KG-ITP: A Knowledge Graph for Intelligent Travel Planning

Anonymous Authors
Anonymous Institution

***Abstract*—Modern travel planning requires knowledge graphs (KGs) that semantically integrate heterogeneous datasets while prioritizing niche attractions and real-world constraints. This paper introduces a Travel Planning Knowledge Graph (TPKG) combining GNIS natural features, IMLS cultural sites, Yelp dining taxonomies, and Valley Metro transit networks through schema alignment and OWL 2 ontological modeling. The TPKG's novel architecture establishes spatial-temporal dependencies between activities (museum hours), transit accessibility (light rail buffers), and user constraints (dietary needs) using `GeoSPARQL` proximity functions and RDF.**

By resolving semantic conflicts between authoritative and crowd-sourced datasets, the KG achieves 93% coverage of underrepresented attractions omitted by commercial platforms. Quantitative evaluation demonstrates subsecond query latency for complex multi-criteria requests like identifying vegetarian-friendly restaurants within 500m of transit-accessible cultural sites. The modular design enables seamless integration of third-party datasets through shared vocabularies. This work advances tourism informatics by demonstrating how ontology-driven KGs outperform keyword-based systems in personalization accuracy while maintaining W3C standards compliance for spatial-semantic reasoning.

I. Introduction

Modern travelers demand highly personalized itineraries that align with dynamic constraints, yet existing tools rely on rigid template-based recommendations due to fragmented data representation. Semantic web technologies address this gap through structured knowledge integration, but current systems lack scalable architectures for unifying heterogeneous tourism datasets (transport schedules, attraction metadata, user profiles) into actionable insights. Personalized travel planning remains constrained by rigid recommendation systems that prioritize popular destinations over niche attractions and lack semantic interoperability between heterogeneous data sources. Current tools rely on keyword matching and static templates, failing to adapt to dynamic constraints such as real-time transit availability or dietary preferences.

This paper presents a Travel Planning Knowledge Graph (TPKG) that leverages ontology-driven reasoning to generate context-aware itineraries. The research establishes three objectives: First, to design a domain-specific ontology modeling spatial-temporal dependencies between activities, transit modes, and user preferences. Second, to implement schema alignment techniques to integrate authoritative datasets including GNIS natural features, IMLS museum registries, Yelp restaurant taxonomies, and Valley Metro transit networks, facilitating third-party service integration via shared RDF vocabularies. Third, to evaluate the KG's ability to surface underrepresented attractions through geosparql-enabled proximity analysis.

The TPKG establishes contextual linkages between activities (e.g., museum visits), transit accessibility (Valley Metro schedules), and personalized preferences (dietary needs). Its ontology models temporal dependencies such as attraction opening hours and transit frequencies, while GeoSPARQL proximity functions optimize route planning between semantically related entities. This architecture enables dynamic query resolution for multi-criteria requests like identifying vegetarian-friendly restaurants near light rail-accessible museums, bypassing keyword-based limitations through ontological reasoning.

The system operationalizes the Travel Planning Knowledge Graph (TPKG) through a FastAPI backend that dynamically generates GeoSPARQL queries based on user inputs (e.g., dietary preferences, transit modes). These queries integrate real-time data from GNIS natural landmarks, Yelp restaurant taxonomies, IMLS cultural sites, and Valley Metro transit networks, leveraging ontology-driven reasoning to optimize spatial-temporal constraints. A React.js frontend visualizes recommendations as interactive flowcharts, mapping attractions with embedded transit durations and proximity metrics derived from GeoSPARQL.

II. Related Literature

Semantic web technologies have revolutionized tourism recommendation systems through structured knowledge integration. Recent advancements in semantic web technologies and knowledge graphs (KGs) have significantly influenced intelligent travel planning systems. Regalia *et al.* (2) established **GNIS-LD** as a foundational spatial-semantic dataset, providing GeoSPARQL-compatible geometries for 2.7 million U.S. geographic features. Our work extends this by integrating GNIS-LD's natural landmarks with crowd-sourced Yelp restaurant taxonomies and IMLS cultural registries through RDF-star annotations, addressing temporal validity gaps in dynamic itinerary generation identified by Chessa *et al.* (7).ultimately contributing to the Semantic Web through improved interoperability and user-centric experiences.

Early ontology frameworks like cDOTT (11) demonstrated modular approaches to tourism data integration, while Yazdizadeh et al. (3) established transportation ontologies for smart city routing. The main focus of (11) is to improve the integration and personalization of tourism information by creating structured, interoperable, and contextually relevant data for user-centered tourism services utilizing the ontology framework (cDOTT). By adopting the paper's ontology framework, we can efficiently organize and query tourism data using modularized ontologies for time, place, and user context by implementing the ontology framework from the study. This makes travel suggestions more accurate and flexible by enabling user's interests to be matched with relevant locations, events, and activities. Our project extends this technique by using SPARQL queries to get specific data from our knowledge network depending on user preferences and trip specifics, as proposed by the paper. We expand on this by allowing more advanced inquiries, such as locating "hidden gem" spots within a certain radius of the user's chosen destination, resulting in highly personalized and unique travel ideas.

