SELECTED WORK / RECRUITER READING COLLECTION

# From a problem to a considered product.

Srinidhi Bhat / UX Designer

27 product stories about independent tools, games and the decisions behind them.

I care about the decisions between a problem and an interface: what to prioritise, what a person needs to understand, and what the design asks of the system behind it.

## Reading this collection

Each chapter explains the challenge, contribution, approach, three consequential decisions, the delivered scope, and the next evaluation. Early concepts are labelled as concepts. Research counts come from original artifacts; product counts describe product scope. Proposed studies are future work, and historical targets are not presented as achieved results.

## A note on this edition

These are retrospective accounts written in October 2026. They connect visible decisions in the original artifacts with my current UX analysis. Product stories describe my ideation and design direction; they do not claim code authorship or framework selection. The original design files are preserved separately, and sources are listed in every chapter.

Read the individual stories online

## CONTENTS / CHOOSE A STORY

**Arkitype** / Change a token. See what it changes.

**AppleCider** / Room to sketch before committing to a design.

**PlanIt** / A trip plan that leaves room for people.

**Earthlog** / What is happening around this place?

**HexChess** / Learning chess on a different board.

**DashF1** / Follow the driver. Keep the race in view.

**Hued** / One colour. A reason to look closer.

**Yuva Panaji** / An event page is only the beginning.

**DeckForge** / A PDF deck you can work with again.

**Vansh Vriksha** / Family connections with a source to follow.

**PreMoney** / A bank message is not always a clean transaction.

**SoundBox** / The song waits until you are ready.

**Wheelie** / Different starting points. One place to meet.

**Matinee** / Less scrolling. A film worth saving.

**BirthSky** / Turn a time and place into a sky to keep.

**ScaleSee** / How much bigger? Show the difference.

**Where Would I Be...?** / Where would this place be as the plates move?

**The Blog** / A home for my writing, in the form it needs.

**The Office Games** / Get everyone in. Keep the round moving.

**Forma** / Health readings with the context attached.

**OhCaptain** / Sail your ship. Coordinate the fleet.

**Halo** / Keep the action close to the pointer.

**Binnder** / Review the file. Keep a way back.

**PureEQ** / Find a sound you can adjust and keep.

**Hidden Identity Chess** / Familiar chess. Very different information.

**5 Dead** / Keep the table moving. Keep the turn clear.

**Alibi** / Watch privately. Give just enough away.

Every chapter is also bookmarked in this PDF.

**ARKITYPE / DESIGN SYSTEMS · WORKING PRODUCT**

# Change a token. See what it changes.

A working design-system studio that links foundations, component behaviour, previews, and exports through one model.

MY CONTRIBUTION / Product idea, design-system UX & interaction direction

SCOPE / Public beta

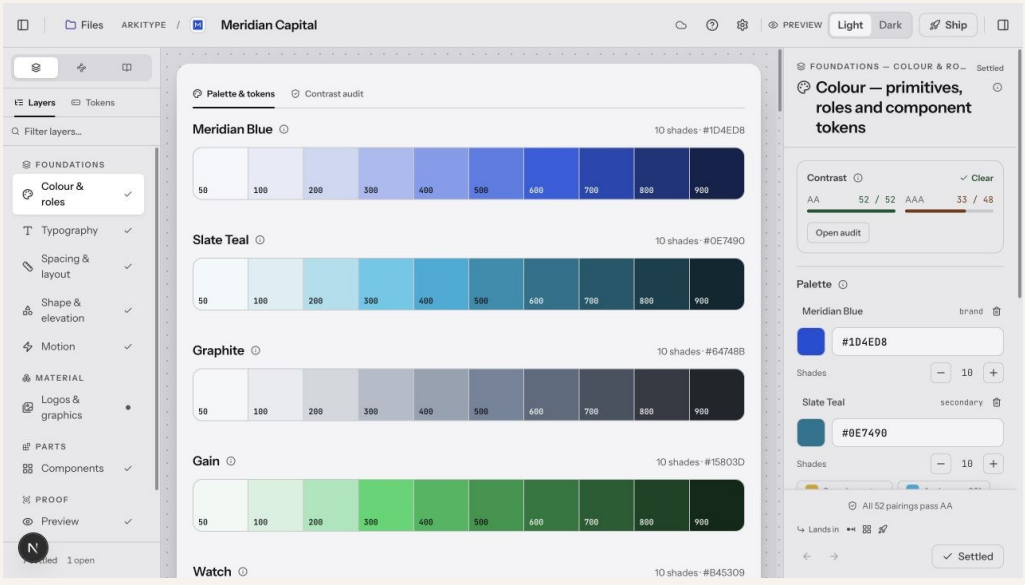
**90**
components in the system registry

**8**
decision areas

**Shared**
token-to-export model

## The challenge

A palette, a component library, and a handoff document can each look convincing while disagreeing with one another. I approached Arkitype as a connected workspace: decisions about colour, type, space, shape, and motion should remain visible in the components and artifacts they produce.




Foundations in context: the palette, navigation and contrast inspector share one workspace.

ARKITYPE / APPROACH & DECISIONS

# How I approached it

I framed the workspace around eight connected design areas, rather than a compulsory wizard. A designer should be able to revisit a foundation while considering a component and still understand what depends on that decision.

The UX goal is continuity: foundations, component states, previews, and handoff should feel like parts of one system. The current 90-component library is product scope; it does not establish that a new user understands the workspace.

## 01 / Use dependencies to guide, rather than enforce a sequence

Foundations, assets, components, and preview remain accessible, with relationships expressed as Built from and Lands in links.

**Why it matters.** The navigation reflects iterative design work instead of pretending every system starts and ends in the same order.

**The trade-off.** An open workspace gives less automatic direction. First-use guidance must help a person find a useful starting point.

## 02 / Keep design changes consistent across surfaces

The preview, component workspace, and handoff reflect the same design decisions.

**Why it matters.** A change should carry through to the places a team actually uses, rather than depend on repeated manual updates.

**The trade-off.** A designer needs to know which properties carry through and which need review before handoff.

## 03 / Let people inspect appearance and behaviour

The studio distinguishes controlled component specimens from interactive previews.

**Why it matters.** A design system needs comparable states and realistic behaviour. One view cannot do both equally well.

**The trade-off.** Interactive examples add state and complexity; the underlying specimen must remain predictable for documentation and export.

ARKITYPE / OUTCOME & REFLECTION

# What came out of it

I shaped Arkitype around the relationship between foundations, component states, previews and handoff. The beta brings those surfaces into one workspace, with a documented library of 90 components. The design question I still want to test is whether a new user can follow a change through the whole system.

# What I would do next

I would ask a designer to change a foundation, find the affected component states and prepare a handoff. If they need a second explanation at every step, the workspace is still making them carry too much of the system in their head.

Proposed study: ask a designer to start from one brand colour, adjust a component, and produce a usable handoff. Record confusion, corrections, and output mismatches.

Proposed measure: time to the first coherent system and the number of inconsistencies discovered after export, with a defined task and baseline.

## Artifacts & evidence

**Repository README and lib/systemAreas.ts**
Eight-area navigation, dependency graph, specimen/interactive distinction, beta scope and export pipeline.

**lib/componentLanes.ts and docs/GETTING_STARTED.md**
Component registry and the documented first-use journey. Reviewed 15 October 2026.

Read the individual web case study

APPLECIDER / DESIGN TOOLS · WORKING PRODUCT

# Room to sketch before committing to a design.

A sketch-like wireframing workspace that carries rough screens into clickable prototypes and editable handoff artifacts.

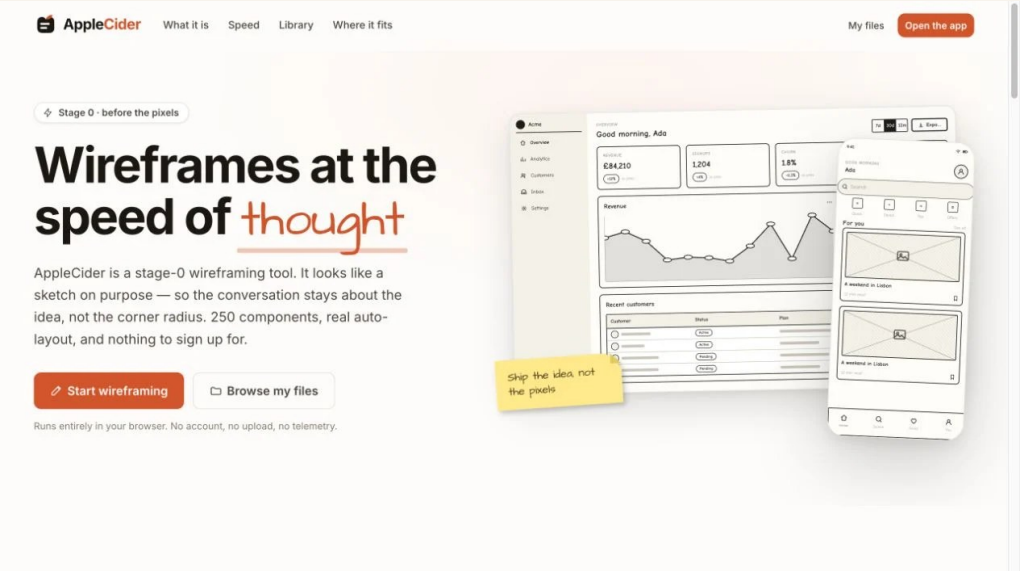
MY CONTRIBUTION / Product idea, sketching workflow & interaction design

SCOPE / Working product

| 250 | 3 | Local |
| --- | --- | --- |
| documented library components | sketch styles | browser-owned files |

## The challenge

Early design conversations can become arguments about polish before the underlying idea is settled. I framed AppleCider around an intentionally unfinished visual language: the speed and social permission of a sketch, with the editing capabilities of a digital workspace.




The archived homepage demonstrates the intentionally rough visual language. These are the project's own wireframe examples.

APPLECIDER / APPROACH & DECISIONS

# How I approached it

I separated the experience into a welcoming introduction, a files workspace, and the editing canvas. This gives each moment a clear task: understand the tool, find the work, then explore an idea.

The design loop is sketch, revise, demonstrate, and hand off. Components, undo, snapping, links, and export matter because they let someone change their mind without starting again. Formal user validation remains a next step.

## 01 / Make the unfinished quality deliberate

Wobbly strokes, a sketch-like type treatment, and three visual styles preserve a low-fidelity feel.

**Why it matters.** The interface should invite changes to the idea before encouraging arguments about pixel-level finish.

**The trade-off.** A sketch aesthetic can also reduce legibility. Labels, focus states, and interactions still need the clarity of a serious tool.

## 02 / Keep the work portable

The files surface stores documents locally and supports a plain project-file export, alongside image and vector outputs.

**Why it matters.** A person should be able to keep and move their work without an account or a proprietary cloud workflow.

**The trade-off.** Browser storage is device-specific and can be cleared. File ownership needs an understandable backup and export path.

## 03 / Carry structure into the next artifact

The handoff offers editable output and a written specification, alongside the clickable prototype.

**Why it matters.** The transition from a rough screen to a working conversation should preserve the decisions already made.

**The trade-off.** Generated structure is a starting point. Placeholder copy and sketch styling should not be presented as finished product requirements.

APPLECIDER / OUTCOME & REFLECTION

# What came out of it

I used an intentionally rough visual language across the canvas, files workspace and prototype flow. The product includes 250 sketch components and editable handoff formats, so an early idea can move forward without pretending to be a finished interface.

# What I would do next

The roughness needs to make revision easier, not make the tool harder to use. I would test whether someone can change a screen, demonstrate its flow and hand it over without losing the decisions they already made.

Proposed evaluation: give participants an early product idea, ask them to revise it after feedback, and then explain the exported prototype to someone else.

Proposed measure: time to the first understandable screen, effort to revise its structure, and corrections needed after export.

## Artifacts & evidence

**Repository README and docs/architecture.md**
Visual intent, component library, three surfaces, local file model and storage trade-off.

**src/io/export.ts and src/store/store.ts**
Export and document-state behaviour supporting the documented workflow. Reviewed 15 October 2026.

Read the individual web case study

**PLANIT / GROUP PLANNING · WORKING PRODUCT**

# A trip plan that leaves room for people.

A browser-based itinerary workspace for overlapping plans, different travellers, and different time zones.

MY CONTRIBUTION / Product framing, group-planning journeys & interaction design

SCOPE / Working product

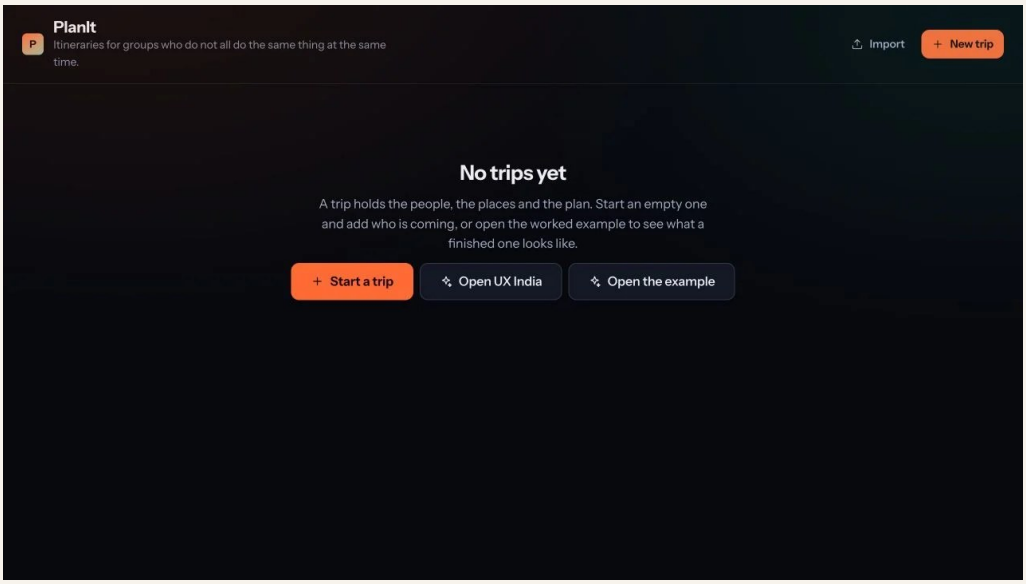
## Shared
plan across multiple views

## Explicit
board-to-time action

## Local
no account required

## The challenge

A group trip is not a single ordered checklist. Different people arrive at different times, attend parallel activities, and move between places. I approached PlanIt as several useful views of one plan, with a board for thinking and a time model for checking whether that plan can work.




The archived PlanIt interface. Ideas, scheduled commitments and location are treated as different views of a shared trip.

PLANIT / APPROACH & DECISIONS

# How I approached it

I distinguished tentative ideas from scheduled commitments. The canvas supports exploration, while itinerary, timeline, map, and people views help the group understand the consequences of a plan.

The key interaction is the deliberate move from arrangement to schedule. Pins protect an existing commitment; loose ideas can stay undecided. The UX needs to explain what Resolve changes and make conflicts understandable.

## 01 / Give the board a clear but limited grammar

Day frames, ordering, and connectors convey meaning; loose cards can remain uncommitted ideas.

**Why it matters.** A planning surface needs room for uncertainty instead of assigning a date and time to everything immediately.

**The trade-off.** Spatial meaning must be taught. A frame or connector that looks decorative can otherwise change a plan unexpectedly.

## 02 / Make scheduling an intentional action

Tidy turns the existing itinerary into a board; Resolve reads the board back into the schedule.

**Why it matters.** A person should know when arrangement becomes a commitment. Automatic rearrangement would undermine the value of a free board.

**The trade-off.** Two explicit directions add a concept to learn, so the product needs visible feedback about what changed and why.

## 03 / Protect decisions the planner has already made

Pinned times win, untouched loose cards opt out, and resolution only moves times forward when constraints require it.

