

Codex Prompt: BookWords Black-and-White Mobile Wireframe Prototype

Prepared prompt for building BookWords inside the existing MINERVA Next.js GitHub repo. This PDF is intended to be copied into Codex or used as a reference while starting the build.

Codex Prompt

You are working inside my existing MINERVA Next.js GitHub repo.

I want you to create a new standalone mobile wireframe prototype called BookWords.

BookWords should live inside the existing repo, but it should be visually and functionally standalone. Do not change the existing MINERVA reader, library, auth, Supabase behavior, database schema, environment variables, or current app flows.

Before making changes, inspect the repo structure and determine whether the project uses the App Router or Pages Router. Use whichever routing system already exists. Do not force a new routing pattern.

Create a new route such as:

```
/bookwords
```

Use BookWords-specific components only, ideally in a contained folder such as:

```
/components/bookwords
```

or another structure that fits the existing repo.

This is a black-and-white mobile wireframe prototype, not production code. The goal is to validate the user flow and screen structure before visual design, Figma polish, OCR, AI, payments, login, or real camera integration.

GitHub / Codex / Security Instructions

Do not ask me for my GitHub password.

Do not ask me for or modify any secrets, API keys, access tokens, or .env values.

Do not use or request:

- GitHub password
- OpenAI API key
- Supabase anon key
- Supabase service role key
- Vercel token
- Stripe/payment secrets
- Google OAuth client secret
- Amazon login credentials
- Shipping/vendor credentials
- Any other private credential

This task should not require credentials because it uses mocked data only.

If running in Codex Cloud, assume GitHub access is handled through the official GitHub authorization/connector flow, where Codex can work with selected repositories and create pull requests.

If running locally, assume I have selected the local MINERVA repo folder or Git repository in the Codex app/CLI.

Use a branch or isolated working changes. Do not commit directly to main unless I explicitly ask.

If possible, create or recommend a branch name like:

feature/bookwords-wireframes

Before making edits

- Inspect the repo structure.
- Identify the routing system.
- Identify relevant styling conventions.
- Confirm the files you intend to add/change.
- Keep the scope limited.

After making edits

- Summarize files created or changed.
- Explain how to run/view /bookwords.
- List supported click paths.
- List what remains mocked.
- List any assumptions.
- Do not claim real OCR, AI, login, payments, shipping, or camera functionality exists.

Product Concept

BookWords is a free mobile app for parents reading physical books with their children.

The app helps parents quickly create vocabulary flashcards from unknown words while reading a real, physical book with a child.

Example use case

A parent and 10-year-old child are reading Harry Potter. The child does not know the word "craning." The parent pauses reading, points to the word on the physical page, opens BookWords, takes a photo of the page while pointing to the word, sees a tiny confirmation that says "Saved: craning," closes the phone, and continues reading.

The app's purpose is not to become the reading experience. The purpose is to make the phone disappear as quickly as possible.

Core Product Principle

The parent-child reading experience is sacred.

The app must not distract from the physical book. It should behave more like the native iPhone Camera app than an education app.

The user should feel:

I can capture this unknown word in two seconds and get back to reading.

Primary User

The parent is the primary user.

The child is not meant to hold the phone or use the app during reading. The child is reading the physical book. The parent owns the phone, captures the word, and later reviews the flashcards with the child.

Do not design this like a kid screen-time app.

Non-Negotiable Design Rules

1. App opens directly to active camera.
2. No login before first capture.
3. Parent is the primary user.
4. Child stays focused on the physical book.
5. Capture must be extremely fast.
6. Camera stays live after photo.
7. Confirmation shows exact detected word, such as: "Saved: craning."
8. AI creates the flashcard automatically.
9. Most high-confidence cards become Ready without parent work.
10. Wrong word capture is not acceptable as normal behavior.
11. Needs Review exists only as a backup path.
12. No games, badges, streaks, avatars, coins, or noisy animations.
13. Free digital flashcard review on phone.
14. Paid physical 4x6 flashcards: 10-card pack for \$9.99.
15. Photos are temporary and deleted after card approval.
16. No notifications in MVP.

What This App Must Not Become

- Not a game.
- Not a kid screen-time app.
- Not a noisy app with badges, streaks, coins, avatars, or animations.
- Not a full e-reader.
- Not a chatbot during reading.
- Not a replacement for the parent reading with the child.
- Not a generic flashcard app.
- Not a photo archive.

MVP Scope

The MVP is:

Capture -> AI-generated draft card.

The parent takes a photo while pointing to an unknown word. The app identifies the word and creates a draft flashcard automatically.

However, the wireframe prototype should also include the near-future flows that will quickly follow:

- Card Library
- Mobile flashcard review
- Edit/regenerate card
- Order physical 4x6 cards
- Account/settings

This prototype should show the full expected product flow, but use mocked data only.