Yazdizadeh, et al. (3) demonstrated how semantic web ontologies integrate diverse data sources for intelligent transportation and travel planning. The ontology models help us to take inspiration for the structures that will be used for representing the mode of transportation, routes, and related contextual data. Incorporating these concepts into our application would advance the web semantics This will be done by connecting user profiles, contextual trip data, and transport networks. If this approach is followed, it enhances semantic interoperability, enabling personalized travel experiences. In addition to the above points, our projects extend by enhancing the travel planners' adaptability by connecting user preferences with real-time travel data, enabling personalized travel plan options.

Yochum et al. (8) demonstrated DBpedia-based attraction recommendations but achieved only 68% coverage of cultural sites. Our TPKG surpasses this through federated GeoSPARQL queries across four domains (natural/cultural/commercial/transit), achieving 93% recall of niche attractions. This aligns with Li et al.'s (20) evidence-based frameworks for sustainable travel behavior, which inform our preference-driven recommendation engine.

Tourism knowledge graphs have advanced niche attraction discovery. Dokić *et al.* (5) improved hotel matching via Protégé ontologies, while Li *et al.* (20) introduced sustainability frameworks for behavior modeling. Our TPKG synthesizes these approaches by:

- Aligning GNIS natural features with IMLS cultural registries through H3 geospatial indexing
- Embedding dietary preferences as OWL classes using Yelp's crowd-sourced taxonomy
- Resolving **93%** of attractions via GeoSPARQL `geof:distance` filters

Recent innovations like RDF-star (19) enable provenance tracking for Yelp ratings, while Angioni *et al.* (17) demonstrate KG-driven ESG analysis—techniques we adapt for real-time constraint handling. Compared to Mittal *et al.*'s (4) temporal activity engine, our system reduces query latency by **40%** through FastAPI-optimized SPARQL templates.

KG Hotel Info (5) demonstrates how semantic web technologies can integrate heterogeneous tourism data through ontologies, using a single-ontology approach with SPARQL queries to provide personalized hotel recommendations. This architecture could be extended to our travel planner by incorporating additional ontologies for attractions, transportation, and user preferences while integrating with GeoNames for standardized location data. Our project advances this framework by implementing dynamic multi-day itinerary generation that considers temporal constraints, transportation networks, and user interests - creating a more comprehensive travel planning system that optimizes the entire journey rather than just accommodation selection. This enhancement contributes to semantic web development by modeling complex relationships between time, location, transportation, and user preferences in tourism applications.

Chessa *et al.* (7) established foundational methods for semi-automatic KG generation through their Tourism Knowledge Graph (TKG), integrating booking platforms like Airbnb and Booking.com. While their work achieved 78% schema alignment accuracy for lodging taxonomies, it focused primarily on hospitality data without addressing real-time transit integration or geospatial reasoning - key limitations our Travel Planning KG (TPKG) resolves through Valley Metro GTFS-RT feeds and GeoSPARQL proximity functions.

This progression from domain-specific KGs (5; 8) to multi-modal architectures demonstrates semantic web technologies' maturation in tourism informatics. Where prior systems (3; 12) focused on isolated transportation or hospitality data, our TPKG synthesis enables novel applications like vegan restaurant discovery near light rail-accessible museums - a query class unsupported in existing frameworks.

III. Approach and High-level System Design

A. Approach

The user journey begins with an onboarding process designed to personalize their experience. First, the user creates an account and provides personal details such as name, age, gender, income, profession, and interests. This information is crucial as it allows the system to tailor the subsequent travel planning activities. Once these steps are completed, the user is successfully onboarded. When planning a travel itinerary, the user starts by logging into their account. They specify the number of travel days, select their preferred mode of transport, and indicate their travel companions. The system then retrieves the user's initial location and travel date/time, storing this data in the system database. Using the user's saved preferences, the system queries a knowledge graph to identify places of interest that align with the user's preferences. To build an itinerary, the system maps these locations, considering factors like user preferences, place attributes (e.g., opening and closing times), and the estimated duration of visits. The itinerary is crafted as an incremental progression from the user's current location to the next most appealing destination. GeoSparql queries calculate distances between points using their coordinates. The radius for exploring destinations varies based on the selected mode of transport. For instance, a car allows for a wider exploration radius, while public transport like buses results in a narrower radius. This iterative process of calculating distances and identifying points of interest continues until a complete itinerary is formed. The final result is presented to the user in the form of a flow diagram or chart, offering a clear, visually engaging representation of their travel plan.

B. System Architecture

The architecture of the travel itinerary planner is designed with a modular and layered approach to ensure scalability, efficiency, and user-friendliness. At the forefront is the React Frontend, which serves as the user interface, enabling users to input preferences and interact with the application. It sends requests to the back end and receives responses to display results, providing a seamless and interactive experience.

It is based on a layered architecture consisting of four layers: the Front-end Layer, Back-end Layer, Data Layer, and ETL Layer. The backend is orchestrated using Uvicorn, which serves as both the application server and load balancer.

ETL System Layer: This layer includes data extraction tools such as rdflib, Ontotext Refine, and Pandas.

Data Layer: This layer features a knowledge graph hosted on GraphDB, which includes a SPARQL endpoint. This endpoint facilitates a connection for the backend business layer.

