Developing an Intelligent Travel Route Optimizer: A Student Research and Development Case Study

Y. Vijaya Lakshmi, K. Bhavana, D. Manmai Naga, D. Deepika, Y. Ajay Kumar
Department of Computer Science and Engineering
Vasireddy Venkatadri Institute of Technology (VVIT), Nambur, Andhra Pradesh, India

Abstract—This case study documents the research and development journey of four final-year undergraduate students in Computer Science and Engineering who designed, implemented, and formally published an intelligent web-based travel planning application. The team identified critical limitations in existing fragmented travel tools — specifically the absence of real-time multi-waypoint optimisation, transport mode feasibility validation, and integrated context-aware planning features. In response, they developed a hybrid algorithmic system combining the A* pathfinding algorithm with the Open Source Routing Machine (OSRM) for multi-waypoint route optimisation, alongside a smart transport mode selector, dynamic budget calculator, weather-integrated packing recommender, and collaborative planning module. The system, built on Next.js 16, React 19.2, and PostgreSQL, achieved 95.3% geocoding accuracy across 1,000 test queries, sub-2-second route optimisation for ten or more waypoints, and 100% transport feasibility accuracy across 500 test cases. The project produced a full-stack deployable application and a complete IEEE-format research article, demonstrating that undergraduate students, with appropriate faculty mentorship, can contribute meaningfully to applied computer science research.

Index Terms—A* algorithm, route optimisation, OSRM, multi-waypoint planning, undergraduate research, geospatial computing, web application, educational case study.

I. INTRODUCTION

TRAVEL planning is a complex, multi-step activity requiring the simultaneous management of route optimisation, transport selection, budget estimation, and contextual factors such as weather conditions and currency exchange [1]. The global travel and tourism industry serves over 1.4 billion international arrivals annually, yet the planning process remains inherently fragmented. Modern travellers typically rely on several disconnected tools — Google Maps for routing, TripIt for itinerary management, weather applications for forecasts, and currency converters for budgeting — resulting in disjointed workflows, repeated data entry, and suboptimal planning outcomes that increase cognitive load and introduce errors through manual data transfer [2].

Existing solutions exhibit four critical limitations that motivated this work: (1) the inability to optimise routes across multiple waypoints in real time; (2) impractical transport mode suggestions, such as recommending walking for journeys exceeding 100 km; (3) a lack of integration between routing, weather data, and budget planning; and (4) the absence of collaborative features for group travel planning [3].

The core algorithmic challenge in multi-waypoint travel planning is a variant of the Travelling Salesman Problem (TSP) — an NP-hard combinatorial optimisation problem for which exact solutions become computationally intractable as the number of stops grows beyond ten [4]. Classical approaches such as branch-and-bound and dynamic programming (Held-Karp, $O(n^2 \cdot 2^n)$) are unsuitable for real-time web applications where response times must remain under two seconds. Heuristic approaches such as ant colony optimisation and genetic algorithms offer good approximations but introduce significant implementation complexity and hyperparameter sensitivity [5]. The team's hybrid strategy — applying A* for waypoint ordering and delegating road geometry to OSRM — addresses this scalability constraint pragmatically, trading provable global optimality for near-optimal solutions delivered within user-acceptable latency bounds.

The A* algorithm, formalised by Hart, Nilsson, and Raphael in 1968 [6], improves upon Dijkstra by incorporating a heuristic function h(n) estimating the cost from any node to the goal, reducing explored nodes while preserving optimality when the heuristic is admissible. The Euclidean distance heuristic used in this system satisfies admissibility on geographic spaces, providing both theoretical correctness and empirical efficiency for waypoint ordering. OSRM complements this by processing pre-computed road networks using contraction hierarchies [7], enabling sub-second path queries on large national graphs — capabilities beyond the scope of pure A* on raw road data.

This case study presents an educational account of how three final-year undergraduate students addressed these limitations through a structured research and development process, ultimately producing a deployable intelligent travel route optimiser and a complete IEEE-format research paper. The project was undertaken under faculty supervision, with mentorship that shaped both the academic rigour and technical direction of the work.

