Street Name Etymology Website: A Complete Production Setup Guide

1. Executive Summary

This report provides a comprehensive, production-ready setup guide for a street name history and etymology website. It is designed for non-technical founders, offering a weekend-to-MVP plan that costs less than £1 for the first month and is engineered to scale to 10,000 monthly users for under £20/month. The recommended technology stack maximizes the use of free tiers from best-in-class providers, ensuring a low-cost, scalable, and low-maintenance solution.

Recommended Tech Stack & Costs:

- **Hosting: Cloudflare Pages** - Free, with unlimited static asset bandwidth, a global CDN, and generous limits for builds and functions.
- **Database: Supabase (PostgreSQL with PostGIS)** - Free tier includes a 500MB database, which is sufficient for the initial UK dataset of ~790,000 street records. The Pro plan starts at $25/month for 8GB of storage, which will be needed as the data grows.
- **Authentication & Community: Supabase Auth** - Free for up to 50,000 monthly active users, with integrated social logins. Community features can be built using Supabase's database with Row Level Security.
- **Mapping:**
  - **Basemap Tiles: MapTiler** or **Stadia Maps** - Both offer generous free tiers for production use.
  - **Map Rendering: MapLibre GL JS** - A high-performance, open-source library for rendering interactive maps with large datasets.
  - **Geocoding: LocationIQ** or **OpenCage** - Both have free tiers suitable for initial development.
- **AI/ML Etymology:** A hybrid approach using local processing with open-source tools and selective API calls to **OpenAI's GPT-4o mini** for complex cases. This keeps costs to a minimum while providing high-quality etymology suggestions.
- **Domain & SSL:** A `.org` domain from a registrar like **Namecheap** (around £8 for the first year) with free SSL from **Cloudflare** or **Let's Encrypt**.

Initial Cost: The MVP can be launched for the cost of the domain registration only, making it a true weekend project with minimal financial outlay. Ongoing costs will be negligible until the website scales beyond the generous free tiers of the recommended services.

This guide provides a step-by-step plan for building and launching the website, a detailed cost breakdown for the first two years, a scalability roadmap, a data strategy, and best practices for backup, monitoring, and risk mitigation.

2. Technical Architecture

The proposed architecture is a modern, serverless, Jamstack-style application that is designed for performance, scalability, and low maintenance.

System Design

The system consists of the following components:

1. **Frontend (Web Application):** A static website built with a modern JavaScript framework (like Next.js, SvelteKit, or Astro) and hosted on Cloudflare Pages. The frontend will be responsible for rendering the user interface, including the interactive map, street information pages, and community features.
2. **Backend (API):** For dynamic functionality, the application will use serverless functions, also hosted on Cloudflare Pages (as Pages Functions, which use Cloudflare Workers). These functions will handle tasks like:
    - Proxying requests to the database to fetch street data.
    - Handling user authentication and community contributions.
    - Interacting with the AI/ML service for etymology suggestions.
3. **Database:** A PostgreSQL database hosted on Supabase. The database will store all the street name data, etymology information, user accounts, and community contributions. The PostGIS extension will be enabled to allow for efficient geographic queries (e.g., finding streets within a certain area on the map).
4. **Mapping Service:**
    - **MapLibre GL JS** will be used in the frontend to render the interactive map.
    - Basemap tiles will be provided by **MapTiler** or **Stadia Maps**.
    - **LocationIQ** or **OpenCage** will be used for geocoding (turning addresses or place names into coordinates).
5. **AI/ML Service:** A hybrid model for etymology analysis.
    - **Local Processing:** A script will pre-process the street name data, using open-source libraries to normalize names and look up etymologies from a local database created from Wiktionary data.
    - **Cloud API:** For more complex names, a serverless function will call the **OpenAI GPT-4o mini** API to generate etymology suggestions.