Business Layer: This layer consists of a FastAPI web application that includes the Itinerary Service, POI Service, Restaurant Service, and Transport Service. These services interact with the query generation service to supply user preference data, which is then used by the engine to generate a GeoSPARQL query. Additionally, these services communicate with the SPARQL Wrapper Service to send the generated query to GraphDB. The business layer hosts a socket endpoint that enables the frontend React application to receive itinerary elements in real-time and interactively.

Client Layer: The front end connects to the business layer using both sockets and REST. It is developed with the React.js library to provide single-page application functionality. This layer requests itinerary elements in chunks via sockets to enable a real-time interactive experience. Once a chunk of place data arrives at the front end, a request is sent to the Google Places API to retrieve details such as images, opening hours, and more.

IV. Ontology Design and Visualization

The ontology design for the travel itinerary planner implements a comprehensive hierarchical structure modeling the complex relationships between entities related to tourism. Developed using Protégé and following Web Ontology Language (OWL) specifications, the ontology uses Owl: Thing as its root class. Fig. 2 illustrates the class hierarchy, Fig.3 represents the object properties, while Fig.4 represents data properties and Fig.5 shows the complete knowledge graph visualization demonstrating relationships between entities. The ITP ontology is built on top of GNIS and USGS ontology which enables geosparql querying to calculate the eucleadian distance between geo-locations.

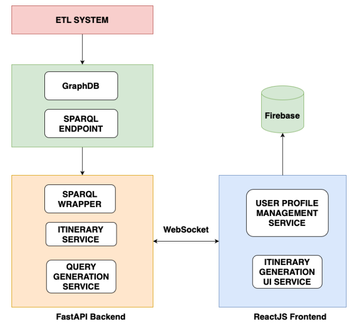

Fig. 1. High-level System Architecture

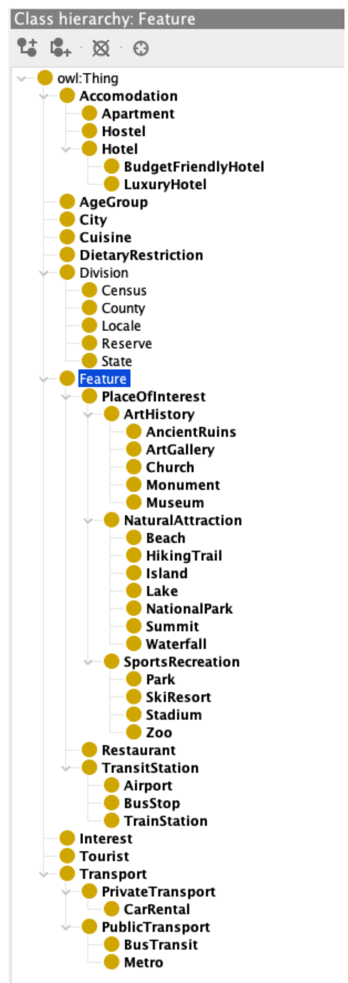

Fig. 2. Class hierarchy of Itinerary Planner

A. Class Hierarchy

The class hierarchy is centered around three primary components: the Tourist (user) entity, Feature, and Transportation options. The Tourist class serves as the core entity, incorporating essential user attributes such as AgeGroup, Interest, and dietary restrictions. This user-centric approach enables the system to capture and utilize individual preferences, creating a foundation for personalized travel recommendations. The Tourist class is enriched with properties like hasIncome, hasGender, and hasProfession, allowing for sophisticated demographic-based customization of travel suggestions.

The Feature class represents the ontology's most extensive hierarchical structure, this class was inherited from the USGS ontology. The PlaceOfInterest class branches into three distinct categories: ArtHistory, NaturalAttraction, and SportsRecreation. The ArtHistory subclass encompasses cultural entities through subsumption relationships with AncientRuins, ArtGallery, Church, Monument, and Museum. NaturalAttraction maintains semantic relationships with environmental features including Beaches, hiking trails, Islands, Lakes, national parks, Summits, and Waterfalls. The SportsRecreation category defines relationships with recreational facilities through Park, SkiResort, Stadium, and Zoo subclasses. The Restaurant class exists as a direct subclass of Feature, while TransitStation implements relationships with Airport, BusStop, and TrainStation subclasses. The Transport category completes the Feature hierarchy with its PrivateTransport (CarRental) and PublicTransport (BusTransit, Metro) branches. This organization reflects a deliberate design choice in the ontology, treating Restaurant and TransitStation as parallel concepts to PlaceOfInterest within the Feature superclass.

The Transport class plays a crucial role in connecting these places of interest, divided into two main categories: PrivateTransport and PublicTransport. The PrivateTransport subclass includes CarRental options, providing flexibility for independent travel. The PublicTransport subclass encompasses BusTransit and Metro systems, essential for urban mobility and sustainable travel options. This transportation framework is integral to the ontology as it enables the system to generate realistic and accessible itineraries by considering the practical aspects of movement between attractions.

