

## Phase 1: Foundational Setup (MVP)

**Goal:** Create a simple, functional website displaying bachata festivals worldwide, leveraging existing data (Google Sheets) and enabling basic filtering.

### 1. Data Gathering & Management

#### ○ Data Source Integration:

- Start with the Google Sheet containing festival info (festival name, location, date, ticket price range, promo codes, website link).
- Use the Google Sheets API (free tier) to directly fetch data dynamically, so you can update the sheet and see changes reflected on the site.

#### ○ Data Validation & Cleanup:

- Ensure festival entries have consistent formatting and mandatory fields (name, location, date, link).
- For missing data, manually fill some initial key festivals for major regions (e.g., Europe, North America, Asia).

### 2. Basic Web App (Static Site + JS)

#### ○ Technology Stack:

- Frontend: Basic HTML/CSS/JS or a lightweight framework like Vue.js or React (hosted on free GitHub Pages or Netlify free tier).
- No backend server necessary at this stage—just client-side requests to the Google Sheets API (Google Sheets as a “database”).

#### ○ Core Features:

- Festival Listing Page: Show a simple table or card layout of festivals with name, location, date, price range, and link to the festival’s official page.
- Basic Filters: Filter by region (continent), month, and price range.

#### ○ Map Integration (Optional at MVP):

- Start with a link: Each festival entry can have a “View on Map” link using a simple Google Maps query URL. This requires no API key at first.

### 3. Internationalization (Basic)

- Start with English only. Later, add a language switcher (e.g., English + Spanish) by using translation files (JSON) and a simple library like i18next if needed.
- Initially keep it simple to avoid complexity.

### 4. User Feedback for Missing Festivals

- Add a Google Form link or a simple form (that emails you) where users can suggest missing festivals. No login needed at this point.
- All free tools (Google Forms for collecting submissions).

## Cost Considerations:

- Hosting on GitHub Pages or Netlify: Free.
- Google Sheets API: Free usage for initial scale.
- Development Tools: Free (your time + open-source frameworks).

This gets you a working MVP quickly and cheaply.

---

## Phase 2: Enhanced User Experience & Basic Accounts

**Goal:** Add login, saved favorites, basic personalization, and improved map integration.

### 1. User Accounts & Authentication

- Introduce simple email/password or social login with a free-tier authentication service like Firebase Auth.
- Store user preferences (e.g., preferred region or budget range) in Firebase's free tier database (Firestore).

### 2. Favorites & Basic Personalization

- Logged-in users can "star" or "favorite" festivals to save them.
- Keep track of user's past favorites, so on return visits they see relevant suggestions first.
- Allow users to mark festivals as "Going," "Attended," or "Interested." This expands beyond just "favorites" and becomes a mini event manager.
- Display these categories in the user's profile page.
- Recommendations can now consider the user's event preferences and history (e.g., "You attended these festivals last year, here are similar ones.")

### 3. Map Integration

- Integrate a map component on the homepage using a free mapping solution:
  - Use Leaflet.js (open source) + OpenStreetMap data to show festival markers on a map.
  - When user clicks a marker, display festival info in a tooltip or popup.
  - Map Integration: Ensure full venue addresses for accurate map pins.

### 4. Filters & Search Improvements

- Include a dedicated section on each festival's detail page to show the instructor/artist lineup. This will also allow filtering by specific artists or instructors.
- Add advanced filtering: by instructor lineup, date range, country/city.
- Add a basic search bar that queries festival name or location.

### 5. Promo Codes Integration

- Highlight festivals that have known promo codes in the listing.
- Let authenticated users copy promo codes easily. For the MVP stage of this, just show the code from the sheet.
- As a future step, consider a small database table or Firestore collection to store and validate promo codes if needed.

## Cost Considerations:

- Firebase Auth & Firestore free tiers are generous enough for early prototypes.
- Leaflet + OSM: Free and open source.

---

## Phase 3: Data Expansion & Semi-Automation

**Goal:** Improve the data pipeline, add missing festivals more easily, allow scraping, and start exploring premium features.

### 1. Data Source Automation

- Use simple Python scripts + Google Cloud Functions (on a free tier) or GitHub Actions to scrape known festival listing websites or social media event pages to auto-update your festival database.
- Integrate suggestions from users (via Google Forms) directly into the sheet or a moderation dashboard.
- Automate data sourcing: Use a scheduled Cloud Function or GitHub Action to scrape a few trusted festival listing sites every week. Validate data before adding it to the main database.
- Implement an in-app calendar view for each user where they can see all the festivals they're 'Going,' 'Attended,' or 'Interested' in. Add an 'Export to Google Calendar' button for each event or for the entire set of selected festivals.
- Improve calendar in-app

### 2. Admin/Moderator Dashboard

- A simple admin panel (password-protected) for you to approve new festivals and edit existing entries.
- User-submitted festival suggestions go into a moderation queue in the admin dashboard. Once approved, they appear on the site
- This can be built on Firebase UI or a simple password-protected page.
- Add tagging for "official promoter" or "community-submitted" festivals.

