunk size defines how large each piece of the educational document is, with a value of 800 ensuring sufficient context is captured for university pricing information and program descriptions within each segment. The chunk overlap is set to 100, meaning that adjacent chunks share some content to preserve important context at the boundaries, which helps improve retrieval accuracy. The tokens per chunk, set to 128, determine how much of each chunk is used during the generation phase, balancing detail with model input constraints. To generate embeddings for the RAG system, the distiluse-base-multilingual-cased-v1 sentence transformer model [12] is used. This model is a lightweight and efficient version of multilingual Universal Sentence Encoder (USE), capable of handling over 15 language and producing high-quality sentence embeddings. It is particularly well-suited for multilingual and cross-lingual retrieval tasks. These parameters and the embedding model were carefully selected based on our RAG performance results for educational document retrieval and were fine-tuned to align with the level of accuracy and relevance we aim to achieve in the education cost analysis section of the final cost of living advice report.

C. Report Generation Methodology

The report generation process follows a template-based approach based on user-specific parameters. Reports are structured into standardized sections covering user profile details, executive summary, monthly rental cost comparison between current and target city that contains monthly saving rate, education costs, market prices, transportation, and fuel price comparisons between current and target cities. Lastly the large language model advice part where the model provides conclusions and recommendations.

TABLE III
SAMPLING PARAMETERS FOR AGENTS

Agent	Top p	Top k	Temperature
Report Generator Agent	0,8	20	0,3
Education Agent	0,85	15	0,2

The values in Table III outline the selected generation configurations for each large language model. In their testing, Ahmed et al. [13] assessed how different temperature and top p settings influence model behavior across 24 test configurations using multiple LLMs and RAG scenarios. Their study identifies these two parameters as critical for controlling output characteristics. Temperature modulates stochastic variation, while top p governs token diversity.

We set the parameters of the Large Language Model that is used by the Report Generator Agent carefully. The temperature parameter is set to 0,3, which encourages the model to favor high-probability outputs, enhancing consistency. A top p value of 0,8 enables moderate diversity by excluding low-probability tokens while retaining flexibility. This combination provides a balanced trade-off between determinism and variation. The top k value of 20 further restricts the token selection to the 20 most likely candidates, providing an additional filtering mechanism. This combination provides a balanced trade-off between determinism and variation.

For the education agent, the lower temperature of 0,2, when combined with the RAG system, encourages outputs that closely align with retrieved text chunks. The top-p value of 0,85 further narrows the token selection, while the top k value of 15 creates a more restrictive vocabulary pool that minimizes the risk of generating irrelevant or hallucinated content by ensuring only the most contextually appropriate tokens are considered during generation.

D. System Redundancy & Data Validity

The report generation system utilizes data from official government and university websites to ensure the reliability of the information used in report generation. The careful selection of these sources enhances the accuracy and trustworthiness of the final outputs. An agent-level retry mechanism is implemented, allowing the system to automatically attempt regeneration when failures occur, based on captured error messages. These errors are systematically logged to support debugging and continuous system improvement over time.

IV. IMPLEMENTATION

This section describes the system architecture development and implementation of the cost-of-living advisor, as well as the integration details. It covers the development of each functional component, their data retrieval mechanisms, and the coordination with Google Gemini AI for report generation.

