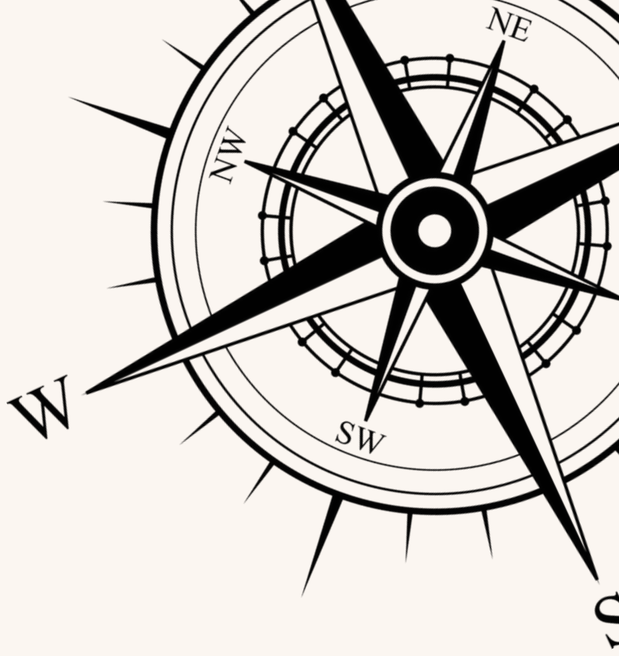


ECO-FRIENDLY TRAVEL PLANNER

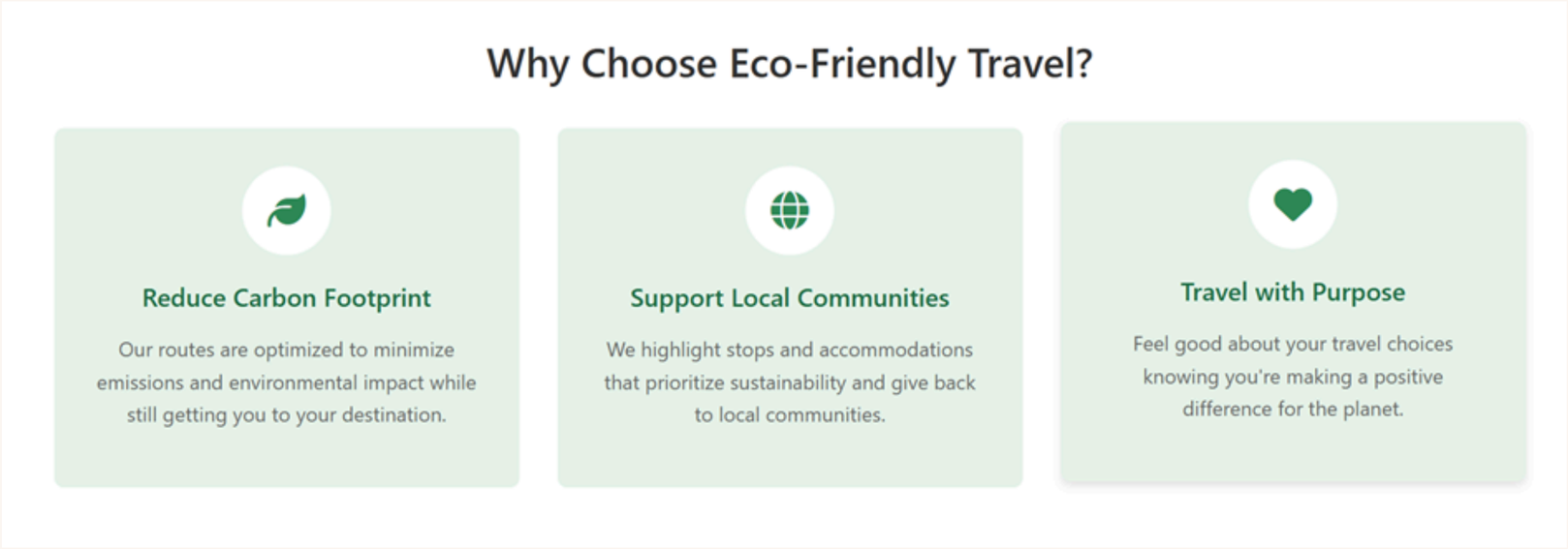
Discovering unexplored corners

Introduction

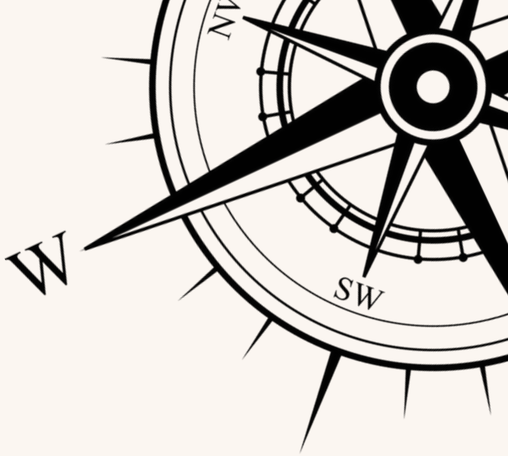
The Eco-Friendly Travel Planner is a smart route optimization system developed using Java, Spring Boot, and JavaScript. It computes the most efficient travel routes based on distance and time by implementing Dijkstra’s algorithm over a graph modeled with an adjacency list. The planner provides a seamless user experience through a robust backend–frontend integration, enabling users to select optimal paths for sustainable and efficient travel.

