INDEPENDENT PORTFOLIO PROJECT / WEB APP

DETOUR

Product Requirements Document

A free, no-login web app that turns a time budget into a walking or driving route.

Field	Value
Document	Detour Product Requirements Document
Document ID	DTR-PRD-001
Version	1.0
Status	V1 baseline. Detour V1 is built and deployed on Netlify; this document records its requirements and the candidates for the next version.
Date	Nov 2026
Platform scope	Web app for phone and desktop browsers, installable to the home screen
Product owner	Independent portfolio project by Jahaan Desai

> **SCOPE NOTE**
>
> This PRD records the requirements for Detour V1 as built and deployed in Nov 2026, and the deferred candidates for a next version. Detour is an independent portfolio project, not a commercial product. It is developed and initially tested in India, particularly Mumbai and its suburbs, but it is not limited to Indian users.

Contents

1 Product Overview; 2 Problem Statement and Evidence; 3 Goals, Non-Goals and Principles; 4 Users and Core Scenarios; 5 Functional Requirements; 6 Edge Cases and Errors; 7 Non-Functional Requirements and Architecture; 8 Release Quality and Testing; 9 Success Metrics; 10 Launch Plan and Next Version; 11 Risks, Dependencies and Open Questions; Appendix A Terminology; Appendix B Sources.

1. Product Overview

TL;DR

Detour turns “I have 45 minutes” into a route. The user picks walk or drive, a time between 10 minutes and 4 hours, and one of three trip types: a loop that ends where it started, a there-and-back that returns the same way, or a one-way trip that ends somewhere new. An optional stay-within limit keeps the whole route inside a chosen radius. The route can be opened in Google Maps, downloaded as a GPX file or shared as a link that rebuilds it. It needs no account and runs at zero cost.

1.1 Product summary

Detour is a free, no-login web app. The user gives it a start point, a mode, a time and a trip type, and it returns a route on a map that can be handed to Google Maps, Strava or Garmin, or shared as a link. It works on phones and desktops and can be installed to the home screen.

1.2 Product hypothesis

HYPOTHESIS

If a route planner takes the time a person has, rather than a distance, and plans a walk or a drive in one step, people who think in minutes will be able to get a usable route without converting time to distance themselves or signing up for anything.

1.3 V1 scope

V1 covers walking and driving; a time budget from 10 minutes to 4 hours; three trip types (loop, there and back, one-way); an optional stay-within radius; results on a map; hand-off to Google Maps, GPX and share links; and phone, desktop and home-screen use, with no login and no backend database. Section 5 specifies each behaviour.



The three trip types. The orange pin marks the turnaround of a there-and-back route.

2. Problem Statement and Evidence

2.1 Problem definition

Most route tools ask for a distance. People planning a walk or a short drive usually start from time instead (“I have 45 minutes”), and converting that into a distance, then into a route that returns to the start or stays within a radius, is left to the user.

2.2 Evidence

Source	What it shows	Type
Personal need	The product owner's repeated question: how to plan an X-minute walk or drive, one-way or return, within Y km.	Personal observation
r/walking, Jul 2026	A maker post for a random-loop tool drew 40 upvotes and 10 comments, including a request for time-based A-to-B routes.	Community signal
r/cycling, Apr 2026	A thread asking for an app that generates a random route drew 12 replies naming tools.	Community signal
Existing tools	Several loop-route generators exist. Most are distance-led and focus on walking, running or cycling; drive loops are served mainly by motorcycle apps.	Secondary research

No validated findings exist yet: no user research has been run and V1 has no usage data. The signals show that people look for generated routes and that a time-based request has come up. They do not show how many people would use Detour, or that time-based input suits most users better than distance.

2.3 Competitive context

Time-based input already exists in a narrow form: LoopWalk accepts a walk time but produces walking loops only. Detour's scope is one page that takes minutes as input for both walking and driving, with three trip types. The full competitive comparison is in the Detour case study and is not repeated here.

3. Goals, Non-Goals and Principles

3.1 Product goals

- Let a user get a walking or driving route from a time budget, without converting time to distance.
- Support the three trip shapes people ask for: back to the start, back the same way, or ending somewhere new.
- Keep the whole route within a chosen radius when the user sets one.