**Why it matters.** Automation should respect existing intent and explain an impossible plan rather than silently rewrite it.

**The trade-off.** A conservative resolver may leave conflicts unresolved. The interface must help the person decide what to relax.

PLANIT / OUTCOME & REFLECTION

# What came out of it

I designed the board, itinerary, timeline, map and people views as different ways of working with the same trip. Scheduling is deliberate, pinned commitments stay protected, and conflicts remain visible. That behaviour is the part of PlanIt I would explain first to a new group.

# What I would do next

I would try the planner with a late arrival, two parallel activities and a booking that cannot move. The useful test is whether people can predict what Resolve will change and spot a conflict before it becomes a problem on the trip.

Proposed study: plan a trip with parallel sessions, a pinned departure, and travellers in two zones; ask participants to predict Resolve before using it.

Proposed measure: unexpected schedule changes, correctly identified conflicts, and confidence explaining the final plan.

## Artifacts & evidence

**Repository README**
Planning views, board grammar, sharing model and scheduling constraints.

**src/core/resolve.ts and src/core/store.ts**
Priority of pins, opt-in cards and forward-only resolution. Reviewed 15 October 2026.

Read the individual web case study

EARTHLOG / DATA EXPLORATION · WORKING PRODUCT

# What is happening around this place?

Explore earthquakes, weather, and the night sky from one place on a globe.

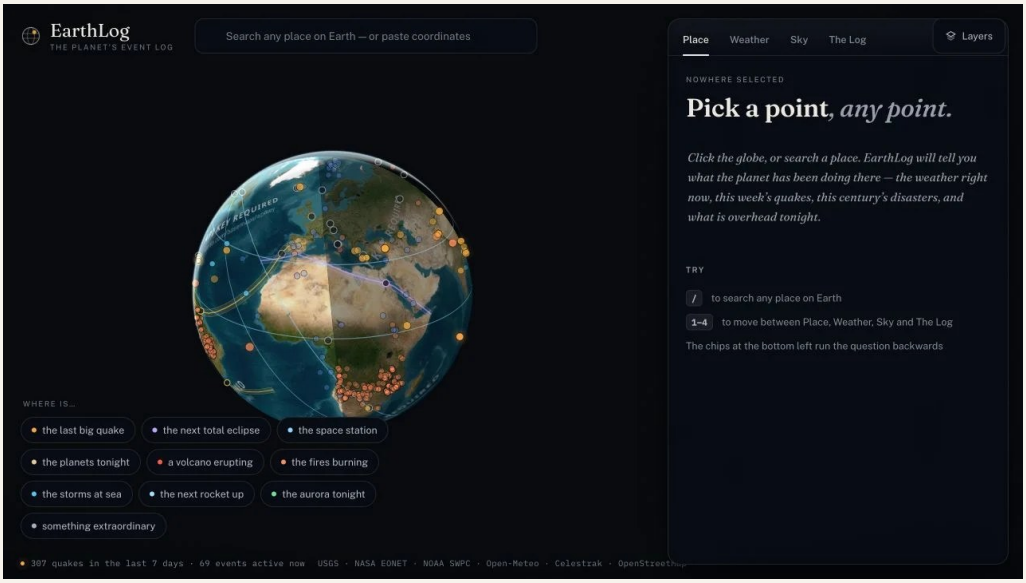
MY CONTRIBUTION / Experience framing, information hierarchy & map interactions

SCOPE / Working product

| 2 | Place-led | Labelled |
|---|---|---|
| directions of exploration | event context | models and estimates |

## The challenge

A place can have recent earthquakes, old incident records, current weather, and predicted sky events. Bringing them onto one map makes exploration easier, but can also make very different information look equally current and reliable.




The actual exploration surface: globe and event entry points on the left; place, weather, sky and history in the inspector.

EARTHLOG / APPROACH & DECISIONS

# How I approached it

I framed two ways into exploration: start from a place, or follow an interesting event to its location. Both routes should arrive at a readable local picture, rather than another layer to decode.

Freshness and uncertainty are part of the information hierarchy. An observation, an older open record, and a forecast need different labels so the map does not make them look equally current or certain.

## 01 / Start with a place or a question

People can search for a place, select the globe, or follow an event shortcut.

**Why it matters.** Someone may be curious about the latest earthquake without knowing where it happened.

**The trade-off.** Every route needs to lead back to the same place context.

## 02 / Show when the information was observed

The documented event model uses observation recency instead of treating every open incident as current.

**Why it matters.** A source's administrative status can outlive the actual event. The label should communicate what is known.

**The trade-off.** Freshness rules remain approximations; source links and timestamps must stay visible.

## 03 / Explain the limits beside the result

Eclipse paths, station passes, and aurora layers carry model limitations in the documented product.

**Why it matters.** A compelling map should not appear more precise than the calculation behind it.

**The trade-off.** Caveats can overwhelm the visual story, so they belong beside the relevant result rather than only in documentation.

EARTHLOG / OUTCOME & REFLECTION

# What came out of it

I brought place search and event discovery together on the globe. A visitor can start with a location or an event, then inspect dates, sources and the relevant model limits. I kept that context close to the result rather than making the globe's appearance carry the explanation.

# What I would do next

I would ask someone to explain one result in their own words: what happened, when it was observed and where the information came from. Finding an event is only part of the task.

Proposed evaluation: ask visitors to distinguish a recent observation from a modelled forecast and locate the source behind each result.

Proposed measure: correct interpretation of freshness and uncertainty, alongside success finding a relevant local event.

## Artifacts & evidence

**Repository README, What it answers and Honesty notes**
Bidirectional exploration, source freshness and model limitations. Reviewed 15 October 2026.

Read the individual web case study

**HEXCHESS / LEARNING & PLAY · WORKING PRODUCT**

# Learning chess on a different board.

A 91-cell chess variant designed around guided learning, readable interaction, and games that survive interruptions.

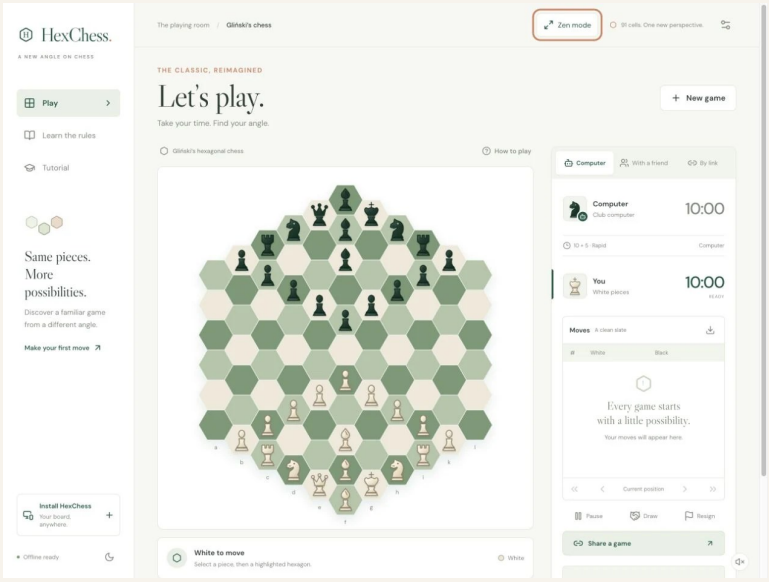
MY CONTRIBUTION / Learning journey, onboarding & game interactions

SCOPE / Working product

| 91 | 6 | Offline |
|---|---|---|
| board cells | guided piece lessons | after first successful load |

## The challenge

A hexagonal board changes movement patterns enough to make familiar chess knowledge unreliable. I designed HexChess around learning by trying: clear targets, feedback on moves, and a route into untimed practice instead of expecting a beginner to absorb a rules document first.




The original playing room: a 91-cell board, legal-move guidance and game controls.

HEXCHESS / APPROACH & DECISIONS

# How I approached it

I connected a six-part introduction to the first practice game. A learner should encounter a rule on the board, try a move, and carry that understanding into play.

I treated interruption and sharing as parts of the experience. A saved game needs a clear return path, and a correspondence link needs to explain that the other person replies with an updated game rather than joins a live room.

## 01 / Make movement teachable on the board

Lessons give a specific target, neutral markers, retries, and progress based on actual practice.

**Why it matters.** A learner can connect a rule to a move instead of memorising abstract directions.

**The trade-off.** Move previews and legal moves serve different purposes; the interface must explain when inspection ignores king safety.

## 02 / Keep assistance appropriate to the context

Hints and takebacks belong to computer practice; human games remove assistance and pause controls.

**Why it matters.** Learning support should not silently change the rules of a friendly competitive game.

**The trade-off.** Assistance rules need to be clear to both players; the interface is not evidence of tournament-grade enforcement.

## 03 / Describe sharing accurately

Correspondence links carry move history and require an updated reply after a turn.

**Why it matters.** People should understand whether they are entering a synchronized room or receiving a game snapshot.

**The trade-off.** Long histories create long URLs, and a shared snapshot is readable by anyone given the link.

HEXCHESS / OUTCOME & REFLECTION

# What came out of it

I connected the 91-cell board to six guided lessons, practice, saved sessions and correspondence sharing. Legal-move feedback helps explain the unfamiliar geometry where the move happens. I have not measured whether the lessons improve learning.

# What I would do next

I would follow a new player from a lesson into their first game. A completed tutorial is less useful evidence than seeing them recognise a move, recover from a mistake and continue without an explanation.

Proposed study: observe a new player complete a piece lesson and use the same movement unaided in practice.

Proposed checks: reload during a game, resume offline, and exchange a correspondence reply while confirming both players understand the snapshot model.

## Artifacts & evidence

**Repository README**
Learning, board interactions, saves, offline support, sharing boundaries and existing test scope. Reviewed 15 October 2026.

Read the individual web case study

DASHF1 / LIVE DATA · WORKING PRODUCT

# Follow the driver. Keep the race in view.

A race dashboard connecting timing, track context, and replay while distinguishing measured data from estimated positions.

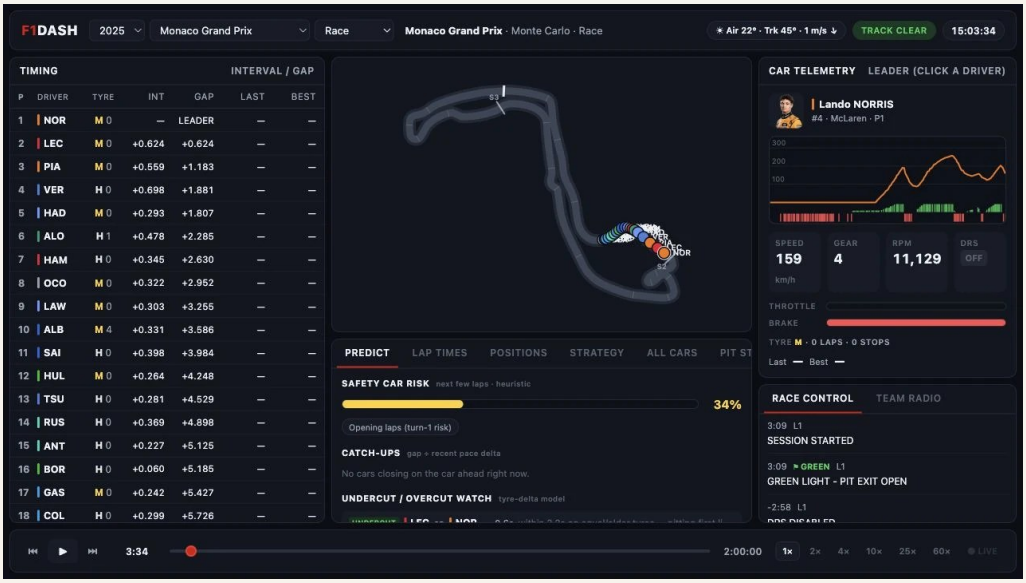
MY CONTRIBUTION / Dashboard hierarchy, driver focus & information design

SCOPE / Working product

| 1 | Live / replay | Estimated |
|---|---|---|
| driver focus across panels | distinct data modes | free-feed track positions |

## The challenge

A race weekend produces more signals than a person can follow at once, and the available feed changes between live and replay sessions. DashF1 uses one shared driver focus across the timing tower, map, and detail views so the dashboard supports a question rather than a wall of independent charts.




Actual local dashboard replaying the 2025 Monaco race from OpenF1 data. The prediction panel labels its heuristic estimates.

DASHF1 / APPROACH & DECISIONS

# How I approached it

I organised the experience around one driver focus. Timing, track context, and detail panels should answer related questions without requiring the same selection repeatedly.

I separated live and replay expectations. When a position is estimated, the label belongs near the moving marker; missing coverage should be understandable at the moment someone tries to use it.

## 01 / Carry one selection through the dashboard

Choosing a driver updates the relevant map and telemetry context.

**Why it matters.** The person should not have to repeat the same selection in several panels.

**The trade-off.** The relationship must remain clear when the selected driver retires or a particular data channel is unavailable.

## 02 / Let the mode determine the promise

Live mode shows supported channels; fuller replay data becomes available through a separate path.

**Why it matters.** Missing data should change the interface's promise instead of being replaced with plausible-looking numbers.

**The trade-off.** A mode change can alter familiar panels, so the dashboard needs an understandable explanation.

## 03 / Label an estimate at the point of use

The documented free-feed map derives position from mini-sector timing and gaps, and calls it estimated.

**Why it matters.** Visual continuity is useful, but it should not be mistaken for location precision the feed does not provide.

**The trade-off.** Race interruptions and stale timing can challenge the estimate; smooth motion must not conceal those limits.

DASHF1 / OUTCOME & REFLECTION

# What came out of it

I organised timing, track position, telemetry and replay around one selected driver. The selection carries across panels, while session mode and data coverage explain what each view can show. Estimated positions need to stay visibly different from recorded data.

# What I would do next

I would check what someone believes when a feed is incomplete. An empty or estimated view should be understandable without making them leave the race to read a technical explanation.

Proposed study: ask a viewer to follow a battle and explain which values are observed, historical, or estimated.

Proposed checks: missing telemetry, stale updates, interrupted streams, safety-car periods, and the transition back to replay.

## Artifacts & evidence

**Repository README, What's on screen and Race mode**
Layout, shared focus, data-source boundaries and position-estimation limitations. Reviewed 15 October 2026.

**Interface capture · 31 October 2026**
Local built F1DASH, captured 31 October 2026; historical replay, not a live session.

Read the individual web case study

**HUED / CREATIVE HABITS · WORKING PRODUCT**

# One colour. A reason to look closer.

A daily photography prompt with a shared constraint, on-device scoring, and a warm visual language.

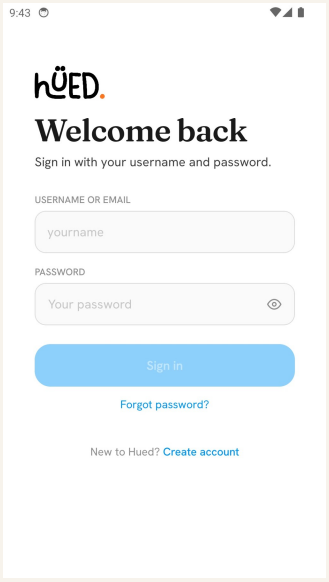
MY CONTRIBUTION / Product concept, creative loop & visual direction

SCOPE / Mobile beta

| 1 | 48 | On-device |
|---|---|---|
| shared daily colour | colours in the documented palette | colour scoring |

## The challenge

A broad prompt to be creative can be harder to act on than a small constraint. Hued gives everyone the same daily colour and asks them to find and photograph it in the real world. The product connects a simple creative action to a repeatable daily rhythm.




Actual Android sign-in screen. The daily photography flow is described from the project documentation; no account or photographs were used for this capture.

HUED / APPROACH & DECISIONS

# How I approached it

I shaped the concept around one shared colour and a small creative invitation: notice it in the world, photograph it, and see other interpretations. The daily boundary needs a clear local explanation.