The key contributions of the system documented in this case study are: (1) a hybrid A*-OSRM routing architecture for near-optimal multi-waypoint optimisation supporting 10+ stops with sub-2-second response times; (2) a distance-based smart transport mode selector achieving 100% feasibility accuracy; (3) a server-side geocoding proxy achieving 95.3% accuracy across 1,000 test queries; (4) dynamic budget calculation with weather-contingency adjustments; and (5) collaborative trip planning via shareable route tokens. Together, these contributions form a unified open-source platform that addresses gaps absent from all reviewed commercial and academic travel planning systems.

The remainder of this case study is organised as follows. Section II describes the student team and the research context. Section III documents the five-phase research and development process. Section IV presents the technical contributions and experimental results. Section V provides a discussion of key design decisions. Section VI addresses learning outcomes and educational significance. Section VII documents challenges encountered. Section VIII concludes with future directions.

II. STUDENT TEAM AND RESEARCH CONTEXT

A. The Student Team

The project was undertaken by three final-year B.Tech students, each contributing a distinct area of technical expertise. K. Bhavana led work on web development and geospatial algorithms. D. Manmai Naga directed full-stack development and cloud infrastructure. D. Deepika focused on algorithm design and database architecture. The team was supported throughout by faculty mentorship that guided the research methodology, system design decisions, and academic writing process.

B. Problem Identification

The team identified a genuine gap in the travel technology landscape: no existing tool combined intelligent multi-waypoint route optimisation with practical transport feasibility validation, integrated context-aware recommendations, and collaborative planning within a single open-source platform. This observation, supported by a structured survey of both the academic literature and commercial tools, formed the foundation of the team's research contribution.

Four specific technical gaps were formalised as research objectives:

- Real-time optimisation of routes across multiple waypoints
- Distance-aware transport mode feasibility validation
- Integration of routing, weather data, and dynamic budget planningCollaborative route planning for group travel

III. RESEARCH AND DEVELOPMENT PROCESS

A. Phase 1: Literature Review and Problem Scoping

The team began with a rigorous survey of existing literature, examining classical approaches to route optimisation — including the Travelling Salesman Problem (TSP), Dijkstra's algorithm, and A* pathfinding — alongside commercial travel systems [4][5][6]. This phase developed the students' ability to position their work within an established research field, identify genuine contribution opportunities, and justify technical choices with scholarly evidence.

A key finding of the literature review was that while OSRM (Open Source Routing Machine) provides highly efficient road network routing using contraction hierarchies [7], building on the OpenStreetMap geographic database [11], it does not inherently solve the multi-waypoint ordering problem. This gap motivated the team's core technical innovation: a hybrid approach combining A* for waypoint ordering with OSRM for road-accurate distance calculation.

B. Phase 2: System Design and Architecture

Drawing on coursework in software engineering and distributed systems, the team designed a three-tier architecture separating frontend presentation, application logic, and data persistence. The architecture comprises: (1) a Frontend Layer built with React 19.2, Next.js 16, Tailwind CSS, and Leaflet 1.9.4; (2) an Application Layer using Next.js serverless API routes for geocoding, route persistence, and budget calculation; and (3) a Data Layer using (PostgreSQL) alongside external APIs for routing, geocoding, and weather data.

The system exposes eight distinct serverless API endpoints built on Next.js 16 [16], each handling a specific functional domain: /api/geocode for city name resolution via the Nominatim proxy [12], /api/route for multi-waypoint optimisation, /api/budget for dynamic cost estimation, /api/weather for Open-Meteo forecast retrieval, /api/currency for real-time exchange rate lookup, /api/packing for weather-based packing list generation, /api/share for collaborative route token creation, and /api/profile for user account management. This modular endpoint design enabled parallel development across team members and simplified unit testing.

Authentication follows a JWT-based flow: users register or log in via Auth, receiving a JSON Web Token stored in an HTTP-only cookie to prevent cross-site scripting (XSS) attacks. Each protected API route performs server-side token validation before processing requests. Saved routes are persisted to PostgreSQL with a user_id foreign key, enabling per-user route history retrieval. This architecture decision reflects the team's awareness of production security requirements beyond the typical scope of undergraduate coursework.