Data Flow

1. **Initial Data Load:** The initial UK open data for street names (~790,000 records) from sources like OS OpenNames will be cleaned, processed, and loaded into the Supabase PostgreSQL database.
2. **User Visits Website:** A user accesses the website, and the static frontend is served from Cloudflare's global CDN.
3. **Map Interaction:** The user interacts with the map. The frontend, using MapLibre GL JS, requests map tiles from MapTiler/Stadia Maps. When the user zooms or pans, the frontend sends a request to a serverless function with the map's bounding box.
4. **Fetching Street Data:** The serverless function queries the Supabase database (using PostGIS for spatial filtering) to get the streets within the user's view. The data is returned to the frontend and displayed on the map.
5. **Viewing Street Etymology:** The user clicks on a street. The frontend requests the etymology from a serverless function.
6. **AI/ML Etymology Generation:** If the etymology is not already in the database, the serverless function triggers the hybrid AI/ML service. Simple cases are handled locally. For complex cases, a call is made to the OpenAI API. The result is saved in the database and returned to the user.
7. **User Contributions:** A logged-in user (authenticated via Supabase Auth) can submit etymology suggestions or historical photos. These submissions are sent to a serverless function, which saves them to the database for moderation.

This architecture is highly decoupled, allowing each component to be scaled or replaced independently. The use of serverless functions and managed services minimizes the need for server maintenance.

3. Storage Solutions for Images and Media

A comprehensive street etymology website requires robust storage solutions for both historical map images and user-contributed photographs. The storage strategy must balance cost-effectiveness, performance, and scalability while maintaining the project's low-budget requirements.

Storage Requirements

The platform will need to handle two distinct types of media:

1. **Historical Map Images:** High-resolution archival maps, street photography from local archives, and heritage documentation. These images are typically larger (2-10MB each) but accessed less frequently, making them ideal candidates for infrequent access storage tiers.

2. **User-Uploaded Photos:** Community-contributed street photos, etymology documentation, and historical images. These require fast upload/download speeds, thumbnail generation, and moderation workflows.

Recommended Storage Services

After analyzing multiple providers, three solutions emerge as optimal for this project:

Primary Recommendation: Supabase Storage

Why Supabase Storage is ideal for this project:
- **Seamless Integration:** Native integration with Supabase Auth and database, eliminating cross-service authentication complexity
- **Cost-Effective Free Tier:** 1GB storage and 5GB egress bandwidth monthly, sufficient for MVP launch
- **Row Level Security:** Built-in access control that integrates with user permissions
- **Automatic Optimization:** Smart CDN with global edge locations for fast image delivery

Technical Specifications:
- Max file size: 50MB (suitable for historical maps)
- Supported formats: JPEG, PNG, WebP, GIF, SVG
- API-driven uploads with progress tracking
- Automatic metadata extraction and indexing

Cost Structure:
- Free tier: 1GB storage + 5GB egress/month
- Pro tier: $0.021/GB storage +$ 0.09/GB egress
- Projected Year 1 cost: £0-£8/month (depending on usage)

Integration Example:

```
// Upload user photo with metadata
const { data, error } = await supabase.storage
  .from('street-photos')
  .upload(`<span class="math-inline" style="display: inline;"><math
xmlns="http://www.w3.org/1998/Math/MathML"
display="inline"><mrow><mrow><mi>s</mi><mi>t</mi><mi>r</mi><mi>e</
mi><mi>e</mi><mi>t</mi><mi>I</mi><mi>d</mi></mrow><mo>&#x0002F;</mo></
mrow></math></span>{timestamp}-${filename}`, file, {
    metadata: { streetName, contributorId, description }
  })
```

Alternative: Cloudinary (For Advanced Image Processing)

When to consider Cloudinary:
- Need for automatic image optimization and transformation

- Advanced features like AI-powered tagging and content moderation
- High-volume image processing requirements

Cost Structure:
- Free tier: 25GB managed storage + 25,000 transformations/month
- Plus plan: $99/month (25GB storage, 100,000 transformations)
- Estimated cost for 10k users: £0-£79/month

Advanced Features:
- Automatic WebP/AVIF conversion (40-60% size reduction)
- AI-powered auto-tagging and content analysis
- Advanced lazy loading and responsive image delivery
- Built-in moderation and quality scoring

Enterprise Alternative: AWS S3 (For Large Scale)

When to consider AWS S3:
- Planning for 100k+ users
- Need for multi-region replication
- Integration with other AWS services