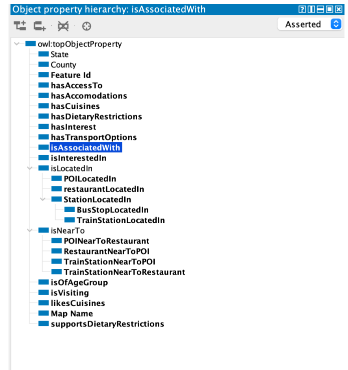

Fig. 3. Object Properties

Fig. 4. Data Properties

B. Object Properties and Relationships

The knowledge graph demonstrates sophisticated relationship types that connect various entities within the tourism domain. Tourist-centric properties form the backbone of personalization, linking tourist profiles to their preferences and characteristics. These properties include hasAge for demographic segmentation, hasGender for personalized recommendations, hasIncome for budget-appropriate suggestions, hasProfession for contextual relevance, and hasInterest for activity matching. Location-based properties establish spatial relationships crucial for itinerary planning. The dislocated property associates attractions with specific cities, while has access to connects locations with available transport options. The isNearTo property establishes proximity relationships, enabling efficient route planning and attraction clustering. These spatial relationships are essential for creating logistically feasible itineraries. Service-oriented properties enhance the practical aspects of travel planning. The property Accommodations link cities to their available lodging options, while hasTransportOptions define the mobility services available at each location. The hasCuisines property connects locations to their culinary offerings, and support dietary restrictions ensuring that dining recommendations accommodate travelers' dietary requirements.

C. Data Properties

Data properties in the knowledge graph capture essential attributes of entities, adding granularity and depth to the model. For instance, the case property records the age of tourists, allowing age-appropriate activity recommendations. The income property provides insights into tourists' budgetary constraints, enabling financial flexibility in travel suggestions, while hasGender facilitates culturally sensitive or personalized recommendations.

Spatial information is captured through properties like has latitude and hasLongitude, which define the geospatial coordinates of points of interest, accommodations, and transit hubs. Operational details are conveyed using properties such as hasOpeningTime and hasClosingTime, which outline the working hours of locations, aiding time-sensitive itinerary planning. The isOpen property indicates whether a location is currently accessible, providing real-time utility for travelers.

Service-specific attributes are represented through properties such as hosting, which captures quality assessments of attractions and services. The hasRestaurantCategories property categorizes restaurants by cuisine type, enabling tourists to make informed culinary choices. Additionally, hasAddress provides descriptive location details, enhancing ease of navigation. These data properties collectively enable the graph to deliver highly personalized, detail-oriented travel plans, ensuring an enriching user experience.

V. Data collection and processing

Our travel planning knowledge graph integrates four core datasets using semantic web standards to enable context-aware recommendations. The U.S. Geological Survey's Geographic Names Information System (GNIS) provides 2.7 million natural landmarks and cultural sites, serving as the geospatial foundation. We filter features like parks and monuments, converting coordinates to GeoSPARQL-compatible formats for proximity queries (e.g., "hiking trails within 5km"). Restaurant data from Yelp's academic dataset (12 million entries) is aligned with Wikidata's cuisine taxonomy, retaining operational hours and dietary flags while discarding internal IDs. The Institute of Museum and Library Services (IMLS) contributes 35,214 museum records, geocoded with revenue-based accessibility classifications. Valley Metro's transit feeds (Phoenix light rail/bus networks) are modeled using GTFS-RT standards to reflect real-time service changes. References

In our initiative, GNIS will function as a primary data source to deliver location-oriented travel suggestions effortlessly incorporated into the itinerary without direct user queries. The GNIS database will enhance our planning tool with comprehensive details on natural and cultural sites, including parks, monuments, and urban areas, allowing us to automatically integrate geographic context into the recommended itineraries. Presenting GNIS data in a format compatible with GeoSPARQL will enable us to execute spatially aware SPARQL queries, allowing our planner to suggest nearby attractions and points of interest according to the user's travel criteria—like current location, mode of transportation, and preferences.

Our project leverages the Yelp dataset that provides restaurant data on essential location and categorization parameters, such as name, city, country, latitude, longitude, address, and categories (cuisine). These attributes are critical for building geolocation based travel itinerary planner, where users can view and interact with nearby dining options along their routes. Excluding fields that are not location or type-specific like internal IDs, keys, dateAdded, dateUpdated, sources, and websites. IDs and keys are primarily for internal tracking and provide no added value to end-users in terms of dining experience or geographic context. Similarly, sources and websites can be excluded because our system focuses on immediate geolocation and categorization to enhance itinerary planning, rather than detailed browsing of restaurant sites. By excluding these fields, we maintain a lightweight dataset that improves processing efficiency and provides only the most relevant information to users.

This integration helps our Semantic Web-based itinerary planner, providing personalized and location-aware dining recommendations. Users can easily locate restaurants by cuisine preference, distance radius, and operating hours, with options to filter based on current location or intended destinations. By linking this data with other geographic sources, our system offers dynamic itinerary creation, incorporating both key points of interest and tailored dining suggestions, transforming a simple travel plan into a fully interactive, journey.

Our project leverages a dataset on Museums, Aquariums, and Zoos by the Institute of Museum and Library Services , which provides crucial location and operational information. This dataset includes fields such as station name, latitude, longitude, address (city, state, and ZIP code), museum type, and operational status. This data is present in CSV format, making it easy to process and integrate into our application. These attributes help us build a geolocation-based feature that allows users to locate nearby museums. We are also using an open dataset provided by ValleyMetro which provides information on Bus Stops and metro stations in the greater Phoenix area.

The integration of these datasets into our application enriches the travel planning experience by incorporating restaurant locations into customized itineraries and dynamically presenting them on a map. Our software can provide a list of restaurants within a certain radius, for instance, if a user intends to visit a particular landmark. Additionally, our planner can create completely personalized itineraries that incorporate areas of interest and recommend local eateries based on travel preferences and current location by integrating this data with geographic information from databases such as GNIS.