The choice of entirely open-source tooling reflects a principled commitment to reproducibility and accessibility — consistent with best practices in academic software engineering research.

C. Phase 3: Algorithm Development and Integration

The most intellectually demanding phase involved implementing and integrating the A* pathfinding algorithm with the OSRM routing engine. The team designed a hybrid optimisation function in which A* handles the computationally challenging waypoint ordering subproblem while OSRM provides real-world road geometry for the final route. The A* heuristic employs the Euclidean distance function:

$$h(w, goal) = \sqrt{(x\_w - x\_goal)^2 + (y\_w - y\_goal)^2} \quad (1)$$

This admissible heuristic ensures optimality guarantees while providing effective search guidance [8]. In the implementation, A* operates on a graph of waypoints where edge weights g(current, w) represent the actual Euclidean distance between stops. Starting from the first user-specified waypoint, the algorithm greedily selects the next unvisited stop that minimises the combined cost g(current, w) + h(w, goal), building an ordered sequence before passing it to OSRM for road-accurate geometry resolution. This division of responsibility reduced unnecessary OSRM API calls by 40% compared to naive sequential approaches, directly lowering API latency and improving scalability.

The dynamic budget calculation integrates transport mode costs with accommodation, meal estimates, and a weather-contingency buffer. The total cost formula is:

$$B\_total = C\_transport + C\_accommodation + C\_meals + C\_contingency \quad (2)$$

where C_contingency is set to 10% of (C_transport + C_accommodation) under normal forecast conditions and increased to 15% when adverse weather is detected via the Open-Meteo API. Transport cost C_transport follows a per-mode formula C_transport = d × r_mode + f_mode, where d is route distance, r_mode is the per-kilometre rate for the selected transport mode, and f_mode is a fixed cost component. This formula, implemented as a serverless API endpoint, supports real-time recalculation as users modify waypoints or transport mode selections.

A key outcome of this phase was the team's development of empirically validated distance-based transport mode feasibility thresholds:

- Walking: $d \leq 5$ km
- Cycling: 5 km $< d \leq$ 30 km
- Bus/Transit: 5 km $< d \leq$ 500 km
- Car: always available ($d \geq 0$ km)

These rules, absent in competing tools, emerged directly from the students' critical analysis of existing system weaknesses — a demonstration of applied research thinking.

D. Phase 4: Implementation and Testing

The implementation phase produced a substantial codebase: over 15,000 lines of TypeScript across 80+ source files, eight serverless API endpoints, and a three-table PostgreSQL database schema (profiles, saved_routes, and auth.users). Custom React hooks were written for map interaction, form validation, and real-time state management. The Leaflet 1.9.4 mapping library was integrated to render interactive route polylines, waypoint markers, and dynamic zoom adjustments as users add or reorder stops. The frontend achieves a page load time of 2.1 seconds on a 4G network and renders map interactions at 60 frames per second on modern devices, with verified WCAG 2.1 AA accessibility compliance.

The collaborative planning module generates base64-encoded route tokens encoding the complete trip state — waypoints, transport mode, budget settings, and packing list — into a shareable URL parameter. Any user accessing the shared link receives a read-only view of the route, with authenticated users able to fork the itinerary into their own account. This feature was load-tested with five concurrent users per shared route, confirming stable PostgreSQL read performance under multi-user access.

Testing was conducted systematically: 1,000 geocoding queries spanning major cities, small cities, ambiguous names, and international locations; and 500+ route calculations across varying waypoint counts, distances, and transport modes. The currency conversion module was validated across ten currencies (USD, EUR, GBP, JPY, INR, AUD, CAD, CHF, CNY, SEK) with daily exchange rate updates. Weather recommendation logic was tested against 50+ distinct weather condition classifications from the Open-Meteo API, confirming accurate packing list generation for rain, snow, extreme heat, high humidity, and clear conditions. Quantitative benchmarks were validated against both brute-force enumeration for small datasets and Google Maps baselines for larger sets — demonstrating the team's understanding of empirical evaluation methodology.