A quiet, warm interface keeps the photographs in front. Scores and streaks are supporting feedback; whether they encourage curiosity or pressure is a question for evaluation, not an established habit-formation result.

## 01 / Make the prompt shared and specific

One date-derived colour creates a common constraint for the day.

**Why it matters.** A small, concrete invitation can make starting easier and make different responses interesting to compare.

**The trade-off.** A UTC boundary may feel surprising across time zones; the deadline needs to be understandable locally.

## 02 / Keep the creative result in the foreground

The feed emphasises photographs within a quiet, warm visual system.

**Why it matters.** The person's interpretation of the colour should carry more attention than surrounding interface decoration.

**The trade-off.** Ranking and streaks can compete with exploration; their value needs to be judged from behaviour and feedback.

## 03 / Treat scoring as guidance with limits

Colour matching runs locally; suspicious-image heuristics are documented as tentative rather than proof.

**Why it matters.** A score can provide feedback without pretending to reliably determine how an image was made.

**The trade-off.** Lighting, camera processing, and scene composition affect colour. An unexplained penalty can feel arbitrary.

HUED / OUTCOME & REFLECTION

# What came out of it

I centred the app on a shared daily colour, a photograph and feedback on the attempt. The warm interface gives the photographs most of the attention; scores, the feed and streaks support the daily activity. Participation and the scoring experience still need user evaluation.

# What I would do next

I want the prompt to encourage noticing things, not just protecting a streak. I would ask people what they chose to photograph and how they interpreted the score before deciding whether to make the competitive features more prominent.

Proposed study: compare how people interpret the prompt and score under different lighting conditions, including whether they understand a low score.

Proposed measure: repeat participation with self-reported creative value; evaluate deadline understanding across time zones.

## Artifacts & evidence

**Repository README and src/lib/dailyColor.ts**
Daily prompt, palette and beta scope.

**src/lib/scoring.ts and README limitations**
On-device colour evaluation and best-effort heuristics. Reviewed 15 October 2026.

**Interface capture · 31 October 2026**
Hued release APK, captured in an isolated Android emulator, 31 October 2026.

Read the individual web case study

YUVA PANAJI / COMMUNITY OPERATIONS · WORKING PRODUCT

# An event page is only the beginning.

A community website and operations portal linking event discovery with registration and committee workflows.

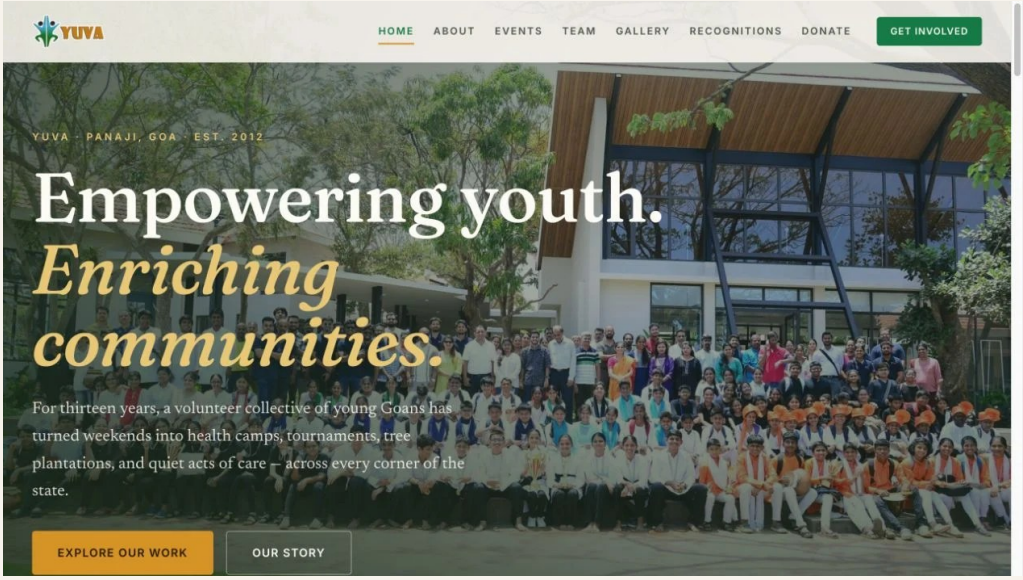
MY CONTRIBUTION / Information architecture, registration & volunteer workflows

SCOPE / Working system

| 2 | Public | Private |
| --- | --- | --- |
| connected surfaces | events and registration | committee operations |

## The challenge

A community organisation needs a public account of what it does and a practical way to run the next event. The reviewed Yuva source brings those responsibilities together: public discovery and registration on one side, event administration and committee workflows on the other.




Original public homepage for Yuva Panaji. Community activity introduces the organisation before visitors enter its event and registration routes.

YUVA PANAJI / APPROACH & DECISIONS

# How I approached it

I mapped the service from public event discovery and registration through committee review and the attendance desk. Participants and volunteers need different views of the same event information.

The design priority is continuity: information entered on a registration form should remain useful when a volunteer checks someone in, corrects a record, or prepares a printed list.

## 01 / Connect discovery to the event's operational record

The public registration journey and admin roster belong to the same event model.

**Why it matters.** A committee should not have to reinterpret a public sign-up into a separate attendance workflow.

**The trade-off.** Custom events need flexible fields without making every registration form equally complex.

## 02 / Give public and committee users different surfaces

Visitors see event and organisation information, while management workflows sit behind admin access.

**Why it matters.** The interface should reflect different responsibilities and information needs.

**The trade-off.** A clean separation in navigation is only part of the requirement; permissions and data handling must be verified independently.

## 03 / Include the work that happens away from the screen

The documented system supports desk attendance sheets and spreadsheet exports.

**Why it matters.** Community operations often happen at an event desk, where a printable list can be more useful than another dashboard.

**The trade-off.** Exported files can become stale. The workflow should explain when a printout was produced and how corrections return to the system.

YUVA PANAJI / OUTCOME & REFLECTION

# What came out of it

I connected public event information and registration with the committee's follow-up work: rosters, corrections, exports and attendance. The public website and operations portal have different jobs, but the information needs to carry from one to the other.

# What I would do next

I would review the handoff at an actual event: who checks a registration, who corrects it and what the attendance desk receives. That is where duplicate entry and outdated lists become visible.

Proposed evaluation: follow one event from discovery through registration, roster correction, and check-in with both participant and committee perspectives.

Proposed measure: duplicate corrections, incomplete registrations, and handoff effort at the event desk, with a documented baseline.

## Artifacts & evidence

**Repository README and src/lib/db.js**
Current public/admin architecture, event operations and Supabase-backed data layer. Reviewed 15 October 2026.

Read the individual web case study

DECKFORGE / PRESENTATION TOOLS · WORKING PRODUCT

# A PDF deck you can work with again.

A browser presentation tool that turns PDF drawing instructions into editable objects, animation, and presenting workflows.

MY CONTRIBUTION / Product idea, editing experience & presentation flows

SCOPE / Working product

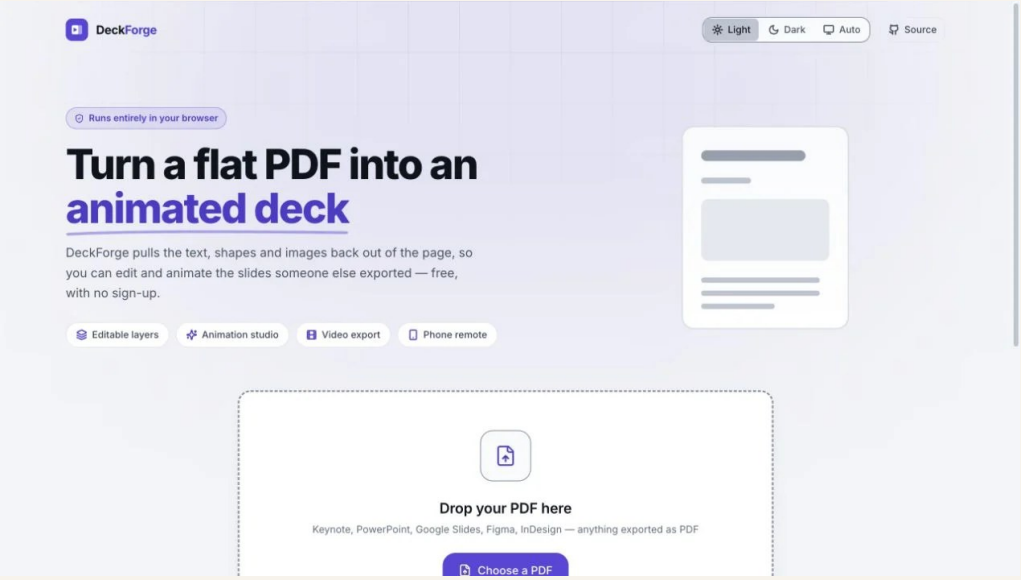
**Editable**
reconstructed page objects

**40**
documented animation presets

**Browser**
core document processing

## The challenge

A PDF preserves how a slide looks but usually loses the convenient objects a presenter wants to edit. DeckForge approaches that gap as a reconstruction problem: recover meaningful text, images, and vector objects, then let a person work with them rather than repainting a screenshot.




Original DeckForge starting screen for opening a PDF presentation. Editing and animation begin with an existing deck.

DECKFORGE / APPROACH & DECISIONS

# How I approached it

I framed the editor around objects a presenter can recognise: a text block, a logo, a shape, or a chart. A technically selectable fragment is not useful if its meaning is unclear.

The flow needs to explain what can be edited, preserve a usable slide when recovery is limited, and distinguish live presentation pacing from an exported recording.

## 01 / Make layers correspond to recognisable objects

Paths are grouped into likely logos, dividers, shapes, and chart series, while text is reconstructed into blocks.

**Why it matters.** An editable layer is useful only if the person recognises what it represents.

**The trade-off.** Object recovery is heuristic. Preserving appearance can be more valuable than forcing misleading editability.

## 02 / Let a presenter control the pace

Animations can wait for an advance action, while timed outputs play sequentially.

**Why it matters.** A live talk and an exported recording need different behaviour from the same slide.

**The trade-off.** That difference must be visible before export so the presenter knows what an audience will receive.

## 03 / Keep the core document workflow local

PDF reconstruction and editing run in the browser; phone pairing has a separate connection mechanism.

**Why it matters.** Local processing supports a lightweight workflow and avoids requiring a document-upload service.

**The trade-off.** Phone pairing still uses signalling, and some remote features exchange preview frames. The privacy explanation should distinguish those paths.

DECKFORGE / OUTCOME & REFLECTION

# What came out of it

I organised recovered slide content into recognisable objects that can be edited, sequenced, presented and exported. Some content remains an image, so the interface needs to explain what was recovered and what can actually be changed.

# What I would do next

I would use a difficult deck to test the explanation, not just a clean example. Someone should be able to tell why one element is editable and another is not, then choose a workable next step.

Proposed benchmark: import a defined set of native-text, scanned, multi-column, and dense-vector PDFs; record editability and visible fidelity separately.

Proposed study: revise a real slide and present it, measuring manual corrections and surprises between editor, audience view, and exported output.

## Artifacts & evidence

**Repository README, How extraction works and Known limitations**
Reconstruction pipeline, presentation model, export boundaries and remote-pairing caveat.

**src/engine and src/store**
Shared extraction, animation, export and document-state behaviour. Reviewed 15 October 2026.

Read the individual web case study

VANSH VRIKSHA / COMMUNITY KNOWLEDGE · WORKING PRODUCT

# Family connections with a source to follow.

A community genealogy archive that treats relationships, provenance, and contribution review as parts of the same experience.

MY CONTRIBUTION / Archive concept, relationship exploration & contribution UX

SCOPE / Working archive

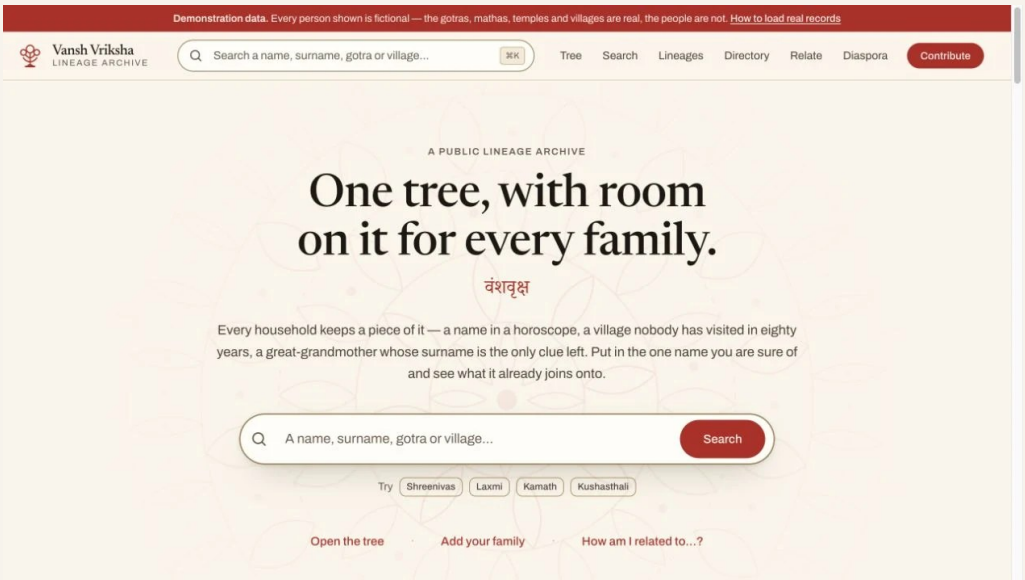
## Graph
relationship model

## Provenance
per-record intent

## Reviewed
contribution workflow

## The challenge

A family archive is not simply a diagram of names. Relationships can be disputed, spellings vary, and historical records represent some people much better than others. Vansh Vriksha is structured around exploring those connections while preserving the basis for a record.




The original archive interface introduces people, families and the connections between records.

VANSH VRIKSHA / APPROACH & DECISIONS

# How I approached it

I treated the archive as a set of connected records with several entry points: a person, a family, a village, or a relationship. The UX should help someone follow a connection without losing its source.

The demonstration content lets the experience be explored; it is not verified family evidence. Missing records, uncertain relationships, and contribution review need visible explanations.

## 01 / Model relationships beyond a simple tree

The graph includes family unions and parent links rather than forcing every connection into one hierarchy.

**Why it matters.** A plain tree can misrepresent remarriage and half-sibling relationships. The data model needs to preserve the actual relationship.

**The trade-off.** A more faithful model is harder to lay out and explain, especially on small screens.

## 02 / Keep a record's basis alongside the record

The documented model includes provenance and versioned family files.

**Why it matters.** Someone reading a family claim should have a way to understand where it came from and how it changed.

**The trade-off.** A source field does not guarantee accuracy. Disagreement and corrections still need a human review process.

## 03 / Separate contribution from publication

Visitors can submit without an account, while a maintainer reviews additions.

**Why it matters.** A low-friction contribution path can coexist with an accountable published archive.

**The trade-off.** Moderation becomes an operational commitment; the interface should communicate what happens while a submission waits.

VANSH VRIKSHA / OUTCOME & REFLECTION

# What came out of it

I designed relationship browsing, record context and contribution review as connected parts of the archive. A person can explore a connection and inspect the basis for it. The demonstration records show the interaction; they do not establish verified family histories.

# What I would do next

I would make gaps in the archive easier to recognise. A missing branch can mean incomplete records, so it should not quietly look like a statement about which families or people matter.

Proposed evaluation: ask a family contributor to submit a relationship, inspect its provenance, and correct a disputed connection.

Proposed review: distinguish demonstration records from sourced records in every public count and assess living-person privacy across search and sharing.

## Artifacts & evidence

**Repository README and src/lib/archive.ts**
Versioned record storage, graph exploration, moderation, demonstration data and known biases. Reviewed 15 October 2026.