Cost Structure:
- Free tier: 5GB storage for 12 months
- Standard tier: $0.023/GB storage,$ 0.09/GB data transfer
- Intelligent tiering: 25-40% cost savings for archival content

Storage Strategy Implementation

Phase 1: MVP Launch (Supabase Storage)
1. Configure Supabase storage buckets:
- `historical-maps` (public read, admin write)
- `user-photos` (public read, authenticated write)
- `thumbnails` (auto-generated, public read)

   1. Implement client-side image compression:
      - Compress uploads to 85% quality JPEG
      - Generate thumbnails at upload time (200px, 400px, 800px)
      - Use WebP format for modern browsers
   2. Set up storage policies with Row Level Security:

```
-- Allow users to upload photos for verified streets
CREATE POLICY "Users can upload street photos" ON storage.objects
FOR INSERT TO authenticated
WITH CHECK (bucket_id = 'user-photos' AND auth.uid() IS NOT NULL);
```

Phase 2: Growth Optimization
1. Implement progressive image loading
2. Add CDN caching headers (max-age: 31536000 for static images)
3. Monitor storage usage and optimize with WebP conversion
4. Consider migration to Cloudinary if image processing becomes complex

Cost Integration with Overall Budget

Adding storage costs to the project's budget framework:

Year 1 Updated Costs:
- MVP (0-1k users): £0-£2/month additional for storage
- Growth (10k users): £2-£8/month for storage
- Total project cost remains under £15/month including storage

Optimization Strategies:
1. **Image Compression:** Implement client-side compression to reduce storage by 25-40%
2. **Lazy Loading:** Only load images when needed to reduce bandwidth costs
3. **Smart Caching:** Use browser caching and CDN to minimize repeated downloads
4. **Archival Strategy:** Move historical maps to infrequent access tiers after 90 days

This storage strategy ensures the platform can handle both historical preservation needs and community engagement features while maintaining the project's cost-effectiveness goals. The modular approach allows for easy migration between services as the platform scales.

4. Detailed Cost Breakdown

The following is a detailed cost breakdown for the first two years of operation, with scenarios for an MVP, a growing website (10,000 monthly users), and a larger-scale platform. The costs are in GBP (£), assuming an exchange rate of £1 = $1.25.

Year 1 Cost Breakdown

Service	MVP (0-1k users/ month)	Growth (10k users/month)	Notes
Domain (.org)	£8	£8	First-year promotional price from Namecheap.
Hosting (Cloudflare Pages)	£0	£0	Free tier is sufficient.
Database (Supabase)	£0	£0	Free tier (500MB) is sufficient for the initial dataset.

Service	MVP (0-1k users/ month)	Growth (10k users/month)	Notes
Authentication (Supabase Auth)	£0	£0	Free for up to 50,000 monthly active users.
Mapping (MapTiler/ Stadia)	£0	£0	Free tiers are sufficient for this level of traffic.
Geocoding (LocationIQ/ OpenCage)	£0	£0	Free tiers are sufficient.
AI/ML (OpenAI GPT-4o mini)	£0 - £1	£1 - £5	Hybrid model keeps costs very low. Costs depend on the percentage of names needing AI analysis.
Storage (Supabase Storage)	£0 - £2	£2 - £8	Free tier (1GB) sufficient for MVP. Scales with image uploads and historical archives.
Total (Monthly)	**£0 - £3**	**£3 - £13**	
Total (Year 1)	**£8 - £44**	**£44 - £164**	

Year 2 Cost Breakdown

Service	Growth (10k users/month)	Scale-Up (50k users/month)	Notes
Domain (.org)	£13	£13	Standard renewal price from Namecheap.
Hosting (Cloudflare Pages)	£0	£0	Free tier is still likely sufficient.
Database (Supabase)	£20	£20	Upgrade to Pro plan ($25/ month) for 8GB storage.
Authentication (Supabase Auth)	£0	£0	Free for up to 50,000 monthly active users.
Mapping (MapTiler/ Stadia)	£0 - £16	£16 - £64	May need to upgrade to a paid plan (20− 80/month).
Geocoding (LocationIQ/ OpenCage)	£0	£0 - £40	May need a paid plan ($50/ month).
AI/ML (OpenAI GPT-4o mini)	£1 - £5	£5 - £25	Costs scale with usage.