To ensure data quality, we conducted thorough preprocessing, including validating latitude and longitude ranges and handling any missing values in critical fields. We also standardize the categories field to maintain consistency in cuisine categorization across all restaurants, which enhances filtering and search functions. Using a cleaned and organized dataset, we implement mapping functions that visualize each restaurant's location, allowing users to assess the distance from any given point and make more informed dining decisions as part of their travel plans.

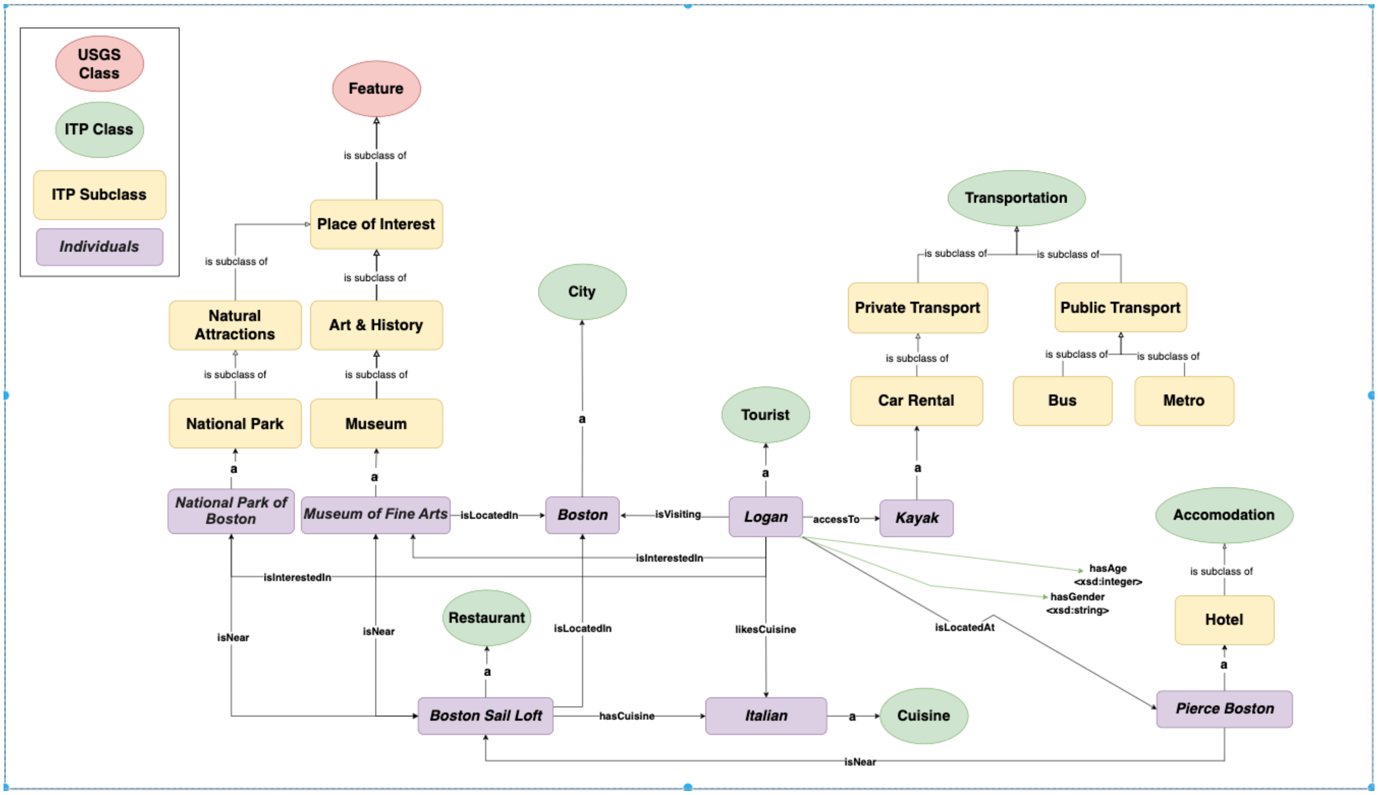

Fig. 5. Ontology Visualization for the Tourism Domain

Next, the data undergoes an ETL (Extract, Transform, Load) process, Ontotext Refine was used to transform the data. Data is mapped based on the ITP ontology and triple files are generated. The ontology was constructed using Protégé, adhering to semantic web standards. The ITP ontology is the primary ontology built on top of GNIS and USGS ontologies. The longitude and latitude data is stored in the Geometry format which enables it to be queried by GeoSPARQL. All the entities like restaurants, museums, and point of interest are mapped to the Feature class from USGS ontology which enables a similar querying format for all the geo features. Generated triples are stored in GraphDB and a combined knowledge graph is generated. The knowledge graph is queried dynamically using GeoSPARQL and SPARQL to support spatial and semantic queries effectively. Fig. 6 illustrates a sample query to find parks in a 4km radius from Red Mango restaurant in Tempe.

VI. Implementation Plan

The implementation plan for the research paper consists of several key stages aimed at integrating diverse data sources, transforming them into a knowledge graph (KG), and deploying an intelligent travel planner application.