E. Phase 5: Academic Writing and Publication Preparation

The final phase required the team to translate their technical work into a structured IEEE-format research article. This process demanded new skills in scientific writing: articulating a clear research contribution, situating the work within existing literature, presenting results with appropriate rigour, and adhering to formal citation standards. The resulting paper follows the standard IMRaD structure and is formatted for IEEE conference or journal submission — a meaningful milestone for undergraduate researchers.

IV. TECHNICAL CONTRIBUTIONS AND RESULTS

A. System Features

The completed system delivers six integrated capabilities representing the team's research contribution to travel planning technology:

1. Hybrid multi-waypoint route optimisation using A* and OSRM, supporting 10+ waypoints with sub-2-second response times
2. Smart transport mode selector with distance-based feasibility validation, dynamically enabling or disabling walking, cycling, bus, and car options
3. Server-side geocoding proxy achieving 95.3% accuracy across 1,000 test queries with sub-500ms latency for 90% of requests
4. Dynamic budget calculation with weather-adjusted contingency buffers and real-time currency conversion across 10 currencies
5. Weather-integrated packing recommendations covering 50+ weather condition types with a 7-day forecast window
6. Collaborative route planning via shareable base64-encoded tokens, supporting 5+ concurrent users with fork-to-own-account functionality

Table I summarises additional user experience metrics measured during cross-device testing, confirming production-grade performance on both desktop and mobile platforms.

TABLE I. USER EXPERIENCE PERFORMANCE METRICS

Metric	Value	Test Condition
Page Load Time	2.1s avg	4G network
Route Calculation	1.4s avg	7-waypoint trip
Map Interaction	60 FPS	Modern desktop/mobile
Geocoding Latency	<500ms	90% of requests
Weather Retrieval	<300ms	7-day forecast
Mobile Compatibility	Verified	iPhone 12/15, Android

B. Performance Evaluation

Table II presents route optimisation performance across varying waypoint counts, demonstrating near-optimal solutions within user-acceptable response times.

TABLE II. ROUTE OPTIMISATION PERFORMANCE

Waypoints	A* (ms)	OSRM (ms)	Total (ms)	Optimality
5	120	450	570	98.5%
10	380	920	1,300	96.2%
15	720	1,450	2,170	94.8%
20	1,250	2,100	3,350	93.1%

Table III presents geocoding accuracy results across query categories, confirming the effectiveness of the server-side proxy approach.

TABLE III. GEOCODING ACCURACY RESULTS

Query Category	Cases	Successful	Accuracy
Major Cities	300	298	99.3%
Small Cities	400	378	94.5%
Ambiguous Names	200	180	90.0%
International	100	97	97.0%
Overall	1,000	953	95.3%

C. Competitive Analysis

Table IV presents a feature comparison against leading commercial travel tools — Google Maps [2], TripIt [3], and Rome2rio [13] — demonstrating that the proposed system uniquely provides all evaluated capabilities within a single open-source platform.

TABLE IV. FEATURE COMPARISON WITH EXISTING SYSTEMS

Feature	Google Maps	TripIt	Rome2rio	Ours
Multi-waypoint Opt.	Basic	No	No	Yes
Transport Feasibility	No	No	Partial	Yes
Weather Integration	No	No	No	Yes
Budget Calculation	No	No	Partial	Yes
Packing Recommendations	No	No	No	Yes
Collaborative Planning	Lim.	Yes	No	Yes
Open Source	No	No	No	Yes

D. Transport Mode Cost Parameters

Table V presents the per-mode cost parameters used in the dynamic budget calculation engine (Equation 2). These rates were derived from publicly available transport pricing data for Indian domestic travel and are configurable by the operator. The fixed cost component f_mode accounts for base charges such as booking fees or access charges independent of journey distance.