OBJECTIVE

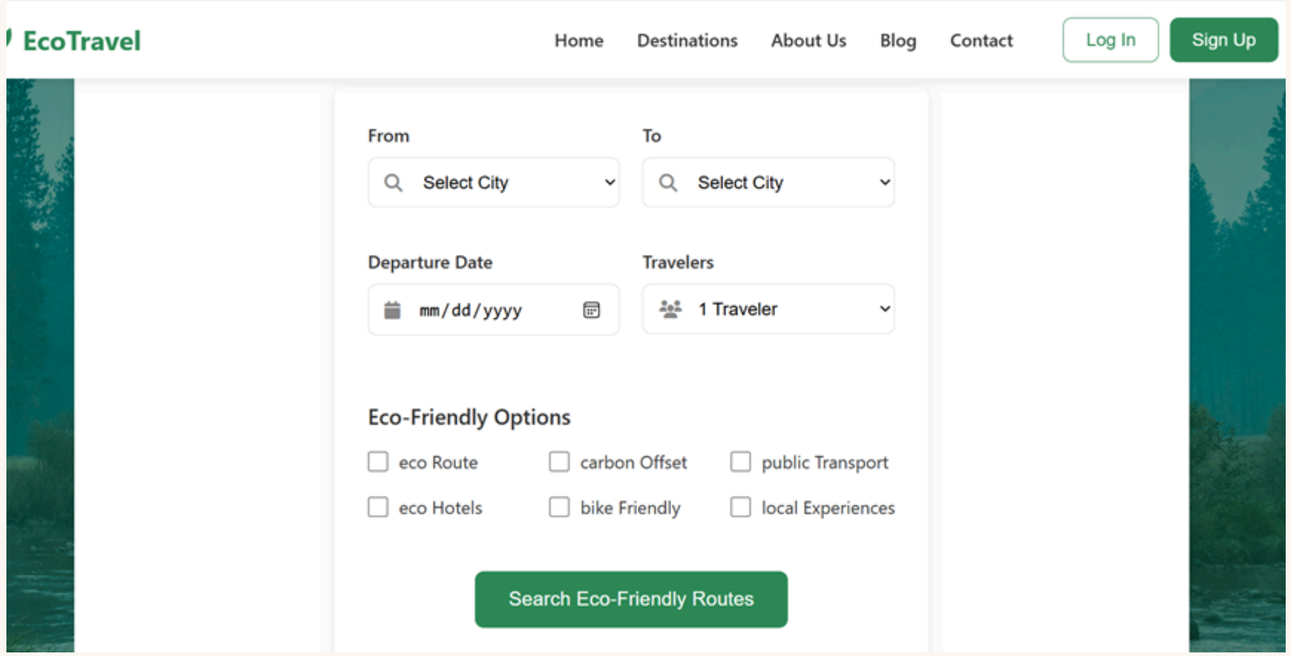
- To help users find the most efficient travel routes based on distance and time, reducing travel effort and confusion.
- To promote eco-friendly travel by minimizing fuel consumption and carbon emissions through optimized routing.
- To apply graph algorithms (Dijkstra's) practically in real-world route planning systems.
- To create a fully functional system with seamless integration of frontend and backend technologies.
- To provide a platform that can be expanded in the future to include scenic routes, toll avoidance, real-time traffic updates, and more.