- Make the route easy to take away: Google Maps, GPX for Strava or Garmin, and a share link.
- Run at zero cost and keep the routing key out of the browser.
- Fail clearly: when a request cannot be met or the free limit is busy, say so instead of breaking.

3.2 Non-goals for V1

- Fitness tracking, turn-by-turn navigation or a social route library. Google Maps provides navigation once the route is handed over.
- Accounts, sign-in or saved routes.
- A backend database or any server-side storage of user routes.
- Live traffic in drive times.
- Cycling, elevation and road-type preference options.
- A user-chosen destination for one-way trips.
- Paid APIs or paid hosting.

3.3 Product principles

Principle	What it means for the product
Time is the input	The user states how long they have; Detour performs the conversion to distance.
One page, one job	Set a few inputs and get a route, with no sign-up or onboarding.
Honest results	If a route runs shorter than requested or a limit cannot be met, the app states it.
Hand off, do not replace	Google Maps, GPX and share links carry the route to the tools people already use.
Zero cost, key on the server	Free tiers only, and the routing key never reaches the browser. See section 7.

4. Users and Core Scenarios

4.1 Primary user context

The primary user has a fixed amount of free time and wants a walk or a short drive without planning it: a lunch-break walk, an evening loop around the neighbourhood, or an hour's drive that should not go too far. A secondary user is a runner or walker who plans in one app, records in another and moves routes between them as GPX files.

4.2 Core scenarios

ID	Scenario	Expected behaviour	FRs
S-01	Walking loop. A user with 45 minutes sets a start point with place search and chooses Walk, 45 minutes and Loop.	The route ends at the start without retracing its steps. Time, distance and the furthest point from the start are shown.	FR-01, FR-03 to FR-05, FR-11, FR-18
S-02	There and back with a limit. A driver chooses Drive, 1 hour, There and back, and stay within 5 km.	The route returns along the same path, the turnaround is marked and the route stays inside the dashed circle. If the hour cannot be filled inside the limit, the best route that fits is shown and flagged as shorter than requested.	FR-06, FR-12, FR-14, FR-15, FR-19, FR-20
S-03	One-way trip. A user chooses a 30-minute one-way walk.	The route, its end point and the straight-line distance from the start are shown. The user does not choose the destination.	FR-13, FR-21
S-04	GPX export and Google Maps. A runner wants the route in Strava or Garmin; another user wants to start it in Google Maps.	Download GPX produces a valid file with the full track. Open in Google Maps passes eight points along the route.	FR-22, FR-23
S-05	Shared route. A user sends the route to a friend.	The share link opens the same route on any device, including the units.	FR-24
S-06	Shuffle. The user does not like the first route.	One tap builds a new route for the same settings.	FR-17
S-07	No start point. The user taps generate before setting a start.	A prompt asks for a start point; no routing request is made.	FR-02

5. Functional Requirements

V1 has 32 functional requirements. Must means V1 is not complete without it; Should means it is part of V1 but the product still works without it. Numeric generation values are defined once, in section 7.2.

ID	Requirement	Priority
5.1 Inputs		
FR-01	The user sets a start point using place search before a route can be generated.	Must
FR-02	If the user generates without a start point, Detour shows a clear prompt to set one and makes no routing request.	Must

ID	Requirement	Priority
FR-03	The user chooses a mode: Walk or Drive.	Must
FR-04	The user chooses a time between 10 minutes and 4 hours.	Must
FR-05	The user chooses a trip type: Loop, There and back, or One-way.	Must
FR-06	The user can optionally set a stay-within limit as a radius from the start.	Must
FR-07	The user can switch between kilometres and miles. Every distance on screen, including the limit, converts.	Must
5.2 Route generation		
FR-08	Detour converts the time into a target distance using a speed guess for the mode (4.8 km/h walking, 22 km/h driving) and a road factor of 1.35. Values: section 7.2.	Must
FR-09	Detour corrects the target using the travel time the router returns, and retries up to four times. Values: section 7.2.	Must
FR-10	Detour accepts a route within 12% of the requested time, and otherwise keeps the attempt closest to the target that also respects the limit. Values: section 7.2.	Must
FR-11	Loop uses the router's round-trip option with three to five turning points, so the user does not retrace their steps.	Must
FR-12	There and back routes to a point about half the time away in a random direction, then returns along the same path.	Must
FR-13	One-way routes to a point the full time away.	Must
FR-14	When a stay-within limit is set and any part of the route leaves the circle, Detour shortens the target and tries again.	Must
FR-15	When the time cannot be met inside the limit, Detour shows the best route that fits and says it runs shorter than asked, instead of failing.	Must
FR-16	Round trips are limited to between 0.2 and 100 km, within the router's round-trip cap. Values: section 7.2.	Must
FR-17	Every route carries a seed number. Shuffle picks a new seed and builds a new route for the same settings.	Must
5.3 Results on the map		
FR-18	The route is drawn on the map with its time, distance and furthest point from the start.	Must
FR-19	A stay-within limit is drawn on the map as a dashed circle.	Must
FR-20	A there-and-back route marks its turnaround with an orange pin.	Should