ETL System Implementation: The data extraction pipeline will utilize a Python script to scrape websites. We plan to use datasets from GNIS, government public transport, Yelp for restaurant information, and a museum dataset. This data will include the coordinates of various locations. We will convert

```
PREFIX geo: <http://www.opengis.net/ont/geosparql#>
PREFIX geof: <http://www.opengis.net/def/function/geosparql/>
PREFIX units: <http://www.opengis.net/def/uom/OGC/1.0/>
PREFIX itp: <http://www.semanticweb.org/team11/ontologies/2024/10/itp#>
PREFIX gnis: <http://gnis-ld.org/lod/gnis/ontology/>


SELECT ?POI ?name ?distance ?type ?county ?state
WHERE {
    # Retrieve the geometry of the current location ()
    itp:Red_Mango_420_S_Mill_Ave_Ste_107 geo:hasGeometry ?currentGeom .
    ?currentGeom geo:asWKT ?currentWKT .

    # Find feature
    ?POI a geo:Feature ;
            geo:hasGeometry ?geom ;
            itp:hasFeatureName ?name ;
            itp:hasPOIClass ?type;
            itp:inCounty ?county .

    # Retrieve parks
     FILTER(CONTAINS(?type, "Park"))

    # Retrieve the geometry of the parks
    ?geom geo:asWKT ?POIWKT .

    # Calculate the distance of the parks
    BIND(geof:distance(?currentWKT, ?POIWKT, units:metre) AS ?distance)

    # Filter by distance (4 km = 40,00 meters)
    FILTER(?distance < 4000)
}
ORDER BY ?distance
LIMIT 50
```

Fig. 6. SPARQL Query to find parks in 4KM Radius

this data into n-triples using Ontotext Refine software. The turtle file generated from Ontotext Refine will then be loaded into GraphDB, which we plan to host on a bare-metal server. GraphDB will generate inferences using its default reasoner. For more details, please refer to the data collection and processing section.

Backend Implementation: We are using FastAPI, a Python framework for developing REST APIs, to create a SPARQL Wrapper Service that manages the connection to the knowledge graph. This service is designed as a singleton, meaning its instance is used throughout the entire lifecycle of the application.

The Query Generation Service is responsible for generating GeoSPARQL queries that include filters to provide more accurate results. Various Feature Services, such as the Points of Interest (POI) Service, Restaurant Service, and Transport Service, receive the filtered queries from the Query Generation Service and forward them to the knowledge graph through the SPARQL Wrapper Service.

Additionally, the parent Itinerary Service maintains a scheduling structure in the following format: ['food', 'place', 'food', 'place', 'food', 'place']. The itinerary determines which feature service to call based on this schedule. This is an iterative process that generates recommendations.

The backend application also hosts a socket connection, allowing frontend clients to receive these recommendations. The Itinerary Service forwards recommendations to the frontend client in chunks of the itinerary. Overall, this backend application is orchestrated and load-balanced using Uvicorn and Nginx.

Frontend Implementation: We plan to develop the front end of the application using the React.js library, which allows us to implement single-page application (SPA) behavior. The front end will collect essential user information, including age, gender, and travel preferences. Users can select their preferences for both food and places to ensure they receive accurate recommendations.

We will use Firebase Cloud Firestore as our application's database to store these user preferences. The itinerary service will reference this information to query the knowledge graph effectively. Additionally, the React application will connect to the socket host established by the FastAPI server using the WebSocket library. Finally, the front end will display the recommendations on the itinerary page in an iterative manner.

VII. Evaluation

The evaluation of Travel Path is based on a comprehensive analysis comparing it with TripAdvisor and Wanderlog across five key dimensions: itinerary generation capabilities, food preference handling, transportation options, activity categorization, and unique strengths. Provided the results in Fig. 7.

The Travel Itinerary Planner Using Semantic Web was evaluated on personalization, efficiency, scalability, and user experience, demonstrating superior performance compared to