Read the individual web case study

**PREMONEY / PERSONAL FINANCE · WORKING PRODUCT**

# A bank message is not always a clean transaction.

A personal-finance app that converts bank and UPI messages into reviewable transactions, categories, and summaries.

MY CONTRIBUTION / Product concept, transaction review & correction flows

SCOPE / Interactive app

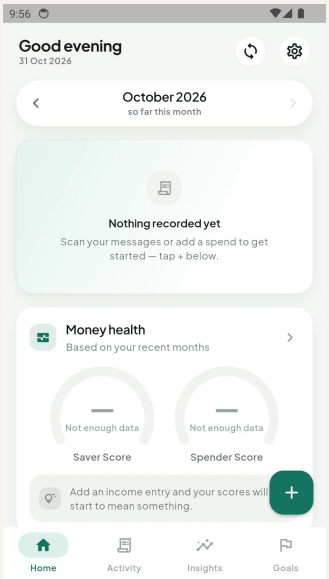
**SMS**
transaction input

**Confidence**
visible in parser model

**Review**
unresolved merchants

## The challenge

Manual expense entry asks people to repeat information already present in transaction alerts. PreMoney explores a more useful starting point: extract likely transactions from messages, then make uncertainty and correction part of the workflow.




Actual Android dashboard with no transactions. Scores stay unavailable until there is enough information; SMS access was skipped for this capture.

PREMONEY / APPROACH & DECISIONS

# How I approached it

I framed the core journey as evidence, suggested transaction, review, correction, and summary. A person should be able to understand why an alert became a record and recover when the interpretation is wrong.

Privacy explanations belong alongside optional categorisation and synchronisation. The experience needs to make those choices understandable without relying on a blanket claim that all processing stays on the device.

## 01 / Prefer an uncertain result to a fabricated detail

The parser lowers confidence when a field is uncertain and records when a merchant is unresolved.

**Why it matters.** A plausible but incorrect financial record can be harder to notice than a clear request for review.

**The trade-off.** Conservative parsing can miss legitimate alerts. The product needs a recoverable route for false rejections.

## 02 / Make correction useful beyond a single transaction

Transaction editing and review make suggested categories and corrections inspectable.

**Why it matters.** Repeated mistakes should become easier to diagnose than manually editing the same pattern forever.

**The trade-off.** A learned rule can also propagate a wrong assumption, so its effect must remain inspectable.

## 03 / Treat enrichment as a separate data decision

Optional categorisation and synchronisation need a separate explanation from reading a transaction alert.

**Why it matters.** People need an understandable account of which data is used by which feature.

**The trade-off.** The product documentation describes external processing paths; disclosures and controls need review before making privacy promises.

PREMONEY / OUTCOME & REFLECTION

# What came out of it

I designed a route from the original message to a suggested transaction, correction and summary. The important part is being able to inspect an entry's origin and fix it when the interpretation is wrong. Accuracy needs to be evaluated against a defined set of messages.

# What I would do next

I would start with a wrong amount, a duplicate alert and a message that is not a transaction. If correcting those cases is difficult, adding more charts will not make the financial record more dependable.

Proposed benchmark: use a consented, redacted alert corpus to measure false accepts, false rejects, duplicate errors, and corrections by bank format.

Proposed release review: document the external-processing and sync paths, then test whether a person understands and controls them.

## Artifacts & evidence

**lib/domain/sms/sms_parser.dart and lib/domain/categorize**
Confidence model, rejection reasons and merchant categorisation.

**lib/domain/ai/groq_service.dart and lib/ui/settings/sms_sandbox_sheet.dart**
External enrichment payload and diagnostic review surface. Reviewed 15 October 2026.

**Interface capture · 31 October 2026**
PreMoney 1.13.0 release APK, captured in an isolated Android emulator, 31 October 2026. No bank messages or personal transactions.

Read the individual web case study

**SOUNDBOX / MUSIC LEARNING · WORKING PRODUCT**

# The song waits until you are ready.

A piano-learning app that connects familiar songs, visible keys, and listening-based progression at the player's pace.

MY CONTRIBUTION / Learning concept, practice flow & feedback design

SCOPE / Working app

| 3 | Wait mode | On-device |
|---|---|---|
| arrangement levels | default practice rhythm | audio processing |

## The challenge

Someone who wants to play a familiar song may not yet read notation or keep up with a timed lesson. SoundBox starts from the song and a picture of the keyboard, then waits for the player to produce the expected note or chord.




The practice screen keeps the current chord, keyboard and pace controls together.

SOUNDBOX / APPROACH & DECISIONS

# How I approached it

I organised the learning experience around a song, an approachable arrangement, and a repeatable passage. The default practice mode waits for the learner instead of making the person chase a clock.

Listening feedback needs a clear ready, heard, and retry state. Recognition reliability has not been established across real instruments and rooms, so recovery deserves as much design attention as successful progression.

## 01 / Make progression depend on readiness

The default mode advances after the expected musical input rather than at a fixed tempo.

**Why it matters.** A beginner can concentrate on finding the right movement without constantly falling behind the interface.

**The trade-off.** Recognition errors can still interrupt the experience, so recovery and a clear input state are essential.

## 02 / Offer more than one way into a song

Chord, simplified, and full-melody arrangements support different levels of challenge.

**Why it matters.** A learner can begin with something recognisable and increase complexity when ready.

**The trade-off.** An automatically simplified arrangement still needs musical review; lower note density is not always easier to play.

## 03 / Support repetition without punishment

Section looping and restrained progress mechanics accompany local practice history.

**Why it matters.** Practice often means repeating one difficult passage, not completing a song from the beginning.

**The trade-off.** Streaks and XP can distract from learning if they reward activity without meaningful progress.

SOUNDBOX / OUTCOME & REFLECTION

# What came out of it

I designed practice around visible keys, listening-based progression and repeatable passages. Different arrangements give learners more than one way into a song. The app lets them stay with a difficult note; recognition reliability and learning outcomes need separate evaluation.

# What I would do next

I would watch the full practice loop: listening, finding a key, playing it, interpreting the feedback and trying again. A passage-completion count would miss much of what makes that experience frustrating or useful.

Proposed study: observe a beginner learning one short passage with the default wait mode, including how they recover from missed detections.

Proposed benchmark: test recognised notes and chords under varied room noise, microphones, and keyboards; report false advances separately from missed inputs.

## Artifacts & evidence

**Repository README, How the listening works and Practising**
Arrangements, paced learning, recognition approach, local processing and practice loops. Reviewed 15 October 2026.

Read the individual web case study

**WHEELIE / GROUP MOBILITY · WORKING PRODUCT**

# Different starting points. One place to meet.

A group-riding beta that connects rendezvous planning, rider location, helmet voice, and ride summaries.

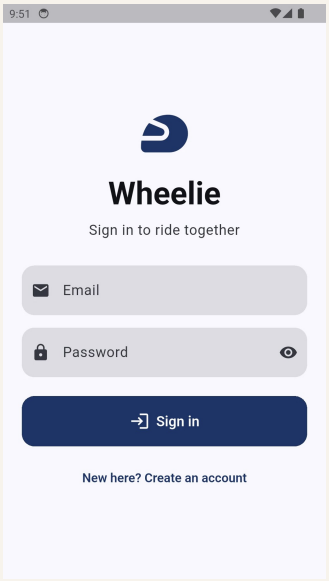
MY CONTRIBUTION / Service idea, coordination journeys & interaction constraints

SCOPE / Beta · accessibility polish in progress

| 3 | 64px+ | Beta |
|---|---|---|
| documented product pillars | primary target intent | not a safety validation |

## The challenge

Group riding begins before everyone reaches the same road. People start in different places, need a meeting point, and can lose contact after setting off. Wheelie brings planning, group location, and helmet voice into one service concept, with large, simple controls as an explicit design constraint.




Actual Android sign-in screen. Ride planning and group features are described from the project documentation; no account or rider location was used.

WHEELIE / APPROACH & DECISIONS

# How I approached it

I framed three moments: meeting before a ride, coordinating with the group, and looking back afterwards. Each moment needs a different level of detail and attention.

The constrained physical context informs large targets, restrained choices, and clear connection states. The next evaluation belongs in stationary or controlled settings; the beta is not evidence of safe in-motion interaction.

## 01 / Begin with a rendezvous, not just a destination

The service computes a meeting point for people starting from different places.

**Why it matters.** The group's coordination problem exists before shared navigation can begin.

**The trade-off.** A fair computed point may not be a suitable physical stopping place; people need to review the suggestion.

## 02 / Reduce reliance on precise gestures

The design calls for large primary targets and simple tap/drag interactions.

**Why it matters.** A constrained physical context should affect the interface model from the start.

**The trade-off.** Large buttons alone cannot establish safe in-motion use. Evaluation should begin in stationary or controlled settings.

## 03 / Separate coordination from reflection

Live map and voice support the current group; ride summaries serve the post-ride story.

**Why it matters.** Different moments deserve different information and levels of attention.

**The trade-off.** Social summaries can create incentives around speed; the content priorities deserve review before wider use.

WHEELIE / OUTCOME & REFLECTION

# What came out of it

I organised the beta around meeting before a ride, coordinating with the group and reviewing the ride afterwards. Meeting points, group location, helmet voice and summaries serve those different moments. Large controls are a design intention, not evidence that the app is safe to operate while moving.

# What I would do next

I would move as much setup as possible into the time before the ride. The next checks belong in a stationary or controlled setting, with particular attention to connection status and anything that asks the rider to look away.

Proposed evaluation: test rendezvous planning and headset controls while stationary, then assess attention demands under an appropriately controlled protocol.

Proposed checks: poor connectivity, location permission changes, unavailable voice, battery use, and clarity about whether rider positions are stale.

## Artifacts & evidence

**Repository README and design.md**
Product pillars, beta status, target-size intent and unfinished accessibility/glove review. Reviewed 15 October 2026.

**Interface capture · 31 October 2026**
Wheelie release APK, captured in an isolated Android emulator, 31 October 2026.

Read the individual web case study

MATINEE / ENTERTAINMENT DISCOVERY · WORKING PRODUCT

# Less scrolling. A film worth saving.

A cinema-tracking product connecting discovery, a personal watchlist, ratings, and locally generated recommendations.

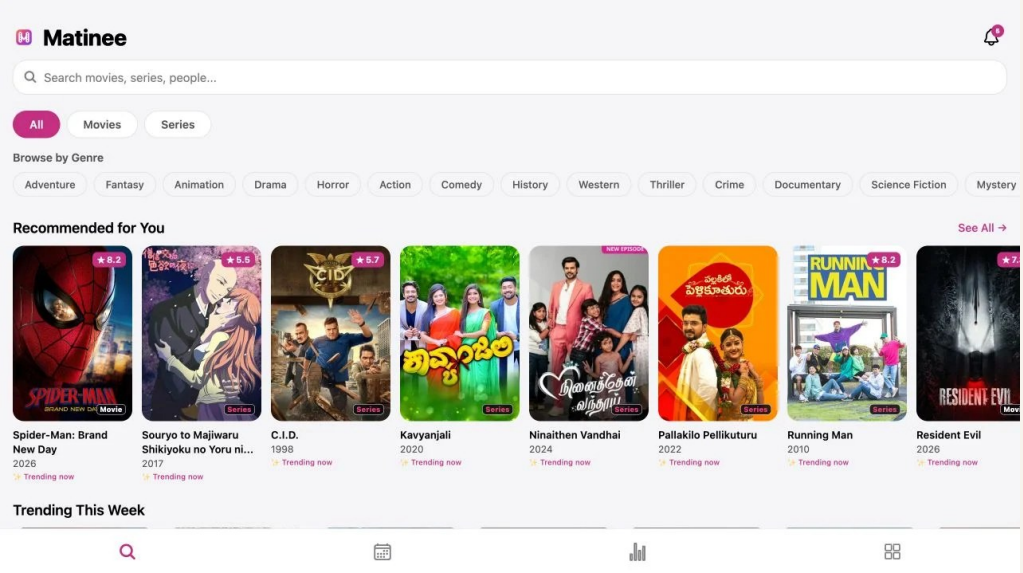
MY CONTRIBUTION / Discovery concept, browsing journeys & responsive UX

SCOPE / Interactive app

| 2 | Local | Adaptive |
|---|---|---|
| media types: films and series | recommendation scoring | phone and wider layouts |

## The challenge

Finding a title and choosing what to watch are related but different tasks. Matinee connects browsing with a personal record of saved and watched titles, using that history to shape the next set of suggestions.




Original Matinee discovery screen. Posters and titles provide the starting point for browsing films and series.

MATINEE / APPROACH & DECISIONS

# How I approached it

I connected browsing with the personal record of saved, rated, and watched titles. The design question is how that history can make the next discovery session more relevant without becoming repetitive.

Language preferences, variety, and recognisable film/series labels inform the browsing experience. Navigation and poster density should adapt to the screen while keeping the selected title and next action clear.

## 01 / Keep a title's type visible

Movie/series identifiers appear in discovery and combined lists.

**Why it matters.** A person choosing a two-hour film has a different commitment from someone starting a series.

**The trade-off.** A badge helps scanning but does not replace runtime and season information near a decision.

## 02 / Let personal history influence discovery

Ratings and watch patterns feed a local scoring process, with controls against repetitive franchises or genres.

**Why it matters.** Recommendations should reflect preference without becoming a list of near-duplicates.

**The trade-off.** A scoring rule is not evidence of relevance. Cold-start and unexpected recommendations need separate evaluation.

## 03 / Adapt the structure to the screen

Phone navigation becomes a side rail at wider sizes, and poster grids expand.

**Why it matters.** The same task can use space differently without turning a larger screen into a stretched phone layout.

**The trade-off.** Density should still preserve readable text, meaningful focus order, and a clear selected title.

MATINEE / OUTCOME & REFLECTION

# What came out of it

I connected discovery, watchlists, history and ratings so previous choices can inform the next one. Film and series context, language preferences and variety all have a place in that flow. I have not established a measured recommendation-accuracy result.

# What I would do next

I would ask whether people found something they wanted to watch, not just something they clicked. I would also check whether suggestions become too familiar after a few ratings.

Proposed study: compare a new profile with a rated-history profile and ask participants why each suggestion feels relevant or repetitive.

Proposed measure: useful selections, unwanted repetition, and confidence choosing across preferred languages.

## Artifacts & evidence

**Repository README**
Current mobile architecture, adaptive layout, watchlist rules and local recommendation factors. Reviewed 15 October 2026.

Read the individual web case study

**BIRTHSKY / PERSONALISED VISUALISATION · WORKING PRODUCT**

# Turn a time and place into a sky to keep.

An astronomy-inspired experience that turns a time and place into a customisable star-map wallpaper.

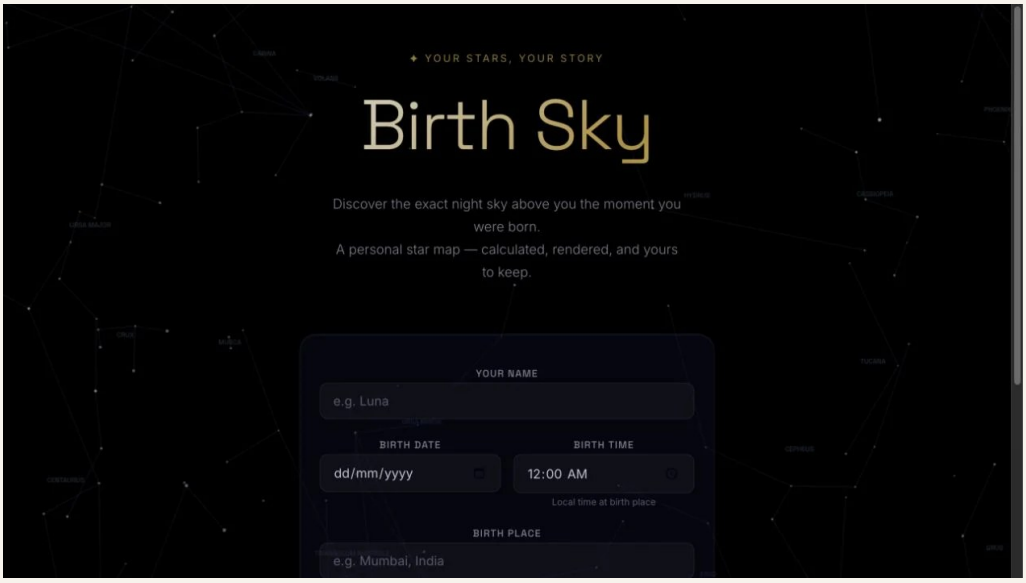
MY CONTRIBUTION / Experience concept, input journey & visual direction

SCOPE / Working browser project

| Time + place | 3 | 2 |
| --- | --- | --- |
| core input | documented visual modes | wallpaper formats |

## The challenge

A star map can be a technical chart or a personal object. BirthSky connects a simple biographical prompt to a visual reveal, then gives the person control over labels and the final wallpaper format.