ID	Requirement	Priority
FR-21	A one-way route shows the end point and the straight-line distance from the start.	Must
5.4 Taking the route away		
FR-22	Open in Google Maps passes eight points along the route to Google Maps.	Must
FR-23	Download GPX produces a valid file containing the full route track.	Must
FR-24	A share link encodes the settings and seed, and opening it rebuilds the same route, including the units.	Must
5.5 Layout and installation		
FR-25	Detour works on phone and desktop browsers. On a phone, the controls collapse to a summary bar with a Start in Maps action.	Must
FR-26	Detour can be installed to the home screen. A service worker caches the page itself, but never routes or map tiles.	Should
5.6 Server function and key protection		
FR-27	Every route request goes through a server function at /api/route that adds the routing key. The key never reaches the browser.	Must
FR-28	The function accepts only POST requests, only the foot-walking and driving-car profiles, at most five points, and round trips between 0.2 and 100 km. Anything else is rejected.	Must
FR-29	The router's base address can be changed with an ORS_BASE_URL setting, without a code change.	Should
5.7 Error handling		
FR-30	When the free routing limit is reached, the user sees a “limit busy, try again” message rather than a crash.	Must
FR-31	Configuration and service problems show a specific message: missing key, invalid key, or routing service unreachable.	Must
5.8 Privacy and attribution		
FR-32	Detour needs no login and stores no routes or user data on a server. Map tiles carry OpenStreetMap attribution.	Must

6. Edge Cases and Errors

6.1 Edge cases

Conditions fully specified by their requirements and the error table: no start point set (FR-02); no attempt within the time tolerance (FR-10); a request outside the allowed shape (FR-28); a shared link opened on another device, which needs no server state (FR-24); the daily or per-minute limit reached, which resets on its own (FR-30, section 6.2); and a moved router address or a service outage (FR-29, FR-31, section 6.2). The table lists only behaviour or context that the requirements do not already state.

Condition	Requirement	Additional behaviour or context
Long time with a tight limit	FR-14, FR-15	Retries made to stay inside the limit count toward the four-attempt maximum. The shorter route is flagged on screen; it is not treated as an error.
Awkward or odd loop	FR-11, FR-17	Turning points are randomised, so some loops are awkward. Shuffle is the in-product fallback.
Thin footpath or lane data	None (data dependency)	OpenStreetMap footpath and lane data is less complete in some Indian cities, so some walking routes may take odd turns. Quality varies by city and is a test item for Mumbai suburbs (section 8).
Google Maps draws a different path	FR-22, FR-23	Google Maps reconstructs its own path between the eight points it receives. The GPX file contains the generated track.
Drive times in traffic	None (service limit)	Drive times come from the routing service and exclude live traffic, so travel in heavy traffic can take longer than shown.

