



Birthday Passport

What I'm working on — project summary

An iPhone app that turns real local birthday freebies into a collectible adventure: discover what's active near you, plan a route for the day, and stamp what you claimed in your passport — without the feel of a random coupon dump.

The idea

People love birthday freebies, but finding them means digging through Instagram lists and outdated blog posts, then guessing which ones still work. Birthday Passport curates **verified** offers, tells you what to do ahead of time (sign up for a rewards app, bring ID), helps you plan an efficient day, and lets you keep a fun record of your "birthday haul."

Who it's for	Adults (and families) with a birthday coming up in the next ~45 days
Launch area	Phoenix East Valley — Queen Creek, Gilbert, Chandler, Mesa, San Tan Valley
Focus	Coffee, desserts, food and local spots; quality over quantity (20–40 great offers)
Business model	Free at launch. Premium features and clearly-labeled featured listings are later ideas

What's built so far

Piece	What it does
iOS app (SwiftUI)	Onboarding, a Today dashboard that changes as your birthday nears, Explore with search/filters, Map , offer details with trust signals, save / claim (passport stamps), birthday route Plan , My Birthday hub and recap, submit-a-deal and report-an-update forms, reminders, themes and settings.
Backend (Supabase)	Live offer catalog that syncs to the app (so offers update without an App Store release), plus intake for user submissions and reports, with community "trust scoring" and auto-hiding of offers that get flagged.
Discovery pipeline (Python)	Scheduled weekly job that checks an allowlist of sources and runs web searches to find <i>candidate</i> offers. It never publishes automatically — everything goes into a review queue.
Admin tools	A local admin dashboard plus a shareable web version (Netlify) so a second person can help review candidates, split up "list" articles, and publish offers.
Docs & launch prep	Product brief, audit, catalog schema, content curation playbook, analytics and notification plans, draft privacy policy & terms, TestFlight tester script, App Store checklist, app icon.

By the numbers

~12,000	lines of Swift across ~58 app files
6 suites / 50	unit tests (availability windows, route planning, search, catalog, trust scoring)
0	third-party app dependencies — Apple frameworks only
25	offer candidates found in the first discovery run

Principles I'm building around

- **Trust first.** Never make up offer terms; link official sources; show a "confirm before you go" note.
- **Privacy-light.** No accounts. Profile, saves and stamps stay on your phone; location is only used while the app is open.
- **Honest about coverage.** A small, accurate catalog beats a big, unreliable one. No fake nearby results.

Where it stands & what's next

The app experience, backend and review tooling are largely built. The remaining work is mostly content and launch logistics:

- Curate and verify the first real set of East Valley offers (the review pipeline is in place for this).
- Host the privacy policy and terms, and set up a support email.
- Set up App Store Connect, upload a build, and run a TestFlight beta.
- Collect tester feedback, then submit to the App Store.

Summary based on project files last updated July 2026.