A. System Workflow

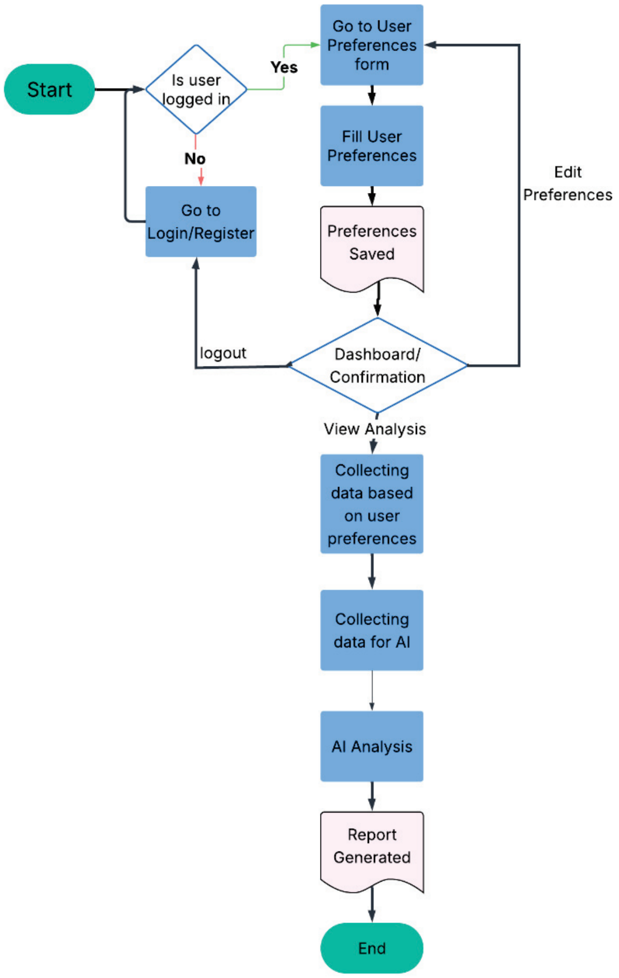

Fig. 3. System Workflow Diagram

This section describes the workflow architecture of the AI-Based Domestic Relocation Assistant system, detailing the sequential processing from user input acquisition to final report generation. The workflow illustrates data collection, and report generation utilizing Google Gemini AI's natural language processing capabilities. The complete system workflow is illustrated in Fig. 3 which demonstrates the interconnected process from initial user authentication through final report generation.

The workflow initiates with Flask-based REST API authentication utilizing JSON Web Token (JWT) for secure session management and user credential verification. Upon successful authentication, the system presents a preference form through Angular frontend components that capture essential relocation parameters including current and target locations, information such as age, family size and monthly net income.

User preferences are structured into categories encompassing housing preferences (property type, room configuration), vehicle ownership details (fuel type, tank capacity, monthly consumption), education requirements (target university, department), and shopping list for grocery price analysis. The system validates geographical data against the Utilities Turkey MySQL [15] database containing price information about utility and transportation prices of specific provinces in Turkey.

Following successful preference validation, when the user initiates report generation, the Report Generator Agent begins the data gathering process by coordinating with several domain specific Python classes. The Report Generator Agent class functions as the central orchestrator that both executes method calls to these domain specific classes and generates the AI analyzed report based on user preferences and the collected data from information sources.

These calculations mentioned above are going to be analyzed detailed in following sections.

B. Market Price Calculation

The market price calculation operates as one of the five main functions utilized by Report Generator Agent to fetch grounded pricing data based on user preferences. When invoked, the calculation method receives the user's shopping list from the input form and returns a JSON-formatted price analysis that is later provided to Gemini before final report generation.

The system maintains specific search links corresponding to the product options provided in the user form. These links contain real-time pricing data sourced from akakce.com [14], a comprehensive Turkish price comparison platform. The price calculation utilizes Crawl4AI to web scrape and retrieve HTML content after JavaScript execution, targeting specific JSON blocks that contain the website's internal query results.

The extracted JSON data is filtered to isolate product names and their corresponding pricing information. Subsequently, the function calculates optimal pricing by using the user-specified products with the lowest available prices from the product listings. This process is executed for each item in the user's shopping list, ultimately returning a structured JSON output containing total grocery cost.

C. Transportation Price Calculation

The transportation price module focuses on retrieving location-specific public transportation pricing based on user specified target region. Upon invocation, this function receives the user's preferred city from the input form and returns structured JSON data containing transportation cost information for the specified location.

The function utilizes the Utilities Turkey database to access transportation pricing information. This database-driven approach provides grounding for the Report Generator Agent by gathering municipal transportation data. The system executes targeted SQL queries against the database and structures the retrieved results into standardized JSON format, enabling integration with the LLM's report generation process.

D. Fuel Price Calculation

The fuel price calculation is used to acquire real-time fuel pricing data across different fuel types and distributors. The system sources this information from doviz.com [16], a platform that maintains current fuel price listings organized by distributor and geographic location within Turkey.

This function operates conditionally, activating only when the user form indicates vehicle ownership. The function employs Crawl4AI to retrieve markdown-formatted content from the target website, which is then parsed into a JSON.

The resulting data structure follows a standardized format containing location identifiers, search timestamps, and regional pricing breakdowns. Each regional entry includes distributor information and pricing for three primary fuel types: gasoline, diesel, and LPG, providing fuel cost analysis for the specified geographic area.

E. Real Estate Price Calculation

The real estate price function retrieves current property listings along with relevant utility pricing information. When activated, this function processes the user's housing preferences and target city from the input form and returns structured JSON data containing matching properties.

The function conducts search on emlakjet.com [17] real estate platform. The system utilizes Selenium to scrape price and room information. The standardized output format includes essential property details including rental prices, room configurations, square footage, property links.

F. Utility Price Calculation

The utility price calculation uses the data collected from various government websites about water, electricity, natural gas, and internet tariffs in Turkey. The collected data uploaded into the Utilities Turkey database to easily update and delete through time.

G. Education Price Calculation

The education price calculation collects university pricing data according to private university prices in Turkey. Upon invocation, this processes the university name provided in the user form and returns structured JSON data suitable for integration into the cost analysis report.