Service	Growth (10k users/month)	Scale-Up (50k users/month)	Notes
Storage (Supabase Storage)	£2 - £8	£8 - £32	Pro tier storage for growing image archive and community uploads.
Total (Monthly)	**£36 - £62**	**£62 - £194**	
Total (Year 2)	**£432 - £744**	**£744 - £2328**	

Cost Assumptions and Notes:

- The costs are estimates and will vary based on actual usage.
- The AI/ML costs assume a hybrid model where only a fraction of street names require analysis by the OpenAI API.
- Storage costs include both user-uploaded photos and historical map archives. Supabase Storage's free tier (1GB) provides an excellent foundation, with costs scaling predictably as community engagement grows.
- Image optimization strategies (WebP compression, thumbnails, CDN caching) can reduce storage costs by 25-40% while improving performance.
- The "Scale-Up" scenario in Year 2 shows that even with significant growth, the total monthly cost can be kept under £200 by continuing to leverage the cost-effective services in the recommended stack.
- The budget of <£20/month for 10,000 monthly users remains achievable in Year 1 with careful storage optimization. In Year 2, due to database, mapping service, and storage upgrades, the cost will likely be in the £36-£62/month range for the same traffic.

4. Weekend MVP Setup Guide

This guide provides a high-level, step-by-step plan for a non-technical founder to launch the Minimum Viable Product (MVP) of the street name etymology website over a single weekend.

Day 1: Setup and Data Loading

Morning (2-3 hours): Accounts and Domain

1. **Register Accounts:** Create free accounts for the following services:
    - **GitHub:** To store your website's code.
    - **Cloudflare:** For hosting and DNS.
    - **Supabase:** For your database and authentication.
    - **MapTiler:** For your map tiles.
    - **OpenAI:** For the AI/ML etymology service.
2. **Buy a Domain:** Go to **Namecheap** and purchase a `.org` domain for your website (e.g., `streetnamehistory.org`).
3. **Connect Domain to Cloudflare:** In your Namecheap account, change the domain's nameservers to the ones provided by Cloudflare. This will allow Cloudflare to manage your domain's DNS.

Afternoon (3-4 hours): Database and Data

1. **Create a Supabase Project:** In your Supabase account, create a new project. This will be your database.
2. **Enable PostGIS:** In your Supabase project settings, go to the "Database" section and enable the **PostGIS** extension.
3. **Download UK Street Name Data:** Download the **OS OpenNames** dataset from the OS Data Hub. You will need the CSV file.
4. **Clean and Upload Data:**
    - Open the OS OpenNames CSV file in a spreadsheet program (like Google Sheets or Microsoft Excel).
    - Remove any columns you don't need, keeping the street name, location (latitude/longitude), and any other relevant information.
    - In your Supabase project, create a new table for your street names. Make sure the columns in your Supabase table match the columns in your CSV file.
    - Upload the cleaned CSV file to your Supabase table.

Day 2: Building and Launching the Website

Morning (3-4 hours): Website Template and Configuration

1. **Choose a Website Template:** Find a simple, free, open-source website template that includes a map. You can search on GitHub for "map website template" or similar terms. Look for templates that use MapLibre GL JS.
2. **Clone the Template to GitHub:** Once you've chosen a template, clone it to your own GitHub account.

3. **Configure the Template:**
   - **Connect to Supabase:** In the website's code, you will need to add your Supabase project URL and API key. This will allow the website to connect to your database.
   - **Connect to MapTiler:** You will also need to add your MapTiler API key to the code to load the map tiles.
   - **Connect to OpenAI:** Add your OpenAI API key for the etymology suggestions.

Afternoon (3-4 hours): Deployment and Launch

1. **Deploy to Cloudflare Pages:**
   - In your Cloudflare account, go to "Pages" and create a new project.
   - Connect your GitHub account and select the repository with your website template.
   - Cloudflare will automatically build and deploy your website.
2. **Connect Your Custom Domain:** In your Cloudflare Pages project settings, add the custom domain you purchased from Namecheap.
3. **Test Your Website:** Once deployed, visit your domain and test the website. Make sure the map loads, you can see the street names from your database, and the etymology suggestions are working.