6.2 User-visible errors

Message	Cause	Resolution
Server is missing ORS_API_KEY	The variable is not set, or the site was not redeployed after setting it	Set it, then redeploy
Invalid API key	A typo or stray space in the key	Re-paste it and redeploy
Could not reach the routing service	The API address moved or the service is down	Set ORS_BASE_URL, or wait
The free routing limit is busy	The daily or per-minute cap was reached	Wait; it resets on its own

7. Non-Functional Requirements and Architecture

7.1 Requirements and technical constraints

Area	Requirement
Cost and hosting	Zero running cost. A static site and one server function on Netlify's free tier, routing from the openrouteservice free Standard plan and map tiles from OpenStreetMap.
Service limits and capacity	The free plan allows 2,000 directions a day and 40 a minute. A route uses up to four requests, so about 500 routes a day is a planning estimate, not a guaranteed capacity.
Generation limits	At most four routing requests per route, which bounds generation time and quota use. Settings in section 7.2.
Security and key handling	ORS_API_KEY exists only in the server function's environment. The function at /api/route adds it to requests and accepts only the narrow request shape in FR-28.
Resilience	The routing API address is configurable through ORS_BASE_URL. Reaching the service limit produces a message, not a crash.
Technology	Front end in plain HTML, CSS and JavaScript, with no framework and no build step. Map display with Leaflet 1.9.4, bundled with the site, using OpenStreetMap tiles with attribution. Routing, round trips and place search from openrouteservice.
Install and offline shell	A web app manifest and a service worker. The service worker caches the page only, never routes or map tiles.
Privacy and storage	No accounts, no database and no stored routes. A share link carries the settings and seed needed to rebuild a route.
Units	Metric by default: kilometres, with miles available (FR-07). All technical values in this document are metric.

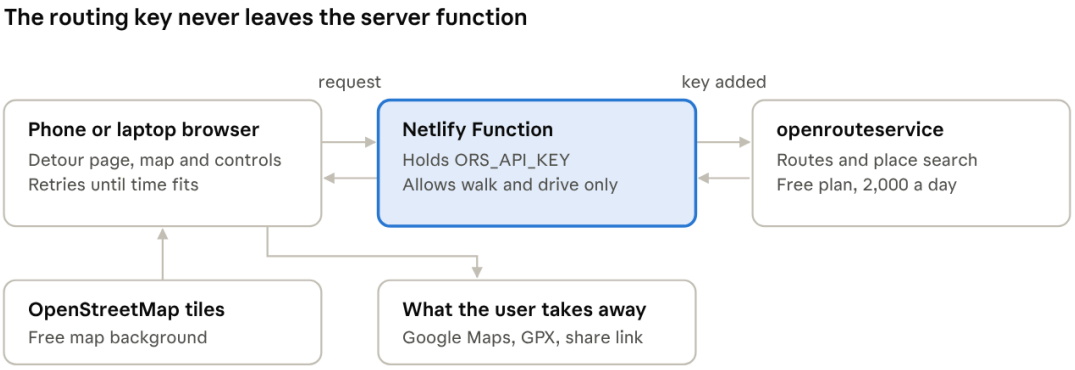
7.2 Generation settings

This table is the authoritative source for generation values referenced in section 5.

Setting	Value	Notes
Walking speed guess	4.8 km/h	Initial estimate of walking pace, not a validated Indian average. Corrected per route by the router's travel time.
Driving speed guess	22 km/h	Initial estimate of city driving speed, not a validated Indian average. Corrected per route by the router's travel time.
Road factor	1.35	Roads run longer than a straight line.
Time tolerance	12%	Close enough to the requested time; avoids endless retries.

Setting	Value	Notes
Maximum attempts	4	Bounds generation time and keeps use within free limits.
Round-trip cap	100 km	The openrouteservice limit for round trips.

7.3 Architecture



Every route request goes through the Netlify Function, which adds the key and forwards only walking and driving requests.

8. Release Quality and Testing

8.1 Pre-launch tests against a stand-in service

Before launch, the main flows were tested end to end in a headless browser against a stand-in routing service, because the build environment could not reach openrouteservice. All ten checks below passed with no page errors. These results confirm application behaviour against the stand-in, not the quality of live routes.

Check	Result
Generate with no start point	Clear prompt to set one
Place search, then a 45-minute walking loop	Route, time, distance and furthest point shown
Shuffle	New seed and new route
1-hour drive, there and back, within 5 km	Stayed inside the limit and flagged that it ran shorter than asked
30-minute one-way walk	End point and straight-line distance shown
Switch km to miles	All figures converted

Check	Result
GPX download	Valid file with the full track
Shared link opened fresh	Same route rebuilt, including units
Phone layout	Controls collapse to a summary bar with Start in Maps
Server function	Rejects unknown modes, oversized loops and non-POST calls

8.2 Tests against the live routing service and known limitations

Testing against the deployed openrouteservice integration began after deployment. Live route quality has not been fully tested. Open items: ten real walking routes in Mumbai suburbs before local release, a review of the 22 km/h driving estimate in city conditions, and continued spot checks of generated routes.