### 3. Improved Language Support

- Add a language toggle with translations for key UI elements into Spanish and maybe French (popular languages in these communities).
- Store translation JSON files and load dynamically based on user selection.

### 4. Basic Recommendation Logic

- Implement a rules-based recommendation engine:
  - For example: if user's favorite festivals are mostly in Europe, prioritize showing European festivals coming soon.
  - If a user sets a preference for price range or festival size, show those first.
  - User preferences (e.g., location, budget).
  - Festival popularity (most viewed/bookmarked).
  - Filters (e.g., by dates, styles, beginner-friendly, with promo-codes).

## Cost Considerations:

- Slight increase in complexity with Cloud Functions or small hosting fees if the free tier is exceeded, but still minimal.

---

## Phase 4: Premium Tier & Monetization

**Goal:** Introduce premium features, affiliate links, and possibly a paid subscription model.

### 1. Premium Features

- Offer advanced festival recommendations powered by a more sophisticated algorithm (e.g., using a free-tier ML service or a simple local model).
- Possibly add a feature like a chatbot (fine-tune a model on festival data) to suggest which festivals to attend based on user's dance style, date availability, budget, and location. The chatbot could be a premium feature.
- Premium users can get alerts when new promo codes are added or when discounted passes become available.

### 2. Affiliate & Ticket Sales Integration

- Where possible, integrate affiliate links for ticket sales. Earn a small commission if users buy through your links.
- Add a "Buy Now" or "Discounted Passes" section that leads to partner sites.
- This might require reaching out to festival organizers or ticket platforms to form partnerships.

### 3. Ads (Low-Key)

- Introduce non-intrusive ads (e.g., a banner on the bottom) from Google AdSense or similar.
- Ensure ads are relevant or at least not disruptive to user experience.

---

## Phase 5: Mobile App Conversion (If Successful)

**Goal:** Provide a native mobile experience, potentially boost user engagement and convenience.

### 1. Cross-Platform App Development

- Use a framework like React Native or Flutter to build a mobile app.
- Reuse existing backend and database. Connect via APIs.

### 2. Push Notifications & Offline Mode

- Send push notifications about new festivals, promo codes, or last-minute discounted tickets to premium users.
- Offer offline caching of festival data so users can browse without an internet connection, helpful when traveling.

---

## Marketing & Growth Strategy (Ongoing Throughout Phases)

### 1. Community Outreach

- Share your tool in bachata Facebook groups, WhatsApp groups, and dance community newsletters.
  - Reach out to known bachata instructors or influencers to mention the platform.
  - 2. Early Feedback & Iteration**
    - Encourage early adopters to give feedback. Regularly improve UI/UX and add the most requested features.
    - Add basic site analytics (e.g., Google Analytics or Plausible) early on to understand user behavior and inform future feature decisions
  - 3. Organizer Self-Service Portal (Future)**
    - Eventually allow festival organizers to claim their event listing and update info, add their promo codes, and possibly pay for “Featured” placement on the festival map or listings page.
    - This can be a source of revenue and an incentive for organizers to keep the data accurate.
    - In the future, organizers can register to update their own festival details and pay for featured listings. This enhances data accuracy and creates a revenue opportunity.
    - Should the platform grow significantly, consider a dedicated backend (Node.js/Django) and a more robust database (e.g., Firestore or Postgres) for handling complex queries and higher traffic volumes.
  - 4. Partnerships**
    - Once established, approach festival organizers, instructors, and promoters to add exclusive promo codes or affiliate deals.
    - Offer special listings or highlight top festivals in exchange for collaboration.
- 

## Scalability & Future Plans

- 1. Scaling Infrastructure**
    - If user load grows, move from client-side fetching to a dedicated backend using Node.js or Python + a managed database (e.g., Firestore or a small Postgres instance on a low-cost cloud provider).
  - 2. Advanced Personalization**
    - Implement machine learning-based recommendations if it becomes beneficial.
    - Analyze user behavior to improve suggestions and refine premium offerings.
  - 3. Expanding Scope**
    - After establishing a solid Bachata festivals database, consider adding other Latin dances if it makes sense and your user base requests it.
- 

## Summary of Costs & Tools

- **Phase 1 & 2:** Almost entirely free-tier (GitHub Pages/Netlify, Firebase free tier, Google Sheets/API).
- **Phase 3+:** Minimal costs if scaling up (basic hosting, domain name, maybe a small budget for scraping tools or map API keys if usage grows).
- **Monetization:** Affiliate links, premium subscriptions, mild ad placement, eventually organizer partnerships.