Original BirthSky interface for entering a birth date, time and place to explore the sky.

BIRTHSKY / APPROACH & DECISIONS

# How I approached it

I organised the experience into a familiar prompt, a personal reveal, optional exploration, and a keepsake. Name, date, time, and place give the abstract sky a recognisable starting point.

The reveal needs understandable inputs. Unknown birth times, time zones, and any collection of personal details deserve clear explanations before a precise-looking image encourages confidence.

## 01 / Begin with a recognisable personal question

A short form collects the time and location before revealing the sky.

**Why it matters.** The input makes an otherwise abstract astronomical chart meaningful to the person requesting it.

**The trade-off.** Unknown birth times and historical time zones require careful explanation; a precise-looking map can still rest on uncertain inputs.

## 02 / Make annotation a choice

The result supports alternative modes and toggles for constellation lines and names.

**Why it matters.** A person may want to learn from the sky or simply use it as a visual keepsake.

**The trade-off.** Switches need clear effects without making the result feel like a complex scientific console.

## 03 / Design the output as part of the journey

Phone and desktop wallpaper downloads follow the reveal.

**Why it matters.** A useful result should fit where the person intends to use it.

**The trade-off.** Export must preserve legibility and composition across aspect ratios rather than just scale one screenshot.

BIRTHSKY / OUTCOME & REFLECTION

# What came out of it

I designed the journey from entering a time and place to adjusting a star map and exporting a wallpaper. Annotation and output choices let the person decide what belongs in the final image. Time uncertainty and data handling need to stay clear throughout.

# What I would do next

I would make it easier to work with an approximate birth time before adding another visual style. The result is personal, so the person should understand which inputs shaped it and what those inputs can reasonably tell them.

Proposed validation: compare known dates and locations with an authoritative sky reference, including historical daylight-saving and time-zone cases.

Proposed iteration: explain any logging before submission and make uncertain birth-time choices visible in the final result.

## Artifacts & evidence

**StarMap/index.html and js/app.js**
Input flow, result controls, modes, export entry points and configured logging integration.

**StarMap/js/astronomy.js and js/exporter.js**
Coordinate calculations and wallpaper generation. Reviewed 15 October 2026.

Read the individual web case study

**SCALESEE / QUANTITATIVE EXPLANATION · WORKING PRODUCT**

# How much bigger? Show the difference.

A visual comparison tool that turns quantities into scenes while keeping the meaning of scale explicit.

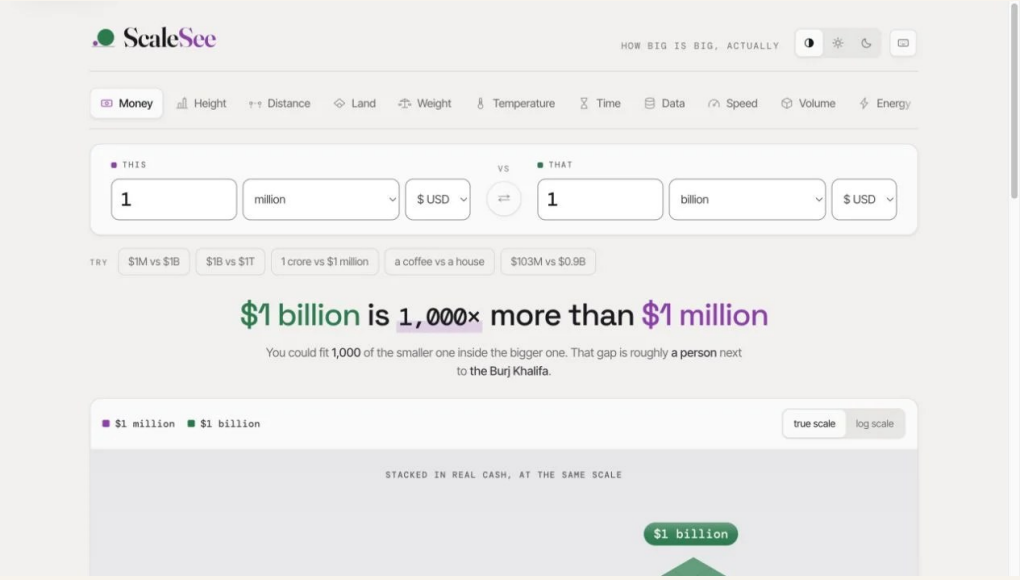
MY CONTRIBUTION / Visual explanation, comparison journeys & interaction design

SCOPE / Working browser product

| 11 | 2 | Shareable |
|---|---|---|
| categories listed in the source | scale modes | comparison URLs |

## The challenge

A ratio can be mathematically clear and still be difficult to picture. ScaleSee turns two quantities into a shared visual comparison, choosing a representation that fits the kind of quantity instead of using the same chart for everything.




Original ScaleSee comparison surface, with two quantities drawn against the same scale.

SCALESEE / APPROACH & DECISIONS

# How I approached it

I selected a visual reference appropriate to each kind of quantity: a ground line for height, nesting for land, or a shared course for speed. The reference is part of the explanation.

An extreme ratio creates a useful design tension: showing true scale can make one quantity nearly disappear. A compressed alternative needs an explicit label that remains with a shared comparison or exported image.

## 01 / Choose the picture to fit the quantity

Height shares a ground line, land can be nested, and speed uses a common course.

**Why it matters.** A meaningful reference can make a ratio easier to interpret than a generic bar.

**The trade-off.** Every visual metaphor introduces assumptions; those assumptions need to be stated where they affect the comparison.

## 02 / Keep scale transformations explicit

True scale and logarithmic scale are separate modes.

**Why it matters.** When a small quantity almost disappears, that can be the honest answer; a compressed alternative should declare the change.

**The trade-off.** A log view can make a scene more legible while reducing intuitive proportionality.

## 03 / Let the comparison leave the tool

A URL carries the selected comparison and an image export preserves the current drawing.

**Why it matters.** An explanation is more useful when someone can bring it into the conversation that prompted it.

**The trade-off.** An exported image needs labels and scale context because it no longer has interactive controls beside it.

SCALESEE / OUTCOME & REFLECTION

# What came out of it

I matched comparison scenes to the quantity being explored and kept the scale mode explicit. Sharing and image export let a comparison leave the tool with its context. Extreme ratios still need testing for correct interpretation.

# What I would do next

Sometimes a proportional picture is too small to read and an adjusted view is easy to misread. I would check whether someone can identify the scale mode and explain what the picture does to the original numbers.

Proposed study: ask readers to estimate the ratio before and after each view, including an extreme comparison and a logarithmic one.

Proposed check: confirm labels, units, assumptions, and scale mode survive in every shared or exported artifact.

## Artifacts & evidence

**Repository README, What's in it and Notes on the numbers**
Category inventory, visual metaphors, scale-mode distinction and sharing/export. Reviewed 15 October 2026.

Read the individual web case study

**WHERE WOULD I BE...? / SCIENTIFIC STORYTELLING · WORKING PRODUCT**

# Where would this place be as the plates move?

An interactive globe that lets a chosen place move with a plate-motion model and makes the model's limits part of the story.

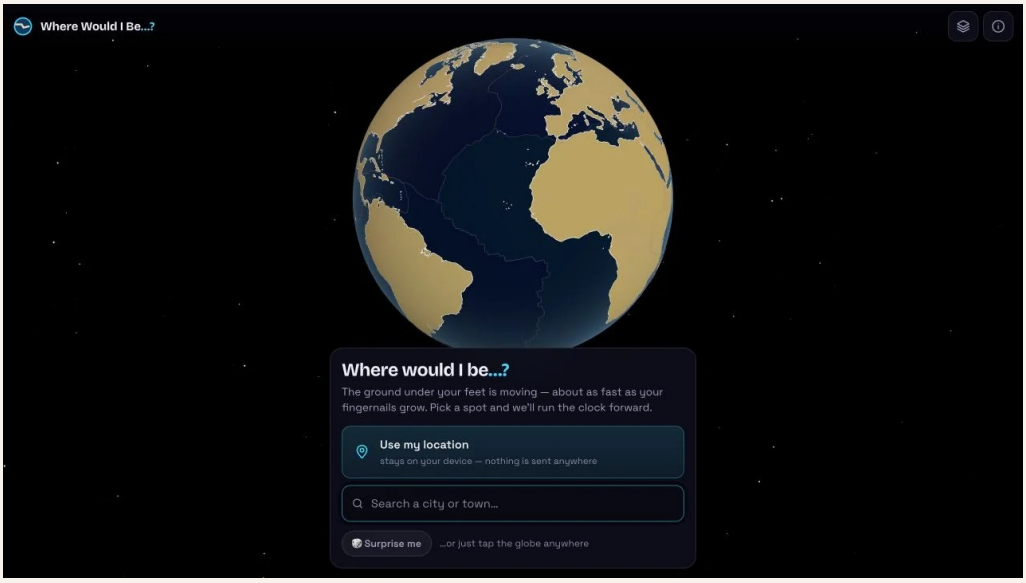
MY CONTRIBUTION / Experience framing, interaction sequence & spatial explanation

SCOPE / Working browser product

| 1 | Progressive | Modelled |
|---|---|---|
| place-led starting question | details on demand | extrapolation, not prediction |

## The challenge

Plate motion is slow enough to be invisible in everyday life and large enough to reshape a world over time. This project gives that scale a personal anchor: choose a place, move the clock, and follow the ground beneath it.




Original location entry screen. Choosing a place starts the exploration of plate motion through time.

WHERE WOULD I BE...? / APPROACH & DECISIONS

# How I approached it

I gave the opening one task: choose a place. A time control and a short result then introduce motion, with optional detail kept out of the main globe.

The visual story needs to carry its qualification with it. Constant-motion extrapolation can invite exploration, but the result should not read as a reconstruction or prediction of the continents.

## 01 / Give each stage one main task

Choose a place first, then explore time, with optional comparison later.

**Why it matters.** A clear sequence can make a complex subject approachable without presenting every control at once.

**The trade-off.** Progressive disclosure should not hide the model's most important qualification until after someone has formed a false impression.

## 02 / Protect the space where the answer is drawn

The design keeps the main globe clear and groups controls in a bottom dock.

**Why it matters.** The selected point and its path are the explanation; a floating panel over them would compete with the result.

**The trade-off.** A dock must adapt to available height so it does not consume the same space on a small screen.

## 03 / Make the scientific boundary legible

The source describes plate movement as a constant-motion extrapolation with increasing limits over long horizons.

**Why it matters.** Personal storytelling can encourage exploration without claiming to know the future shape of continents.

**The trade-off.** A smooth animation can still imply certainty, so the qualification belongs near the time result.

WHERE WOULD I BE...? / OUTCOME & REFLECTION

# What came out of it

I used a chosen location as the starting point for exploring time on the globe. Time controls, comparison and optional layers support the exploration, while the model limit stays near the result. The long-term path is an extrapolation, not a geological forecast.

# What I would do next

I would ask someone to explain why the point moves and why the long-term path is conditional. The interaction should leave them with a useful question about plate motion, not misplaced confidence in a prediction.

Proposed study: ask a visitor to describe what changed and what assumptions the model makes after a short exploration.

Proposed checks: narrow-screen globe visibility, reduced-motion behaviour, keyboard interaction, and whether uncertainty stays visible at extreme dates.

## Artifacts & evidence

**Repository README, The layout rule and How much to trust it**
Interaction sequence, unobstructed visual space and constant-motion model limitations. Reviewed 15 October 2026.

Read the individual web case study

THE BLOG / PUBLISHING EXPERIENCE · WORKING PRODUCT

# A home for my writing, in the form it needs.

An editorial site with a browser-based publishing workflow for poems, essays, fiction, and serial writing.

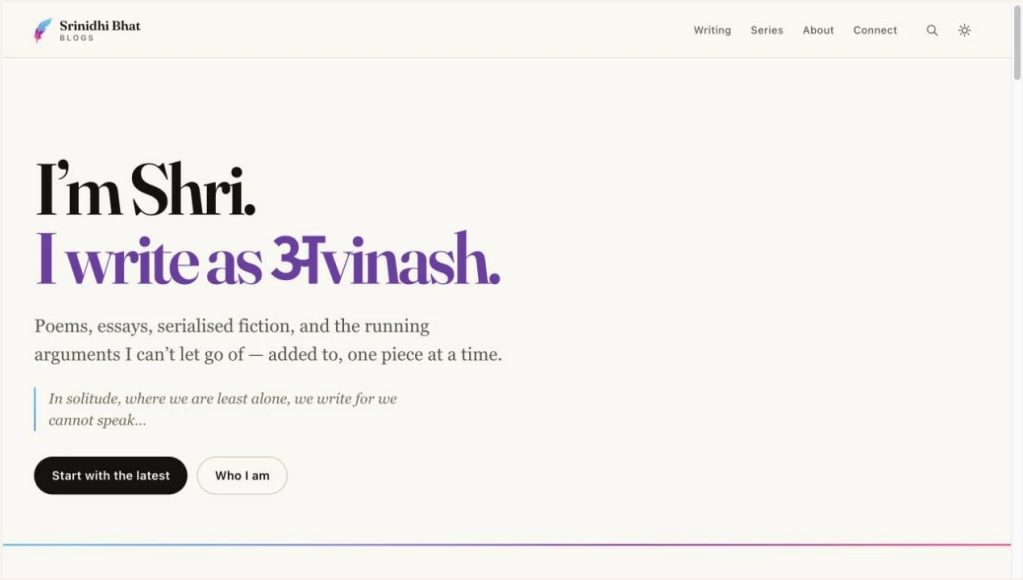
MY CONTRIBUTION / Editorial structure, reading experience & author workflow

SCOPE / Working publishing system

| 2 | Drafts | Verse |
| --- | --- | --- |
| reader and author workflows | explicit publishing control | preserved line breaks |

## The challenge

A personal writing practice should not require a technical deployment ritual for every post. The Blog connects a reading experience with a browser editor that handles content, images, drafts, and publishing through a familiar form.




Original writing homepage under the name अvinash, with poems, essays, fiction and a written podcast.

THE BLOG / APPROACH & DECISIONS

# How I approached it

I framed writing, reviewing, and publishing as separate commitments. The editor should support the author's work without requiring technical knowledge of how a post reaches the website.

The reading experience follows the form of the writing: poetry keeps its line breaks, while topics and series help a reader continue. Publication feedback should make draft and live states easy to distinguish.

## 01 / Give publishing a clear state

Draft and published are separate states, with a deliberate release action.

**Why it matters.** Writing and releasing a piece are different commitments and should remain distinguishable.

**The trade-off.** Publishing can take time. The author needs clear confirmation of what is saved and what is live.

## 02 / Let the content form affect the layout

Poetry preserves line breaks and an appropriate verse treatment, while ordinary articles retain a reading layout.

**Why it matters.** The way a piece is written can be part of its meaning.

**The trade-off.** Formatting conventions must be discoverable in the editor rather than depend on hidden technical knowledge.