Known limitations at release are the data and service conditions in section 6.1.

9. Success Metrics

V1 has no analytics. Routes generated are estimated from daily openrouteservice request counts, used as a proxy. Launch reactions and feature requests are supporting signals. All of these are planned measures, not measured results.

Type	Metric	How it is measured
Primary	Routes generated per day	Daily request counts on the openrouteservice dashboard, divided by up to four requests per route, as a proxy.
Supporting	Reaction to launch posts	Upvotes and comments per Reddit post.
Supporting	Feature requests	The number of distinct requests from first users, and how often each comes up.
Guardrail	Days the free limit is reached	Days on which the daily cap is hit and users see “limit busy, try again”.

10. Launch Plan and Next Version

10.1 Release considerations for V1

- Before local release, test ten real walking routes in Mumbai suburbs (section 8).

- At launch, monitor daily openrouteservice request counts and the days on which the free limit is reached (section 9).
- After launch, ship one visible fix from first-user feedback within a week.

10.2 Deferred candidates

The following are deferred possibilities, not committed requirements or a roadmap. They are ordered by how often the need showed up in research; the decision on what, if anything, to build next depends on first-user feedback.

Candidate	Why	Source of the need
One-way to a chosen destination, stretched to X minutes	Turns a commute into a walk	Requested on the r/walking route tool thread; RouteFast launched with it
Avoid busy roads, prefer parks	Walks feel safer and nicer	Trail Router's main selling point
Elevation profile	Runners and cyclists care about hills	route-random already shows elevation gain and loss
Cycling mode	Cyclists ask for random route apps	r/cycling threads
Free privacy-friendly analytics	Real usage numbers for measuring the launch	Needed to measure the launch

11. Risks, Dependencies and Open Questions

11.1 Key risks and mitigations

Risk	Mitigation
Route quality in Indian cities	Quality varies by city; Mumbai suburbs are tested before local release (section 8.2). Shuffle is the in-product fallback.
Free quota exhausted during a usage spike	FR-30 message instead of a crash; limits reset daily, and openrouteservice accepts free applications for a higher quota.
Routing API address changes	ORS_BASE_URL, with no code change (FR-29).
API key misuse	Key held server-side and a narrow request shape (FR-27, FR-28).
Live-service validation	Pre-launch tests used a stand-in; live routes need continued spot checks (section 8.2).
Driving-time estimates	No live traffic, and 22 km/h is an initial estimate (section 7.2). Review whether it should vary by city.

11.2 Dependencies

Dependency	Used for
openrouteservice	Routing, round trips and place search on the free Standard plan
OpenStreetMap	Map tiles and the underlying road and footpath data
Leaflet 1.9.4	Map display, bundled with the site
Netlify	Static hosting and the /api/route server function on the free tier

11.3 Open questions

- How usable are walking routes in Mumbai suburbs in practice?
- Is 22 km/h a reasonable estimate for city driving, or should it vary by city?

Appendix A. Terminology

Term	Meaning
One-way	A route from the start to a point the full time away. The user does not choose the destination in V1.
Stay within	An optional radius, measured from the start, that the whole route must stay inside.
Seed	A number that makes a generated route repeatable; Shuffle changes it.
GPX	A standard route file format that apps such as Strava and Garmin can import.

Appendix B. Sources

1. openrouteservice plans and limits. https://openrouteservice.org/?p=176
2. openrouteservice API restrictions. https://openrouteservice.org/restrictions/
3. Random loop route tool thread, r/walking. https://www.reddit.com/r/walking/comments/1v92lwo/
4. Random route app question, r/cycling. https://www.reddit.com/r/cycling/comments/1sgeqt2/
5. LoopWalk. https://loopwalk.netlify.app/
6. RouteFast on Product Hunt. https://www.producthunt.com/products/routefast