

AIRTABLE DATABASE DESIGN GUIDE

Free-Florida-Free Application // Backend Architecture & Setup

Rodney Biddle LLC • Rev 1.0 • Internal Technical Document

1. Platform Overview

Airtable is a cloud-based platform that blends the simplicity of a spreadsheet with the power of a relational database. Data is organized into bases, which contain tables of records — each row being a record with customizable field types including text, numbers, dates, attachments, checkboxes, dropdowns, linked records, and more.

Teams can switch between multiple views of the same data — Grid, Kanban, Calendar, Gallery, and Gantt — without altering the underlying records.

1.1 Relational Linking

Tables within a base can reference each other, enabling normalized data structures without SQL. A Products table can link to Vendors and Orders tables; lookup and rollup fields pull summarized data across those relationships automatically. This makes Airtable a viable lightweight CMS or operational database for small-to-mid-size workflows.

1.2 Automation & Integration

Airtable includes a built-in automation and integration layer. Triggers based on field changes or schedules can fire actions such as sending email, creating records, or calling webhooks. Its REST API is well-documented and serves as a common data source for orchestration tools including n8n, Zapier, and Make.

Tier	Key Features	Limits
Free	2 bases, basic automations, API access	1,000 records/base
Team	Unlimited bases, revision history, permissions	50,000 records/base
Business	Advanced RBAC, audit log, SAML SSO	125,000 records/base

2. Project Overview

This document defines the Airtable base architecture that will serve as the backend for the Free-Florida-Free web application. The database tier must satisfy the following design requirements:

Cost	Operate within Airtable Free tier — supports up to 2 bases per account.
Access	All tables must be accessible via REST API using Personal Access Tokens or OAuth tokens.
Security	Moderate security posture required. No PII will be stored; high-security controls are out of scope.
Formulas	Leverage Airtable formula fields to reduce application-layer logic.
Automations	Simple record-triggered automations only — no complex multi-step flows required.
Interface	All data management performed through the Airtable web UI. No SQL required.
Forms	Native Airtable Form views used for Contact Us and Suggest a Venue inputs.

3. Table Definitions

All tables are to be provisioned within the primary base and populated via CSV import during initial setup. The following tables are required:

Table Name	Description
pages	Website page content including title, subtitle, slug URL, header image URL, and SEO/Social Media meta tags.
blogs	Blog post records. Stores article content if the blog feature is enabled.
contactus	Inbound contact form submissions received from site visitors.
places	Directory of Florida locations to be displayed and filtered on the Home page.
users	Accounts permitted to authenticate into the application.
banners	Banner images assigned per page with configurable sort order.
email_out	Outbound email queue. Records are created by automation and trigger email delivery.

4. Form Configuration

Two Airtable Form views are required. Forms are native to Airtable and map directly to table records — no external form library is needed.

4.1 Contact Us Form

Implemented as a Form view on the contactus table. Form submission creates a new record and triggers the Contact Us automation (see Section 6.1). This form will be styled and embedded via vibe-coding.

4.2 Suggest a Venue Form

Implemented as a Form view on the places table (or a staging variant). Allows community members to submit venue suggestions for review before publication. This form is designed and managed entirely within the Airtable interface.

5. Page Inventory

The following pages will be constructed during the vibe-coding phase. Each page record should be pre-provisioned in the pages table so that slugs, SEO tags, and meta content are available to the application at build time.

Page	Notes
Home	Displays the places directory with filter controls: Free/Cheap, County, and Region (Panhandle, North Florida, Central Florida, East Coast, West Coast, South Florida).
Map	Renders a map using the address data from the places table.
About	Describes the project mission and lists the producers.
Privacy	Contains the site privacy statement.
Terms of Service	Legal terms governing site usage.
Contact	Introductory text for the Contact page; form is embedded separately.

6. Formula Reference

The following formula fields should be configured in their respective tables. All formulas are native to Airtable and require no external scripting.

6.1 Slug — pages table

Generates a URL-safe slug from the title field by lowercasing all characters and replacing non-alphanumeric sequences with hyphens.

```
LOWER(  
  REGEX_REPLACE(  
    {title},  
    "[^a-zA-Z0-9]+",  
    "-"  
  )  
)
```

6.2 RecID — all tables

Returns the unique Airtable record identifier for the current row. Useful as a stable foreign key when cross-referencing tables or integrating with external systems.

```
RECORD_ID()
```

6.3 Msg — contactus table

Composes a formatted notification string from form submission fields. This value is written to the email_out table by automation to populate the email body.

```
"A new message has been received\n\n"  
& "From: " & {Name}  
& "\nEmail: " & {email}  
& "\nSubject: " & {subject}  
& "\nMessage: " & {message}
```

7. Automation Definitions

Two automations are configured within the Airtable base. Both are trigger-action flows managed natively in Airtable — no external automation platform is required for these specific workflows.

7.1 Contact Us → email_out

Whenever a new record is created in the contactus table (i.e., a visitor submits the Contact Us form), this automation copies the record to the email_out table and queues it for outbound email delivery.

TRIGGER	ACTION 1	ACTION 2
When record is created	Create record in email_out	Update record (status flag)

7.2 email_out → Send Email

When a new record is created in email_out, Airtable sends an outbound email using the field values populated in Action 1 above. A record update step marks the record as sent to prevent duplicate delivery.

TRIGGER	ACTION 1	ACTION 2
When record is created	Send email (via Airtable)	Update record (mark sent)

END OF DOCUMENT — Rodney Biddle LLC — Internal Use Only