## 03 / Make ongoing writing navigable

Topics and series connect related pieces within the editorial structure.

**Why it matters.** A reader should be able to follow a thread without knowing the order it was published.

**The trade-off.** The documented title-based series matching is convenient but implicit; a visible series field would make mistakes easier to avoid.

THE BLOG / OUTCOME & REFLECTION

# What came out of it

I gave poems, essays and ongoing stories a reading structure suited to each form. The publishing flow separates drafts from release, while topics and series help a reader continue. Poetry keeps its line breaks instead of being squeezed into an ordinary article template.

# What I would do next

I would make draft status and series membership more explicit in the editor. Those small details affect whether I can publish the right version and whether someone can find the next part of a story.

Proposed evaluation: publish a draft, update an existing post, and add a series instalment without using development tools.

Proposed measure: accidental publication, broken links after slug edits, and clarity about whether the latest save is live.

## Artifacts & evidence

**Repository README, Writing a post and Things worth knowing**
Browser editor, draft/build publishing path, verse formatting and title-based series conventions. Reviewed 15 October 2026.

Read the individual web case study

THE OFFICE GAMES / SOCIAL PLAY · WORKING PRODUCT

# Get everyone in. Keep the round moving.

A team-trivia experience exploring quick room entry, visible roles, and shared game state for a short social break.

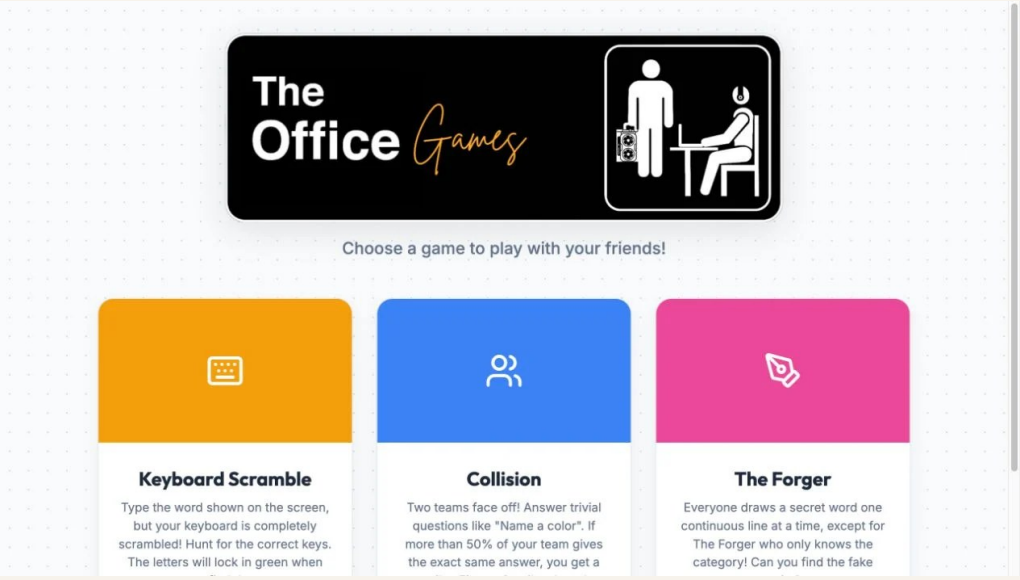
MY CONTRIBUTION / Social game concept, room journey & turn-taking UX

SCOPE / Working project · trivia module

| 4 | Team-based | Explicit |
|---|---|---|
| letters in a room code | turn and score model | local or network mode |

## The challenge

A short team game loses momentum when people are unsure how to join or whose turn it is. The reviewed Office Games module centres on a simple room flow, a lobby, team roles, and a visible answering/stealing sequence.




Original Office Games entry screen. A group chooses a game before joining a shared room.

THE OFFICE GAMES / APPROACH & DECISIONS

# How I approached it

I mapped the reviewed trivia journey from joining a room to understanding teams, turns, answering, and stealing. The broader games catalogue is outside the evidence available for this case study.

The UX needs to explain the connection mode before a group commits to a game. Trying a local demonstration and joining colleagues on separate devices are different experiences.

## 01 / Make joining a small action

A four-letter room code and a player name provide a concise entry path.

**Why it matters.** A social break should spend more of its time playing than explaining setup.

**The trade-off.** A short code still needs clear invalid-room and full-room handling, and does not establish secure identity.

## 02 / Expose responsibility in the game state

The interface distinguishes answering and stealing teams, active players, and scores.

**Why it matters.** People need to see who can act and why the round is waiting.

**The trade-off.** Role-based interaction becomes confusing if waiting states or connection loss are not explained.

## 03 / Keep connection mode understandable

The setup distinguishes a local demonstration from connected play.

**Why it matters.** A demo can help someone try the game, but should not promise multi-device play it cannot provide.

**The trade-off.** A connected game needs clear waiting, conflict, reconnection, and host-recovery states.

THE OFFICE GAMES / OUTCOME & REFLECTION

# What came out of it

For the trivia flow documented here, I focused on room entry, teams, scores and the answering-and-stealing sequence. The interface gives the group a shared view of who can act next. Local demo mode and connected play need different expectations.

# What I would do next

I would run a short session and pay attention to the pauses: joining, waiting for a team, recovering a connection and changing the host. The answer screen is only one part of whether the break feels easy to join.

Proposed study: run a short trivia session with several participants and record setup explanations, confused turns, and waiting time.

Proposed checks: separate-device joining, reconnection, duplicate actions, host departure, and a clearly labelled local demo mode.

## Artifacts & evidence

**src/App.jsx and src/gameService.js**
Reviewed room/lobby/trivia journey, short codes, team state, local-storage mock service and Firebase path. Reviewed 15 October 2026.

Read the individual web case study

FORMA / PERSONAL HEALTH

# Health readings with the context attached.

A native Android health app that brings activity, sleep and personal trends together while making sources, missing data and experimental estimates visible.

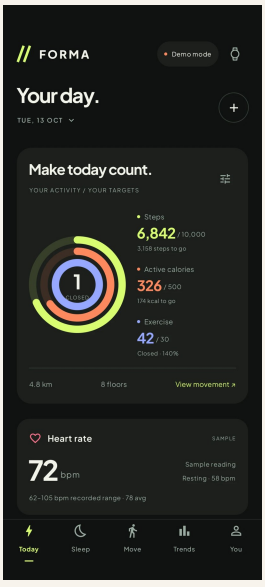
MY CONTRIBUTION / Product concept, health-data journeys & interaction design

SCOPE / Native Android app · experimental wellness estimates

| 5 | 44 | Recorded |
|---|---|---|
| main destinations | native metric views | measurement times retained |

## The challenge

A watch can collect more measurements than a person can comfortably interpret. A number also changes meaning when it is old, incomplete, imported or estimated. I framed Forma around a daily overview that stays useful without making uncertain data look more reliable than it is.




Archived native Today screen with clearly marked demo values. This capture predates the later Health-tab and personal-tool cleanup.

FORMA / APPROACH & DECISIONS

# How I approached it

I organised the experience around five destinations: Today for a daily overview, Sleep for the recorded night, Move for activity and workouts, Trends for comparison, and Health for the measurement library. The native app is the product; the React version in the repository is a design preview.

The data constraints shaped the UX. Samsung Health must share records through Health Connect, permissions can be partial, and not every watch measurement is available. Connection setup therefore needs to explain availability rather than treat an empty reading as a failure or a zero.

The later function audit separated Devices from Your data, moved optional personal tools into settings, removed controls without a working purpose, and put formulas behind expandable details. The archived screens illustrate the native visual direction; the audit and current source establish the later navigation and scope. This is an artifact-based retrospective, without a participant study or clinical validation.

## 01 / Distinguish recorded, missing and estimated values

Keep source labels and measurement times attached to readings. Preserve gaps in histories, identify sample data, and distinguish sleeping heart rate from a recorded resting-heart-rate value.

**Why it matters.** A recent refresh does not mean a sensor reading is recent. People need to understand what a value represents before comparing it with yesterday or using it to interpret their day.

**The trade-off.** These labels add density. The hierarchy has to keep essential provenance visible while leaving longer explanations available on demand.

## 02 / Let a score open into its explanation

Show the score and its contributing values first, then offer the calculation and diagnostics in an expandable sheet. Recovery requires recorded sleep and a usable baseline vital; missing inputs can leave the estimate unavailable.

**Why it matters.** An unexplained score can look authoritative. Showing its inputs makes the estimate inspectable and lets the interface explain why the same score can accompany different activity records.

**The trade-off.** Transparency does not validate the model. Sleep, recovery and body-energy estimates remain experimental wellness interpretations, distinct from Samsung proprietary scores and medical assessment.

## 03 / Separate connecting a device from managing data

Devices leads with connection state and the next available action. Your data contains coverage, export, optional import and local reset. Targets and personal tools have their own settings paths.

**Why it matters.** Someone trying to connect a watch should not have to interpret a coverage report or navigate file-management controls before granting access.

**The trade-off.** Separating the tasks adds destinations. Clear names and back paths matter, particularly when permissions are partial or a watch has not synced.

FORMA / OUTCOME & REFLECTION

# What came out of it

I organised the native app around a daily overview, dedicated sleep and movement views, histories and a searchable health library. A score opens into its contributors; device setup and data management have separate destinations. The estimates remain experimental, and Samsung sharing depends on the device and permissions.

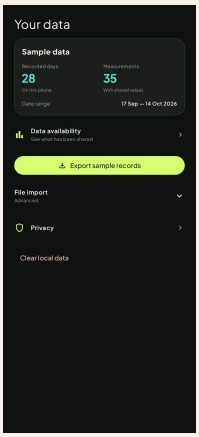
# What I would do next

Separating Devices from Your data made the next action clearer. I would now check whether someone can explain an unavailable or older reading as readily as a complete one, and whether they can find the inputs behind a score.

Proposed study: ask people to distinguish sample, shared, direct-watch, older and unavailable readings, then explain what an experimental recovery score does and does not mean.

Proposed device checks: follow granted, denied and partial-permission journeys on a Samsung phone and watch, including delayed sync, expired direct readings and corrected history.

Proposed measures: connection-task completion, mistaken interpretations of missing data, and success finding a score explanation or exporting records.




The later Your data panel groups coverage and file controls separately from device setup. All values shown are sample data.

## Artifacts & evidence

**Forma README and native interface source**
Reviewed 16 October 2026. Native Android scope, five destinations, Samsung Health/Health Connect integration, measurement timestamps and experimental score limits. The web project is a design preview.

**docs/DESIGN.md and docs/UX-AUDIT.md**
The later 1.5 audit supersedes the older 46-measurement design note: 44 native metric views, distinct Devices/Your data flows, optional personal tools and expandable score explanations.

**Archived native captures · sample data**
Today rings and vitals captures from the project output illustrate the native visual system. The Today capture predates the Health tab rename and personal-tool cleanup; the later Your data capture shows the separated data-management flow. Values are demonstration data.

Read the individual web case study

**OHCAPTAIN / NAVAL CAPTURE-AND-HOLD · OPERATION BLACKWATER**

# Sail your ship. Coordinate the fleet.

A 3-v-3 naval game exploring shared command, role-specific decisions and a combat interface that keeps the sea in view.

MY CONTRIBUTION / Game concept, captain/operator journeys & tactical interface design

SCOPE / Playable browser prototype · keyboard & mouse

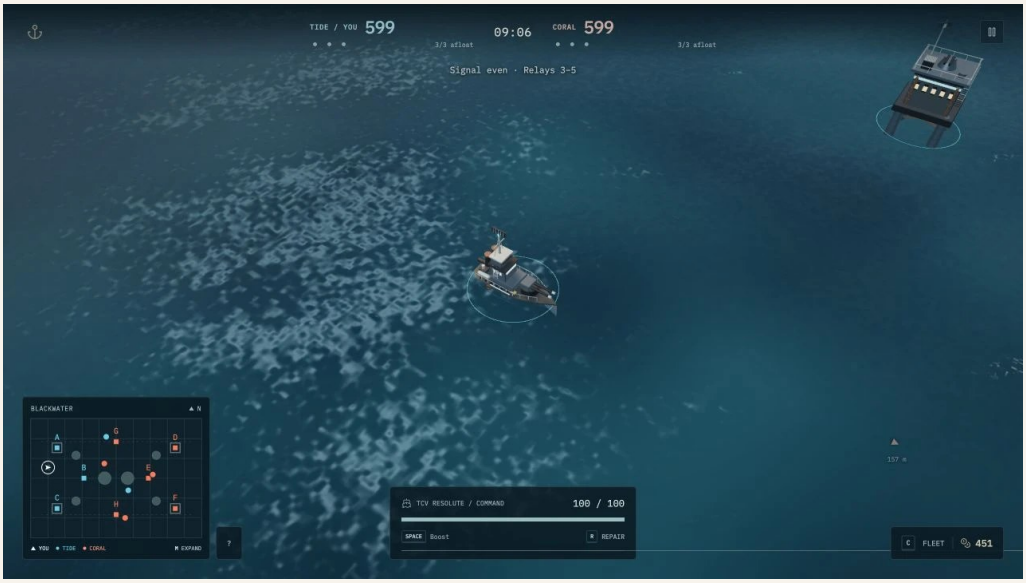
**3 v 3**
captain and two operators per crew

**8**
capture-and-hold relays

**10 min**
maximum match window

## The challenge

In OhCaptain, sailing and firing are only part of the job. Each crew shares supplies, but its captain decides how to spend them. Operators need to request help while both roles keep track of a changing battle. I framed the interface around the next decision a player can actually make.




Actual local gameplay with an AI-filled crew: compact battle status, a persistent map, the Fleet control and the player's hull.

OHCAPTAIN / APPROACH & DECISIONS

# How I approached it

I separated the crew journey into choosing a station, entering solo play or assembling a room, deploying, and managing the battle. A captain still sails and fights; the extra responsibility is distributing shared supplies. Operators request upgrades or repairs rather than spending the treasury themselves.

The documented Clear Deck revision gives combat a compact score/clock, fleet-alive counts, hull status and a persistent map. Spending moves into the Fleet drawer, while the expanded overview holds rosters, recent events and relay details. Warnings compete for a small number of attention slots instead of becoming a wall of banners.

I reviewed the local playable build, documentation and awareness rules. Solo play with AI fill and peer-to-peer rooms are distinct product paths. Online scope depends on a host keeping the page open; the prototype has no dedicated match service or host migration. Rule checks and seeded scenarios do not establish balance or usability with human crews.

## 01 / Make responsibility visible before deployment

Offer Captain and Vessel Operator as distinct stations. Captains distribute shared supplies through the Fleet drawer; operators use the same crew context to request an upgrade or repair.

**Why it matters.** A shared resource needs a clear decision owner. Role-specific actions let people understand why they can request something but cannot buy it directly.

**The trade-off.** The captain can become a bottleneck while also sailing and fighting. Pending requests need to remain discoverable without forcing a panel over combat.

## 02 / Keep essential battle state on the field

Keep the score, clock, relay balance, fleet-alive counts, hull and corner map available. Put spending in a drawer and detailed rosters, events and defenses in the expanded overview.

**Why it matters.** The map helps connect local action to the relay objective. Separating routine awareness from deliberate inspection leaves room to aim and navigate.

**The trade-off.** An overview can interrupt action. Solo overview pauses the match; online play continues, so players need to understand that distinction. The prototype targets keyboard and mouse rather than claiming a touch-control experience.

## 03 / Give urgent feedback a limited attention budget

Show at most one off-screen enemy bearing and reserve one text cue for critical hull, safe healing or an endangered home relay. Limit outgoing damage feedback to two numbers and keep routine radio outside combat.

**Why it matters.** An alert is useful only if it helps a player choose an action. A priority order makes a received hit or low hull easier to notice among effects, ships and objectives.

**The trade-off.** A quieter interface can hide context that an experienced player wants. The map and overview must provide a dependable route to that detail, and the priority policy needs human playtesting.