The function leverages a ChromaDB [18] vector database containing university pricing documentation stored as individual PDF files separated into chunks. The system employs semantic similarity search using dot product calculations on text embeddings to identify relevant pricing information matching the user's educational requirements.

The search algorithm retrieves the ten most similar results based on embedding proximity, which are subsequently formatted into JSON structure for integration with the LLM's report generation workflow. This vector-based approach enables flexible matching of user queries with available educational pricing data, accommodating variations in terminology and program descriptions.

H. Report Generation

The report generation process represents the final stage of the system implementation, where all collected data is synthesized into a comprehensive cost of living analysis document. This process is orchestrated by the Report Generator Agent, which integrates information from all domain-specific calculation functions to produce a structured, user-friendly report.

The Report Generator Agent receives JSON-formatted data from each calculation module including real estate prices, market costs, transportation expenses, fuel prices, utility costs, and education fees. This structured data is then processed through Google Gemini AI to generate analysis and recommendations.

The final report maintains a standardized structure consisting of multiple sections: user profile summary, executive overview, detailed cost breakdowns for each category, comparative analysis between current and target locations, and personalized recommendations based on the user's budget and preferences. As seen in Table IV, each cost category is showing current versus target location expenses, absolute differences, and percentage changes to facilitate easy comparison.

TABLE V
SAMPLE REPORT COST ANALYSIS

Cost Category	Current City (TL)	Target City (TL)	Difference (TL)	Change (%)
Monthly Rent	15000	47333	+32333	+216
Water	1000	860	-140	-14
Electricity	1000	1436	+436	+44
Natural Gas	1000	946	-54	-5
Internet	1000	349	-651	-65
Total Monthly Cost	19000	50924	+31924	+168

V. TESTING AND METRICS

The testing process focuses on evaluating the quality and efficiency of the generated reports by comparing AI-based evaluations with those from real human evaluators. This approach assesses both the quantitative performance and the qualitative aspects of the reports produced by the system.

Testing involved generating a series of reports, for which generation duration and page count were systematically recorded. Initially, an AI model was employed to evaluate the reports based on predetermined criteria. Subsequently, multiple human evaluators independently assessed the same reports against identical criteria. This comparative analysis facilitated the identification of strengths and weaknesses, enabling iterative improvements to the report generation process. The success metrics utilized to establish evaluation criteria included: report generation duration of less than two minutes per report, report page count of approximately four pages, and report evaluation ratings on a scale of 1 to 5 across five dimensions clarity, addressing core questions, completeness, accuracy, and consistency.

The testing results demonstrates that the average time to generate a report is 84 seconds in 20 report generations. The speed of the internet connection, capabilities and specs of the computer or servers being utilized, and the complexity of the report being generated are some of the variables that can affect these report generating times. The system regularly produces reports within the allotted period, indicating a medium level of efficiency that requires enhancement. The objective is to further optimize this process to generate reports under a minute.

TABLE VI
REPORT GENERATION PERFORMANCE METRICS

Rating Categories	Evaluators			
	AI	***Human 1***	***Human 2***	***Human 3***
Clarity	4,3	4,7	4,6	4,3
Addressing Core Question	4,7	4,5	4,6	4,5
Completeness	2,3	3	2,8	2,2
Accuracy	4	4,5	3,7	3,5
Consistency	3,6	3,3	2,8	2,9
Total Average Ratings	3,78	4	3,7	3,48

As seen in Table V the AI scored highest in Addressing the Core Question (4,7) and lowest in "Completeness" (2,3), with an overall average of 3,78. Human evaluators, with individual total averages ranging from 3,48 to 4,0 generally outscored the AI in "Completeness" and "Accuracy". However, the AI was competitive in Clarity and Addressing Core Question. The varied human ratings indicate subjective differences in judgement. Statistical analysis showed high consensus on "Addressing Core Question" with a mean of 4,57 and standard deviation of 0,10, but significant disagreement on "Completeness" with a mean of 2,58 and standard deviation of 0,39. Overall ratings averaged 3,74 with moderate variability (standard deviation of 0,21), indicating that while reports were generally well-received, "Completeness" and "Consistency" require improvement.

VI. CONCLUSION

This study presents an AI-Based Domestic Relocation Assistant system for Turkey that integrates web scraping, RAG, and database queries with Google Gemini AI to generate personalized relocation reports within two minutes. The modular agent-based architecture enables domain-specific data collection and natural language report generation, empowering users to make informed relocation decisions through detailed cost comparisons across rental, education, transportation, utility, and grocery expenses, while the open-source project [19] encourages academic reuse and future development.

Despite the system's contributions, several limitations warrant consideration. Regional data availability varies significantly across Turkish cities, with some areas having limited real-time pricing information. The system's reliance on web scraping has potential vulnerabilities to website structure changes, which could affect data collection reliability.

