

CS4530 Final Project: CoTrip

Group 503: Nimita, Caleb, Jonah, Joomee

CoTrip?

CoTrip is a full-stack travel community platform designed to help travelers connect, share experiences, and build meaningful communities through interactive, real-time features. Built with JavaScript, TypeScript, React, Node.js, and MongoDB, the platform enhances user engagement by combining social interaction with smart travel tools.

The platform includes a **Global Map Interface** that displays all existing travel communities, allowing users to **filter by tags**, explore driving and flight routes, and click directly into any community for more details. Each **Travel Community Page** offers **group messaging, photo sharing with moderation**, and collaborative trip-planning features that make coordination intuitive and social. Users can also submit a **Travel Match Form**, where an **algorithm recommends communities** that align with their interests, trip duration, age group, and destination. To keep interactions active, a **Notification Hub** alerts users when someone joins their community or sends a message, ensuring they stay connected throughout their travel planning.

Future Work

CoTrip plans to introduce an **AI powered travel chat assistant** displayed as a small widget in the bottom right corner of the website. This assistant will allow travelers to ask questions about cost estimates, accommodations, transportation options, and activity recommendations, and receive clear and personalized answers based on their destination and preferences. It will serve as an intelligent guide that helps users compare options, refine their trip plans, discover new experiences, and make more confident decisions through natural, conversational guidance.

Demo and Source

Our web application is available at <https://cs4530-f25-503.onrender.com>, and our code at <https://github.com/neu-cs4530/fall25-project-fall25-project-group-503>.

Technology Stack and Design

CoTrip introduced a Travel Form schema that stores user profiles, trip dates, GeoJSON locations, and tags, which power our matching algorithm and map features. The Community schema was expanded with GeoJSON coordinates, validated age ranges, tags, and travel dates, enabling geographic mapping, filtering, and personalized community recommendations.

A new Notification schema supports event-driven alerts such as new members or submitted forms. It includes sender and recipient data, event types, timestamps, visibility, and metadata, and is delivered through a modular backend service and a polling frontend hook for real-time updates.

For media handling, a dedicated Picture schema manages community photo uploads with metadata, visibility control, moderation status, storage location, and timestamps. Images pass through Azure's moderation API before being stored on local disk in development and an external Render disk in production. This design maintains security, performance, and consistent access control.

The All Communities page was refactored into an interactive Leaflet-based map interface, using client-side rendering to avoid heavy server load. Communities are displayed as map markers populated from database GeoJSON fields, and users can filter by tags, search locations, and view metadata through custom React hooks. A stateless MapPage component focuses solely on rendering, while logic for filtering and navigation remains in parent components to keep the system modular and fault tolerant.

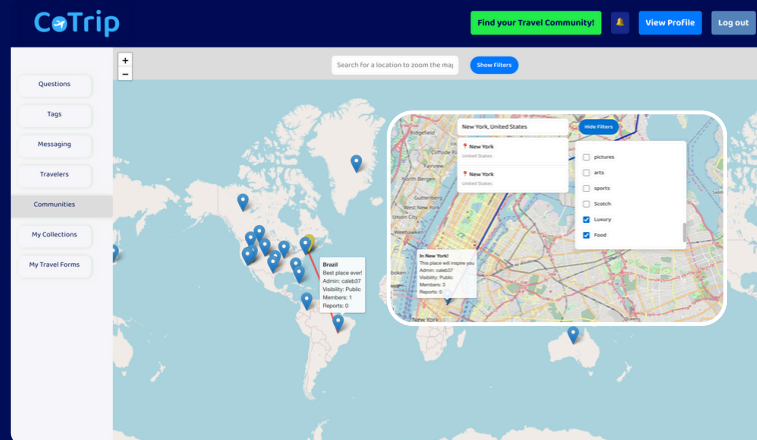


Image 1: Global Map Interface

Image 2: Travel Community Page

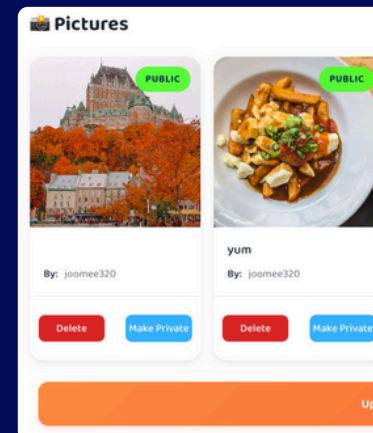
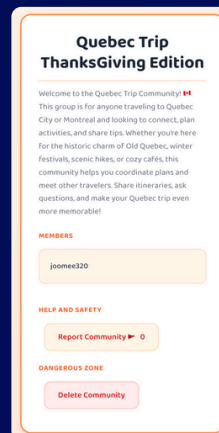
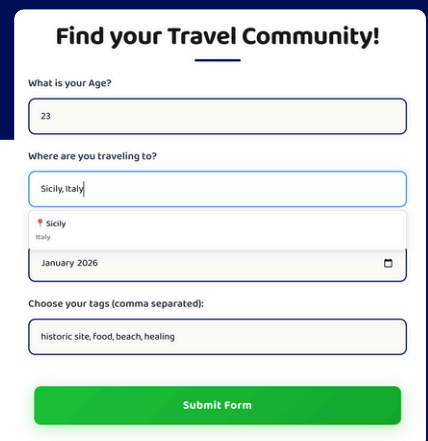


Image 3: Travel Match Form



Find your Travel Community!

What is your Age?

Where are you traveling to?

Sicily
Italy

January 2026

Choose your tags (comma separated):

Submit Form