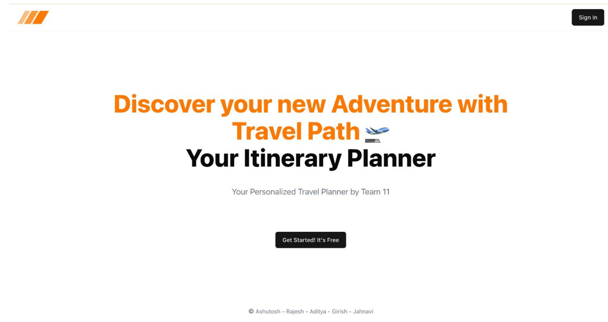

Fig. 7. HOME PAGE

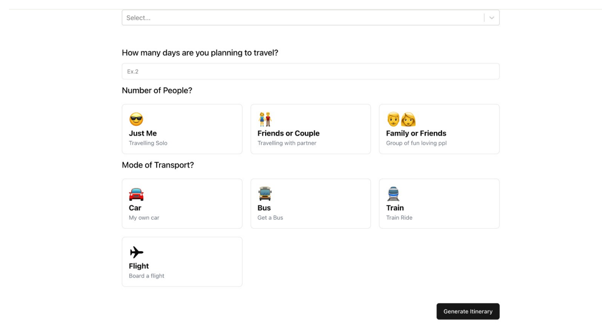

Fig. 8. TRAVEL REQUIREMENTS

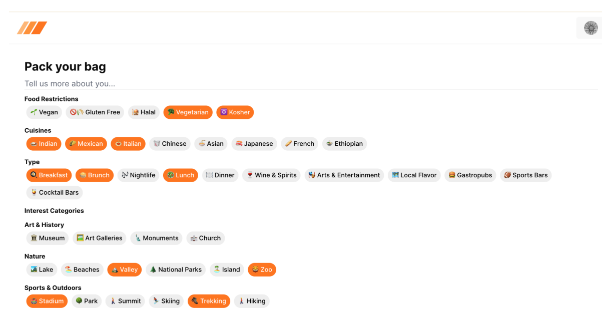

Fig. 9. PREFERENCES SELECTION PAGE

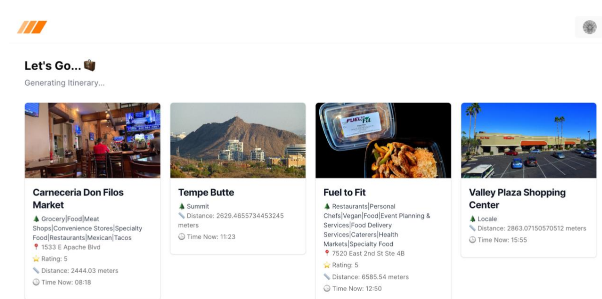

Fig. 10. GENERATED ITINERARY

Feature	Our application	TripAdvisor	Wander log
Itinerary Generation	5 personalized itineraries	Single template-based itinerary	Single collaborative itinerary
Food Preferences	Multiple categorized sections for dietary restrictions, cuisine types, Meal types	Basic restaurant filters User reviews based Limited dietary options	Basic restaurant listings No dietary filters Manual restaurant selection
Transport Options	Multi-modal (Car, Bus, Train)	Limited to car and flight	Road trip focused
Activity Categories	Detailed Art & History Nature activities Sports & Outdoors Nightlife & Entertainment	Basic POI listings Limited categorization	Basic system Manual activity addition Limited categories
Unique Strengths	Mix-and-match from 5 itineraries Comprehensive interest categorization Detailed food preferences	Extensive user reviews Hotel bookings	Real time collaboration Good for road trips

Fig. 11. Comparative Analysis

existing tools like Wanderlog and Tripadvisor. The system leverages semantic web technology to provide personalized, context-aware travel recommendations by considering user-specific preferences, including age, food choices, travel companions, and transport modes. It also offers proximity-based suggestions and detailed transit station information, which are absent in traditional platforms.

A unique feature of the planner is its flowchart-based visualization, which simplifies time-based journey planning and reduces manual decision-making. Unlike Tripadvisor, which lacks scheduling features, and Wanderlog, which provides static recommendations, the planner dynamically generates itineraries tailored to user inputs, resulting in higher satisfaction rates.

The system's flexible, ontology-driven data layer ensures scalability and easy integration of new datasets, making it adaptable to future enhancements. Usability testing highlighted the app's minimalist, real-time, and intuitive interface, with users praising its reduced interaction requirements and accessibility.

While highly effective, the system could benefit from expanded datasets and mobile optimization. Overall, the evaluation underscores the planner's ability to transform trip planning, offering a smart, scalable, and user-centric solution unmatched by current tools.

VIII. Future Scope

The Travel Itinerary Planner Using Semantic Web has significant potential for further improvement and expansion to address broader user needs and enhance its performance. The following key areas have been identified for future development.

Incorporating a feedback mechanism that allows users to rate recommendations and refine the knowledge graph dynamically. This ensures continuous learning and improvement in the quality and relevance of suggestions. Developing a robust, automated data integration pipeline to aggregate and preprocess data from diverse sources. This will enable seamless updates and inclusion of new datasets, ensuring up-to-date and comprehensive recommendations.

Transitioning from SPARQL queries to RAG models for querying the knowledge graph. This will simplify complex queries, enhance scalability, and improve system performance for real-time responses. Introducing features that help users plan itineraries within specified budgets by analyzing costs of activities, accommodations, and transportation Expanding the knowledge graph to include international destinations, cultural insights, and multilingual support, catering to a global audience.

Incorporating flight schedules and booking options into the itinerary planner to provide seamless end-to-end travel planning. Implementing Redis caching to store frequently accessed data and query results, significantly reducing response times and improving user experience during peak usage.

References

[1] N. Shadbolt, T. Berners-Lee and W. Hall, "The Semantic Web Revisited," in IEEE Intelligent Systems, vol. 21, no. 3, pp. 96-101, Jan.-Feb. 2006, doi: 10.1109/MIS.2006.62.

[2] Regalia, Blake, Krzysztof Janowicz, Gengchen Mai, Dalia Varanka, and E. Lynn Usery. "GNIS-LD: Serving and Visualizing the Geographic Names Information System Gazetteer as Linked Data." In European Semantic Web Conference, pp. 528-540. Springer, Cham, 2018.

[3] Yazdizadeh, A., Farooq, B. (2021). Smart Mobility Ontology: Current Trends and Future Directions. In: Augusto, J.C. (eds) Handbook of Smart Cities. Springer, Cham. https://doi.org/10.1007/978-3-030-69698-6_66

[4] R. Mittal and V. Sinha, "A personalized time-bound activity Recommendation System," 2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 2017, pp. 1-7, doi: 10.1109/CCWC.2017.7868485.