TECH STACK

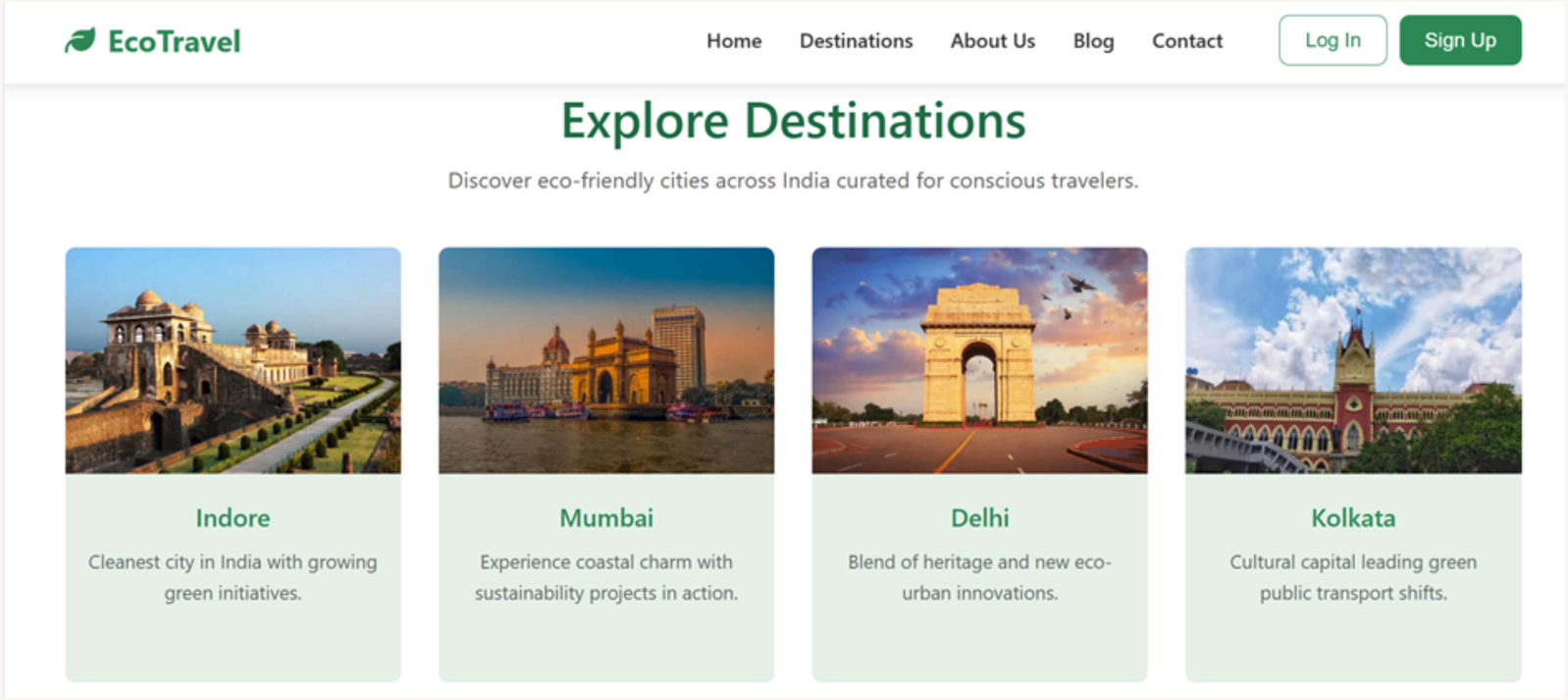


- Backend: Java, Spring Boot
- Frontend: JavaScript, HTML, CSS
- Tools: Eclipse IDE, VS Code
- Algorithm: Dijkstra’s Algorithm
- Data Structure: Adjacency List (Graph)
- Communication: REST APIs
- Package Manager: Node.js (npm)