Congratulations! You have launched your MVP.

5. Scalability Roadmap

The initial MVP is built on free tiers and is designed to handle a modest amount of traffic. As your website grows, you will need to scale your infrastructure. Here is a roadmap for scaling from the MVP to 10,000+ monthly users.

Stage	Trigger	Action	Estimated Cost
1. MVP	0 - 1,000 users/ month	All services on free tiers.	£0 - £1/ month
2. Growth	1,000 - 10,000 users/ month	* Monitor Supabase database size. If approaching 500MB, prepare to upgrade.	
* Monitor MapTiler/Stadia Maps usage.	£1 - £5/ month		

Stage	Trigger	Action	Estimated Cost
3. Scaling	10,000 - 50,000 users/ month	* **Upgrade Supabase:** Move to the Pro plan ($25/month) for 8GB of storage.	
* **Upgrade Mapping Service:** If you exceed the free tier limits, upgrade to a paid plan on MapTiler (25/ month)orStadiaMaps(20/month).	£34 - £54/ month		
4. High Traffic	50,000+ users/ month	* **Optimize Database Queries:** Ensure your database queries are efficient to handle the increased load.	
* **Implement Caching:** Use Cloudflare's caching features to reduce the load on your database and API functions.			
* **Consider a Higher Tier Mapping Plan:** Depending on traffic, you may need to upgrade to a higher-tier plan on MapTiler or Stadia Maps.			
* **Scale AI/ML:** If AI/ML usage is high, consider optimizing your prompts or using a dedicated inference endpoint on a service like Hugging Face.	£54 - £162+/ month		

Key Scaling Considerations:

- **Database:** Supabase offers seamless scaling. You can upgrade your plan with a few clicks as your data grows.
- **Hosting:** Cloudflare Pages is built for scale and will handle traffic spikes without any issues.
- **Cost Management:** Keep an eye on your usage for each service. Most services have dashboards where you can monitor your consumption and set alerts to avoid unexpected costs.

6. Data Strategy

A robust data strategy is crucial for the success of the website. The strategy should cover both the initial integration of UK open data and the ongoing expansion through community contributions.

UK Open Data Integration

1. **Primary Data Source:** The initial dataset will be **OS OpenNames** from the Ordnance Survey, which contains over 870,000 named and numbered roads in Great Britain.
2. **Data Enrichment:** The primary dataset will be enriched with other UK open data sources:
   - **Postcodes: Code-Point Open** will be used to add postcode information.
   - **Administrative Boundaries: OS Boundary-Line** will be used to add information about local authorities and other administrative areas.
   - **Historical Information: Historic England's National Heritage List for England (NHLE)** will be used to identify streets with historical significance.
3. **Data Licensing and Attribution:** All data will be used in accordance with the **Open Government Licence (OGL)** and other applicable licenses. Proper attribution will be given to all data sources on the website.
4. **Data Processing:** A data processing pipeline will be created to:
   - Download the latest versions of the datasets.
   - Clean and normalize the data.
   - Combine the different datasets.
   - Load the final, clean dataset into the Supabase database.
     This pipeline should be automated to run quarterly to coincide with the update schedule of OS OpenNames and Code-Point Open.

Community-Driven Expansion

The website will be designed to grow through community contributions. This will allow the dataset to expand beyond the initial UK data and to be enriched with local knowledge.

1. **Contribution Features:** Users will be able to:
   - Suggest etymologies for street names.
   - Provide historical context and sources.
   - Upload historical photos of streets.
   - Suggest new street names to be added to the database (especially for areas outside the UK).
2. **Moderation:** All community contributions will be subject to moderation before being published. A simple moderation queue will be built into the application, where administrators can review and approve submissions.
3. **User Roles:** A simple role-based system will be implemented:
   - **User:** Can submit suggestions.
   - **Moderator:** Can review and approve submissions.
   - **Admin:** Can manage users and the website.