OHCAPTAIN / OUTCOME & REFLECTION

# What came out of it

I separated immediate battle awareness, fleet spending and tactical inspection across the combat view, Fleet drawer and overview. The prototype supports solo AI-filled matches and crew rooms in one capture-and-hold arena. The captures here show the actual game; the illustrated harbor is menu artwork.

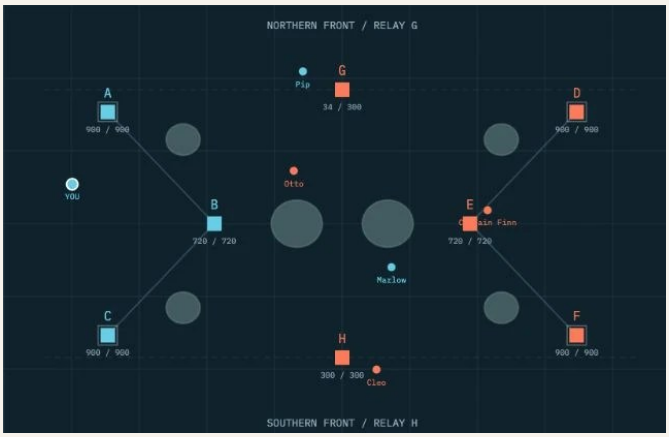
# What I would do next

I would put a new crew through requesting a repair, spending supplies and choosing a relay without coaching. Missed requests and time spent searching the overview would help me decide what to simplify next.

Proposed playtest: form two crews and ask players to explain their role, request or buy a repair, and identify the relay that matters next without coaching.

Proposed checks: invitation and room entry, role reassignment, a guest disconnecting, restrictive peer-to-peer networks, and an honest explanation when a host leaves.

Proposed observations: missed requests, map-to-world confusion, unnoticed critical alerts and time spent in the overview. Compare these before changing the attention policy.




A detail of the actual expanded tactical map. Both fronts, ship positions and relay ownership share the same view; the solo overview pauses play.

## Artifacts & evidence

**OhCaptain README · Play, online limits and Clear Deck interface**
Reviewed 16 October 2026. Captain/operator responsibilities, AI fill, peer-to-peer rooms, eight-relay objectives, ten-minute match window and host/network limitations.

**src/awareness.js, src/main.js and src/combat.css**
Battle interpretation, role-specific controls, attention priority, local damage feedback and the single off-screen enemy-bearing policy.

**Local playable build and docs/BLACKWATER-ART.md**
Actual solo combat and expanded-map captures from 16 October 2026. The harbor capture includes the project's generated cinematic key art; it is menu presentation, distinct from the procedural playable arena.

Read the individual web case study

**HALO / POINTER-BASED SHORTCUTS**

# Keep the action close to the pointer.

A desktop action ring for frequently used shortcuts, with app-specific layouts and a visual editor.

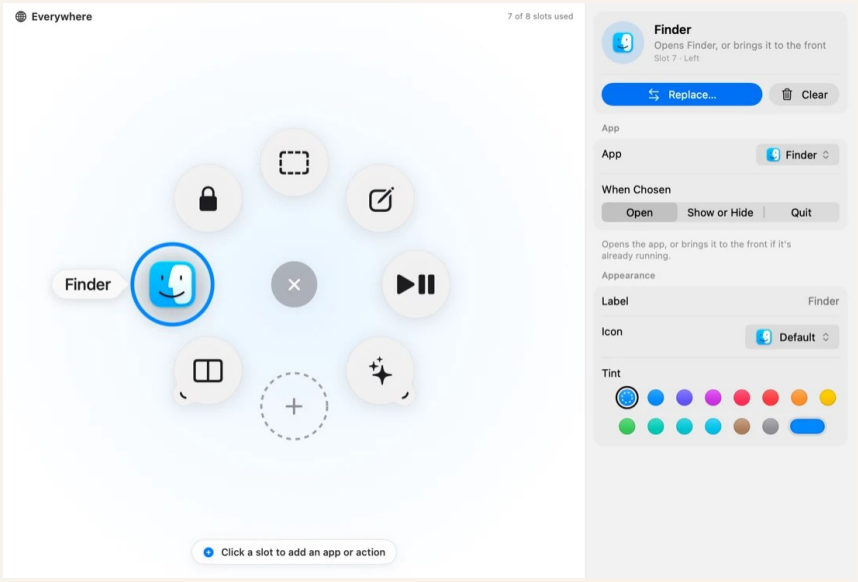
MY CONTRIBUTION / Interaction direction, action organisation & configuration UX

SCOPE / Desktop utility · macOS and Windows

| 8 | Per app | Click or flick |
|---|---|---|
| slots in a ring | different action sets | two ways to choose |

## The challenge

A useful shortcut is one I can reach without breaking the task I am doing. Halo puts a ring of actions around the pointer, instead of asking me to hunt through menus or remember another keyboard combination. The design problem was keeping that first choice quick while still allowing the ring to grow.




Original Halo editor capture from the project documentation. The action's position and settings are visible together.

HALO / APPROACH & DECISIONS

# How I approached it

I organised the experience around opening the ring, choosing a direction and returning to the original task. Clicking supports a deliberate choice; holding the shortcut, moving towards an action and releasing supports a quicker gesture once the layout is familiar.

The editor exposes the same ring spatially. Selecting a slot opens its action, app behaviour, label and appearance settings. App-specific rings and folders hold more choices without putting every action into the first ring.

Halo is an independent project inspired by Logitech's Actions Ring. I focused on the interaction and visual configuration. Comparing click and flick selection with familiar keyboard shortcuts is still a useful next test.

## 01 / Make the first choice spatial

Keep actions in stable positions around the pointer, with labels available while choosing.

**Why it matters.** A direction can become familiar through use, while labels help someone who is still learning the layout.

**The trade-off.** Moving actions frequently would work against that familiarity. Customisation needs to feel deliberate.

## 02 / Let the editor mirror the ring

Select a slot in the ring and edit that action beside it.

**Why it matters.** The person can see which position they are changing instead of translating between a list and the final layout.

**The trade-off.** A visual editor still needs clear names and keyboard access; position alone is not enough.

## 03 / Keep specialised actions out of the first layer

Use app-specific rings and folders for actions that belong to a particular context.

**Why it matters.** A general ring can stay small while a specialist workflow has its own useful choices.

**The trade-off.** Context changes and nested folders must remain predictable. More actions do not automatically make the tool faster.

HALO / OUTCOME & REFLECTION

# What came out of it

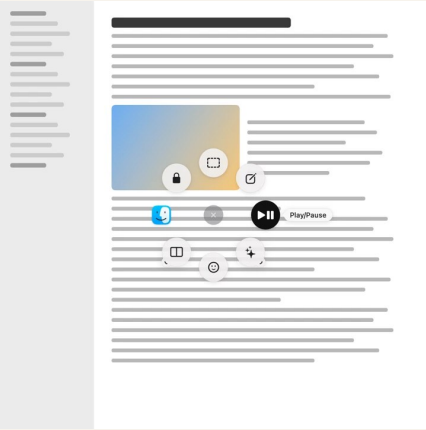
I gave the ring and its editor the same spatial model, with click and hold-and-flick selection, per-app configurations and nested action folders. The app supports actions such as app control, media, window management and snippets. Some actions depend on the operating system and the permissions available.

# What I would do next

I would start a comparison with a small, familiar task set rather than a ring full of possibilities. I want to know which positions people remember, when they hesitate and whether app-specific rings help or surprise them.

Proposed study: configure three frequently used actions, repeat the task after a break and compare hesitation with an existing shortcut method.

Proposed checks: keyboard-only configuration, ring changes between apps, folder backtracking and permission-denied actions.




Original ring example from the project documentation, shown over a neutral illustrated page rather than a live document.

## Artifacts & evidence

**Halo README · using and customising the ring**
Click/flick selection, app-specific rings, folders, platform support and permission boundaries. Reviewed 31 October 2026.

**Original editor and ring documentation images**
Action-Ring/docs/images/editor.png and classic.png. Existing project graphics, not newly generated interface concepts. The ring example uses a neutral illustrated background.

Read the individual web case study

**BINNDER / STORAGE REVIEW**

# Review the file. Keep a way back.

A native Android file-review app that combines large previews with explicit keep, bin and permanent-delete actions.

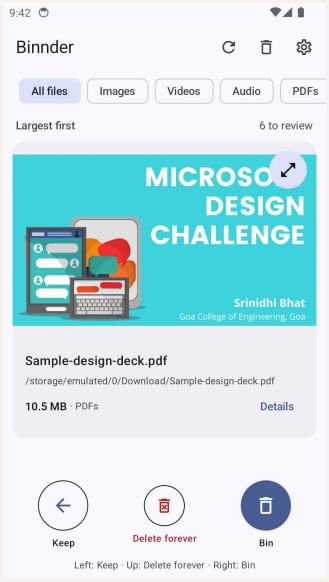
MY CONTRIBUTION / File-review flow, destructive-action UX & information hierarchy

SCOPE / Native Android app · early release

| 5 seconds | 30 days | 7 days |
|---|---|---|
| undo window | bin retention | before kept files return |

## The challenge

Storage cleanup is easy to rush and difficult to undo. I wanted the file itself to be the main thing on screen, with a clear difference between keeping it, moving it to a bin and deleting it permanently. The interaction should make those consequences visible before a gesture becomes an operation.




Native Android capture using a sample portfolio PDF in an isolated emulator. Fixed controls provide the same actions as swiping.

BINNDER / APPROACH & DECISIONS

# How I approached it

I put the largest files first and used a large preview with fixed actions. File-type filters help narrow the queue, while matching buttons and accessibility actions make swiping optional.

I separated reversible binning from permanent deletion. An upward swipe asks for confirmation, and every deletion action has a five-second Undo window before the operation begins. A kept file is hidden from review for seven days rather than treated as permanently resolved.

The bin reports its retention period and supports preview and restore. Moving a file there does not free storage, so the interface must not count those bytes as recovered space. The captures use sample portfolio files in a separate emulator, not a scan of someone's phone.

## 01 / Keep the file visible while deciding

Use a large preview, file name, size and fixed Keep, Bin and Delete forever controls.

**Why it matters.** The decision should be about a recognisable file, not an unfamiliar path in a long list.

**The trade-off.** Some formats need another installed app. The preview must explain that limitation without implying a file is broken.

## 02 / Give permanent deletion its own commitment

Ask for confirmation and name the file before queuing permanent deletion, then offer the individual Undo window.

**Why it matters.** A gesture alone is too easy to misinterpret when the result cannot be recovered through the app.

**The trade-off.** Extra confirmation slows cleanup. It belongs to the more consequential action rather than every decision.

## 03 / Describe the bin honestly

Show retention and restoration separately from bytes actually deleted.

**Why it matters.** A recoverable file still occupies storage. The person needs to understand both recovery and capacity.

**The trade-off.** Retention depends on the device being able to run cleanup and reach the volume; it is not an exact deletion appointment.

BINNDER / OUTCOME & REFLECTION

# What came out of it

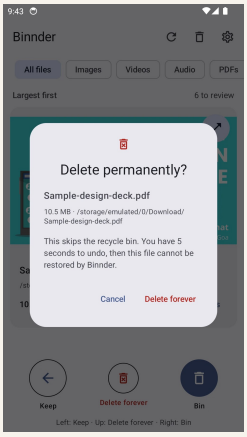
I designed one-file-at-a-time review with visible alternatives to gestures, a recoverable bin and an explicit permanent-delete path. The native app also accounts for interrupted operations and changed files. I still need to check those recovery paths across Android devices and removable storage.

# What I would do next

I would test a mistaken swipe, a removed storage volume and a restore destination that already exists. A cleanup tool earns trust when it explains what happened and preserves a way to recover, especially when it cannot finish an operation.

Proposed study: review a small set of test files, undo an unintended action and restore a file without coaching.

Proposed checks: permission loss, app restart during a pending action, missing SD card and a file changed since scanning.




The actual confirmation names the sample file, explains that the bin is skipped and describes the five-second Undo window. No deletion was confirmed for this capture.

## Artifacts & evidence

**Binnder README and ReviewPolicy/Storage/SafeDisk source**
Documented five-second Undo, seven-day keep, thirty-day bin, delayed operations, file identity checks and storage-accounting limits. Reviewed 31 October 2026.

**Native Android captures · isolated test files**
Captured from Binnder 0.1.1 in a separate Android emulator. The displayed design deck is a sample copy of an existing portfolio PDF; no personal phone files were read.

Read the individual web case study

**PUREEQ / PERSONAL AUDIO TUNING**

# Find a sound you can adjust and keep.

An Android sound-tuning app that combines a per-ear listening check, music preferences and editable EQ profiles.

MY CONTRIBUTION / Setup journey, listening-test interaction & profile controls

SCOPE / Native Android app

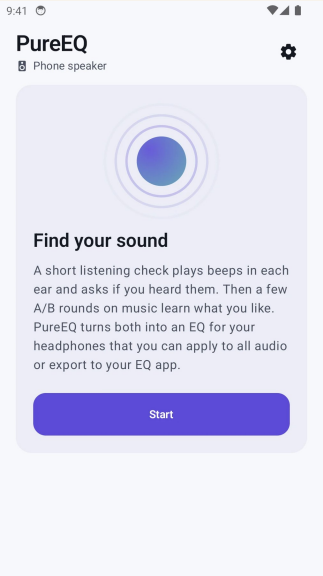
### Per ear
separate listening input

### A/B
music preference choices

### Export
take the profile elsewhere

## The challenge

A row of equaliser sliders assumes that someone already knows which frequencies they want to change. I approached PureEQ as a guided starting point: explain the setup, gather a few listening responses, compare music and leave the result open to adjustment.




Actual native welcome screen from an isolated Android emulator. It explains the two inputs before a profile is created.

PUREEQ / APPROACH & DECISIONS

# How I approached it

I separated the listening check from the taste test. The first asks whether a short tone is audible through the current headphones; the second asks which of two music treatments the person prefers. Neither needs to be presented as an expert diagnosis.

Preparation is part of the flow: a quiet environment, suitable headphones, stable volume and a left/right check. The result can then be adjusted with named controls, compared with EQ off and saved as a profile.

Export gives the profile a useful life beyond the setup. The documented formats have different capabilities, so a combined curve or per-ear adjustment needs explanation. The captures show the actual native starting flow, not a completed hearing test or a clinical result.

## 01 / Explain the two kinds of input

Describe listening responses and music preferences as separate steps before asking for either.

**Why it matters.** A person should know whether they are reporting audibility or choosing a sound they like.

**The trade-off.** A short explanation must not make the listening check sound like a calibrated audiogram.

## 02 / Leave the suggestion editable

Offer named fine-tuning controls, an EQ on/off comparison and a way back to suggested settings.

**Why it matters.** The suggested curve is a starting point. People need to hear and revise it rather than accept it as an unquestionable result.

**The trade-off.** Several controls can become confusing together. Reset and suggested-value feedback need to make changes traceable.

## 03 / Carry context into export

Describe the target format and explain when an app combines left and right curves.

**Why it matters.** A technically valid export can still behave differently from the per-ear result shown in PureEQ.

**The trade-off.** Compatibility depends on the receiving app and device. Exporting a file does not guarantee the same playback everywhere.

PUREEQ / OUTCOME & REFLECTION

# What came out of it

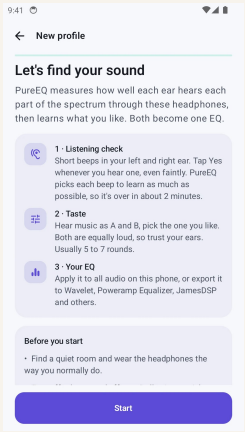
I connected preparation, listening responses, A/B choices and an editable result into a profile flow. The app can apply the EQ or export it for another player, with limitations explained for each format. This is personal sound tuning, not a clinical hearing assessment.