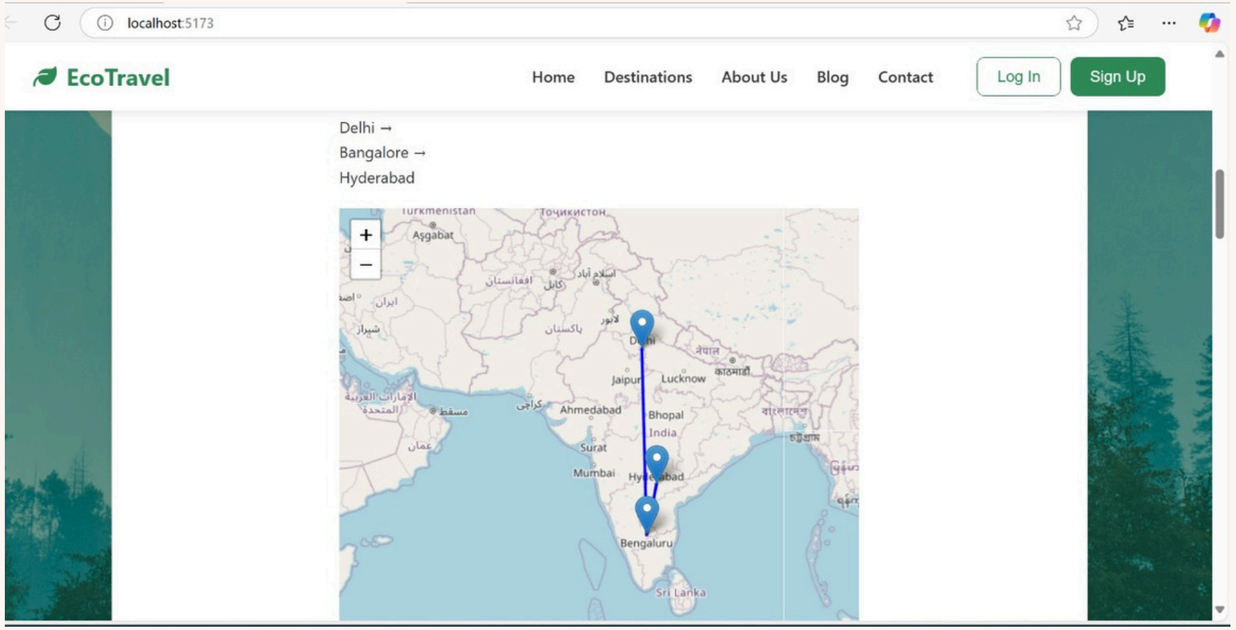
FEATURES

- User-Friendly Interface: Simple and interactive frontend for easy travel planning.
- Route Optimization: Calculates the best route based on shortest distance or shortest time.
- Graph-Based Pathfinding: Uses Dijkstra's algorithm on a graph modeled with an adjacency list.
- Customizable Travel Preferences: Users can select their priority — distance or time optimization.
- Eco-Friendly Focus: Reduces unnecessary travel, indirectly supporting environmental sustainability.
- Cross-Platform Development: Developed using popular tools like Eclipse and VS Code.



