

# Wedding Website – Development Roadmap

## (Starting with UI Development)

**Stack lock-in:** React + Tailwind + TypeScript on Cloudflare Pages; Pages Functions (TypeScript) for APIs; Cloudflare D1 (SQLite) for data; Cloudflare R2 for photos; Resend for emails; Cloudflare Turnstile for spam protection; Cloudflare Access for `/admin`.

---

### 0) Objectives & Success Criteria

- **Core goals**
    - Guests can RSVP easily.
    - Guests can upload photos.
    - Automated reminders for pending RSVPs.
    - Admin tools for exports and moderation.
  - **Success metrics**
    - $\geq 95\%$  RSVP completion.
    - Photo upload success  $\geq 99\%$ .
    - Zero production data loss.
- 

### 1) Architecture Overview

- **Frontend:** React (Vite) + Tailwind, static build.
  - **Backend/APIs:** Cloudflare Pages Functions.
  - **Data:** D1 (households, guests, photos metadata, email log).
  - **Files:** R2 bucket for photos.
  - **Email:** Resend.
  - **Security:** Turnstile + Cloudflare Access.
  - **Scheduling:** Cron Triggers for reminders.
- 

### 2) Environments & Config

- **Dev:** Preview deploys with dev DB + bucket.
  - **Prod:** Main branch with prod DB + bucket.
  - **Secrets:** `EMAIL_API_KEY`, `FROM_EMAIL`, `TURNSTILE_*`.
-

### 3) Project Phases & Milestones

#### Phase A — UI Foundation (MVP Skeleton)

**Deliverables** - React + Tailwind app scaffolded. - Navigation shell with Home, RSVP, Photos, Admin pages. - Placeholder components for RSVP form, photo upload, gallery, and admin. **Tasks** - [ ] Setup Vite + Tailwind project. - [ ] Create `App` shell with navigation. - [ ] Basic responsive layout and footer. - [ ] Add stub pages with mocked data. **Acceptance Criteria** - Preview deploy renders all pages with placeholder UI.

#### Phase B — RSVP UI (Frontend-first)

**Deliverables** - `/rsvp` page with lookup form and editable guest list UI (mocked). **Tasks** - [ ] Lookup form (household code OR last name + ZIP). - [ ] Guest table with RSVP dropdown + meal choice. - [ ] Local mock data used for state changes. **Acceptance Criteria** - RSVP flow works end-to-end with mock data only.

#### Phase C — Photos UI (Frontend-first)

**Deliverables** - `/photos` page with upload form and gallery grid (mocked). **Tasks** - [ ] Upload component (drag/drop or file input). - [ ] Gallery grid with lazy load + lightbox (mock data). - [ ] Validation on file type/size (client-side only for now). **Acceptance Criteria** - Photos page feels complete with fake uploads/gallery.

#### Phase D — Admin UI (Frontend-first)

**Deliverables** - `/admin` dashboard with tabs (Guests, Photos, Exports). - Mocked tables and export buttons. **Tasks** - [ ] Create Access-protected admin page shell. - [ ] Guests table view (mocked data). - [ ] Photos moderation view (mock toggles). - [ ] Export buttons (no backend yet). **Acceptance Criteria** - Admin dashboard renders with placeholder content.

#### Phase E — Backend Foundations

**Deliverables** - D1 schema created and applied. - R2 bucket created and wired. - Wrangler config with bindings. **Tasks** - [ ] Write `schema.sql`. - [ ] Apply to dev DB. - [ ] Test R2 put/get via function. **Acceptance Criteria** - D1 + R2 usable in preview env.

#### Phase F — Wire RSVP to Backend

**Deliverables** - `/api/rsvp/lookup` + `/api/rsvp/update`. - RSVP page integrated with real data. - Emails sent to you on update. **Tasks** - [ ] Functions implemented. - [ ] UI calls APIs and handles responses. - [ ] Resend integration for notifications. **Acceptance Criteria** - RSVP flow works with real DB + email.

#### Phase G — Wire Photos to Backend

**Deliverables** - Photo upload streams to R2. - Metadata recorded in D1. - Gallery shows approved photos. **Tasks** - [ ] Implement `/api/photos/presign` + `/api/photos/record`. - [ ] Integrate UI to call Functions. - [ ] Admin approve toggle updates D1. **Acceptance Criteria** - Photo upload + moderation works end-to-end.

## Phase H — Reminder Emails (Cron)

**Deliverables** - Daily cron triggers reminders. - Logged in `email_log`. **Tasks** - [ ] Scheduled Function queries D1. - [ ] Sends via Resend. - [ ] Logs entries. **Acceptance Criteria** - Cron sends test reminders correctly.

## Phase I — UX Polish & SEO

**Deliverables** - Final design tweaks. - SEO meta + social cards. - Registry/FAQ/Schedule pages. **Tasks** - [ ] Style refinements. - [ ] Add content pages. - [ ] Add OpenGraph, favicon, analytics. **Acceptance Criteria** - Lighthouse  $\geq 95$  across metrics.

## Phase J — Reliability & Backups

**Deliverables** - Nightly D1 exports to R2. - Restore tested. **Tasks** - [ ] Backup script. - [ ] Document restore. **Acceptance Criteria** - Backups verified.

---

## 4) Security & Testing

- Turnstile on forms.
  - Access on admin.
  - Input validation.
  - Unit + integration tests with mock + dev DB.
  - E2E RSVP/photo flows with Cypress.
- 

## 5) Launch Checklist

- Domain + HTTPS active.
  - Prod env vars set.
  - Admin Access working.
  - RSVP + photos + reminders functional.
  - Backups working.
  - SEO + analytics live.
- 

## 6) Stretch Goals

- QR codes prefill RSVP.
- Magic link RSVPs.
- HEIC  $\rightarrow$  JPEG conversion.
- Guestbook comments.
- Multi-language.
- Download-all photos.