# What I would do next

I would test whether people can tell what came from their listening responses, what came from preference choices and what they changed afterwards. If those differences are unclear, the final curve may look more certain than it should.

Proposed study: complete setup with headphones, explain each stage and compare a saved result with EQ off at matched loudness.

Proposed checks: interrupted setup, a volume change, incorrect channel identification, resetting one adjustment and exporting a combined curve.




The actual preparation screen separates listening, taste and the resulting EQ. No hearing-test result is shown.

## Artifacts & evidence

**PureEQ README and native UI/theme source**
Listening and preference stages, preparation, editable profiles, reset behaviour and export limitations. Reviewed 31 October 2026.

**Native Android captures · no listening result**
Captured from PureEQ 1.5.2 in a separate Android emulator. The welcome and preparation screens are genuine app UI; no human listening responses or clinical measurements are represented.

Read the individual web case study

**HIDDEN IDENTITY CHESS / INDEPENDENT ADAPTATION · TRIPLE S GAMES RULES**

# Familiar chess. Very different information.

A digital adaptation of Hidden Identity Chess, with private setup, bluffing, challenges and a focused play surface.

MY CONTRIBUTION / Digital adaptation, private setup & game interaction design

SCOPE / Playable web game

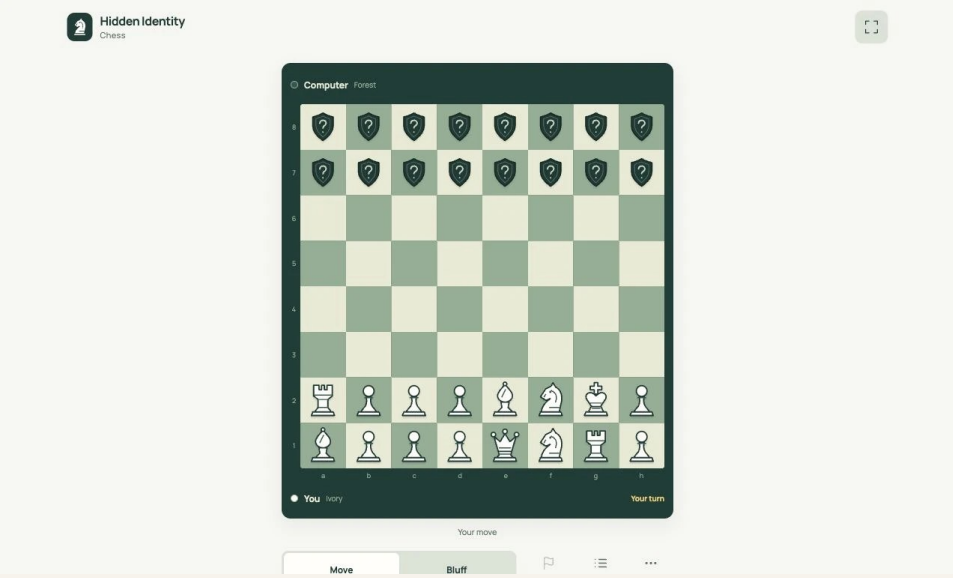
## Private
opponent identities stay hidden

## 3 modes
computer, shared device and online

## Move / bluff / challenge
explicit turn actions

## The challenge

A familiar chessboard is useful orientation, but it can also create the wrong expectations when piece identities are hidden and bluffing is allowed. I focused the adaptation on explaining what is private, what a player can do now and how a challenge changes the game.




Actual local computer-mode interface. The opponent's identities remain hidden while the player's own army is visible.

# How I approached it

I separated arranging an army from playing it. In same-device mode, a covered handoff protects the next player's view. Online play needs the same distinction in what each player receives, not just a visual cover over shared data.

Zen mode keeps the board, players, turn and Move, Bluff and Challenge controls in view. History, hints and secondary tools move into sheets so they can be inspected without permanently shrinking the board.

The rules and original game concept are credited to Triple S Games. This is an independent digital adaptation. I treated computer play, shared-device play and online rooms as separate flows because they have different privacy and connection requirements.

## 01 / Make hidden information explicit

Show the player's own identities and keep the opponent's pieces visually hidden.

**Why it matters.** Someone should understand the information available to each side before making a move.

**The trade-off.** Shared-device privacy depends on a clear handoff; it cannot stop someone looking at the screen at the wrong time.

## 02 / Name the unusual actions

Keep Move, Bluff and Challenge as separate controls with rules available nearby.

**Why it matters.** The familiar board should not make a new player assume ordinary check-and-checkmate behaviour.

**The trade-off.** A challenge can refer to earlier play, so its target and consequence need careful explanation.

## 03 / Give the board the space

Use the focused play view and move history and secondary tools into deliberate inspection surfaces.

**Why it matters.** The board is easier to read when optional tools do not compete for its space.

**The trade-off.** Hiding history can make challenges harder to investigate. The route back to a past move must remain obvious.

HIDDEN IDENTITY CHESS / OUTCOME & REFLECTION

# What came out of it

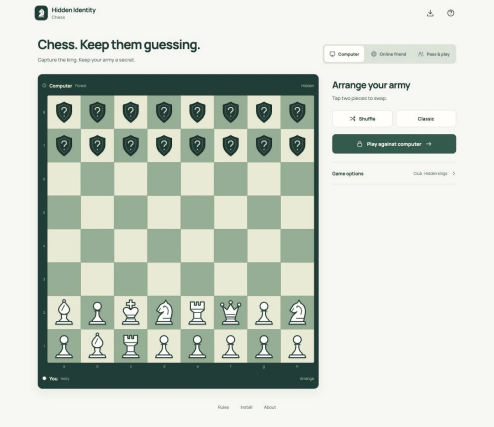
I designed setup, private handoff and the focused game view around the information each player can use. The game supports computer, same-device and online play, with explicit challenge controls and local saving. I still want to test whether a new player understands the handoff and can investigate a challenge.

# What I would do next

I would watch where an experienced chess player applies an ordinary rule that does not belong here. Those moments would tell me where the adaptation needs a clearer explanation rather than another hint.

Proposed study: arrange an army, make a bluff and investigate a past move with a first-time player.

Proposed checks: shared-device handoff, keyboard square navigation, a disconnected online player and restoring a local game.




Actual setup screen with army arrangement and the separate computer, online and shared-device modes.

## Artifacts & evidence

**Project README and original rules credit**
Private setup, computer/shared-device/online modes, focused play, challenge handling and credit to Triple S Games. Reviewed 31 October 2026.

**Actual local game captures**
Captured from the local built game under /game/ in computer mode. Positions illustrate interaction states; no human playtest result is represented.

Read the individual web case study

**5 DEAD / INDEPENDENT GAME INSPIRED BY 5 ALIVE**

# Keep the table moving. Keep the turn clear.

An independent digital card game with bot play, shared-device turns and online rooms, built around a visible running total and card actions.

MY CONTRIBUTION / Game presentation, joining flow & turn feedback

SCOPE / Playable card game

| 21 | 3 modes | 2-24 |
| --- | --- | --- |
| safe running-total limit | bots, pass-and-play and online | extended party-table scope |

## The challenge

A card table has several things to explain at once: the current total, whose turn it is, what a card will do and who is still in the hand. I wanted the presentation to give each moment a clear focus without making players work through a rules panel on every turn.




Original game capture. The arena, avatars and skull card art are generated assets used by the playable game; labels and controls are part of the actual interface.

5 DEAD / APPROACH & DECISIONS

# How I approached it

I kept the running total and current discard near the centre, opponents around the table and the local hand close to the player. Selection and a Play action help separate considering a card from committing it.

The three entry modes make different promises. Bot play and local pass-and-play can work offline after the app is ready; online rooms depend on a connection and wait for a disconnected player. Shared-device turns use a privacy screen between hands.

This is an independent game inspired by Hasbro's 5 Alive. Larger tables use a disclosed house rule. The generated arena and card artwork are assets used in the game itself; I kept the running total, hand and turn feedback distinct from that background.

## 01 / Keep the running total central

Place the total, discard and turn state where the next decision is made.

**Why it matters.** A player needs to connect the chosen card to the current state of the hand.

**The trade-off.** Effects and animations can obscure that relationship if they linger or hide the next turn.

## 02 / Separate choosing from playing

Show a selected card and an explicit Play action, alongside the supported drag interaction.

**Why it matters.** Considering a card should not accidentally commit it, especially on a crowded phone hand.

**The trade-off.** An extra action can slow experienced play. Both methods need clear feedback and usable touch targets.

## 03 / Explain the mode before the table starts

Distinguish bot, pass-and-play and connected rooms, including their waiting and privacy states.

**Why it matters.** A group should know whether it needs one device, several devices or a network before starting.

**The trade-off.** Local privacy covers are a social convention; online connection loss can pause the table.

5 DEAD / OUTCOME & REFLECTION

# What came out of it

I brought the hand, table, turn feedback and joining flow into one playable presentation. Bot and local play work separately from connected rooms. Larger tables follow the documented house rules, with waiting and privacy states suited to each mode.

# What I would do next

I would test the smallest phone hand and the largest table before adding more effects. The useful questions are whether the next player notices their turn and whether someone can tell what a special card just changed.

Proposed study: start a bot hand, play a special card and explain the next action without opening the full rules.

Proposed checks: overlapping cards, shared-device handoff, a disconnected seat and the clarity of the extended-table rule.




Original game capture showing the phone hand, special cards and the Play control.

## Artifacts & evidence

**5 Dead README · modes, rules and extended tables**
Mode boundaries, the 21 limit, hand privacy, online waiting and the disclosed larger-table variant. Reviewed 31 October 2026. Exact deck counts for a particular physical edition have not been verified.

**Original game captures and DESIGN.md**
Existing outputs/5dead-arena-v2.png and 5dead-card-faces-phone.png. These show the actual game presentation, including its generated arena, avatar and card art.

Read the individual web case study

**ALIBI / SOCIAL DEDUCTION · 3-8 PLAYERS**

# Watch privately. Give just enough away.

A social video-imposter game that takes a group through private clips, one-word clues, discussion, voting and a reveal.

MY CONTRIBUTION / Game direction, private-viewing flow & group interaction design

SCOPE / Playable browser game

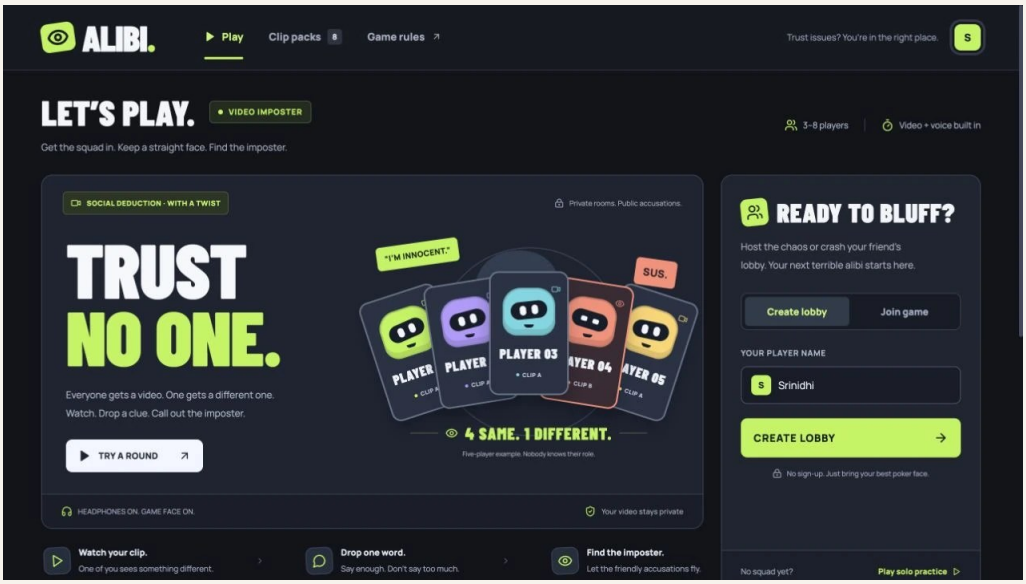
**3-8**
players in a room

**One word**
clue from each player

**Private**
clips and votes before reveal

## The challenge

A social-deduction game loses its rhythm when someone misses a step or sees information too soon. Alibi gives most players the same clip and one player a related alternative, then moves the group through clues, discussion and voting. The flow needs to protect the mystery without making the group wait without an explanation.




Actual local browser interface. The five-player diagram explains the rules before a room starts.

ALIBI / APPROACH & DECISIONS

# How I approached it

I structured each round as private viewing, one-word submission, shared clues, discussion, a hidden vote and the reveal. Players need a shared phase and a private view at the same time.

I used a solo practice path with clearly labelled simulated opponents so the whole sequence can be tried before inviting friends. A practice lobby should not look like proof that other people are connected.

The project includes room chat and built-in calling, with external calls as another option. Playback depends on YouTube availability and embedding, and the server keeps room state in memory. Those limits make readiness, replacement clips and host recovery controls part of the design.

## 01 / Keep private viewing separate from the group phase

Show each assigned clip privately and release clues together after submission or timeout.

**Why it matters.** Early clues or role information could change the game before everyone has had a chance to act.

**The trade-off.** A shared phase still needs a clear explanation when a player is loading or has not submitted.

## 02 / Make practice an honest invitation

Label simulated players throughout solo practice and let the person try the real sequence.

**Why it matters.** Someone can learn the rhythm without assembling a group or mistaking demo activity for real participants.

**The trade-off.** Practice can explain controls but cannot reproduce the conversation and hesitation of a group.

## 03 / Explain waits and recovery

Keep the current phase, readiness and host controls visible when playback or connection needs attention.

**Why it matters.** The group needs a next action rather than an unexplained delay.

**The trade-off.** A replacement clip or returning player can affect the round. The interface needs to say when play restarts instead of hiding that change.

ALIBI / OUTCOME & REFLECTION

# What came out of it

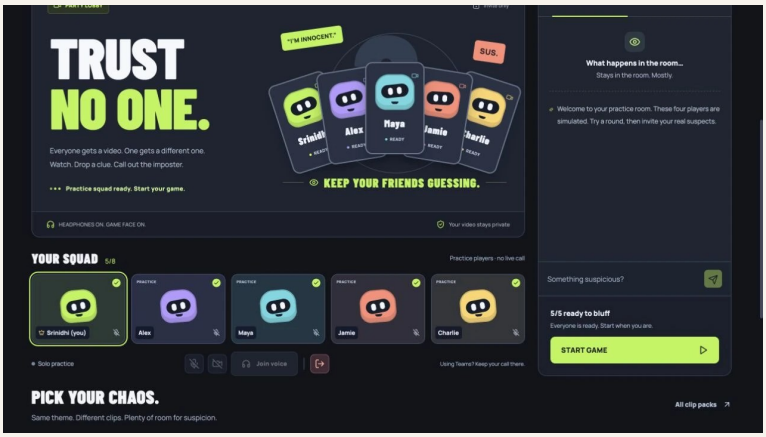
I connected the lobby, private watching, clues, discussion, votes and reveal into one group flow. Solo practice explains the steps with simulated opponents; connected rooms support real players. A study with a real group is the next step for evaluating the social experience.

# What I would do next

I would watch a real group through a failed clip and a late player, not just a successful reveal. Those pauses would show whether the instructions preserve the social rhythm or turn the host into technical support.

Proposed study: run a short group session and observe phase explanations, accidental information sharing and waiting time.

Proposed checks: a blocked YouTube embed, a reconnecting player, a departing host and switching between built-in and external calling.




Actual solo-practice lobby with clearly labelled simulated opponents.

## Artifacts & evidence

**Alibi README and round-state source**
Private assignments, simultaneous clues, secret votes, practice labelling, playback and connection limits. Reviewed 31 October 2026.

**Actual browser captures · solo practice**
Captured from the local VideoImposter application. The practice capture uses its clearly labelled simulated opponents, with no camera, microphone or personal-call recording.

Read the individual web case study