Technical Boundaries

Do not build real functionality yet.

Use mocked state and placeholder UI.

Do not add:

- Real OCR
- Real AI calls
- Real camera integration
- Real Supabase schema changes
- Real login
- Real payment
- Real shipping
- Real notification system

Use simple local mocked data in React state.

The camera screen should be a placeholder wireframe that looks like a live camera view. It does not need to access the device camera.

Required Screens

- Launch / Active Camera Screen
- Capture Confirmation State
- Card Library
- Mobile Flashcard Review
- Card Detail / Edit / Regenerate Flow
- Order Physical Cards Flow: Select Cards, Preview Physical Cards, Shipping Address, Payment, Order Confirmation
- Account / Settings

Use iPhone-sized mobile layout. Keep everything black, white, gray, and wireframe-like. No decorative polish yet.

Screen 1: Launch / Active Camera Screen

The /bookwords route should open directly into the active camera wireframe.

Do not show:

- Dashboard
- Login wall

- Onboarding
- Child selector
- Book selector
- Account setup
- Payment setup

The parent should land directly on a camera-like screen.

Required elements:

- Full-screen camera placeholder
- Large capture button at bottom center
- Flash toggle
- Small X/back button
- Optional quiet account/library access
- Instruction text: "Point to the word and take photo"
- Manual word entry fallback

The screen should feel like:

Open app -> camera already active -> point at word -> take photo -> continue reading.

Screen 2: Capture Confirmation State

When the parent taps the capture button:

- The live camera view should remain visible.
- Do not show a modal.
- Do not navigate away.
- Do not interrupt the reading moment.
- Show a small bottom-left thumbnail or saved-word chip, similar to the iPhone Camera thumbnail.

The confirmation should show the exact detected word:

Saved: craning

This matters because if the app detects the wrong word, the parent can immediately retake the photo.

For this prototype, mock the captured word as:

craning

Clicking the saved-word thumbnail should open the card detail/edit screen.

Screen 3: Card Library

The Card Library should be simple and parent-friendly.

Use a simple list grouped by status:

- Needs Review
- Ready
- Ordered

Status	Meaning
--------	---------

Needs Review	App is uncertain, word may be wrong, or card needs parent attention
Ready	Card is prepared and usable for phone review or physical card ordering
Ordered	Card has already been included in a mailed physical card order

High-confidence AI-generated cards should become Ready automatically. Uncertain cards go to Needs Review.

The Card Library should not feel like a database or admin system.

Each list item should show:

- Word
- Short definition preview
- Status
- Edit option
- Review option if Ready

Search/filter can be shown as a future secondary feature, but do not make it central.

Screen 4: Mobile Flashcard Review

The free phone review flow must be included.

This is used by parent and child together after reading, the next day, or a few times per week.

Use a calm, parent-led flashcard stack.

Behavior:

- Front of card shows vocabulary word only.
- Parent asks the child what it means.
- Parent taps the card to flip.
- Back shows learning content.
- Parent taps/swipes Next to move to the next card.
- No score.
- No badges.
- No streaks.
- No gamification.
- No “correct/incorrect” pressure.

Front of mobile card:

craning

Back of mobile card should include:

- Vocabulary word
- Pronunciation text
- Audio speaker button placeholder
- Short definition
- Kid-friendly definition
- Book-context meaning

- Sentence from the book
- Synonyms

No illustration in MVP. No photo from the book page on the card.

Screen 5: Card Detail / Edit / Regenerate Flow

The parent should have full editing control, but editing should feel optional.

The product promise is:

The app removes the pain, time, and friction from making flashcards. The parent should not feel like they are manually building study materials. AI should do almost everything automatically, and the parent should only step in when something is wrong.

Editable fields:

- Vocabulary word
- Short definition
- Kid-friendly definition
- Book-context meaning
- Sentence from the book
- Pronunciation
- Synonyms
- Status

When the parent edits the vocabulary word, show a prompt:

Regenerate this card?

Updating the word will refresh the definition, kid-friendly definition, pronunciation, synonyms, and context.

[Cancel] [Yes, regenerate]

If the parent clicks Yes:

- Mock regeneration.
- Update the card content.
- Show a simple regenerated state.
- Assume the app tries to find the corrected word in the original photo/sentence.

AI Flashcard Content Rules

Field	Include
Vocabulary word	Yes
Short definition	Yes
Kid-friendly definition	Yes
Book-context meaning	Yes
Sentence from the book	Yes
Pronunciation text	Yes
Audio speaker button	Yes

Synonyms	Yes
Illustration	No for MVP
Photo from page	No; temporary only
Book title/page	No; remove from physical cards and do not make central

For a sample card, use:

Word: craning

Pronunciation: KRAY-ning

Short definition: Stretching your neck to see something.