4. **Data Quality:** To maintain data quality, the following measures will be taken:
    - Require users to provide sources for their contributions where possible.
    - Implement a voting or rating system for etymology suggestions, allowing the community to identify the most plausible explanations.
    - Regularly review and clean the community-contributed data.

This hybrid approach, combining authoritative open data with community-driven content, will create a rich and ever-growing resource for street name history and etymology.

7. Implementation Timeline

This timeline provides a week-by-week plan for developing and launching the street name etymology website.

Week	Key Activities	Goal
Week 1	**Project Setup & Data Foundation**	- Register all necessary accounts (GitHub, Cloudflare, Supabase, etc.). - Purchase domain and configure DNS. - Set up Supabase project and load initial UK open data.
Week 2	**Website Frontend Development**	- Choose and configure a website template. - Develop the main user interface, including the interactive map and street information pages.
Week 3	**Backend and API Development**	- Develop serverless functions to fetch data from the database. - Integrate the mapping service (MapTiler/Stadia Maps) and geocoding service (LocationIQ/ OpenCage).
Week 4	**AI/ML Integration & Community Features**	- Develop the hybrid AI/ML service for etymology suggestions. - Implement user authentication (Supabase Auth). - Develop the community contribution and moderation features.
Week 5	**Testing and Deployment**	- Thoroughly test all features of the website. - Deploy the website to Cloudflare Pages. - Test the live website.
Week 6	**Launch and Promotion**	- Announce the launch of the website. - Promote the website through social media and other channels. - Monitor the website for any issues.

Week	Key Activities	Goal
Ongoing	**Maintenance and Improvement**	- Regularly update the data. - Review and approve community contributions. - Add new features based on user feedback.

8. Backup & Monitoring

Even with a low-cost, low-maintenance architecture, it is important to have a plan for backing up your data and monitoring your services.

Backup Strategy

- **Database:** Supabase provides daily automated backups for projects on the Pro plan. For the free plan, you can manually back up your data by exporting your tables to CSV files from the Supabase dashboard. You should perform manual backups at least weekly.
- **Code:** Your website's code is stored in your GitHub repository. GitHub itself is a very reliable backup.
- **Cloudflare Configuration:** Your Cloudflare Pages configuration is stored in your Cloudflare account. You can also keep a record of your settings in a document.

Monitoring Strategy

- **Uptime Monitoring:** Use a free uptime monitoring service like **Uptime Robot** to get alerts if your website goes down. Uptime Robot offers 50 monitors with 5-minute checks for free.
- **Usage Monitoring:** Regularly check the dashboards of your services (Cloudflare, Supabase, MapTiler) to monitor your usage. This will help you to anticipate when you might need to upgrade to a paid plan and to avoid unexpected costs.
- **Error Tracking:** Use a free error tracking service like **Sentry** to get notified of any errors that occur in your frontend or backend code. Sentry has a free plan for developers that is suitable for the MVP.

9. Risk Mitigation

This section outlines potential risks and how to mitigate them.

Risk	Mitigation
Data Quality Issues	- Implement a data cleaning and validation process as part of your data loading pipeline. - Encourage users to report any data errors they find. - Regularly review community-contributed data.

Risk	Mitigation
Service Downtime	- Use reliable services with good uptime records (Cloudflare, Supabase, etc.). - Use an uptime monitoring service to get notified of any downtime immediately.
Unexpected Costs	- Regularly monitor your usage of all services. - Set up billing alerts where possible. - Choose services with predictable pricing and generous free tiers.
Security Vulnerabilities	- Keep your website's code and dependencies up to date. - Use a web application firewall (WAF), which is included in Cloudflare's free plan. - Follow security best practices for user authentication and data handling.
Low Community Engagement	- Actively promote the website to attract users. - Make it easy for users to contribute. - Engage with your community through social media and other channels.
Vendor Lock-in	- The proposed architecture is highly decoupled, which reduces vendor lock-in. For example, it would be relatively easy to switch from Supabase to another PostgreSQL provider, or from MapTiler to another mapping service.

By proactively addressing these risks, you can increase the chances of your street name etymology website being a long-term success.