

PathArt – GPS Art Route Generator

Web Application Development

Introduction:

PathArt is a web-based application designed to transform simple hand-drawn sketches into GPS-compatible routes that align with real-world streets and paths.

Users upload a drawing, position it on an interactive map, and generate a route that can be exported as a GPX file and imported into fitness platforms such as Strava or Komoot.

This document specifies the functional and non-functional requirements of the system.

System Overview

The system provides the following workflow:

1. User uploads a sketch (PNG/JPG).
2. The image is vectorised into a clean outline.
3. The outline is placed on an interactive map.
4. The user adjusts size, position, and rotation.
5. The system generates a street-based route following the outline.
6. User exports the resulting route as a GPX file.

The tool supports creativity, fitness, and personalised map-based artwork.

In-Scope Functionality

1. Uploading and previewing raster images
2. Vectorisation into SVG/paths
3. Interactive map placement
4. Real-world route generation
5. GPX export
6. Optional enhancements (distance, time, variants)

Out-of-Scope Functionality

1. Native mobile applications
2. Social networking features
3. Large public galleries
4. Complex authentication systems
5. Heavy server-side rendering

User Roles:

Guest User (Primary Role)

1. Full access to core creation tools
2. Create, edit, and export GPS art

3. No login required

Future Roles (Optional Enhancements)

1. Registered User: Save history, manage gallery
2. Administrator: Approve shared content

Functional Requirements

Mandatory Features

Upload Sketch

Users can upload PNG/JPG/JPEG drawings.

System validates file size and displays preview.

Vectorise Image

The application converts raster sketches into simplified vector paths (SVG) using Potrace.js or Paper.js.

Map Overlay

The vector outline is displayed on an interactive map using Leaflet or Mapbox.

Shape Manipulation

Users can move, scale, and rotate the outline to align it with the selected geographic area.

GPS Route Generation

System generates a real-street route following the outline:

1. Sample points along vector path
2. Use OpenRouteService/Mapbox Directions for routing
3. Merge partial routes into one continuous GeoJSON route

GPX Export

The final route is converted to GPX using togpX (or server-side equivalent).

User downloads a GPX file compatible with Strava/Komoot.

Enhancements

Distance & Time Estimation

Display total route distance and estimated completion time for walking, running, or cycling.

Routing Profile Selector

Allow user to choose mode: walking, running, cycling.

Alternative Route Variants

Generate multiple versions (Short / Normal / Detailed) for the same artwork.

Fun & Creative Features:

Built-in Shape Library

Users can choose pre-designed shapes (heart, star, arrow, etc.) instead of uploading.

Themed Styles

Visual themes (dark, neon, minimalist) to preview the route for export.

Export Map Snapshot

Download a PNG screenshot of the final GPS artwork.

Decorative POIs

Allow users to add stickers/icons (e.g., coffee stops, checkpoints).

Non-Functional Requirements:

Usability

1. Clear step-by-step workflow
2. Accessible on modern browsers
3. Simple interface for non-technical users

Performance

1. Route generation should complete within 10 seconds
2. Efficient vector processing

Compatibility

1. Works on Chrome, Firefox, Edge
2. Responsive design (desktop-first)

Security

1. API keys stored securely (backend-side)
2. No permanent storage of uploaded images

Reliability

1. Graceful handling of routing failures
2. Informative error messages

External Interfaces:

Frontend

1. HTML, CSS, JavaScript
2. Leaflet.js or Mapbox GL for map display
3. Potrace.js or Paper.js for vectorisation

Backend

1. Python Flask or Node.js
2. REST endpoints for routing and GPX conversion

External APIs

1. OpenRouteService Directions API
2. Mapbox Directions API
3. OpenStreetMap raster tiles

Assumptions & Constraints:

1. Users have internet access
2. Free-tier API limits may affect usage
3. Drawings must be simple outlines for best results
4. Map data availability depends on external providers