Kid-friendly definition: Moving your neck forward or upward because you are trying hard to see.

Book-context meaning: In this sentence, "craning" means he stretched his neck so he could look over the fence.

Sentence: He was craning his neck to look over the fence.

Synonyms: stretching, peering, leaning

Status: Ready

Photo Storage / Privacy Rule

Photos are temporary.

The captured photo exists only to extract the unknown word and surrounding sentence.

Rules:

- Keep photo temporarily until the card is approved.
- Once the card becomes Ready, automatically delete the original photo.
- Do not let the parent permanently store page photos.
- Do not design BookWords as a photo archive.
- Do not show page photos as part of the flashcard.

Account Model

No account is required before first capture.

Do not block any of these with login:

- Open camera
- Capture word
- Generate draft card
- Save locally
- Edit card
- Review flashcards on phone
- Build up 10+ cards

Before login, everything is saved locally on the phone.

Use a progressive account model:

- "Back up my cards" is optional.
- Login is not the first experience.
- Account can be created during backup or checkout.
- During physical card checkout, collect email, shipping address, and payment information.
- The account/order profile can be created in the background.

Use this phrase instead of "Create account" where possible:

Back up my cards

Account/Settings should include:

- Back up my cards
- Google login
- Amazon login
- Parent name/email
- Child name
- Child age/grade
- Shipping address
- Payment method
- Order history
- Privacy/photo deletion explanation

Physical Flashcard Ordering

BookWords is free for digital use.

Parents only pay when they order mailed physical flashcards.

Pricing:

Order 10-card pack - \$9.99

Physical cards are:

4x6 mailed flashcards

The order option should appear once the parent has at least 10 Ready cards.

The parent selects cards, but the app suggests a 10-card pack.

Ordering rule:

Parent selects cards, but the app suggests a 10-card pack.

Do not automatically force the 10 newest cards. The parent should be able to review the suggested pack and swap cards.

Physical Flashcard Layout

Side	Content
Front	Vocabulary word only
Back	Vocabulary word, pronunciation, short definition, kid-friendly definition, book-context meaning, synonyms

Do not include book title/page on the physical card. Do not include the page photo.

Checkout Flow

Create a click-through checkout flow with these five steps:

- Select Cards - Parent selects which Ready cards to print. App suggests a 10-card pack. Show price: Order 10-card pack - \$9.99.

- Preview Physical Cards - Show front/back preview of 4x6 cards. Front: word only. Back: word, pronunciation, definitions, context, synonyms.
- Shipping Address - Parent enters mailing address.
- Payment - Parent enters payment method. Show total: \$9.99.
- Order Confirmation - Confirm order placed. Move selected cards to Ordered status in mocked state.

Notifications

No notifications in MVP.

Do not design:

- Review reminders
- Push notifications
- Streak reminders
- Order nudges
- Daily habit loops

The app should wait for the parent to open it.

Required Click Paths

1. Open /bookwords -> active camera wireframe.
2. Tap capture -> show Saved: craning in bottom-left and add/mock a card.
3. Tap saved thumbnail -> open card detail/edit screen.
4. Tap X/back from camera -> go to Card Library or Account/Settings.
5. Card Library -> grouped by Needs Review, Ready, Ordered.
6. Tap a Ready card -> open mobile flashcard review.
7. Tap flashcard -> flip front/back.
8. Tap Edit -> edit word -> show Regenerate this card?
9. Tap Order 10-card pack - \$9.99 -> checkout flow: Select Cards, Preview Physical Cards, Shipping Address, Payment, Order Confirmation.
10. Account/Settings -> backup, login, address, payment, order history, privacy.

Visual Style

Use basic black-and-white wireframes first.

Do:

- Simple layout
- Clear hierarchy
- iPhone-sized screen frame
- Black, white, gray only
- System-font feel
- Apple-like simplicity
- Premium, minimal, parent-focused structure

Do not:

- Add color branding
- Add playful kid graphics
- Add badges
- Add avatars
- Add coins
- Add streaks
- Add magical/Harry Potter visuals
- Add decorative animations

The app should feel like it could eventually become an Apple Design Award-caliber app: simple, restrained, elegant, and useful.

Deliverable

Create the first working black-and-white React/Next.js mobile wireframe prototype.

Requirements:

- Use existing repo routing style.
- Add /bookwords.
- Keep code contained.
- Use mocked data.
- Make click-through flows work.
- Do not add real integrations.
- Do not modify existing MINERVA functionality.
- Keep the UI mobile-first and wireframe-like.
- Do not request passwords, keys, tokens, or secrets.
- Do not edit .env files.

After implementation, summarize:

- Files created or changed.
- How to run/view the prototype.
- What click paths are supported.
- What remains mocked.
- Any assumptions made.