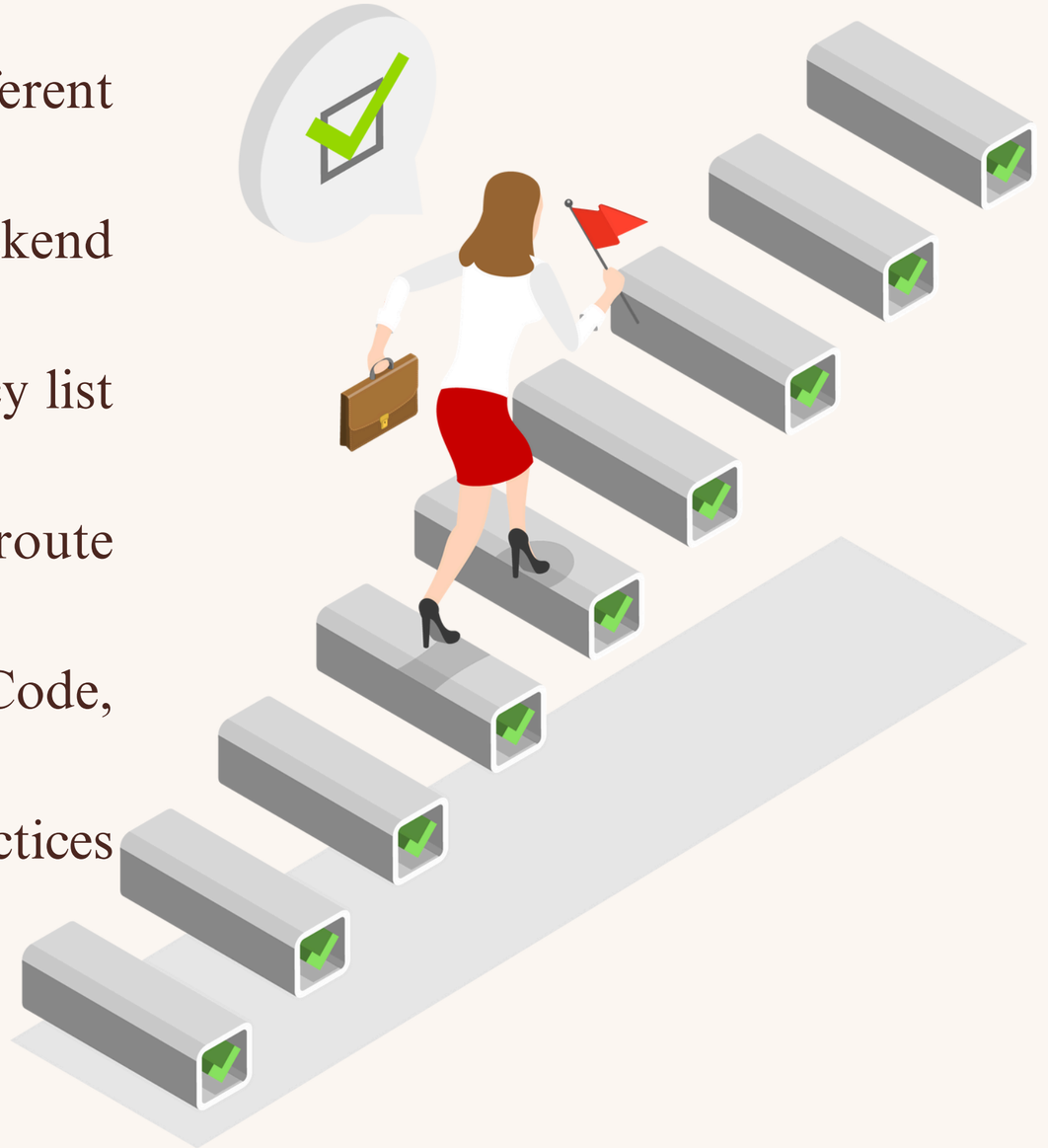
RESULTS

- The system successfully calculates optimal travel routes based on shortest distance or shortest time using Dijkstra's algorithm.
- The backend (Java, Spring Boot) and frontend (JavaScript, HTML, CSS) are fully integrated through REST APIs, ensuring smooth data communication.
- User inputs (starting point and destination) are correctly processed and efficient routes are displayed to the user.
- The adjacency list graph structure efficiently models locations and connections, enabling fast computation even for multiple routes.
- The website delivers accurate and quick results, improving user travel experience and promoting eco-friendly travel by minimizing unnecessary journey length.



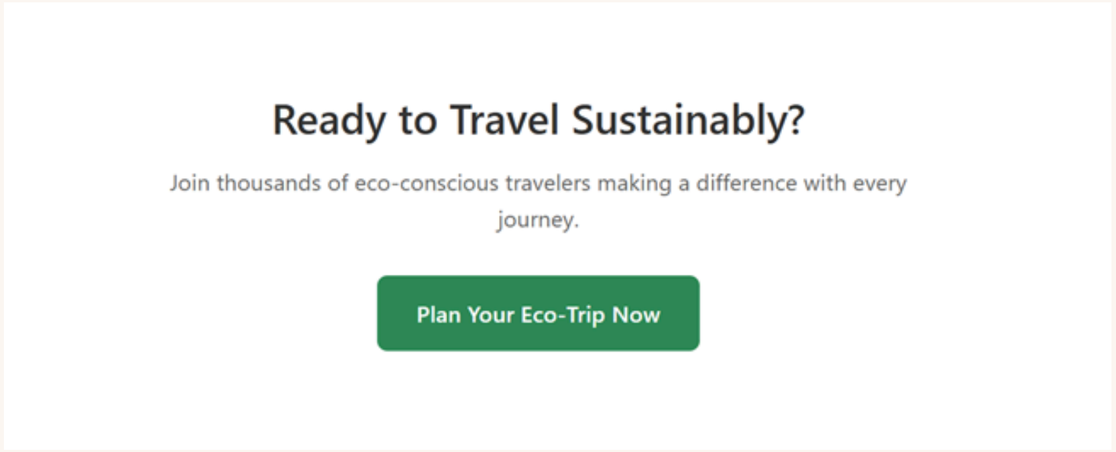
CHALLENGES

- Implementing Dijkstra’s Algorithm efficiently for different user preferences (distance vs. time).
- Ensuring smooth integration between frontend and backend through REST APIs.
- Modeling real-world routes using a graph and adjacency list structure accurately.
- Handling performance optimization to ensure fast route calculations even for larger graphs.
- Setting up development environments (Eclipse, VS Code, Node.js, Vite) and managing dependencies correctly.
- Maintaining code modularity and following best practices for future scalability and easier debugging.



CONCLUSION

- Efficient Travel Planning: Combines Java-Spring Boot backend and JavaScript frontend.
- Route Optimization: Uses Dijkstra’s algorithm for distance and time-based route calculation.
- Modular Architecture: Structured design with separate layers (controllers, services, models, repositories) for scalability.
- Tools Used: Eclipse for backend, VS Code for frontend.
- Future Enhancements: Ability to integrate scenic routes, toll avoidance, and real-time traffic data.
- Platform for Innovation: Strong foundation for continuous improvements in eco-friendly travel planning.



Thank you very much!