

# Ride & Walk

MERN Trip Planning App with GROQ descriptions, weather, and real-world routes

---

## Well Known Bugs:

- The map doesn't reset after selecting a new country/city and clicking "Generate."
- A logged-in session persists after the site is closed; the next visitor sees the previous user still logged in.
- Occasionally, the image doesn't match the selected starting country/city.
- For locations near a body of water, generated waypoints sometimes fall in the water.

- **Quick start**

### Prerequisites

- Node.js 18+ and npm
- MongoDB Atlas cluster and database user
- API keys: GROQ\_API\_KEY and UNSPLASH\_ACCESS\_KEY (Open-Meteo needs no key)

### Server environment (.env in server/)

```
MONGODB_URI="mongodb+srv://<user>:<password>@<cluster>.mongodb.net/<db>?retryWrites=true&w=majority"  
JWT_SECRET="a-long-random-secret"  
JWT_EXPIRES_IN="7d"  
GROQ_API_KEY="your_groq_key"  
UNSPLASH_ACCESS_KEY="your_unsplash_key"  
PORT=4000  
CORS_ORIGIN="http://localhost:5173"
```

---

### **.env example:**

```
PORT=4000

MONGO_URI="insert mongo uri here"

JWT_SECRET=insert jwt secret here

CLIENT_ORIGIN=http://localhost:5173


ORS_API_KEY=insert ORS API key here

OSRM_URL=https://router.project-osrm.org

USER_AGENT_EMAIL=dev@example.com


GROQ_API_KEY=insert GROQ API key here

GROQ_MODEL=llama-3.3-70b-versatile

LLM_TEMPERATURE=0.9


UNSPLASH_ACCESS_KEY=insert UNSPLASH API key here
```

### **Install and run (two terminals)**

```
# Terminal A (server)
cd server
npm install
npm run dev
```

```
# Terminal B (client)
cd client
npm install
npm run dev
```

---

## What this project does

### Auth

- Register, login, logout with JWT cookies; protected API routes; Planner and Saved pages redirect to Login when not authenticated.

### Planner

- Input: city/country and mode (Bike or Trek).
- Bike: two days, each up to 60 km; Trek: loops between 5–15 km per day.
- Realistic paths with distance and duration per day.
- Weather: 3-day forecast starting tomorrow for the start location.
- Media: one Unsplash image per place (credit and exact photo location) plus an About section written by Groq for the place.

### Saved routes

- Expanding list in a single card; selecting a route shows About, image, weather, stats, and the map.
- Edit title and description, or delete.

## Project structure

```
project-root/  
├─ README.md  
├─ server/  
│   ├─ package.json  
│   ├─ tsconfig.json  
│   ├─ .env  
│   └─ src/  
│       ├─ index.ts  
│       ├─ config/env.ts  
│       ├─ middleware/auth.ts  
│       ├─ models/  
│       │   ├─ User.ts  
│       │   └─ Route.ts  
│       └─ routes/
```

```
| | | /- auth.routes.ts
| | | /- plan.routes.ts
| | | | route.routes.ts
| | | | services/
| | | | /- llm.ts
| | | | /- media.ts
| | | | /- weather.ts
| | | | | routing.ts
| | | | client/
| | | | /- package.json
| | | | /- index.html
| | | | /- vite.config.ts
| | | | | src/
| | | | | /- api/http.ts
| | | | | /- state/AuthContext.tsx
| | | | | /- components/
| | | | | | /- Header.tsx
| | | | | | /- MapPlanner.tsx
| | | | | | /- TravelMedia.tsx
| | | | | | | WeatherWidget.tsx
| | | | | | pages/
| | | | | | /- Landing.tsx
| | | | | | /- Planner.tsx
| | | | | | /- SavedRoutes.tsx
| | | | | | /- Login.tsx
| | | | | | | Register.tsx
```

---

## Running in production

```
# Build and run server
cd server
npm ci
npm run build
npm run start # node dist/index.js

# Build client (serve statically or via your server)
cd ../client
npm ci
npm run build
npm run preview
```

---

*Tip: serve the built client with your reverse proxy or add static serving in the server to host client/dist.*

## Key dependencies

Server	Client
Express, TypeScript, Mongoose, bcrypt, jsonwebtoken, cookie-parser, cors, dotenv, groq-sdk, @mapbox/polyline	React, Vite, TypeScript, Tailwind CSS, Framer Motion, lucide-react, Leaflet, Axios

## API overview

Endpoint	Description
POST /auth/register	Create account
POST /auth/login	Set auth cookie (JWT)
POST /auth/logout	Clear cookie
POST /plan/plan	Generate a plan: days[], media, weather
GET /plan/weather?lat=&lng=	3-day forecast starting tomorrow
GET /plan/media?location=...&lat=&lng=	About (Groq) + Unsplash image
GET /routes	List current user routes
POST /routes	Create route (persists aboutHtml, imageUrl, etc.)
PUT /routes/:id	Update title/description (and optionally aboutHtml)
DELETE /routes/:id	Delete route

## Configuration notes

### MongoDB Atlas

- Create a database user and allow your IP or 0.0.0.0/0 during development. Use mongodb+srv in MONGODB\_URI.

### Auth cookies

- Server issues an HTTP-only JWT cookie. Client must send credentials; ensure http.ts sets withCredentials: true.

### CORS

- Set CORS\_ORIGIN to your client origin.

### Groq and Unsplash

- Use your keys on the server. Open-Meteo needs no key.

## **Troubleshooting**

### **“Cannot GET /” on server port**

- That is normal when opening the server directly. Use the client dev URL or serve the client build from the server.

### **Login button does not switch without refresh**

- Ensure the app is wrapped in AuthProvider and requests include credentials.

### **Auth failures to protected routes**

- Verify JWT\_SECRET, cookie domain, and CORS. Confirm browser sends credentials.

### **Unsplash unrelated images**

- Refine the query in server/src/services/media.ts to better target the place.

### **Rate limits**

- Groq and Unsplash are rate-limited. Consider caching.

## **Known limitations**

- Route geometry is simulated with realistic paths and caps; may not match official trails everywhere.
- Weather is a point forecast for the start location.