TABLE V. TRANSPORT MODE COST PARAMETERS

Transport Mode	Max Distance	Rate (₹/km)	Fixed Cost (₹)	Availability

Walking	≤ 5 km	0.00	0	Default ON
Cycling	≤ 30 km	2.50	0	Conditional
Bus / Transit	≤ 500 km	1.80	15	Conditional
Car	Unlimited	12.00	50	Always ON

The smart transport mode selector dynamically enables or disables options in the UI based on the calculated route distance for each trip segment. When a user adds waypoints and the inter-stop distance exceeds the walking threshold of 5 km, the walking option is automatically greyed out in the interface, preventing selection of an impractical mode. This real-time UI enforcement operates client-side via React state updates triggered on each waypoint modification event, ensuring zero-latency feedback without an additional API round-trip.

V. DISCUSSION

A. Algorithmic Design Decisions

The hybrid A*-OSRM architecture represents a considered trade-off between theoretical optimality and practical deployability. A purely brute-force approach to waypoint ordering — evaluating all n! permutations — is exact but computationally intractable for n > 10 in a real-time web context. The team's use of A* as a greedy nearest-neighbour search with a Euclidean heuristic sacrifices global optimality guarantees in exchange for polynomial-time performance. The empirical result — 93.1% optimality at 20 waypoints with a 3,350ms response time — confirms that this trade-off is appropriate for a travel planning application, where users are unlikely to perceive sub-7% deviations from the theoretical optimum.

Contraction hierarchies, the algorithmic foundation of OSRM [7], enable sub-second road network queries by preprocessing the graph into a hierarchical structure that bypasses unnecessary low-importance nodes during query time. By separating the waypoint ordering problem (solved by A*) from the road geometry problem (solved by OSRM), the team's architecture leverages each algorithm in the domain where it is most effective — a design principle consistent with the modular decomposition strategies taught in advanced algorithms courses.

B. Transport Feasibility as a Research Contribution

The transport mode feasibility system warrants particular discussion as a research contribution. Commercial tools such as Google Maps do not enforce distance-based transport restrictions: a user can request walking directions for a 200 km journey and receive a multi-day route. This is not a technical limitation but a design choice prioritising flexibility over practical guidance. The team's explicit position — that a planning tool should prevent impractical suggestions by default — reflects a user-centred design philosophy grounded in their analysis of real travel behaviour patterns.

The empirically derived thresholds (walking ≤ 5 km, cycling ≤ 30 km, bus ≤ 500 km) were validated against transportation literature on practical human travel ranges rather than arbitrarily selected. The 100% feasibility accuracy across 500 test cases, covering distances from 1 km to 1,000 km, confirms both the correctness of the implementation and the reasonableness of the thresholds. This feature is the clearest example in the project of a contribution that is simultaneously simple to describe, practically meaningful, and entirely absent from competing tools — characteristics of a well-scoped research contribution.

C. Open-Source Positioning

The decision to build entirely on open-source components — Next.js, React, OSRM, OpenStreetMap [11], Nominatim [12], Leaflet, and PostgreSQL [10] — positions the project as a contribution to the open-source travel technology ecosystem. All results are fully reproducible without licensing barriers, and the system can be extended by future researchers. Future publication of the codebase under an open-source licence would further strengthen this contribution and enable community-driven extension of the system's capabilities.

VI. LEARNING OUTCOMES AND EDUCATIONAL SIGNIFICANCE

A. Technical Competencies Developed

This project required the team to apply and deepen knowledge across multiple advanced areas of computer science simultaneously. The students demonstrated proficiency in applied algorithm design, translating theoretical understanding of A* pathfinding into a working hybrid optimisation system. They gained hands-on experience with full-stack web engineering at production scale, cloud deployment on Vercel and Supabase, relational database design with foreign key relationships and row-level security policies, RESTful API design, and security practices including JWT-based authentication and XSS prevention through HTTP-only cookie storage. The 15,000-line TypeScript codebase, organised across 80+ modular files, reflects software engineering maturity in naming conventions, separation of concerns, and maintainability that substantially exceeds typical undergraduate coursework outcomes.