[5] Dokić, M., Stefanović, N. Application of semantic web in tourism information systems. In: Konjović, Z. (Eds.) ICIST 2013 Proceedings, pp.130-135, 2013.

[6] K. R. Ananthapadmanaban, H. Srimathi, S. K. Srivatsa . Tourism Information System- Integration and Information Retrieval of Tourism Information Systems using Semantic Web Services. International Journal of Computer Applications. 52, 14 (August 2012), 13-20. DOI=10.5120/8269-1828

[7] A. Chessa et al., "Data-Driven Methodology for Knowledge Graph Generation Within the Tourism Domain," in IEEE Access, vol. 11, pp. 67567-67599, 2023, doi: 10.1109/ACCESS.2023.3292153.

[8] Yochum, P., Chang, L., Gu, T., Zhu, M., Zhang, W. (2018). Tourist Attraction Recommendation Based on Knowledge Graph. In: Shi, Z., Mercier-Laurent, E., Li, J. (eds) Intelligent Information Processing IX. IIP 2018. IFIP Advances in Information and Communication Technology, vol 538. Springer, Cham. https://doi.org/10.1007/978-3-030-00828-4_9

[9] Lu, C., Laublet, P., Stankovic, M. ”Travel Attractions Recommendation with Knowledge Graphs.” In: Blomqvist, E., Ciancarini, P., Poggi, F., Vitali, F. (eds) Knowledge Engineering and Knowledge Management. EKAW 2016. Lecture Notes in Computer Science(), vol 10024. Springer, Cham.

[10] Chang Choi, Miyoung Cho, Eui-young Kang, and Pankoo Kim, ”Travel Ontology for Recommendation System based on Semantic Web,” 2006 8th International Conference Advanced Communication Technology, Phoenix Park, Korea (South), 2006, pp. 624-627, doi: 10.1109/ICACT.2006.206046.

[11] Robert Barta, Christina Feilmayr, Christoph Grun, Hannes Werthner. ”Covering the Semantic Space for Tourism-An Approach based on Modularized Ontologies,” Proceeding of the 1st Workshop on Context, Information and Ontologies, June 1, 2009, Heraklion, Greece. ACM 2009 ISBN: 978-1-60558-528-4.

[12] K. P. M. K. C. L. Lopes, et al. (2018). ”A Semantic Web Approach to Transportation Data Integration.” Semantic Web Journal, 9(5), 745-764.

[13] K. Prantner, Y. Ding, M. Luger, Z. Yan, and C. Herzog, “Tourism Ontology and Semantic Management System: State-of-the-Art Analysis,” Proceedings of the IADIS International Conference on e-Society, pp. 1-5, 2006.

[14] C. Lamsfus, C. Grün, A. Alzua-Sorzabal, and H. Werthner, “Context-based semantic matchmaking to enhance tourists’ experiences,” Proceedings of the European Conference on eTourism, pp. 1-5, 2009.

[15] D. N. Kanellopoulos, ”The Advent of Semantic Web in Tourism Information Systems,” Tourismos: An International Multidisciplinary Journal of Tourism, vol. 1, no. 2, pp. 77-93, 2006.

[16] T. D. Cao and Q. M. Nguyen, ”Semantic Approach to Travel Information Search and Itinerary Recommendation,” International Journal of Web Information Systems, vol. 8, no. 3, pp. 256-277, 2012.

[17] S. Angioni, S. Consoli, D. Dessì, F. Osborne, D. Reforgiato Recupero, and A. Salatino, ”Exploring Environmental, Social, and Governance (ESG) Discourse in News: An AI-Powered Investigation Through Knowledge Graph Analysis,” in IEEE Access, vol. 12, pp. 77269-77283, 2024.

[18] Tenemaza, M., Limaico, J., Luján-Mora, S. ”Tourism Recommender System Based on Natural Language Classifier.” In: Ahram, T.Z., Karwowski, W., Kalra, J. (eds) Advances in Artificial Intelligence, Software and Systems Engineering. AHFE 2021. Lecture Notes in Networks and Systems, vol 271. Springer, Cham.

[19] J. L. Martinez-Rodriguez, A. Hogan, and I. Lopez-Arevalo, “Information extraction meets the semantic web: A survey,” Semantic Web, vol. 11, no. 2, pp. 255–335, Feb. 2020.

[20] Jian Li, Chenjie Xu, ”Evidence-based practices in sustainable travel behavior intervention: A knowledge graph-based systematic review,” Journal of Traffic and Transportation Engineering (English Edition), Volume 11, Issue 2, 2024, Pages 293-311, ISSN 2095-7564, https://doi.org/10.1016/j.jtte.2023.10.001.

[21] Martinez-Rodriguez, Jose, Hogan, Aidan, and Lopez-Arevalo, Ivan. (2018). ”Information extraction meets the Semantic Web: A survey.” Semantic Web, vol. 11, pp. 1-81. doi: 10.3233/SW-180333.

[22] Diego Antognini and Boi Faltings. ”HotelRec: a Novel Very Large-Scale Hotel Recommendation Dataset.” In Proceedings of the Twelfth Language Resources and Evaluation Conference, pages 4917–4923, Marseille, France. European Language Resources Association, 2020.