VII. ACKNOWLEDGEMENT

This study was carried out as part of the SENG 472 – Building AI Applications with Large Language Models course, under the supervision of Dr. Kasım Murat Karakaya at TED University.

REFERENCES

[1] TÜİK, "İç Göç İstatistikleri, 2023," Turkish Statistical Institute, 2024. [Online]. Available: https://data.tuik.gov.tr/Bulten/Index?p=Ic-Goc-Istatistikleri-2023-53676. [Accessed Jun. 1, 2025].

[2] Google, "Gemini API Reference," Google AI for Developers. [Online]. Available: https://ai.google.dev/api. [Accessed Jun. 1, 2025].

[3] Numbeo, "Cost of living," Numbeo.com. [Online]. Available: https://www.numbeo.com/cost-of-living/. [Accessed Jun. 1, 2025].

[4] Expatistan, "Cost of living comparisons, 2025 data," Expatistan.com. [Online]. Available: https://www.expatistan.com/cost-of-living. [Accessed Jun. 1, 2025].

[5] P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," arXiv preprint arXiv:2005.11401, May 2020. [Online]. Available: https://arxiv.org/abs/2005.11401. [Accessed Jun. 1, 2025].

[6] Y. Gao et al., "Retrieval-Augmented Generation for Large Language Models: A Survey," arXiv preprint arXiv:2312.10997, Dec. 2023. [Online]. Available: https://arxiv.org/abs/2312.10997. [Accessed Jun. 1, 2025].

[7] S. Schulhoff et al., "The Prompt Report: A Systematic Survey of Prompt Engineering Techniques," arXiv preprint arXiv:2406.06608, Jun. 2024. [Online]. Available: https://arxiv.org/abs/2406.06608. [Accessed Jun. 1, 2025].

[8] "Serper API Documentation," Serper.dev. [Online]. Available: https://serper.dev/. [Accessed Jun. 1, 2025].

[9] UncleCode, "Crawl4AI: Open-Source LLM-Friendly Web Crawler & Scraper," GitHub Repository. [Online]. Available: https://github.com/unclecode/crawl4ai. [Accessed Jun. 1, 2025].

[10] N. Kanataria, K. P. Patel, H. N. Patel, P. Goel, K. Patel and D. Garg, "RAG-Enhanced Large Language Model for Intelligent Assistance from Web-Scraped Data," 2024 9th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2024, pp. 1043-1048, doi: 10.1109/ICCES63552.2024.10859894. keywords: {Accuracy;Large language models; Retrieval augmented generation;Semantics;Education;Data models;Explosions;Data mining;Indexing;Generative AI;Large Language models (LLMs);Retrieval augmented generation (RAG);Web scraping;Question-answering;RAGAS)}

[11] C. Iscan, M. F. Ozara and A. Akbulut, "Enhancing RAG Pipeline Performance with Translation-Based Embedding Strategies for Non-English Documents," *2024 Innovations in Intelligent Systems and Applications Conference (ASYU)*, 2024, pp. 1–8, doi: 10.1109/ASYU62119.2024.10756977.

[12] Hugging Face, "distiluse-base-multilingual-cased-v1," Hugging Face Model Hub. [Online]. Available: https://huggingface.co/sentence-transformers/distiluse-base-multilingual-cased-v1. [Accessed Jun. 1, 2025].

[13] B. S. Ahmed, L. O. Baader, F. Bayram, S. Jagstedt and P. Magnusson, "Quality Assurance for LLM-RAG Systems: Empirical Insights from Tourism Application Testing," 2025 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW), Naples, Italy, 2025, pp. 200-207, doi: 10.1109/ICSTW64639.2025.10962487.

[14] Akakçe, "Akakçe Price Comparison Platform," Akakce.com. [Online]. Available: https://www.akakce.com/. [Accessed Jun. 1, 2025].

[15] Oracle Corporation, "MySQL Developer Zone," MySQL.com. [Online]. Available: https://dev.mysql.com/. [Accessed Jun. 1, 2025].

[16] Doviz.com, "Akaryakıt Fiyatları," Doviz.com. [Online]. Available: https://www.doviz.com/akaryakit-fiyatlari. [Accessed Jun. 1, 2025].

[17] Emlakjet, "Emlakjet Real Estate Platform," Emlakjet.com. [Online]. Available: https://www.emlakjet.com/. [Accessed Jun. 1, 2025].

[18] Chroma, "Chroma: The AI-native Open-source Embedding Database," GitHub Repository. [Online]. Available: https://github.com/chroma-core/chroma. [Accessed Jun. 1, 2025].

[19] H. H. Tan, "AI-Powered Regional Cost of Living Advisor For Turkey," GitHub Repository, 2025. [Online]. Available: https://github.com/HarunHudaiTan/AI-Powered-Regional-Cost-of-Living-Advisor-For-Turkey. [Accessed: Jun. 1, 2025]