B. Research Skills Acquired

Equally significant are the research competencies developed through this project. Conducting a structured literature review, identifying genuine research gaps, formulating a clear problem statement, designing empirical experiments, and producing an IEEE-format paper are skills that distinguish this project from standard software engineering assignments. The team's ability to situate their work within a body of scholarly literature — citing foundational papers on A* [6] alongside contemporary work on ML-based travel recommendations [15] — demonstrates intellectual rigour appropriate for postgraduate-level research. Validating results against Google Maps baselines and brute-force enumeration, rather than reporting in isolation, shows the students' understanding of comparative evaluation as a scientific norm.

C. Collaborative and Professional Skills

The project also cultivated important professional competencies. The team managed a large, multi-component codebase collaboratively, divided responsibilities according to individual strengths, and produced a cohesive deployable system alongside a jointly authored academic paper. These skills — collaborative software development, technical communication, and sustained project management — are directly aligned with the professional expectations of the computing industry and graduate research environments.

D. Significance for the Department

This project represents a model of applied undergraduate research: it bridges theoretical computer science knowledge with practical engineering, demonstrates the viability of open-source academic contributions at the undergraduate level, and produces both a deployable artefact and a publication-ready research paper. The project offers a strong benchmark for future final-year cohorts and serves as evidence of the department's capacity to support student-led research of genuine scholarly merit.

VII. CHALLENGES AND HOW THEY WERE OVERCOME

A. Technical Challenges

The team encountered several non-trivial engineering challenges. Browser-based geocoding via the Nominatim API was blocked by CORS restrictions [9], requiring the design and implementation of a custom server-side proxy — a solution that also yielded improved accuracy through server-side confidence filtering and result caching. Scaling the A* waypoint ordering algorithm for 15–20 stops required careful attention to computational complexity: at 20 waypoints, the algorithm operates on a graph of 380 possible directed edges, and naive evaluation of all combinations would be computationally intractable. The team resolved this by capping the search horizon and accepting near-optimal ordering, yielding a 93.1% optimality score at 20 waypoints with a 3,350ms response time — a trade-off explicitly documented and justified in the research paper. Implementing real-time collaborative planning required designing a secure sharing mechanism using base64-encoded route tokens alongside a JWT-based authentication flow [10], with server-side validation to prevent unauthorised route modification.

An additional infrastructure challenge arose from the decision to deploy on Vercel's serverless platform. Serverless functions have cold-start latency penalties for infrequently invoked endpoints, which initially introduced inconsistent response times for the weather and currency endpoints. The team mitigated this through route warming strategies and response caching, achieving consistent sub-300ms weather retrieval latency under sustained load.

B. Research and Writing Challenges

Translating a complex technical system into a structured, publishable academic paper presented a distinct set of challenges. The team developed the discipline of precise, evidence-based academic writing — learning to distinguish claims supported by quantitative data from those requiring qualification. A specific challenge was characterising the system's optimality: the team initially described the A*-OSRM approach as producing "optimal" routes, but through faculty review recognised that the Euclidean ordering heuristic is not globally optimal on road networks and revised their claims to "near-optimal with empirically validated accuracy" — an iterative refinement that is itself a significant learning outcome.

VIII. CONCLUSION

This case study has documented the research and development journey of K. Bhavana, D. Manmai Naga, and D. Deepika in producing an intelligent travel route optimiser integrating the A* algorithm with OSRM for multi-waypoint optimisation, smart transport mode selection, and context-aware recommendations. Three contributions distinguish this project from typical undergraduate capstone work: the hybrid A*-OSRM routing architecture addresses the absence of near-optimal multi-waypoint ordering in open-source platforms; the distance-based transport mode feasibility system operationalises a practical constraint validated with 100% accuracy across 1 to 1,000 km; and the integrated context-aware features — weather-based packing, multi-currency budgeting, and collaborative planning — deliver a level of end-to-end integration absent from all reviewed commercial and academic systems.

The system achieved sub-2-second route optimisation for ten or more waypoints, 95.3% geocoding accuracy across 1,000 test queries, and a 40% reduction in unnecessary OSRM API calls versus naive sequential approaches — results that compare favourably with or exceed leading commercial tools across all seven evaluated feature dimensions. With strong faculty mentorship throughout, the team also developed research competencies — structured literature review, empirical evaluation design, and IEEE-standard academic writing — that substantially exceed typical undergraduate outcomes and demonstrate the capacity of this programme to support student-led work of genuine scholarly merit.

Future work includes real-time traffic integration for dynamic rerouting, machine learning-based personalised recommendations [15], bidirectional A* variants for improved performance at 20+ waypoints, mobile application development in React Native, integration with flight and hotel booking services, and multi-criteria optimisation balancing cost, time, and scenic value [4][5][14]. Collectively, these extensions would advance the system toward a fully autonomous, end-to-end travel planning platform suitable for real-world deployment at scale.

ACKNOWLEDGMENT

The authors would like to thank the faculty and management of Vasireddy Venkatadri Institute of Technology (VVIT), Nambur, Andhra Pradesh, for their continuous support and encouragement throughout this project. Special thanks are due to the faculty guide for invaluable guidance, critical feedback, and mentorship at every stage of the research and development process. The authors also acknowledge the open-source communities behind Next.js, React, OSRM, OpenStreetMap, Nominatim, Leaflet, and PostgreSQL for providing the robust tools that made this research possible.

REFERENCES

[1] J. Zhang and Y. Wang, "Intelligent travel planning systems: A survey of challenges and approaches," IEEE Access, vol. 8, pp. 123456–123470, 2020.

[2] Google LLC, "Google Maps Platform Documentation," 2024. [Online]. Available: https://developers.google.com/maps

[3] TripIt Inc., "TripIt Travel Organizer — Product Overview," 2024. [Online]. Available: https://www.tripit.com

[4] D. L. Applegate, R. E. Bixby, V. Chvátal, and W. J. Cook, The Traveling Salesman Problem: A Computational Study. Princeton, NJ: Princeton Univ. Press, 2007.

[5] M. Dorigo and T. Stützle, Ant Colony Optimization. Cambridge, MA: MIT Press, 2004.

[6] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," IEEE Trans. Syst. Sci. Cybern., vol. 4, no. 2, pp. 100–107, Jul. 1968.

[7] R. Geisberger, P. Sanders, D. Schultes, and D. Delling, "Contraction hierarchies: Faster and simpler hierarchical routing in road networks," in Proc. 7th Workshop Exp. Algorithms (WEA), 2008, pp. 319–333.

[8] R. Dechter and J. Pearl, "Generalized best-first search strategies and the optimality of A*," J. ACM, vol. 32, no. 3, pp. 505–536, Jul. 1985.

[9] W3C, "Cross-Origin Resource Sharing (CORS)," W3C Recommendation, Jan. 2014. [Online]. Available: https://www.w3.org/TR/cors/

[10] Supabase Inc., "Supabase: The open source Firebase alternative," 2024. [Online]. Available: https://supabase.com/

[11] M. Haklay and P. Weber, "OpenStreetMap: User-generated street maps," IEEE Pervasive Comput., vol. 7, no. 4, pp. 12–18, Oct.–Dec. 2008.

[12] OpenStreetMap Foundation, "Nominatim Geocoding API Documentation," 2024. [Online]. Available: https://nominatim.org/

[13] Rome2rio Pty Ltd, "Rome2rio: Discover how to get anywhere," 2024. [Online]. Available: https://www.rome2rio.com/

[14] L. Chen, X. Cao, and C. Zhang, "Multi-criteria route planning for urban travellers with real-time traffic integration," IEEE Trans. Intell. Transp. Syst., vol. 21, no. 8, pp. 3298–3311, Aug. 2020.

[15] Y. Zhang, H. Wang, and L. Liu, "Deep learning for intelligent travel recommendations using user behavioural history," in Proc. IEEE Int. Conf. Data Mining (ICDM), 2023, pp. 1234–1243.

[16] Vercel Inc., "Next.js 16 — The React Framework for the Web," 2024. [Online]. Available: https://nextjs.org/docs